ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БАШКИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Стерлитамакский филиал

Колледж

Рабочая программа профессионального модуля

Наименование профессионального модуля — ***ПМ.02 Осуществление интеграции программных модулей***

Профессиональный цикл (обязательная часть)
цикл дисциплины и его часть

специальность

09.02.07	***Информационные системы и программирование***
код	наименование специальности

квалификация
Администрирование баз данных

Год начала подготовки
2022

Разработчик (составитель)

Зарипова Л.З.
ученая степень, ученое звание, категория, Ф.И.О.

Стерлитамак 2022

ОГЛАВЛЕНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

1.1. Область применения рабочей программы

Рабочая программа профессионального модуля является частью основной профессиональной образовательной программы в соответствии с ФГОС для специальности: 09.02.07 (укрупненная группа специальностей 09.00.00 Информатика и вычислительная техника), для обучающихся очной формы обучения.

Рабочая программа разработана с учетом Профессионального стандарта «Специалист по информационным системам», утвержден приказом Министерства труда и социальной защиты Российской Федерации от 17 сентября 2014 г. № 647н (зарегистрирован Министерством юстиции Российской Федерации 24 ноября 2014 г., регистрационный N 34846).

1.2. Место профессионального модуля в структуре основной образовательной программы

Профессиональный модуль относится к профессиональному циклу, входящей в обязательную часть ППССЗ.

1.3. Цель и планируемые результаты освоения профессионального модуля

В результате изучения профессионального модуля обучающийся должен освоить основной вид деятельности «Осуществление интеграции программных модулей» и соответствующие ему общие компетенции и профессиональные компетенции:

1.3.1. Перечень общих компетенций

Код	Наименование общих компетенций
ОК 01.	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
ОК 02.	Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.
ОК 09.	Пользоваться профессиональной документацией на государственном и иностранном языках.

1.3.2. Перечень профессиональных компетенций

Код	Наименование видов деятельности и профессиональных компетенций
ПК 2.1	Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.
ПК 2.2	Выполнять интеграцию модулей в программное обеспечение.
ПК 2.3	Выполнять отладку программного модуля с использованием специализированных программных средств.
ПК 2.4	Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

В результате освоения профессионального модуля обучающийся должен:

Иметь практический опыт	в интеграции модулей в программное обеспечение; в отладке программных модулей;
уметь	использовать выбранную систему контроля версий; использовать методы для получения кода с заданной

	функциональностью и степенью качества;
знать	модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; основы верификации и аттестации программного обеспечения.

1.4. Количество часов, отводимое на освоение профессионального модуля

Всего 422 ч.
Из них на освоение:
МДК.02.01 Технология разработки программного обеспечения – 80 часов;
МДК.02.02 Инструментальные средства разработки программного обеспечения – 90 часов;
МДК.02.03 Математическое моделирование – 66 часов
в том числе самостоятельная работа 30 часов
Учебная практика 72 часа
Производственная практика 108 часов
Экзамен 6 часов

Вид учебной работы	Объем часов
Объем образовательной программы (всего часов по ПМ)	422
Во взаимодействии с преподавателем (всего по ПМ)	210
в том числе:	
лекции, уроки	70
в том числе в форме практической подготовки (если предусмотрено)	*
практические занятия	140
Практика	
в том числе:	
учебная практика	72
в том числе в форме практической подготовки (если предусмотрено)	*
производственная практика	108
в том числе в форме практической подготовки (если предусмотрено)	
Экзамен по модулю / квалификационный экзамен	6

2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

2.1 Объем профессионального модуля и виды учебной работы

Очная форма обучения

Коды общих и профессиональных компетенций	Наименования разделов профессионального модуля (МДК)	**Объем образовательной программы**	**Работа обучающихся во взаимодействии с преподавателем**						Самостоятельная работа	Консультации	Промежуточная аттестация
			Обучение по МДК, в час.				*Практики*				
			Всего, часов	В том числе, лекции, в час.	В том числе, лабораторных и практических занятий, в час.	Курсовых работ (проектов)	Учебная практика, в час.	Производственная практика, в час.			
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*		
ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5	МДК 02.01 «Технология разработки программного обеспечения»	80	58	20	38	-			12	4	6
ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4,	МДК.02.02 «Инструментальные средства разработки программного обеспечения»	90	68	20	48				12		

ПК 2.5										4	6
ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5	МДК.02.03 «Математическое моделирование»	66	60	30	30				6	4	6
	Практика	180	-	-	-	-	72	108			
	Экзамен по модулю	6									6
	Всего:	422	176	60	116	X	72	108	30	12	24

2.2. Тематический план и содержание профессионального модуля

Наименование разделов и тем профессионального модуля (ПМ), междисциплинарных курсов (МДК)	**Содержание учебного материала**	**Объем часов**	**Осваиваемые компетенции**
1	**2**	**3**	
МДК.02.01 Технология разработки программного обеспечения		***80 ч***	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
Тема 1.1. Программное обеспечение	**Содержание:**	***4***	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	1. Основные понятия. Защита программного обеспечения.	*2*	
	2.Классификация ПО. Операционные системы. Инструментарий технологий программирования	*2*	
Тема 1.2. Разработка программного	**Содержание:**	***4***	ОК 01, ОК 02, ОК 09, ПК
	1. Жизненный цикл программного обеспечения. Основы разработки программного обеспе-	*2*	

обеспечения	чения.		2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	2. Модели жизненного цикла. *Технология программирования в компании Microsoft.*	*2*	
	Тематика практических занятий.	***4***	ОК 01, ОК 02,
	1. Техническое задание на проектирование программы	*4*	ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
Тема 1.3. Проектирование программ	**Содержание:**	***4***	ОК 01, ОК 02,
	1. Начала унифицированного языка моделирования UML.Непрограммные сущности.	*2*	ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	2. Диаграммы UML.Диаграммы объектов. Диаграммы прецедентов. Диаграммы взаимодействий. Диаграммы деятельности	*2*	
	Тематика практических занятий.	***8***	ОК 01, ОК 02,
	1. Стадия разработки программного обеспечения «Эскизный проект».	*4*	ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	2. Стадия разработки программного обеспечения «Технический проект»	*4*	
Тема 1.4. Модульное программирование	**Содержание :**	***4***	ОК 01, ОК 02,
	1. Современные технологии программирования. Объектно-ориентированное программирование.	*2*	ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	2 Технология OLE. Принцип работы .NET	*2*	
	Тематика практических занятий.	***8***	ОК 01, ОК 02,
	1. Использование объектно-ориентированного программирования (ООП) для создания качественного программного обеспечения	*4*	ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	2. Использование визуальных компонент для создания качественных программ.	*4*	
Тема 1.5. Тестирование и сопровождение программ	**Содержание :**	***4***	ОК 01, ОК 02,
	1. Тестирование и отладка программ. Надежность программного обеспечения. Порядок разработки тестов.	*2*	ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	2. Автоматизация тестирования. Надежность программного обеспечения. Количественные характеристики надежности программ	*2*	
	Тематика практических занятий.	***18***	ОК 01, ОК 02,

	1. Средства отладки программ в объектно-ориентированном программировании	*4*	ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	2. Использование стиля программирования	*4*	
	3. Методы оптимальной обработки текстовой информации	*2*	
	4. Оптимальное построение структур данных	*2*	
	5. Структурное программирование с использованием процедур и функций	*2*	
	6. Программирование с использованием средств графической информации	*2*	
	7. Использование OLE- и COM-технологий программирования	*2*	
Самостоятельная работа при изучении МДК 02.01: Работа с Интернет-ресурсами, подготовка к практическим занятиям, дифференцированному зачету, выполнение практических работ и составление по ним отчетов, работа с дополнительными источниками		*12ч*	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
Консультации		*4 ч*	
Промежуточная аттестация: Экзамен по МДК.02.01. Технология разработки программного обеспечения		*6 ч*	
МДК.02.02 Инструментальные средства разработки программного обеспечения		***90 ч***	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
Тема 2.1. Введение в функциональное программирование. Понятие функции и функциональной программы.	**Содержание:**	***6***	
	1. Понятие функции и функциональной программы. История развития языков функционального программирования. Программирование при помощи функций.	*2*	
	2. Императивный, объектно-ориентированный, логический и функциональный подходы к программированию – достоинства, недостатки и основные характеристики.	*2*	
	3. Общие сведения о функциональном подходе к программированию.	*2*	
Тема 2.2. Основы функционального программирования.	**Содержание:**	***4***	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	1. Основы функционального программирования на языке Haskell. Основы языка Haskell. Символы, константы, атомы, логические значения.	*2*	
	2. Базовые функции. Элементарные понятия; приемы программирования.	*2*	
	Тематика практических занятий.	***36***	
	1. "Программирование ветвлений".	*8*	
	2. "Оператор выбора".	*8*	
	3. "Операторы цикла".	*8*	
	4. "Программирование массивов".	*12*	

Тема 2.3. Программирование на Haskell.	**Содержание:**	***4***	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	1. Рекурсивные функции. Примеры применения рекурсивных функций на различных задачах обработки списков. Проблема выбора подфункций.	*2*	
	2. Проблема модульности функциональной программы. Возможность накапливающих параметров на примере инверсии списка. Локальные определения в функциональных программах. Точечная запись выражений.	*2*	
	Тематика практических занятий.	***12***	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	1. Рекурсивные алгоритмы.	*12*	
Тема 2.4. Представление и интерпретация функциональных программ. .Особенности интерпретирования императивных программ.	**Содержание:**	***6***	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	1. Представление и интерпретация функциональных программ. Абстрактная и конкретная формы программ.	*2*	
	2. Проблема связывания значений и переменных. Особенности интерпретирования императивных программ. Особенности интерпретирования императивных программ.	*2*	
	3. Функциональные эквиваленты императивных программ. Аппаратное обеспечение функциональных программ.	*2*	
Самостоятельная работа при изучении МДК 02.02: Работа с Интернет-ресурсами, подготовка к практическим занятиям, дифференцированному зачету, выполнение практических работ и составление по ним отчетов, работа с дополнительными источниками		*12 ч*	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
Консультации		*4 ч*	
Промежуточная аттестация: экзамен по МДК.02.02. Инструментальные средства разработки программного обеспечения		*6 ч*	
МДК.02.03 Математическое моделирование		***66 ч***	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
Тема 3.1. Основы моделирования. Детерминированные задачи	**Содержание:**	***10***	
	1. Понятие решения. Множество решений, оптимальное решение. Показатель эффективности решения. Математические модели, принципы их построения, виды моделей.	*2*	
	2. Задачи: классификация, методы решения, граничные условия. Общий вид и основная задача линейного программирования. Симплекс – метод. Транспортная задача. Методы нахож-	*2*	

	дения начального решения транспортной задачи. Метод потенциалов.		
	3. Общий вид задач нелинейного программирования. Графический метод решения задач нелинейного программирования. Метод множителей Лагранжа.	*2*	
	4. Основные понятия динамического программирования: шаговое управление, управление операцией в целом, оптимальное управление, выигрыш на данном шаге, выигрыш за всю операцию, аддитивный критерий, мультипликативный критерий. Простейшие задачи, решаемые методом динамического программирования.	*2*	
	5. Методы хранения графов в памяти ЭВМ. Задача о нахождении кратчайших путей в графе и методы ее решения. Задача о максимальном потоке и алгоритм Форда–Фалкерсона.	*2*	
	Тематика практических занятий:	***14***	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	1. «Построение простейших математических моделей.»	*2*	
	2. «Сведение произвольной задачи линейного программирования к основной задаче линейного программирования»	*2*	
	3. «Решение простейших однокритериальных задач графическим методом». «Решить графическим методом задачу программирования.»	*2*	
	4. «Решение простейших однокритериальных задач симплекс-методом». «Решение простейших однокритериальных задач симплекс-методом с искусственным базисом»	*2*	
	5. «Нахождение начального решения транспортной задачи методом северо-западного угла.» «Нахождение начального решения транспортной задачи методом минимальной стоимости.»	*2*	
	6. «Решение транспортной задачи методом потенциалов»	*2*	
	7. «Графический метод решения задач нелинейного программирования» «Метод множителей Лагранжа.»	*2*	
Тема 3.2 Задачи в условиях неопределенности	**Содержание:**	***10***	ОК 01, ОК 02, ОК 09, ПК 2.1, ПК 2.2, ПК 2.3, ПК 2.4, ПК 2.5
	1. Системы массового обслуживания: понятия, примеры, модели. Основные понятия теории марковских процессов: случайный процесс, марковский процесс, граф состояний, поток событий, вероятность состояния, уравнения Колмогорова, финальные вероятности состояний.	*2*	
	2. Схема гибели и размножения. Метод имитационного моделирования. Единичный жребий и формы его организации. Примеры задач. Понятие прогноза.	*2*	
	3. Количественные методы прогнозирования: скользящие средние, экспоненциальное сглаживание, проектирование тренда. Качественные методы прогноза. Предмет и задачи теории игр.	*2*	
	4. Основные понятия теории игр: игра, игроки, партия, выигрыш, проигрыш, ход, личные и случайные ходы, стратегические игры, стратегия, оптимальная стратегия. Антагонистические матричные игры: чистые и смешанные стратегии. Методы решения конечных игр: сведение игры mxn к задаче линейного программирования, численный метод – метод итераций.	*2*	

	5. Область применимости теории принятия решений. Принятие решений в условиях определенности, в условиях риска, в условиях неопределенности. Критерии принятия решений в условиях неопределенности. Дерево решений.	*2*	
	Тематика практических занятий:		
	1. «Исследование характеристик случайного потока требований в телекоммуникационной системе».	*2*	
	2. «Исследование характеристик случайного потока освобождений каналов в телекоммуникационной системе».	*2*	
	3. «Прогнозирование временных рядов с использованием скользящей средней»	*2*	
	4. «Прогнозирование временных рядов с использованием тренда.»	*2*	
	5. «Элементы теории игр»	*2*	
	Самостоятельная работа при изучении МДК 02.03: Решение графическим методом задачи программирования	**6**	
	Консультации	**4**	
	Промежуточная аттестация: экзамен по МДК.02.03. Математическое моделирование	*6*	
	Учебная практика **Виды работ:** 1. Ознакомление с базой практики. 2. Изучение аппаратно-программного обеспечения базы практики. 3. Выполнение индивидуальных заданий. 4. Выполнение поручений руководителя практики. 5. Оформление отчета по практике.	**72**	ОК 01, ОК 02, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6
	Промежуточная аттестация: *дифференцированный зачет.*	**-**	
	Производственная практика **Виды работ:** 1. Ознакомление с базой практики. 2. Изучение аппаратно-программного обеспечения базы практики. 3. Выполнение индивидуальных заданий. 4. Выполнение поручений руководителя практики. 5. Оформление отчета по практике.	**108**	ОК 01, ОК 02, ОК 09, ПК 1.1, ПК 1.2, ПК 1.3, ПК 1.4, ПК 1.5, ПК 1.6
	Промежуточная аттестация: *дифференцированный зачет.*	**-**	
	Курсовая работа (проект)	**-**	
	Промежуточная аттестация: *экзамен по модулю.*	**6**	
	Всего	**422**	

3. ФОРМЫ КОНТРОЛЯ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРЫ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, ПРАКТИЧЕСКОГО ОПЫТА, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ

– включает контрольные задания и критерии их оценки, а также описания форм и процедур для *экзамена по модулю/квалификационного экзамена* по ПМ**,** предназначен для определения качества освоения обучающимися профессионального модуля (готовность к выполнению вида профессиональной деятельности, владение ПК и ОК). Фонд оценочных средств по профессиональному модулю представлен в Приложении № 2.

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

4.1. Требования к минимальному материально-техническому обеспечению

Реализация программы дисциплины требует наличия учебных аудиторий 35, 36, 37, 24 и лекционных аудиторий.

Аудитории для самостоятельной работы №144.

Оборудование учебного кабинета: учебная мебель, компьютер в сборе, проектор, экран.

4.2. Учебно-методическое и информационное обеспечение профессионального модуля

4.2.1. Перечень основной и дополнительной учебной литературы, необходимой для освоения профессионального модуля

Основная учебная литература:

1. Е.В. Поколодина, Н.А. Долгова, Д.В. Ананьев Ревьюирование программных модулей : учебник для студ. учреждений сред. проф. образования. – М.: Издательский центр «Академия», 2020. – 208 с. IBSN 978-5-4468-8609-8.

2. А.Е. Генельт Учебно-методическое пособие по дисциплине «Управление качеством разработки ПО» - СПб. : Изд-во СПГУ ИТМО, 2021. — 187 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/43557 (дата обращения: 25.09.2023). — Режим доступа: для авториз. пользователей.

3. Е.В. Ковалевская Метрология, качество и сертификация программного обеспечения : учеб. программа, руководство по изучению дисциплины, учебное пособие, практикум по курсу, тестовые задания по дисциплине. – М.: МГУЭСИ, 2021 г. – 96 с. ISBN 978-5-8114-1832-9 — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/61361 (дата обращения: 25.09.2023). — Режим доступа: для авториз. пользователей.

Дополнительная учебная литература:

1. И.И. Мазур, В.Д. Шапиро, Н.Г. Ольдерогге Управление проектами. учеб. пособие для студентов. – М.: Омега – Л, 2004 г. – 960 с. ISBN 978 5 370 01058 3.

4.2.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее - сеть «Интернет»), необходимых для освоения профессионального модуля

	Наименование электронной библиотечной системы
1.	Договор на доступ к ЭБС ZNANIUM.COM между УУНиТ в лице директора СФ УУНиТ и ООО «Знаниум»№ 1151-эбс от 11.07.2023
2.	Договор на доступ к ЭБС ZNANIUM.COM между УУНиТ в лице директора СФ УУНиТ и ООО «Знаниум»№ 223/801 от 23.08.2023 (предоставление доступа к коллекции ЭФУ «Федеральный перечень учебников издательства «Провещение»
3.	Договор на доступ к ЭБС «ЭБС ЮРАЙТ» (полная коллекция) между УУНиТ в лице директора СФ УУНиТ и ООО «Электронное издательство ЮРАЙТ» № 1/23-эбс от 03.03.2023
4.	Договор на доступ к ЭБС «Университетская библиотека онлайн» между БашГУ и «Нексмедиа» № 223-950 от 05.09.2022

5.	Договор на доступ к ЭБС «Лань» между БашГУ и издательством «Лань» № 223-948 от 05.09.2022
6.	Договор на доступ к ЭБС «Лань» между БашГУ и издательством «Лань» № 223-949 от 05.09.2022
7.	Соглашение о сотрудничестве между БашГу и издательством «Лань» № 5 от 05.09.2022
8.	ЭБС «ЭБ БашГУ**», бессрочный** договор между БашГУ и ООО «Открытые библиотечные системы» № 095 от **01.09.2014 г.**
9.	Договор на доступ к электронным изданиям в составе базы данных «НАУЧНАЯ ЭЛЕКТРОННАЯ БИБЛИОТЕКА eLIBRARY.RU» между УУНиТ и ООО НЭБ № SU-20179 /2023 от 28.03.2023
10.	Договор на БД диссертаций между УУНиТ и РГБ № 223-997 от 11.07.2023
11.	Договор о подключении к НЭБ и о предоставлении доступа к объектам НЭБ между БашГУ в лице директора СФ БашГУ с ФГБУ «РГБ» № 101/НЭБ/1438-П от 11.06.2019

4.3.3. Перечень информационных технологий, используемых при осуществлении образовательного процесса по профессиональному модулю, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Наименование программного обеспечения
Пакет Microsoft Office 2019
Visual Studio Code — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS.
Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов.
PyCharm: IDE кроссплатформенная интегрированная среда разработки для языка программирования Python
Dev-C++ — свободная интегрированная среда разработки приложений для языков программирования C/C++

ПРИЛОЖЕНИЕ 1

ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«БАШКИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Стерлитамакский филиал

Колледж

Календарно-тематический план

по профессиональному модулю ***ПМ.02 Осуществление интеграции программных модулей***

специальность

09.02.07 ***Информационные системы и программирование***

код наименование специальности

Квалификация

Администрирование баз данных

Разработчик (составитель)

Зарипова Л.З.

ученая степень, ученое звание, категория, Ф.И.О.

Стерлитамак 2022

№ п/п	Наименование разделов и тем	Кол-во часов	Календарные сроки изучения (план)	Вид занятия	Домашнее задание
МДК 02.01. ***Технология разработки программного обеспечения***		**58 ч**			
1	Основные понятия. Защита программного обеспечения.	2/2	Январь	лекция	составление опорного конспекта
2	Классификация ПО. Операционные системы. Инструментарий технологий программирования	2/4	Январь	лекция	составление опорного конспекта
3	Жизненный цикл программного обеспечения. Основы разработки программного обеспечения.	2/6	Январь	лекция	составление опорного конспекта
4	Модели жизненного цикла. Технология программирования в компании Microsoft.	2/8	Февраль	лекция	составление опорного конспекта
5	Техническое задание на проектирование программы	2/10	Февраль	практическое занятие	решение задач
6	Техническое задание на проектирование программы	2/12	Февраль	практическое занятие	решение задач
7	Начала унифицированного языка моделирования UML. Непрограммные сущности.	2/14	Март	лекция	составление опорного конспекта
8	Стадия разработки программного обеспечения «Эскизный проект».	2/16	Март	практическое занятие	решение задач
9	Стадия разработки программного обеспечения «Эскизный проект».	2/18	Март	практическое занятие	решение задач
10	Диаграммы UML. Диаграммы объектов. Диаграммы прецедентов. Диаграммы взаимодействий. Диаграммы деятельности	2/20	Март	лекция	составление опорного конспекта
11	Стадия разработки программного обеспечения «Технический проект»	2/22	Март	практическое занятие	решение задач
12	Стадия разработки программного обеспечения «Технический проект»	2/24	Март	практическое занятие	решение задач
13	Современные технологии программирования. Объектно-	2/26	Март	лекция	составление опор-

	ориентированное программирование.				ного конспекта
14	Технология OLE. Принцип работы .NET	2/28	Апрель	лекция	составление опорного конспекта
15	Использование объектно-ориентированного программирования (ООП) для создания качественного программного обеспечения	2/30	Апрель	практическое занятие	решение задач
16	Использование объектно-ориентированного программирования (ООП) для создания качественного программного обеспечения	2/32	Апрель	практическое занятие	решение задач
17	Использование визуальных компонент для создания качественных программ.	2/34	Апрель	практическое занятие	решение задач
18	Использование визуальных компонент для создания качественных программ.	2/36	Апрель	практическое занятие	решение задач
19	Тестирование и отладка программ. Надежность программного обеспечения. Порядок разработки тестов.	2/38	Апрель	лекция	составление опорного конспекта
20	Автоматизация тестирования. Надежность программного обеспечения. Количественные характеристики надежности программ	2/40	Апрель	лекция	составление опорного конспекта
21	Средства отладки программ в объектно-ориентированном программировании	2/42	май	практическое занятие	решение задач
22	Средства отладки программ в объектно-ориентированном программировании	2/44	май	практическое занятие	решение задач
23	Использование стиля программирования	2/46	май	практическое занятие	решение задач
24	Использование стиля программирования	2/48	май	практическое занятие	решение задач
25	Методы оптимальной обработки текстовой информации	2/50	май	практическое занятие	решение задач
26	Оптимальное построение структур данных	2/52	май	практическое занятие	решение задач
27	Структурное программирование с использованием процедур и функций	2/54	июнь	практическое занятие	решение задач
28	Программирование с использованием средств графической информации	2/56	июнь	практическое занятие	решение задач
29	Использование OLE- и COM-	2/58	июнь	практиче-	решение

	технологий программирования			ское занятие	задач
МДК 02.02. ***Инструментальные средства разработки программного обеспечения***		**68 ч**			
1	Понятие функции и функциональной программы. История развития языков функционального программирования. Программирование при помощи функций.	2/2	Январь	лекция	составление опорного конспекта
2	Императивный, объектно-ориентированный, логический и функциональный подходы к программированию – достоинства, недостатки и основные характеристики.	2/4	Январь	лекция	составление опорного конспекта
3	Общие сведения о функциональном подходе к программированию.	2/6	Январь	лекция	составление опорного конспекта
4	Основы функционального программирования на языке Haskell. Основы языка Haskell. Символы, константы, атомы, логические значения.	2/8	Февраль	лекция	составление опорного конспекта
Февраль	Базовые функции. Элементарные понятия; приемы программирования.	2/10	Февраль	лекция	составление опорного конспекта
Февраль	Программирование ветвлений.	2/12	Февраль	практическое занятие	решение задач
7	Программирование ветвлений.	2/14	Февраль	практическое занятие	решение задач
8	Программирование ветвлений.	2/16	Февраль	практическое занятие	решение задач
9	Программирование ветвлений.	2/18	Февраль	практическое занятие	решение задач
10	Оператор выбора.	2/20	Март	практическое занятие	решение задач
11	Оператор выбора.	2/22	Март	практическое занятие	решение задач
12	Оператор выбора.	2/24	Март	практическое занятие	решение задач
13	Оператор выбора.	2/26	Март	практическое заня-	решение задач

				тие	
14	Операторы цикла.	2/28	Март	практическое занятие	решение задач
15	Операторы цикла.	2/30	Март	практическое занятие	решение задач
16	Операторы цикла.	2/32	Апрель	практическое занятие	решение задач
17	Операторы цикла.	2/34	Апрель	практическое занятие	решение задач
18	Программирование массивов.	2/36	Апрель	практическое занятие	решение задач
19	Программирование массивов.	2/38	Апрель	практическое занятие	решение задач
20	Программирование массивов.	2/40	Апрель	практическое занятие	решение задач
21	Программирование массивов.	2/42	Апрель	практическое занятие	решение задач
22	Программирование массивов.	2/44	Апрель	практическое занятие	решение задач
23	Программирование массивов.	2/46	Апрель	практическое занятие	решение задач
24	Рекурсивные функции.	2/48	Апрель	лекция	составление опорного конспекта
25	Проблема модульности функциональной программы.	2/50	Апрель	лекция	составление опорного конспекта
26	Представление и интерпретация функциональных программ. Абстрактная и конкретная формы программ.	2/52	май	лекция	составление опорного конспекта
27	Особенности интерпретирования императивных программ.	2/54	май	лекция	составление опорного конспекта
28	Функциональные эквиваленты императивных программ. Аппаратное обеспечение функциональных программ.	2/56	май	лекция	составление опорного конспекта
29	Рекурсивные алгоритмы.	2/58	май	практическое заня-	решение задач

				тие	
30	Рекурсивные алгоритмы.	2/60	май	практическое занятие	решение задач
31	Рекурсивные алгоритмы.	2/62	май	практическое занятие	решение задач
32	Рекурсивные алгоритмы.	2/64	июнь	практическое занятие	решение задач
33	Рекурсивные алгоритмы.	2/66	июнь	практическое занятие	решение задач
34	Рекурсивные алгоритмы.	2/68	июнь	практическое занятие	решение задач
МДК 02.03. Математическое моделирование		**60**			
1	Понятие решения. Множество решений, оптимальное решение. Показатель эффективности решения. Математические модели, принципы их построения, виды моделей.	2/2 ч	январь	лекция	составление опорного конспекта
2	Задачи: классификация, методы решения, граничные условия. Общий вид и основная задача линейного программирования. Симплекс – метод. Транспортная задача. Методы нахождения начального решения транспортной за-дачи. Метод потенциалов.	2/4	январь	лекция	составление опорного конспекта
3	Общий вид задач нелинейного программирования. Графический метод решения задач нелинейного программирования. Метод множителей Лагранжа.	2/6	январь	лекция	составление опорного конспекта
4	Численная модель решения задачи теплопроводности. Моделирование распределения температуры	2/8	январь	лекция	составление опорного конспекта
5	Имитационное моделирование. Математический аппарат имитационного моделирования.	2/10	январь	лекция	составление опорного конспекта
6	Моделирование движения в поле силы тяжести. Компьютерное моделирование свободного падения.	2/12	январь	лекция	составление опорного конспекта
7	Математическая модель задачи баллистики.	2/14	январь	лекция	составление опорного конспекта
8	Графический метод решения задач	2/16	январь	лекция	составле-

	линейного программирования.				ние опорного конспекта
9	«Построение простейших математических моделей.»	2/18	январь	практическое занятие	решение задач
10	«Сведение произвольной задачи линейного программирования к основной задаче линейного программирования»	2/20	январь	практическое занятие	решение задач
11	«Решение простейших однокритериальных задач графическим методом». «Решить графическим методом задачу программирования.»	2/22	февраль	практическое занятие	решение задач
12	Основные понятия динамического программирования: шаговое управление, управление операцией в целом, оптимальное управление, выигрыш на данном шаге, выигрыш за всю операцию, аддитивный критерий, мультипликативный критерий. Простейшие задачи, решаемые методом динамического программирования.	2/24	февраль	лекция	составление опорного конспекта
13	Методы хранения графов в памяти ЭВМ. Задача о нахождении кратчайших путей в графе и методы ее решения. Задача о максимальном потоке и алгоритм Форда–Фалкерсона.	2/26	февраль	лекция	составление опорного конспекта
14	«Решение простейших однокритериальных задач симплекс-методом». «Решение простейших однокритериальных задач симплекс-методом с искусственным базисом»	2/28	февраль	практическое занятие	решение задач
15	«Нахождение начального решения транспортной задачи методом северо-западного угла.» «Нахождение начального решения транспортной задачи методом минимальной стоимости.»	2/30	февраль	практическое занятие	решение задач
16	«Решение транспортной задачи методом потенциалов»	2/32	февраль	практическое занятие	решение задач
17	«Графический метод решения задач нелинейного программирования» «Метод множителей Лагранжа.»	2/34	март	практическое занятие	решение задач
18	Системы массового обслуживания: понятия, примеры, модели. Основные понятия теории марковских процессов: случайный процесс,	2/36	март	лекция	составление опорного конспекта

	марковский процесс, граф состояний, поток событий, вероятность состояния, уравнения Колмогорова, финальные вероятности состояний.				
19	2. Схема гибели и размножения. Метод имитационного моделирования. Единичный жребий и формы его организации. Примеры задач. Понятие прогноза.	2/38	март	лекция	составление опорного конспекта
20	3. Количественные методы прогнозирования: скользящие средние, экспоненциальное сглаживание, проектирование тренда. Качественные методы прогноза. Предмет и задачи теории игр.	2/40	март	лекция	составление опорного конспекта
21	«Исследование характеристик случайного потока требований в телекоммуникационной системе».	2/42	март	практическое занятие	решение задач
22	«Исследование характеристик случайного потока требований в телекоммуникационной системе».	2/44	март	практическое занятие	решение задач
23	«Исследование характеристик случайного потока освобождений каналов в телекоммуникационной системе».	2/46	март	практическое занятие	решение задач
24	«Исследование характеристик случайного потока освобождений каналов в телекоммуникационной системе».	2/48	март	практическое занятие	решение задач
25	Основные понятия теории игр: игра, игроки, партия, выигрыш, проигрыш, ход, личные и случайные ходы, стратегические игры, стратегия, оптимальная стратегия. Антагонистические матричные игры: чистые и смешанные стратегии. Методы решения конечных игр: сведение игры mxn к задаче линейного программирования, численный метод – метод итераций.	2/50	апрель	лекция	составление опорного конспекта
26	5. Область применимости теории принятия решений. Принятие решений в условиях определенности, в условиях риска, в условиях неопределенности. Критерии принятия решений в условиях неопределенности. Дерево решений.	2/52	апрель	лекция	составление опорного конспекта
27	«Прогнозирование временных рядов с использованием скользящей средней»	2/54	апрель	практическое занятие	решение задач
28	«Прогнозирование временных рядов с использованием скользящей средней»	2/56	апрель	практическое занятие	решение задач

29	«Прогнозирование временных рядов с использованием тренда.»	2/58	май	практическое занятие	решение задач
30	«Элементы теории игр»	2/60	май	практическое занятие	решение задач
Всего часов		**186**			

ПРИЛОЖЕНИЕ 2

ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«БАШКИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Стерлитамакский филиал

Колледж

Фонд оценочных средств

по профессиональному модулю ***ПМ.02 Осуществление интеграции программных модулей***

цикл дисциплины и его часть (обязательная, вариативная)

специальность

09.02.07 ***Информационные системы и программирование***

код наименование специальности

Квалификация

Администрирование баз данных

Разработчик (составитель)

Зарипова Л.З.

ученая степень, ученое звание, категория, Ф.И.О.

Стерлитамак 2022

I. Общие положения

1. Фонды оценочных средств предназначены для проверки результатов освоения вида профессиональной деятельности (ВПД) ПМ.02 Осуществление интеграции программных модулей и составляющих его профессиональных и общих компетенций**,** программы подготовки специалистов среднего звена по специальности 09.02.07 Информационные системы и программирование.

Формой аттестации по профессиональному модулю является экзамен по модулю.

Форма проведения экзамена –устный опрос и выполнение задания по билетам.

2. Формы контроля и оценивания элементов профессионального модуля

Таблица 1.1.

Элемент модуля	**Форма контроля и оценивания**	
	Промежуточная аттестация	**Текущий контроль**
МДК 02.01 «Технология разработки программного обеспечения»	Экзамен по модулю (4 семестр).	Наблюдение за выполнением практических работ. Контроль результата выполнения практических работ. Устный опрос на практических занятиях. Самостоятельная работа. Тестирование.
МДК.02.02 «Инструментальные средства разработки программного обеспечения»	Экзамен по модулю (4 семестр).	Наблюдение за выполнением практических работ. Контроль результата выполнения практических работ. Устный опрос на практических занятиях. Самостоятельная работа. Тестирование.
МДК.02.03 «Математическое моделирование»	Дифференцированный зачет (4 семестр), экзамен по модулю	Наблюдение за выполнением практических работ. Контроль результата выполнения практических работ. Устный опрос на практических занятиях. Самостоятельная работа. Тестирование.
УП	Дифференцированный зачет (4 семестр).	Наблюдение за выполнением работ на учебной практике.
ПП	Дифференцированный зачет (4 семестр).	Наблюдение за выполнением работ на производственной практике.

Далее размещаются ФОС по каждому МДК, УП и ПП.

3. Результаты освоения профессионального модуля, подлежащие проверке

В результате аттестации по профессиональному модулю комплексная проверка общих и профессиональных компетенций профессионального модуля осуществляется в форме оценки качества выполнения **заданий на *экзамене по модулю/квалификационном экзамене***:

Планируемые результаты освоения образовательной программы	**Этап**	**Показатели и критерии оценивания результатов обучения**				**Вид оценочного средства**
1.	2.	3.				4.
		неуд.	удовл.	хорошо	отлично	
ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам	1 этап: Знания: актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов ре-	Нет знаний, умений, практического опыта	Значительные затруднения при выборе способов решения задач профессиональной деятельности, применительно к различным контекстам	Незначительные затруднения при выборе способов решения задач профессиональной деятельности, применительно к различным контекстам	Обоснованный выбор способов решения задач профессиональной деятельности, применительно к различным контекстам	Устный опрос

	шения задач профессиональной деятельности.					
	2 этап: Умения: распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия; определить необходимые ресурсы					Тестовые задания
	3 этап: Иметь практический опыт: владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно					Решение

	или с помощью наставника)					ситуационных задач
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной	1 этап: Знания: номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации.	Нет знаний, умений, практического опыта	Значительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач профессиональной деятельности	Незначительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач профессиональной деятельности	Обоснованный поиск, анализ и интерпретация информации, необходимой для выполнения задач профессиональной деятельности	Устный опрос
	2 этап: Умения: определять задачи для поиска информации; определять необходимые источники информации.					Тестовые задания
	3 этап: Иметь практический опыт: планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска.					Решение ситуационных задач

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках	1 этап: Знания: современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности	Нет знаний, умений, практического опыта	Значительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач профессиональной деятельности	Незначительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач профессиональной деятельности	Обоснованный поиск, анализ и интерпретация информации, необходимой для выполнения задач профессиональной деятельности	Устный опрос Тестовые задания Решение ситуационных задач
	2 этап: Умения: применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение					
	3 этап: Иметь практический опыт: применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение					
ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.	1 этап: Знания: Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Виды и вариан-	Нет знаний, умений, практического опыта	Значительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач	Незначительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач профес-	Обоснованный поиск, анализ и интерпретация информации, необходимой для выполнения задач профессиональной деятельности	Устный опрос

	ты интеграционных решений. Современные технологии и инструменты интеграции. Основные протоколы доступа к данным. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Методы отладочных классов. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Графические средства проектирования архитектуры программных продуктов.		профессиональной деятельности	сиональной деятельности		
	2 этап: Умения: Анализировать проектную и техническую документацию. Использовать специализированные графические средства построения и анализа архитектуры программных продуктов.					Тестовые задания

	Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес-процессов. Определять источники и приемники данных. Проводить сравнительный анализ. Выполнять отладку, используя методы и инструменты условной компиляции (классы Debug и Trace). Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии.					
	3 этап: Иметь практический опыт: Разрабатывать и оформлять требования к программным модулям по предложенной документации. Разрабатывать тестовые наборы (пакеты) для программного модуля. Разрабатывать тестовые сценарии программного					Решение ситуационных задач

	средства. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.					
ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.	1 этап: Знания: Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные верификации программного обеспечения. Современные технологии и инструменты интеграции. Основные протоколы доступа к данным. Методы и способы идентефикации сбоев и ошибок при интеграции приложений. Основные методы и виды тестирования программных продуктов. Стандарты качества программной документации Основы организации ин-	Нет знаний, умений, практического опыта	Значительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач профессиональной деятельности	Незначительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач профессиональной деятельности	Обоснованный поиск, анализ и интерпретация информации, необходимой для выполнения задач профессиональной деятельности	Устный опрос

	спектирования и верификации. Приемы работы с инструментальными средствами тестирования и отладки. Методы организации работы в команде разработчиков.					
	2 этап: Умения: использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Организовывать заданную интеграцию модулей в программные средства на базе имеющейся архитектуры и автоматизации бизнес- процессов. Использовать различные транспортные протоколы и стандарты форматирования сообщений. Выполнять тестирование интеграции. Организовы-					Тестовые задания

	вать постобработку данных. Создавать классы-исключения на основе базовых классов. Выполнять ручное и автоматизированное тестирование программного модуля. Выявлять ошибки в системных компонентах на основе спецификаций- Использовать приемы работы в системах контроля версий.					
	3 этап: Иметь практический опыт: Интегрировать модули в программное обеспечение. Отлаживать программные модули. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.					Решение ситуационных задач

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств	1 этап: Знания: Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения.	Нет знаний, умений, практического опыта	Значительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач профессиональной деятельности	Незначительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач профессиональной деятельности	Обоснованный поиск, анализ и интерпретация информации, необходимой для выполнения задач профессиональной деятельности	Устный опрос
	2 этап: Умения: Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию. Использовать инструментальные сред ства отладки программных продуктов. Определять источники и приемники данных. Выполнять тес-					Тестовые задания

	тирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий. Выполнять отладку, используя методы и инструменты условной компиляции. Выявлять ошибки в системных компонентах на основе спецификаций.					
	3 этап: Иметь практический опыт: Отлаживать программные модули. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования					Решение ситуационных задач
ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.	1 этап: Знания: Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию	Нет знаний, умений, практического опыта	Значительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполне-	Незначительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач	Обоснованный поиск, анализ и интерпретация информации, необходимой для выполнения задач профессиональной деятельности	Устный опрос

	программных модулей. Основы верификации и аттестации программного обеспечения. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Методы и схемы обработки исключительных ситуаций. Основные методы и виды тестирования программных продуктов. Приемы работы с инструментальными средствами тестирования и отладки. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков.		ния задач профессиональной деятельности	профессиональной деятельности		
	2 этап: Уме-					

	ния: Использовать выбранную систему контроля версий. Анализировать проектную и техническую документацию. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий. Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии. Выполнять ручное и автоматизированное тестирование программного модуля. Выявлять ошибки в системных компонентах на основе спецификаций					Тестовые задания
	3 этап: Иметь практический опыт: Разрабатывать тестовые наборы (пакеты) для программного модуля. Разрабатывать тестовые сценарии про-					

	граммного средства. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.					Решение ситуационных задач
ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.	1 этап: Знания: Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации рабо-	Нет знаний, умений, практического опыта	Значительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач профессиональной деятельности	Незначительные затруднения при осуществлении поиска, анализа и интерпретации информации, необходимой для выполнения задач профессиональной деятельности	Обоснованный поиск, анализ и интерпретация информации, необходимой для выполнения задач профессиональной деятельности	Устный опрос

	ты в команде разработчиков.					
	2 этап: Умения: Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию. Организовывать постобработку данных. Приемы работы в системах контроля версий. Выявлять ошибки в системных компонентах на основе спецификаций.					Тестовые задания
	3 этап: Иметь практический опыт: Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.					Решение ситуационных задач

4. Структура контрольно-оценочных средств, для *экзамена по модулю*

4.1. Перечень вопросов, выносимых на *экзамен по модулю:*

МДК.02.01 Технология разработки программного обеспечения

1. Критерии качества программного средства. Определение качества ПО в стандарте ISO 9126. Многоуровневая модель качества ПО. Оценочные характеристики качества программного продукта

2. Жизненный цикл программного продукта, фазы жизненного цикла. Этапы классического жизненного цикла, их содержание.

3. Фаза разработки, этапы процесса разработки. Стратегии конструирования ПО: линейная, инкрементная, эволюционная.

4. Стандарт ISO/IEC 12207-95: основные определения – система, модель жизненного цикла, квалификационные требования. Основные процессы, их содержание, работы и задачи процесса разработки.

5. Стандарт ISO/IEC 15504 (SPICE): оценка возможностей разработчика. Связь этого стандарта с моделью зрелости предприятия SEI CMM.

6. Прогностические модели процесса разработки: каскадная, RAD, спиральная.

7. Адаптивные модели процесса разработки: экстремальное программирование, Scrum.

8. Руководство программным проектом. Предварительные оценки проекта. Системный анализ и анализ требований. Анализ рисков. Планирование процесса разработки. Типовая структура распределения работ.

9. Контроль процесса разработки. Размерно- и функционально-ориентированные метрики. Метрические характеристики объектно-ориентированных систем.

10. Структурный и объектно-ориентированный подходы к разработке ПО. Их сравнительный анализ. Сущность объектного подхода к разработке программных средств.

11. Анализ предметной области: цели и задачи. Модели предметной области. Формальные определения. Классификация моделей. Методология IDEF0, синтаксис IDEF0-моделей.

12. Диаграммы потоков данных (DFD-диаграммы) и диаграммы потоков работ (IDEF3-диаграммы), их использование при моделировании предметной области.

МДК.02.02 Инструментальные средства разработки программного обеспечения.

1. Объектно-ориентированный анализ предметной области. Методика определения границ системы и ключевых абстракций. Пример проведения анализа. Функциональные и нефункциональные требования к системе.

2. Функциональные требования к системе. Способ их представления в виде UML-диаграммы. Пример диаграммы с использованием отношений «расширяет» и «включает». Понятие прецедента и сценария.

3. Концептуальная модель системы: концептуальные классы, системные события и системные операции. Способ их представления в виде UML-диаграмм. Пример концептуального описания прецедента.

4. Диаграммы взаимодействия как элементы концептуальной модели. Синтаксис диаграмм взаимодействия.

5. Проектирование программных средств. Цели и задачи этапа проектирования. Понятие модели проектирования, ее отличия от концептуальной модели. Стадии проектирования, их краткая характеристика.

6. Задачи, решаемые на стадии эскизного проектирования. Понятие архитектуры ПС. Проблема выбора архитектуры. Влияние архитектуры на качественные характеристики ПС.

7. Понятие модуля и модульного программирования. Преимущества модульного подхода к разработке ПО. Модули как средство физического структурирования ПО. Свойства модулей.

8. Задачи, решаемые на стадии детального проектирования. Цели и задачи проектирования пользовательского интерфейса.

9. Понятие шаблона. Классификация шаблонов. Стандарт описания шаблонов.

10. Идентификация методов программных классов. Диаграммы классов, способы отображения отношений ассоциации и зависимости. Пример диаграммы классов.

11. Тестирование и отладка программного средства. Стадии тестирования и их характеристика. Основные принципы тестирования. Тесты и тестовые наборы. Понятие тестового покрытия.

12. Отладочное тестирование. Соотношение структурного и функционального подходов. Примеры реализации.

МДК.02.03 Математическое моделирование

1. Интеграционное тестирование. Виды интеграционного тестирования. Критерии полноты тестовых наборов. Регрессионное тестирование. Критерии завершения отладочного тестирования.

2. Системное тестирование. Виды системного тестирования. Критерии полноты тестовых наборов.

3. Особенности объектно-ориентированного тестирования. Расширение области применения тестирования. Критерии тестирования моделей. Тестирование классов. Тестирование кластеров и потоковое тестирование.

4. Понятие автоматизированного тестирования. Автотесты. Достоинства и недостатки автоматизированного тестирования. Средства автоматизированного тестирования.

5. Утилита модульного тестирования NUnit. Средства описания тестов. Утверждения, параметры утверждений.

6. Понятие версии программного продукта и системы контроля версий. Модели версиони-рования, их сравнение.

7. Система Subversion, ее архитектура. Хранилище, его структура, правки. Команды SVN для работы с хранилищем. Понятия рабочей копии и служебного каталога. Сценарий объединения правок. Конфликты и способы их разрешения.

8. Понятие сборки, манифест сборки. Сборка приложения, системы автоматизации сборки.

9. Утилита NAnt, файл сборки и его структура. Цели, зависимость целей, описание целей.

10. Документирование процесса разработки. Типы документов управления.

11. Документирование программного продукта. Документация сопровождения, ее назначение и состав. Пользовательская документация, ее назначение и состав.

Перечень вопросов, выносимых на экзамен по модулю

1. Метод Main – точка входа программы. Дайте описание данного метода. Приведите примеры кода.
2. Инкапсуляция переменных: механизмы её реализации и для чего она используется. Приведите примеры кода.
3. Статические и нестатические методы. Дайте описание данных методов, в чём заключается между ними отличие. Приведите примеры кода.

4. Механизм создания объекта класса. Роль конструктора в данном алгоритме. Вызов методов объекта класса. Приведите примеры кода.
5. get- и set- методы. Механизмы их функционирования. Модифицированный setметод. Цели, которые преследуются при использовании данных методов. Приведите примеры кода.
6. Методы типа void и методы, возвращающие значение. Дайте описание этих методов. Приведите примеры кода.
7. Реляционные базы данных. Первичный ключ, внешний ключ. Дать описание реляционных баз данных, пояснить принципы использования первичного и внешнего ключа. Привести примеры.
8. Технология ASP.NET. Основы веб-программирования. Файл программной логики. Дайте подробное описание данной технологии. Приведите примеры кода.
9. Технология ADO.NET. Подключение базы данных в проект с использование SQL-сервера. Привести конкретный алгоритм действий по подключению и развёртыванию базы данных в проекте.

10. UML-язык для описания сложных систем реального мира. Основные элементы языка UML. Приведите пример простой диаграммы на данном языке.
11. Тестирование программного обеспечения. Основные понятия. Тестирование при отладке.
12. Тестирование программного обеспечения. Структурное тестирование (белый ящик). Способ тестирования базового пути: потоковый граф. Расчёт цикломатической сложности данного графа (все три формулы). Привести примеры.
13. Тестирование программного обеспечения. Структурное тестирование (белый ящик). Способ тестирования условий. Способ тестирования потоков данных.
14. Тестирование программного обеспечения. Структурное тестирование (белый ящик). Способ тестирования циклов.
15. Тестирование программного обеспечения. Функциональное тестирование (чёрный ящик). Способ разбиения по эквивалентности. Способ диаграмм причин и следствий.
16. Тестирование программного обеспечения. Практические советы по проведению тестирования программных комплексов.
17. Платформа .NET Framework. Общее концептуальное описание данной архитектуры.

4.2. Практические задания:

ВАРИАНТ 1

Задание 1

Оборудование эксплуатируется в течение 5 лет, после этого продается. В начале каждого года можно принять решение – сохранить оборудование или заменить его новым. Стоимость нового оборудования p_0 = 4000 руб. После t лет эксплуатации (1 < t < 5) оборудование можно продать за $g(t) = p_0$ т.руб. (ликвидная стоимость). Затраты на содержание в течение года зависят от возраста t оборудования и равны $r(i) = 600(i + 1)$. Определить оптимальную стратегию эксплуатации оборудования, чтобы суммарные затраты с учетом начальной покупки и заключительной продажи были минимальны.

Задание 2

Дана сеть S(X,U)
x_0 — исток сети; x_7 — сток сети, где $x_0 \in X$; $x_7 \in X$.
Значения пропускных способностей дуг $r_{i,j}$ заданы по направлению ориентации дуг:
i j

от индекса к индексу .

$$r[0,1] = 39;\ r[4,7] = 44;\ r[6,3] = 33;\ r[5,7] = 53;\ r[0,2] = 10;$$
$$r[4,2] = 18;\ r[6,7] = 95;\ r[5,4] = 16;\ r[0,3] = 23;\ r[2,5] = 61;$$
$$r[2,1] = 81;\ r[6,5] = 71;\ r[1,4] = 25;\ r[2,6] = 15;\ r[3,2] = 20$$

0	39	10	23	0	0	0	0
0	0	0	0	25	0	0	0
0	81	0	0	0	61	15	0

0	0	20	0	0	0	0	0
0	0	18	0	0	0	0	44
0	0	0	0	16	0	0	53
0	0	0	33	0	71	0	95
0	0	0	0	0	0	0	0

ВАРИАНТ 2

Задание 1

Компания производит полки для ванных комнат двух размеров - А и В. Агенты по продаже считают, что в неделю на рынке может быть реализовано до 890 полок. Для каждой полки типа А требуется 3 материала, а для полки типа В - 5 материала. Компания может получить до 2000 материала в неделю. Для изготовления одной полки типа А требуется 14 мин. машинного времени, а для изготовления одной полки типа В - 25 мин; машину можно использовать 185 час в неделю. Если прибыль от продажи полок типа А составляет 4 денежных единицы, а от полок типа В - 7 ден. ед., то сколько полок каждого типа следует выпускать в неделю?

Задание 2

Из трех холодильников $A_i, i = 1..4$, вмещающих мороженную рыбу в количествах a_i т, необходимо последнюю доставить в пять магазинов $B_j, j - 1..6$ в количествах b_j т. Стоимости перевозки 1т рыбы из холодильника A_i в магазин B_j заданы в виде матрицы C, 4x6.

Написать математическую модель задачи и спланировать перевозки так, чтобы их общая стоимость была минимальной.

ВАРИАНТ 3

Задание 1

Решить графически систему неравенств:

а) $\begin{cases} x1 + x2 \leq 5 \\ 3x1 - x2 \leq 3 \\ x1 \geq 0,\ x2 \geq 0 \end{cases}$

б) $\begin{cases} x1 + x2 \leq 4 \\ 6x1 + 2x2 \geq 6 \\ 1 + 5x2 \geq 5 \\ x1 \geq 0,\ x2 \geq 0 \end{cases}$

Задание 2

Найти максимальный поток и минимальный разрез в транспортной сети, используя алгоритм Форда–Фалкерсона (алгоритм расстановки пометок) Построить граф приращений. Проверить выполнение условия максимальности построенного полного потока. Источник – вершина 1, сток – вершина 8.

ВАРИАНТ 4

Задание 1

Найти оптимальную стратегию замены оборудования возраста 3 года на период продолжительностью 10 лет, если для каждого года планового периода известны стоимость продукции, производимой с использованием этого оборудования, и эксплуатационные расходы . Известны также остаточная стоимость, не зависящая от возраста оборудования и составляющая 4 ден. ед., и стоимость нового оборудования, равная 18 ден. ед., не меняющаяся в плановом периоде.

t	0	1	2	3	4	5	6	7	8	9	10
$r(t$	31	30	28	28	27	26	26	25	24	24	23
$u(t$	8	9	9	10	10	10	11	12	14	16	18

Задание 2

Производственному объединению из четырех предприятий выделяется банковский кредит в сумме 60 млн. ден. ед. для реконструкции и модернизации производства с целью увеличения выпуска продукции. Значения $z_i(u_i)$ (i = 1,4) дополнительного дохода, получаемого на предприятиях объединения в зависимости от выделенной суммы, приведены в таблице. Распределить выделенный кредит между предприятиями так, чтобы дополнительный доход объединения был максимальным.

Выделенные средства xi млн. ден. ед.	Предприятие			
	№1	№2	№3	№4
	Получаемый доход, млн. ден. ед			
	$z_1(u_i)$	$z_2(u_i)$	$z_3(u_i)$	$z_4(u_i)$
20	9	11	16	13
40	18	19	32	27
60	24	30	40	44

ВАРИАНТ 5

Задание 1

Производственное объединение выделяет четырем входящим в него предприятиям кредит в сумме 100 млн. ден. ед. для расширения производства и увеличения выпуска

продукции. По каждому предприятию известен возможный прирост) $z_1(u_i), i = 1,4$ выпуска продукции (в денежном выражении) в зависимости от выделенной ему суммы i u . Для упрощения вычислений выделяемые суммы кратны 20 млн. ден. ед. При этом предполагаем, что прирост выпуска продукции на i-м предприятии не зависит от суммы средств, вложенных в другие предприятия, а общий прирост выпуска в производственном объединении равен сумме приростов, полученных на каждом предприятии объединения.

Выделяемые средства u_i , млн. ден. ед	Предприятие			
	№1	№2	№3	№4
	Прирост выпуска продукции на предприятиях $z_i(u_i)$ млн. ден. ед			
	$z_1(u_i)$	$z_2(u_i)$	$z_3(u_i)$	$z_4(u_i)$
20	10	12	11	16
40	31	26	36	37
60	42	36	45	46
80	62	54	60	63
100	76	78	77	80

Требуется так распределить кредит между предприятиями, чтобы общий прирост выпуска продукции на производственном объединении был максимальным

Задание 2

Из трех холодильников $A_i, i = 1..3$, вмещающих мороженную рыбу в количествах ai т, необходимо последнюю доставить в пять магазинов $B_j, j = 1..5$ в количествах b_j т. Стоимости перевозки 1т рыбы из холодильника A_i в магазин B_j заданы в виде матрицы G

, 3x5.

Написать математическую модель задачи и спланировать перевозки так, чтобы их общая стоимость была минимальной.

ВАРИАНТ 6

Задание 1

В трех пунктах отправления имеется однородный груз в количестве соответственно. Этот груз нужно доставить пяти заказчикам . Потребности в грузе в каждом пункте известны и равны соответственно. Известны также тарифы перевозки - стоимость перевозки единицы груза из пункта в пункт . Нужно найти такой план перевозок, при котором весь груз из пунктов потребления будет вывезен, потребности всех заказчиков будут удовлетворены, и при этом общая стоимость перевозки всего груза будет наименьшей. Данные в таблице, в клетках которой проставлены элементы матрицы тарифов ; в последнем столбце таблицы указаны значения величин , в последней строке - значения величин .

Заказчики Пункты						

A_1	4	9	2	5	3	23
A_2	4	6	2	1	8	25
A_3	6	2	3	4	5	17
b_i	14	10	16	10	15	

Требуется:

- Составить математическую модель задачи.
- Найти оптимальное решение транспортной задачи методом потенциалов.

Задание 2

Сырьё	Затраты сырья на единицу продукции			Запас сырья
	A1	A2	A3	
I	3,5	7	4,2	1400
II	4	5	8	2000
Прибыль от ед. прод.	1	3	3	

1. Сколько изделий каждого вида необходимо произвести, чтобы получить максимум прибыли?
2. Определить статус каждого вида сырья и его удельную ценность.
3. Определить максимальный интервал изменения запасов каждого вида сырья, в пределах которого структура оптимального плана, т.е. номенклатура выпуска, не изменится.
4. Определить количество выпускаемой продукции и прибыль от выпуска при увеличении запаса одного из дефицитных видов сырья до максимально возможной (в пределах данной номенклатуры выпуска) величины.
5. Определить интервалы изменения прибыли от единицы продукции каждого вида, при которых полученный оптимальный план не изменится.

ВАРИАНТ 7

Задание 1

Решить задачу модифицированным симплекс-методом. Для производства двух видов изделий А и Б используется три типа технологического оборудования. На производство единицы изделия А оборудование первого типа используется a_1=4 часов, оборудование второго типа a_2=8 часов, а оборудование третьего типа a_3=9 часов. На производство единицы изделия Б оборудование первого типа используется b_1=7 часов, оборудование второго типа b_2=3 часов, а оборудование третьего типа b_2=5 часов.

На изготовление этих изделий оборудование первого типа может работать не более чем t_1 =49 часов, оборудование второго типа не более чем t_2=51 часов, оборудование третьего типа не более чем t_3=45 часов.
Прибыль от реализации единицы готового изделия А составляет α=6 рублей, а изделия Б – β=5 рублей.
Составить план производства изделий А и Б, обеспечивающий максимальную прибыль от их реализации.

Задание 2

Составить математическую модель задачи и найти решение системы ограничений:

Чулочно-носочная фирма производит и продает два вида товаров: мужские носки и женские чулки. Фирма получает прибыль в размере 10 руб. от производства и продажи одной пары чулок и в размере 4 руб. от производства и продажи одной пары носков. Производство каждого изделия осуществляется на трех участках. Затраты труда (в часах) на производство одной пары указаны в следующей таблице для каждого участка:

Участок производства	1	2	3
Чулки	0,02	0,03	0,03
Носки	0,01	0,01	0,02

Руководство рассчитало, что в следующем месяце фирма ежедневно будет располагать следующими ресурсами рабочего времени на каждом из участков: 60 ч на участке 1; 70 ч на участке 2 и 100 ч на участке 3. Сколько пар носков и чулок следует производить ежедневно, если фирма хочет максимизировать прибыль?

ВАРИАНТ 8

Задание 1

Составить математическую модель следующей задачи. Имеются три пункта поставки однородного груза А1, А2, А3 и пять пунктов В1, В2, В3, В4, В5 потребления этого груза. На пунктах А1, А2 и А3 находится груз соответственно в количестве 200, 450, 250 тонн. В пункты В1, В2, В3, В4, В5 требуется доставить соответственно 100, 125, 325, 250, 100 тонн груза. Расстояние между пунктами поставки и пунктами потребления приведено в таблице:

Пункты поставки	**Пункты потребления**				
	В1	**В2**	**В3**	**В4**	**В5**
А1	5	8	7	10	3
А2	4	2	2	5	6
А3	7	3	5	9	2

Задание 2

Привести к канонической форме задачу линейного программирования

$$\begin{cases} 2x_1 - x_2 + 3x_3 \leq 5 \\ x_1 + 2x_3 = 9 \\ -x_1 - x_2 \geq 1 \end{cases}$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

$$F = x_1 - x_2 + 3 \rightarrow min$$

ВАРИАНТ 9

Задание 1

Компания производит полки для ванных комнат двух размеров - А и В. Агенты по продаже считают, что в неделю на рынке может быть реализовано до 550 полок. Для каждой полки типа А требуется 2 материала, а для полки типа В - 3 материала. Компания может получить до 1200 материала в неделю. Для изготовления одной полки типа А требуется 12 мин машинного времени, а для изготовления одной полки типа В - 30 мин; машину можно использовать 160 час в неделю. Если прибыль от продажи полок типа А составляет 3 денежных единицы, а от полок типа В - 4 ден. ед., то сколько полок каждого типа следует выпускать в неделю?

Задание 2

Из трех холодильников $A_i, i = 1..3$, вмещающих мороженную рыбу в количествах ai т, необходимо последнюю доставить в пять магазинов $B_j, j = 1..5$ в количествах b_j т. Стоимости перевозки 1т рыбы из холодильника A_i в магазин B_j заданы в виде матрицы G

, 3x5.

Написать математическую модель задачи и спланировать перевозки так, чтобы их общая стоимость была минимальной.

ВАРИАНТ 10

Задание 1

Составить математическую модель следующей задачи. Предположим, что для производства продукции вида А и В можно использовать материал трех сортов. При этом на изготовление единицы изделия вида А расходуется a_1 кг первого сорта, a_2 кг второго сорта и a_3 кг третьего сорта. На изготовление продукции вида В расходуется b_1 кг первого сорта, b_2 кг второго сорта, b_3 кг третьего сорта. На складе фабрики имеется всего материала первого сорта c_1 кг, второго сорта c_2 кг, третьего сорта c_3 кг. От реализации единицы готовой продукции вида А фабрика имеет прибыль вида α руб., а от реализации единицы готовой продукции вида В фабрика имеет прибыль вида β руб. Определить максимальную прибыль от реализации всей продукции видов А и В.

$a_1 = 19, a_2 = 16, a_3 = 19, b_1 = 26, b_2 = 17, b_3 = 8, c_1 = 868, c_2 = 638, c_3$

$\alpha = 5, \beta = 4.$

Задание 2

Имеются три пункта поставки однородного груза А1, А2, А3 и пять пунктов В1, В2, В3, В4, В5 потребления этого груза. На пунктах А1, А2 и А3 находится груз соответственно в количестве , a_1, a_2, a_3 тонн. В пункты В1, В2, В3, В4, В5 требуется доставить соответственно b_1, b_2, b_3, b_4, b_5 тонн груза. Расстояние между пунктами поставки и пунктами потребления приведено в таблице:

Пункты поставки	Пункты потребления				
	B1	B2	B3	B4	B5
A1	D11	D12	D13	D14	D15
A2	D21	D22	D23	D24	D25
A3	D31	D32	D33	D34	D35

Найти такой план закрепления потребителей за поставщиками однородного груза, чтобы общие затраты по перевозкам были минимальными.

a_1=300, a_2=250, a_3=200, b_1=210, b_2=150, b_3=120, b_4=135, b_5 =135.	$D = \begin{matrix} 4 & 8 & 13 & 2 & 7 \\ 9 & 4 & 11 & 9 & 17 \\ 3 & 16 & 10 & 1 & 4 \end{matrix}$

4.3. Перечень вопросов по устному опросу

Раздел 1. Технология разработки программного обеспечения

1. Что такое технология разработки ПО?

2. Что явилось предпосылкой становления дисциплины «Технология разработки ПО»? Что явилось причиной стремительного развития ПО?

3. Чем отличаются программа и программное обеспечение?

4. Достаточно ли при работе над проектом большой программной системы быть компетентным в области вычислительной техники и программировании. Почему?

5. Может ли большая программная система быть отлажена до конца и почему?

6. При каких условиях созданный программный комплекс может быть назван про-

граммным продуктом?

7. Что такое системное программное обеспечение?

8. Что такое инструментарий технологии программирования?

Раздел 2. Инструментальные средства разработки программного обеспечения.

1. Необходимые инструментальные средства разработки программ
2. Часто используемые инструментальные средства разработки программ
3. Специализированные инструментальные средства разработки программ
4. Интегрированные среды разработки
5. Средства разработки программного обеспечения
6. Определение «разработка программ»
7. Три этапа разработки программ
8. Средства проектирования приложений
9. Средства реализации программного кода

Раздел 3. Математическое моделирование

1. Что понимается под объектом моделирования?
2. Что такое гипотеза в моделировании?
3. Дайте определение модели.
4. Что такое математическая модель?
5. Приведите пример аналогии в физических процессах.
6. Дайте классификацию процессов как объектов моделирования.
7. Чем отличаются стохастические процессы от детерминированных?
8. Опишите постановку задачи моделирования в общем виде.
9. Дайте общую классификацию математических моделей.
10. Какова структура модели математического программирования?
11. Что понимают под структурно-параметрическим описанием объекта моделирования?
12. В чем состоит различие между линейными и нелинейными моделями?
13. В каких случаях используется корреляционный коэффициент, а в каких − корреляционное отношение как критерий адекватности модели?
14. Дайте классификацию моделируемых процессов по характеру их протекания.
15. Перечислите основные этапы построения математической модели.
16. Опишите метод активного и пассивного эксперимента. Чем они отличаются?
17. Какой математический аппарат используется при синтезе математических моделей детерминированных процессов?
18. Какие системы относят к системам с распределенными параметрами?
19. Что такое сплошная среда?
20. Каким уравнением в частных производных моделируется процесс теплопереноса?
21. В чем состоит идея метода аналогий?
22. Опишите экспериментально-статистический метод моделирования.

МДК.02.01

Практическая работа 1

Техническое задание на проектирование программы

Цель работы: ознакомиться с правилами написания технического задания.

ГОСТ 2.610-2006. Настоящий стандарт устанавливает порядок построения и оформления технического задания на разработку программы или программного изделия для вычислительных машин, комплексов и систем независимо от их назначения и области применения.

Общие сведения

1. Требования к оформлению

1.1. Техническое задание оформляют в соответствии с ГОСТ 2.610-2006 на листах формата А4 и А3, как правило, без заполнения полей листа. Номера листов (страниц) проставляют в верхней части листа над текстом.

1.2. Лист утверждения и титульный лист оформляют в соответствии с ГОСТ 2.610-2006. Информационную часть (аннотацию и содержание), лист регистрации изменений допускается в документ не включать.

1.3. Для внесения изменений и дополнений в техническое задание на последующих стадиях разработки программы или программного изделия к нему выпускают дополнение. Согласование и утверждение дополнения к техническому заданию проводят в том же порядке, который установлен для технического задания.

1.4. Техническое задание должно содержать следующие разделы:

- название программы и область применения;
- основание для разработки;
- назначение разработки;
- технические требования кпрограмме или программному изделию;
- технико–экономические показатели;
- стадии и этапы разработки:
- порядок контроля и приемки;
- приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

2. Содержание разделов

2.1. В разделе «Наименование и область применения» указывают наименование, краткую характеристику области применения программы или программного изделия и объекта, в котором используют программу или программное изделие.

2.2. В разделе «Основание для разработки» должны быть указаны:

- документ (документы), на основании которых ведется разработка;
- организация, утвердившая этот документ, и дата его утверждения;
- наименование и (или) условное обозначение темы разработки.

2.3. В разделе «Назначение разработки» должно быть указано функциональное и эксплуатационное назначение программы или программного изделия.

2.4. Раздел «Технические требования к программе или программному изделию» должен содержать следующие подразделы:

- требования к функциональным характеристикам;
- требования к надежности;
- условия эксплуатации;
- требования к составу и параметрам технических средств;
- требования к информационной и программной совместимости;
- требования к маркировке и упаковке;
- требования к транспортированию и хранению;
- специальные требования.

2.4.1. В подразделе «Требования к функциональным характеристикам» должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т.п.

2.4.2. В подразделе «Требования к надежности» должны быть указаны требования к обеспечению надежного функционирования (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т.п.).

2.4.3. В подразделе «Условия эксплуатации» должны быть указаны условия эксплуатации (температура окружающего воздуха, относительная влажность для выбранных типов носителей данных), при которых должны обеспечиваться заданные характеристики, а также вид обслуживания, необходимое количество и квалификация персонала.

2.4.4. В подразделе «Требования к составу и параметрам технических средств» указывают необходимый состав технических средств с указанием их технических характеристик.

2.4.5. В подразделе «Требования к информационной и программной совместимости» должны быть указаны требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования. При необходимости должна обеспечиваться защита информации и программ.

2.4.6. В подразделе «Требования к маркировке и упаковке» в общем случае указывают требования к маркировке программного изделия, варианты и способы упаковки.

2.4.7. В подразделе «Требования к транспортированию и хранению» должны быть указаны для

программного изделия условия транспортирования, места хранения, условия хранения, условия складирования, сроки хранения в различных условиях.

2.5. В разделе «Технико–экономические показатели» должны быть указаны: ориентировочная экономическая эффективность предполагаемая годовая потребность, экономические преимущества разработки по сравнению с лучшими отечественными и зарубежными образцами или аналогами.

2.6. В разделе «Стадии и этапы разработки* устанавливают необходимые стадии разработки, этапы и содержание работ (перечень программных документов, которые должны быть разработаны, согласованы и утверждены), а также, как правило, сроки разработки и определяют исполнителей.

2.7. В разделе «Порядок контроля и приемки» должны быть указаны виды испытаний и общие требования к приемке работы.

2.8. В приложениях к техническому заданию при необходимости приводят:

- перечень научно-исследовательских и других работ, обосновывающих разработку;
- схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые могут быть использованы при разработке;
- другие источники разработки.

Пример выполнения задания

1. Введение

Работа выполняется в рамках проекта «Автоматизированная система оперативно-диспетчерского управления электро-, теплоснабжением корпусов института».

2. Основание для разработки

2.1. Основанием для данной работы служит договор № ___ от _____ 20___ г.

2.2. Наименование работы:

2.3. «Модуль автоматизированной системы оперативно-диспетчерского управления теплоснабжением корпусов института».

2.4. Исполнители: ОАО «Лаборатория создания программного обеспечения.

2.5. Соисполнители: нет.

3. Назначение разработки

Создание модуля для контроля и оперативной корректировки состояния основных параметров теплообеспечения корпусов Московского института.

4.Технические требования

4.1. Требования к функциональным характеристикам

4.1.2. Состав выполняемых функций. Разрабатываемое ПО должно обеспечивать:

– сбор и анализ информации о расходовании тепла, горячей и холодной воды по данным теплосчетчиков SA-94 на всех тепловых выходах;

– сбор и анализ информации с устройств управления системами воздушного отопления и кондиционирования типа РТ1 и РТ2 (разработки кафедры СММЭ и ТЦ);

– предварительный анализ информации на предмет нахождения параметров в допустимых пределах и сигнализирование при выходе параметров за пределы допуска;

– выдачу рекомендаций по дальнейшей работе;

– отображение текущего состояния по набору параметров –циклически постоянно (режим работы круглосуточный), при сохранении периодичности контроля прочих параметров;

– визуализацию информации по расходу теплоносителя;

– текущую, аналогично показаниям счетчиков;

– с накоплением за прошедшие сутки, неделю, месяц – в виде почасового графика для информации за сутки и неделю;

– суточный расход – для информации за месяц.

Для устройств управления приточной вентиляцией текущая информация должна содержать номер приточной системы и все параметры, выдаваемые на собственный индикатор.

По отдельному запросу осуществляются внутренние настройки.

В конце отчетного периода система должна архивировать данные.

4.1.2. Организация входных и выходных данных

Исходные данные в систему поступают в виде значений с датчиков, установленных в помещениях института. Эти значения отображаются на компьютере диспетчера. После анализа поступившей информации оператор диспетчерского пункта устанавливает необходимые параметры для устройств, регулирующих отопление и вентиляцию в помещениях. Возможна также автоматическая установка

некоторых параметров для устройств регулирования.

Основной режим использования системы – ежедневная работа.

4.2. Требования к надежности

Для обеспечения надежности необходимо проверять корректность получаемых данных с датчиков.

4.3. Условия эксплуатации и требования к составу и параметрам технических средств

Для работы системы должен быть выделен ответственный оператор.Требования к составу и параметрам технических средств уточняются на этапе эскизного проектирования системы.

4.4. Требования к информационной и программной совместимости

Программа должна работать на платформах Windows.

4.5. Требования к транспортировке и хранению

Программа поставляется на лазерном носителе информации. Программная документация поставляется в электронном и печатном виде.

4.6. Специальные требования.Программное обеспечение должно иметь дружественный интерфейс, рассчитанный на пользователя (в плане компьютерной грамотности) средней квалификации.Ввиду объемности проекта задачи предполагается решать поэтапно, при этом модули ПО, созданные в разное время, должны предполагать возможность наращивания системы и быть совместимы друг с другом, поэтому документация на принятое эксплуатационное ПО должна содержать полную информацию, необходимую для работы программистов. Язык программирования выбирает исполнитель, он должен обеспечивать возможность интеграции программного обеспечения с некоторыми видами периферийного оборудования (например, счетчик SA-94 и т.п.).

5. Требования к программной документации

Основными документами, регламентирующими разработку будущих программ, должны быть документы Единой системы программной документации (ЕСПД); руководство пользователя, руководство администратора, описание применения.

6. Технико-экономические показатели

Эффективность системы определяется удобством использования системы для контроля и управления основными параметрами теплообеспечения помещений Московского института, а также экономической выгодой, полученной от внедрения аппаратно–программного комплекса.

7. Порядок контроля и приемки

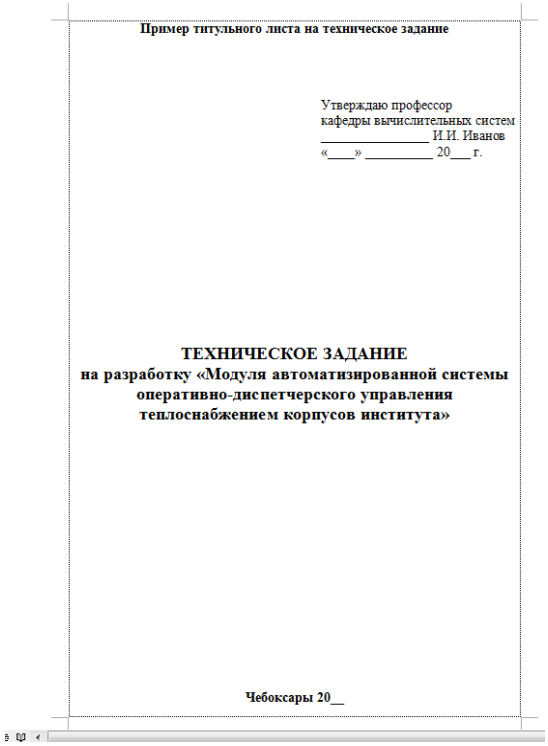
После передачи исполнителем отдельного функционального модуля программы заказчику последний имеет право тестировать модуль в течение семи дней. После тестирования заказчик должен принять работу по данному этапу или в письменном виде изложить причину отказа от принятия. В случае обоснованного отказа исполнитель обязуется доработать модуль.

8. Календарный план работ

Название этапа	Сроки этапа	Чем закачивается этап
1. Изучение предметной области. Проектирование системы. Разработка предложений по реализации системы.	01.02.20__ – 28.02.20__	Предложения по работе системы. Акт сдачи–приёмки
2. Разработка программного модуля по сбору и анализу информации со счётчиков и устройств управления. Внедрение системы для одного из корпусов ЧГУ.	01.03.20__ – 31.08.20__	Программный комплекс
3. Тестирование и отладка модуля. Внедрение системы во всех корпусах ЧГУ.	01.09.20__ – 30.12.20__	Готовая система контроля теплоснабжения, установленная в диспетчерском пункте Программная документация.

		Акт сдачи–приёма работ

Руководитель работ Сидоров А.В.


Пример титульного листа на техническое задание

Утверждаю профессор
кафедры вычислительных систем
_____ И.И. Иванов
«____» _____ 20___ г.

ТЕХНИЧЕСКОЕ ЗАДАНИЕ
на разработку «Модуля автоматизированной системы оперативно-диспетчерского управления теплоснабжением корпусов института»

Чебоксары 20__

Рис. 1.1. Пример титульного листа на техническое задание

Индивидуальные задания

Ниже приведены 15 вариантов программных продуктов. По указанию преподавателя выберите свое индивидуальное задание. Разработайте техническое задание на создание программного продукта по всем требованиям.

1. Разработка программного комплекса «Автотранспорт».
2. Разработка программного комплекса «Деканат института».
3. Разработка программного комплекса «Обслуживание банкомата».
4. Разработка программного комплекса «Управление гостиницей».
5. Разработка программного комплекса «Выдача кредитов в банке».
6. Разработка программного комплекса «Строительная фирма».
7. Разработка программного комплекса «Управление библиотечным фондом».
8. Разработка программного комплекса «АРМ работника склада».
9. Разработка программного комплекса «АРМ администратора ателье по ремонту оргтехники».
10. Разработка программного комплекса «АРМ администратора автосалона».
11. Разработка программного комплекса «АРМ администратора ресторана».
12. Разработка программного комплекса «АРМ сотрудника ЖЭСа».
13. Разработка программного комплекса «АРМ администратора аэропорта».
14. Разработка программного комплекса «АРМ работника отдела кадров».
15. Разработка программного комплекса «АРМ администратора спорткомплекса».

Практическая работа 2. **Стадия разработки программного обеспечения.«Эскизный проект»**

Цель работы: научиться создавать формальные модели и на их основе определять спецификации разрабатываемого программного обеспечения.

Подготовка к лабораторной работе

Ознакомиться с лекционным материалом по теме «Этапы разработки программного обеспечения. Анализ требований и определение спецификаций программного обеспечения» учебной дисциплины «Технология разработки программного обеспечения».

Теоретическая часть

Разработка спецификаций программного обеспечения начинается с анализа требований к

нему. В результате анализа получают спецификации разрабатываемого программного обеспечения, строят общую модель его взаимодействия с пользователем или другими программами и конкретизируют его основные функции.

При структурном подходе к программированию на этапе анализа и определения спецификаций разрабатывают три типа моделей: модели функций, модели данных и модели потоков данных. Поскольку разные модели описывают проектируемое программное обеспечение с разных сторон, рекомендуется использовать сразу несколько моделей, разрабатываемых в виде диаграмм, и пояснять их текстовыми описаниями, словарями и т. п.

Структурный анализ предполагает использование следующих видов моделей:

- диаграмм потоков данных (DFD – Data Flow Diagrams), описывающих взаимодействие источников и потребителей информации через процессы, которые должны быть реализованы в системе;
- диаграмм «сущность – связь» (ERD – Entity-Relationship Diagrams), описывающих базы данных разрабатываемой системы;
- диаграмм переходов состояний (STD – State Transition Diagrams), характеризующих поведение системы во времени;
- функциональных диаграмм (методика SADT);
- спецификаций процессов;
- словаря терминов.

Спецификации процессов обычно представляют в виде краткого текстового описания, схем алгоритмов, псевдокодов, Flow-форм или диаграмм Насси – Шнейдермана.

Словарь терминов представляет собой краткое описание основных понятий, используемых при составлении спецификаций. Он должен включать определение основных понятий предметной области, описание структур элементов данных, их типов и форматов, а также всех сокращений и условных обозначений.

С помощью диаграмм переходов состояний можно моделировать последующее функционирование системы на основе ее предыдущего и текущего функционирования. Моделируемая система в любой заданный момент времени находится точно в одном из конечного множества состояний. С течением времени она может изменить свое состояние, при этом переходы между состояниями должны быть точно определены.

Функциональные диаграммы отражают взаимосвязи функций разрабатываемого программного обеспечения.

Они создаются на ранних этапах проектирования систем, для того чтобы помочь проектировщику выявить основные функции и составные части проектируемой системы и, по возможности, обнаружить и устранить существенные ошибки. Для создания функциональных диаграмм предлагается использовать методологию SADT. Диаграммы потоков данных.

Для описания потоков информации в системе применяются диаграммы потоков данных (DFD – Data flow diagrams). DFD позволяет описать требуемое поведение системы в виде совокупности процессов, взаимодействующих посредством связывающих их потоков данных. DFD показывает, как каждый из процессов преобразует свои входные потоки данных в выходные потоки данных и как процессы взаимодействуют между собой.

Диаграмма «сущность – связь» – инструмент разработки моделей данных, обеспечивающий стандартный способ определения данных и отношений между ними.

Она включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области. Такая диаграмма не слишком детализирована, в нее включаются основные сущности и связи между ними, которые удовлетворяют требованиям, предъявляемым к ИС.

Порядок выполнения работы

1. На основе технического задания из лабораторной работы №1 выполнить анализ функциональных и эксплуатационных требований к программному продукту.
2. Определить основные технические решения (выбор языка программирования, структура программного продукта, состав функций ПП, режимы функционирования) и занести результаты в документ, называемый «Эскизным проектом».
3. Определить диаграммы потоков данных для решаемой задачи.
4. Определить диаграммы «сущность – связь», если программный продукт содержит базу данных.
5. Определить функциональные диаграммы.
6. Определить диаграммы переходов состояний.
7. Определить спецификации процессов.

8. Добавить словарь терминов.

9. Оформить результаты, используя MS Office в виде эскизного проекта.

10. Сдать и защитить работу.

Пояснительная записка к эскизному проекту

Данный документ является эскизным проектом на создание Системы Управления Базой Данных для Библиотечного Фонда Российской Федерации (СУБД «Библиотека»).

Перечень организаций, участвующих в разработке системы, сроки и стадии разработки, а также ее цели и назначение указаны в техническом задании на создание информационной системы.

Рассмотрим пример эскизного проекта, титульный лист которого приведен на рис. 2.1.

Пример эскизного проекта

УТВЕРЖДАЮ

Руководитель (заказчика ИС)
Личная подпись _____ Расшифровка подписи _____
Печать
Дата «___»_____ 2004 г.

УТВЕРЖДАЮ

Руководитель (разработчика ИС)
Личная подпись_____ Расшифровка подписи_____
Печать
Дата «___»_____ 2004 г.

Эскизный проект на создание
информационной системы

Система Управления Базой Данных
(наименование вида ИС)

БИБЛИОТЕЧНЫЙ ФОНД РОССИЙСКОЙ ФЕДЕРАЦИИ
(наименование объекта информатизации)

СУБД «Библиотека»
(сокращенное наименование ИС)

На 8 листах

Действует с «___»_____ 2004 г.

Рис. 2.1. Титульный лист эскизного проекта

Ведомость эскизного проекта

Основные технические решения

Решения по структуре системы СУБД «Библиотека» будет представлять собой персональную систему управления локальной базой данных, работающей на одном компьютере.

Система будет управлять реляционной базой данных, представляющей собой набор связанных между собой таблиц в формате Paradox, доступ к которым осуществляется с помощью ключей или индексов. Сведения в одной таблице могут отражать сведения из другой, и при изменении сведений в первой таблице эти изменения немедленно отображаются во второй. Таким образом будет достигнута непротиворечивость данных.

Общая структура базы данных:

1. Анкеты организации, которые зарегистрированы в данном ПФ:

• тип предприятия (российская организация, физическое лицо, иностранная организация, обособленное подразделение);

• вид предприятия;

• регистрационный номер работодателя в ПФР.

2. Свидетельство: серия, номер.
3. Дата выдачи свидетельства.
4. ИНН.
5. КПП.
6. Наименование.
7. Юридический адрес.
8. Адрес постоянно действующего органа (при отличии от юридического).
9. Анкеты сотрудников этих организаций.
10. Сведения о стаже сотрудников этих организаций.
11. Таблица периодов работы со следующей структурой:

• Начало периода (дата).

• Конец периода (дата).

• Вид деятельности (работа, служба соцстрах, уход-дети, безр, реабилит, уход-инвд, профзаб, пересмотр).

• Наименование организации.

• Должность.

• Территориальные условия.

Работа системы СУБД «Библиотека» будет функционировать в однопользовательском режиме, а также будет способна:

• просматривать записи базы данных (в том числе и при помощи фильтров);

• добавлять новые записи;

• удалять записи;

• при входе в систему будет запрашиваться пароль.

Автоматизированная система должна выполнять следующие функции:

• сделать запись о пенсионном удостоверении;

• удалить информацию о пенсионном удостоверении;

• выдать справку о всех пенсионных удостоверениях;

• зарегестрировать новое предприятие в ПФ РФ;

• удалить предприятие из базы данных;

• выдать справку обо всех предприятиях, зарегистрированных в ПФ РФ;

• подсчитать пенсию для работников предприятий на основании стажа;

• выдать справку о пенсионных накоплениях работника.

Для реализации АС будет использоваться среда программирования Boland Delphi 7.0 и язык программирования Object Pascal.

Для подсчета пенсии будет использоваться следующий алгоритм. Вначале определяется стажевый коэффициент пенсионера. Он полагается равным 0,55 за общий трудовой стаж до текущей даты не менее 25 лет мужчинам и 20 лет женщинам. За каждый полный год стажа больше указанного стажевый коэффициент увеличивается на 0,01, но не более чем на 0,20.

Затем определяется отношение заработка пенсионера к среднемесячной заработной плате в стране. Этот заработок может быть взят за этот отсчетный период или за любые 60 месяцев работы подряд, или тот, из которого была исчислена пенсия на момент реформы. Среднемесячная зарплата в стране берется за тот же самый период.

Отношение заработков учитывается в размере не свыше 1,2. Для пенсионеров, проживающих на Крайнем Севере, учитываемое соотношение выше: от 1,4 до 1,9 в зависимости от установленного в централизованном порядке районного коэффициента к зарплате.

Затем стажевый коэффициент умножается на соотношение заработков и на 1671 руб. – утвержденную для расчетов среднемесячную зарплату в стране за III квартал 2001 г. Это и будет пересчитанный размер трудовой пенсии по новому законодательству в обычном случае. Если он оказался менее 660 руб., то размер пенсии повышается до этого гарантированного минимума.

Если пенсионер является инвалидом I группы или достиг к 1 января 2002 г. возраста 80 лет и более, рассчитанный в этом порядке размер пенсии по старости увеличивается на 450 руб.

Если у пенсионера имеются лица, находящиеся на его иждивении, то рассчитанный размер

пенсии увеличивается на 150 руб. на каждого иждивенца, но не более чем на трех в общей сложности.

Источники разработки

Данный документ разрабатывался на основании ГОСТ34.698 – 90 на написание ТЗ на автоматизированные системы управления от 01.01.1992 г.

Практическая работа 3. **Стадия разработки программного обеспечения. «Технический проект»**

Цель работы: изучить вопросы проектирования программного обеспечения.

Подготовка к работе

Ознакомиться с лекционным материалом по теме «Этапы разработки программного обеспечения. Проектирование программного обеспечения» учебной дисциплины «Технология разработки программного обеспечения».

Теоретическая часть

Технический проект – образ намеченного к созданию объекта, представленный в виде его описания, схем, чертежей, расчетов, обоснований, числовых показателей.

Цель технического проекта – определение основных методов, используемых при создании информационной системы, и окончательное определение ее сметной стоимости.

Техническое проектирование подсистем осуществляется в соответствии с утвержденным техническим заданием.

Технический проект программной системы подробно описывает:

- выполняемые функции и варианты их использования;
- соответствующие им документы;
- структуры обрабатываемых баз данных;
- взаимосвязи данных;
- алгоритмы их обработки.

Технический проект должен включать данные об объемах и интенсивности потоков обрабатываемой информации, количестве пользователей программной системы, характеристиках оборудования и программного обеспечения, взаимодействующего с проектируемым программным продуктом.

При разработке технического проекта оформляются:

- ведомость технического проекта. Общая информация по проекту;
- пояснительная записка к техническому проекту. Вводная информация, позволяющая ее потребителю быстро освоить данные по конкретному проекту;
- описание систем классификации и кодирования;
- перечень входных данных (документов). Перечень информации, которая используется как входящий поток и служит источником накопления;
- перечень выходных данных (документов). Перечень информации, которая используется для анализа накопленных данных;
- описание используемого программного обеспечения.
- перечень программного обеспечения и СУБД, которые планируется использовать для создания информационной системы;
- описание используемых технических средств. Перечень аппаратных средств, на которых планируется работа проектируемого программного продукта;
- проектная оценка надежности системы. Экспертная оценка надежности с выявлением наиболее благополучных участков программной системы и ее узких мест;
- ведомость оборудования и материалов. Перечень оборудования и материалов, которые потребуются в ходе реализации проекта.

Структурная схема

Структурной называют схему, отражающую состав и взаимодействие по управлению частями разрабатываемого программного обеспечения.

Структурная схема определяется архитектурой разрабатываемого ПО:

- структуры обрабатываемых баз данных;
- взаимосвязи данных;
- алгоритмы их обработки.

Функциональная схема

Функциональная схема – это схема взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых фай-

лов и устройств.

Порядок выполнения работы

1. На основе технического задания из лабораторной работы №1 и спецификаций из лабораторной работы №2 разработать уточненные алгоритмы программ, составляющих заданный программный модуль.

2. На основе уточненных и доработанных алгоритмов разработать структурную схему программного продукта.

3. Разработать функциональную схему программного продукта.

4. Оформить результаты, используя MS Office в виде технического проекта.

5. Сдать и защитить работу.

Практическая работа 4. **Использование объектно-ориентированного программирования (ООП) для создания качественного программного обеспечения.**

Цель работы: познакомиться с принципами объектно-ориентированного программирования в среде Delphi. Написать и отладить программу линейного алгоритма, в которой присутствовали бы некоторые критерии и примитивы качественного программного обеспечения.

Общие сведения

Класс – абстрактный тип данных, включающий свойства объекта (поля) и методы. Класс позволяет упростить процесс программирования, так как человеку проще представлять любой объект из реальности, обладающий некоторыми характеристиками (свойствами) и действиями, которые может совершать объект или которые можно совершать над ним.

Класс – это тип данных. Объект класса – переменная типа «класс». Из определения класса следует первое свойство ООП –инкапсуляция. Инкапсуляция данных означает, что они являются не глобальными – доступными всей программе, а локальными – доступными только малой ее части. Инкапсуляция автоматически подразумевает защиту данных. Для этого в структуре class используется спецификатор раздела private, содержащий данные и методы, доступные только для самого класса. Если данные и методы содержатся в разделе public, они доступны извне класса. Раздел protected содержит данные и методы, доступные из класса и любого его *производного класса.*

Интегрированная среда разработки Delphi

Среда Delphi визуально реализуется в виде нескольких одновременно раскрытых на экране монитора окон. Количество, расположение, размер и вид окон может меняться в зависимости от текущих нужд, что значительно повышает производительность работы. При запуске Delphi можно увидеть на экране картинку (рис. 4.1).

*Главное окно*всегда присутствует на экране и предназначено для управления процессом создания программы. Основное меню содержит все необходимые средства для управления проектом. Пиктограммы облегчают доступ к наиболее часто применяемым командам основного меню. Через меню компонентов осуществляется доступ к набору стандартных сервисных программ среды Delphi, которые описывают некоторый визуальный элемент (компонент), помещенный программистом в окно формы. Каждый компонент имеет определенный набор свойств (параметров), которые программист может задавать, например цвет, заголовок окна, надпись на кнопке, размер и тип шрифта и др.

Окно инспектора объектов(вызывается спомощью клавиши F11) предназначено для изменения свойств выбранных компонентов и состоит из двух страниц. Страница Properties(Свойства) предназначена для изменения необходимых свойств компонента, страница Events (События) – для определения реакции компонента на то или иное событие (например, нажатие определенной клавиши или щелчок мышью по кнопке).

*Окно формы*представляет собой проект Windows–окна программы. В это окно в процессе написания программы помещаются необходимые компоненты. Причем при выполнении программы помещенные компоненты будут иметь тот же вид, что и на этапе проектирования.

*Окно текста*программы предназначено для просмотра, написания и корректирования текста программы. В системе Delphi используется язык программирования Object Pascal. При первоначальной загрузке в окне текста программы находится текст, содержащий минимальный набор операторов для нормального функционирования пустой формы в качестве Windows-окна. При помещении некоторого компонента в окно формы текст программы автоматически дополняется описанием необходимых для его работы библиотек стандартных программ (раздел uses) и типов переменных (раздел type).

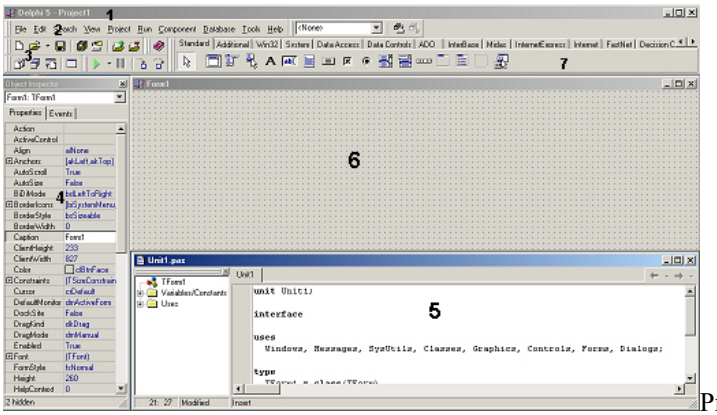


Рис. 4.1. Внешний вид окна Delphi.

1 – главное окно; 2 – основное меню; 3 – пиктограммы основного меню; 4 – окно инспектора объектов; 5 – окно текста программы; 6 – окно пустой формы, 7 – меню компонентов

Программа в среде Delphi составляется как описание алгоритмов, которые необходимо выполнить, если возникает определенное событие, связанное с формой (например, щелчок мыши на кнопке – событие OnClick, создание формы – OnCreate). Для каждого обрабатываемого в форме события с помощью страницы Events инспектора объектов в тексте программы организуется процедура (procedure), в которой записывается на языке Object Pascal требуемый алгоритм.

Переключение между окном формы и окном текста программы осуществляется с помощью клавиши F12.

Меню и команды Delphi

Для того чтобы выдать команду в среде Delphi, можно воспользоваться тремя основными способами:

- с помощью меню;
- с помощью полоски SpeedBar (инструментальной линейки);
- с помощью SpeedMenu (одного из локальных меню, которое активизируется при нажатии правой кнопки мыши).

Меню File

Команды выпадающего меню File можно использовать для работы как с проектами, так и файлами исходного кода.

Ккомандам, работающимспроектами, относятся New, New Application, Open, Reopen, Save Project As, Save All, Close All, Add to Project и Remove from Project. Сфайламиисходногокодаработаютткоманды New, New Form, New Data Module, Open, Reopen, Save As, Save, Close и Print. Основной командой является File/New, которую можно использовать для вызова экспертов, для начала работы с новым приложением, наследования формы из уже существующей и т.д. Для того чтобы открыть проект или файл исходного кода, с которыми вы работали последний раз, используйте команду File/Reopen.

Меню Edit

Стандартные возможности меню Edit применимы как к тексту, так и к компонентам формы. Можно копировать и вставлять тот или иной текст в редакторе, копировать и вставлять компоненты в одной форме или из одной формы в другую. Также можно копировать и вставлять компоненты в другое групповое окно той же формы, например в панель или блок группы; копировать компоненты из формы в редактор, и наоборот. Delphi помещает компоненты в буфер обмена, преобразуя их в текстовое описание. Можно соответствующим образом отредактировать этот текст, а затем вставить его обратно в форму в виде нового компонента. Можно выбрать несколько компонентов и скопировать их как в другую форму, так и в текстовый редактор. Это может пригодиться, когда вам придется работать с рядом схожих компонентов. Вы сможете скопировать один компонент в редактор, размножить его нужное число раз, а затем вставить назад в форму целую группу.

Меню Search

Если вы выберете команду Incremental Search, то вместо того чтобы показать диалоговое окно, где вводится образец для поиска, Delphi переходит в редактор. Когда вы введете первую букву, редактор перейдет к первому слову, которое начинается с этой буквы. Продолжайте набор букв и, курсор будет последовательно переходить к словам, в начале которых будут стоять введенные символы. Эта команда очень эффективна и чрезвычайно быстра. Команда Browse Symbol вызывает Object Browser – инструмент, который можно использовать для просмотра многих деталей при исследовании откомпилированной программы.

Меню View

Большинство команд меню View применяются для отображения какого–либо окна среды Delphi, например Project Manager, Breakpoints List или Components List. Эти окна не связаны друг с другом. Команда Toggle Form/Unit используется для перехода от формы, над которой вы работаете, к ее исходному коду, и обратно. Команда New edit window создает дубликат окна редактирования и его содержимого. В Delphi это единственный способ просмотреть два файла рядом друг с другом, поскольку редактор для показа нескольких загруженных файлов использует ярлыки. После дублирования окна редактирования могут содержать разные файлы. Последние две команды меню View можно использовать для удаления с экрана полоски SpeedBar и палитры Components, хотя при этом среда Delphi становится менее удобнее для пользователя. Команда Build All заставляет Delphi откомпилировать каждый исходный файл проекта, даже если после последней трансляции он не был изменен. Для проверки написанного кода без создания программы можно использовать команду Syntax Check. Команда Information дает некоторые подробности о последней выполненной трансляции. Команда Options применяется для установки опций проекта: опций компилятора и редактора связей, опций объекта приложения и т.д.

Меню Run

Меню Run можно назвать Debug (отладка), так как большинство команд в нем относится к отладке, включая саму команду Run. Программа, запускаемая внутри среды Delphi, выполняется в ее интегрированном отладчике (если не отключена соответствующая опция). Для быстрого запуска приложения используется клавиша F9. Остальные команды применяются в процессе отладки для пошагового выполнения программы, установки точек прерывания, просмотра значений переменных и объектов и т.п.

Меню Component

Команды меню Component можно использовать для написания компонентов, добавления их в библиотеку, а также для конфигурирования библиотеки или палитры компонентов.

Меню Tools

Меню Tools содержит список нескольких внешних программ и инструментальных средств. Команда Tools позволяет сконфигурировать это выпадающее меню и добавить в него новые внешние средства. Меню Tools также включает команду для настройки репозитория и команду Options, которая конфигурирует всю среду разработки Delphi.

Работа с формами

Проектирование форм – ядро визуальной разработки в среде Delphi. Каждый помещаемый в форму компонент или любое задаваемое свойство сохраняется в файле, описывающем форму (DFM–файл), а также оказывает некоторое влияние на исходный текст, связанный с формой (PAS–файл). Можно начать новый пустой проект, создав пустую форму или начать с существующей формы (используя различные доступные шаблоны), или добавить в проект новые формы. Проект (приложение) может иметь любое число форм.

При работе с формой можно обрабатывать ее свойства, свойства одного из ее компонентов или нескольких компонентов одновременно. Для того чтобы выбрать форму или компонент, можно просто щелкнуть по нему мышью или воспользоваться Object Selector (комбинированный список в Object Inspector), где всегда отображены имя и тип выбранного элемента. Для выбора нескольких компонентов можно или нажать клавишу Shift и щелкать по компонентам левой кнопкой мыши, или отбуксировать в форме рамку выбора.

SpeedMenu формы содержит ряд полезных команд. Для изменения относительного расположения компонентов одного вида можно использовать команды Bring To Front и Send To Back. Командой Revert To Inherited можно воспользоваться, чтобы в унаследованной форме установить те значения свойств выбранного компонента, которые были у них в родительской форме. При выборе сразу нескольких компонентов вы можете выровнять их или изменить их размеры.

С помощью SpeedMenu можно также открыть два диалоговых окна, в которых устанавливается порядок обхода визуальных управляющих элементов и порядок создания невизуальных управляющих элементов. Команда Add To Repository добавляет текущую форму в список форм, доступных для использования в других проектах.

Для установки положения компонента кроме применения мыши существуют еще два способа:

- установка значений для свойств Top и Left;
- использование клавиш курсора при нажатой клавише Ctrl.

Метод Ctrl+клавиша курсора особенно удобен при тонкой подстройке положения элемента. Точно также, нажимая клавиши курсора при нажатой клавише Shift, можно подстроить размер компонента.

Палитра компонентов

Для того чтобы добавить в текущую форму новый компонент, можно щелкнуть на одной из страниц палитры Components, а затем щелкнуть в форме, чтобы поместить новый элемент. Причем в форме можно или буксировать мышь с нажатой левой кнопкой, чтобы установить сразу и размер, и положение компонента, или просто щелкнуть один раз, позволяя Delphi установить размер по умолчанию.

Каждая страница палитры содержит компоненты, которые обозначены пиктограммами и именами, появляющимися в виде подсказки. Эти имена являются официальными названиями компонентов. В действительности это названия классов, описывающих компоненты без первой буквы T (например, если класс называется Tbutton, имя будет Button). Если необходимо поместить в форму несколько компонентов одного и того же вида, то при выборе компонента щелчком в палитре удерживайте нажатой клавишу Shift. Затем при каждом щелчке в форме Delphi будет вставлять новый компонент выбранного вида. Чтобы остановить эту операцию, просто щелкните по стандартному селектору (пиктограмма стрелки) слева от палитры Components.

Структура программ Delphi

Программа в Delphi состоит из файла проекта (файл с расширением .dph),одного или нескольких файлов исходного текста (с расширением .pas),файлов с описанием окон формы (с расширением .dfm).

В файле проектанаходится информация о модулях, составляющих данный проект. Файл проекта автоматически создается и редактируется средой Delphi и не предназначен для редактирования.

Файл исходного текста – программный модуль (Unit), предназначенный для размещения текстов программ. В этом файле программист размещает текст программы, написанный на языке PASCAL.

В разделе объявлений описываются типы, переменные, заголовки процедур и функции, которые могут быть использованы другими модулями через операторов подключения библиотек (Uses). В разделе реализации располагаются тела процедур и функций, описанных в разделе объявлений, а также типы переменных, процедуры и функции, которые будут функционировать только в пределах данного модуля. Раздел инициализации используется редко. Модуль имеет следующую структуру:

```
unitUnit1;
interface
// Раздел объявлений
implementation
// Раздел реализации
begin
// Раздел инициализации
end.
```

При компиляции программы Delphi создает файл с расширением .dcu, содержащий результат перевода в машинные коды содержимого файлов с расширением *.pas* и *.dfm.* Компоновщик преобразует файлы с расширением .dcuв единый загружаемый файл с расширением .exe. В файлах, имеющих расширение *.~df, .~dp, .~pa,* хранятся резервные копии файлов с образом формы, проекта и исходного текста соответственно.

Пример написания программы линейного алгоритма

Задание: составить программу вычисления для заданных значений *x, y, z* арифметического выражения:

$$u = tg^2(x+y) - e^{y-z} \cdot \cos x^2 + \sin z^2$$

Настройка формы

Пустая форма в правом верхнем углу имеет кнопки управления, которые предназначены для свертывания формы в пиктограмму  , разворачивания формы на весь экран  и возвращения к исходному размеру и для закрытия формы  . С помощью мыши, «захватывая» одну из кромок формы или выделенную строку заголовка, отрегулируйте нужные размеры формы и ее положение на экране (рис. 4.2).

Новая форма имеет одинаковые имя (Name) и заголовок (Caption) – FORM1. Имя формы менять не рекомендуется, так как оно содержится в тексте программы.

Изменение заголовка формы

Для изменения заголовка вызовите окно инспектора объектов (F11) и щелкните кнопкой мы-

ши на форме. В форме инспектора объектов найдите и щелкните мышью на Properties – Caption . В выделенном окне наберите «Лраб 1».

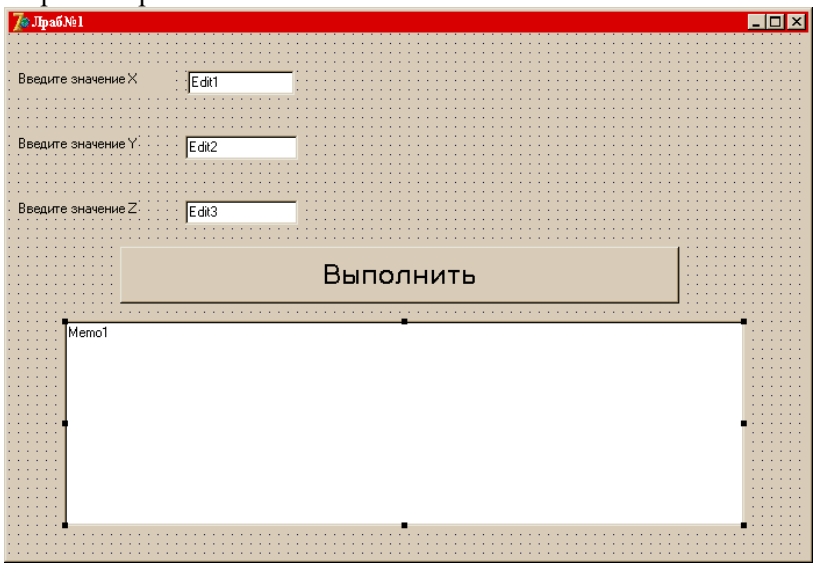


Рис.4.2. Внешний вид формы

Размещение строки ввода (TEdit)

Если необходимо ввести из формы в программу или вывести на форму информацию, которая вмещается в одну строку, используют окно однострочного редактора текста, представляемого компонентом TEdit. В данной программе с помощью однострочного редактора будут вводиться переменные *x, y, z* типа extended или integer.

Выберите в меню компонентов Standard пиктограмму  , щелкните мышью в том месте формы, где вы хотите ее поставить. Вставьте три компонента TEdit в форму. Захватывая их мышью, отрегулируйте размеры окон и их положение. Обратите внимание на то, что в тексте программы появились три новых однотипных переменных Edit1, Edit2, Edit3. В каждой из этих переменных с расширением .Text будет содержаться строка символов (тип String) и отображаться в соответствующем окне Edit.

Так как численные значения переменных *x, y, z* имеют действительный тип дляпреобразования строковой записи числа, находящегося в переменной Edit.Text, в действительное, используется стандартная функция X:=StrToFloat(Edit1.Text).

Если исходные данные имеют целочисленный тип, например integer, то используется стандартная функция X:=StrToInt(Edit1.Text).

При этом в записи числа не должно быть пробелов, а действительное число пишется с десятичной запятой.

С помощью инспектора объектов установите шрифт и размер символов, отображаемых в строке Edit (свойство Font).

Размещение надписей (TLabel)

На форме имеются четыре пояснительные надписи. Для нанесения таких надписей на форму используется компонент TLabel. Выберите в меню компонентов Standard пиктограмму  , щелкните на ней мышью. После этого в нужном месте формы щелкните мышью, появится надпись Label1. Проделайте это для четырех надписей. Для каждой надписи, щелкнув на ней мышью,отрегулируйте размер и, изменив свойство Caption инспектора объектов, введите строку, например "Введите значение X:", а также выберите размер символов (свойства Font).

Обратите внимание на то, что в тексте программы автоматически появились четыре новых переменных типа .TLabel. В них хранятся пояснительные строки, которые можно изменять в процессе работы программы.

Размещение многострочного окна вывода (TMemo)

Для вывода результатов работы программы обычно используется текстовое окно, которое представлено компонентом (TMemo). Выберите в меню компонентов пиктограмму  и поместите компонент TMemo на форму. С помощью мыши отрегулируйте его размеры и местоположение. После установки с помощью инспектора свойства ScrollBars – SSBoth в окне появятся вертикальная и горизонтальная полосы прокрутки.

В тексте программы появится переменная Memo1 типа TMemo. Информация, которая отобра-

жается построчно в окно типа TMemo, находится в массиве строк Memo1.Lines. Каждая строка имеет тип String.

Для чистки окна используется метод Memo1.Clear. Для того чтобы добавить новую строку в окно, используется метод Memo1.Lines.Add (переменная типа String).

Если нужно вывести число, находящееся в переменной действительного или целого типа, то его надо предварительно преобразовать к типу String и добавить в массив Memo1.Lines. Например, если переменная u:=100 целого типа, то метод Memo1.Line.Add сделает это и в окне появится строка «Значение u=100». Если переменная u:=-256,38666 действительного типа, то при использовании метода Meno1.Lines.Add('Значение u=' + FloatToStrF(u.ffFixed,8,2)) будет выведена строка «Значение u= –256.39». При этом под все число отводится восемь позиций, из которых две позиции занимает его дробная часть.

Если число строк в массиве Memo1 превышает размер окна, то для просмотра всех строк используется вертикальная полоса прокрутки. Если длина строки Memo1 превосходит количество символов в строке окна, то в окне отображается только начало строки. Для просмотра всей строки используется горизонтальная полоса прокрутки.

Написание программы обработки события нажатия кнопки(ButtonClick)

Поместите на форму кнопку, которая описывается компонентом TButton, для чего выберем в меню компонентов Standard пиктограмму  . С помощью инспектора объектов измените заголовок (Caption) – Button1 на слово «Выполнить» или другое по вашему желанию. Отрегулируйте положение и размер кнопки.

После этого два раза щелкните мышью на кнопке, появится текст программы, дополненной заголовком процедуры обработчика события – нажатия кнопки (Procedure TForm1.ButtonClick(Sender : TObject);).

Наберите текст этой процедуры, приведенный в примере.

Запуск и работа с программой

Запустить программу можно, нажав Run в главном меню Run, или клавишу F9, или пиктограмму  . При этом происходит трансляция и, если нет ошибок, компоновка программы и создание единого загружаемого файла с расширением .exe. На экране появляется активная форма программы.

Работа с программой происходит следующим образом. Нажмите (щелкните мышью) кнопку «Выполнить». В окне Memo1 появится результат. Измените исходные значения *x, y, z* в окнах Edit и снова нажмите кнопку «Выполнить» – появятся новые результаты. Завершить работу программы можно нажав в главном меню Run или кнопку {}на форме.

Текстпрограммы:

```
unit tema1;
interface
uses
Windows, Messages. SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;
type
TForm1 = class(TForm)
Label1:TLabel;
Edit1:TEdit;
Label2: TLabel;
Edit2:TEdit:
Label3: TLabel;
Edit3: TEdit;
Label4: TLabel;
Memo1:TMemo;
Buttonl : TButfon;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
 private
{ Private declarations }
 public
{ Public declarations }
 end;
var
Form1: TForm1;
Implementation
 {$R*.DFM}

procedure TForm1.Button1Click(Sender: TObject);
var
x, y, z, a, b, c, u : extended;
begin
x:=StrToFloat(Edit1.Text);                    // Считываетсязначение X
Memo1.Lines.Add('X = '+Edit1.Text);        // Вывод X вокно Memo1
```

```
y:=StrToFloat(Edit2.Text);                    // Считывается значение Y
Memo1.Lines.Add('Y = '+Edit2.Text);       // Вывод Y вокно Memo1
z:=StrToFloat(Edit3.Text);                    // Ситывается значение Z
Memo1.Lines.Add('Z = '+Edit3.Text);       // Вывод Z вокно Memo1
// Вычисляем арифметическое выражение
a:=Sqr(Sin(x+y)/Cos(x+y));
b:=Exp(y–z);
c:=Sqrt(Cos(Sqr(x))+Sin(Sqr(z)));
u:=a–b*c;
                              // Выводим результат в окно Memo1
Memo1.Lines.Add('Результат  U = '+FloatToStrF(u,ffFixed,8,3));
end;
end.
```

Индивидуальные задания

Ниже приведены 15 вариантов задач. По указанию преподавателя выберите свое индивидуальное задание. Уточните условие задания, количество, наименование, типы исходных данных. В соответствии с этим установите количество окон Edit, тексты заголовков на форме, размеры шрифтов, а также типы переменных и функции преобразования при вводе и выводе результатов. С помощью инспектора объектов измените цвет формы, шрифт выводимых символов.

1. Найти сумму цифр заданного четырехзначного числа.
2. Определить число, полученное в обратном порядке цифр заданного трехзначного числа.
3. Вывести на экран 1 или 0в зависимости от того, равна ли сумма двух первых цифр заданного четырехзначного числа сумме двух его последних цифр.
4. Вывести на экран 1или 0в зависимости от того, равен ли квадрат трехзначного числа кубу суммы цифр этого числа.
5. Вывести на экран 1или 0в зависимости от того, есть ли среди первых цифр дробной части заданного положительного вещественного числа определенная цифра.
6. Вывести на экран 1 или 0в зависимости от того, есть ли среди цифр трехзначного числа одинаковые.
7. Присвоить целой переменной k третью от конца цифру в записи положительного целого числа n.
8. Присвоить целой переменной k первую цифру из дробной части положительного вещественного числа.
9. Целой переменной S присвоить сумму цифр трехзначного целого числа k.
10. Идет k–я секунда суток. Определить, сколько полных часов (h) и полных минут (m) прошло к этому моменту.
11. Определить f – угол (в градусах) между положением часовой стрелки в начале суток и ее положением в h –часов, m – минут и s – секунд ($0 \le H \le 11$, $0 \le m, s \le 59$).
12. Определить h – полное количество часов и m – полное количество минут, прошедших от начала суток до того момента (в первой половине дня), когда часовая стрелка повернулась на f градусов ($0 \le f < 360$, f – *вещественное число*).
13. Пусть k – целое от 1 до 365. Присвоить целой переменной n значение 1,2,3,...,6 или 7 в зависимости от того, на какой день недели (понедельник, вторник, ..., суббота или воскресенье) приходится k-й день невисокосного года, в котором 1 января – понедельник.
14. Поменять местами значения целых переменных x и y, не используя дополнительные переменные.
15. Вывести на экран 1 или 0в зависимости от того, имеют три заданных числа одинаковую четность или нет.

Практическая работа 5. **Использование визуальных компонент для создания качественных программ**

Цель работы: научиться пользоваться простейшими компонентами организации переключений (TCheckBox, TRadioGroup).

Общие сведения

Операторы if и case языка Паскаль

Для программирования разветвляющихся алгоритмов в языке Pascal используются специальные переменные типа boolean, которые могут принимать только два значения – true и false (да, нет), а также операторы if и case. Оператор if проверяет результат логического выражения, или значение переменной типа boolean, и организует разветвление вычислений.

Например, если bl: boolean, x,y,u:integer, то фрагмент программы с оператором if может быть

таким:

```
bl:=x>y;
if  bl   then u:=x–y
else u:=x–y;
```

Оператор выбора case организует разветвления в зависимости от значения некоторой переменной перечисляемого типа.

Например, если In: integer, то после выполнения

```
case in of
   0: u:=x+y;
1: u:=x–y;
   2: u:=x*y;
   else u=0;
end;
```

В соответствии со значением in вычисляется u. Если in = 0, то u = x + y, если in = 1, то u = x – y, если in = 2, то u = x * y и, наконец, u = 0 при любых значениях in, отличных от 0, 1 или 2.

Кнопки-переключатели в Delphi

При создании программ в Delphi для организации разветвлений часто используются компоненты в виде кнопок-переключателей. Состояние такой кнопки (включено – выключено) визуально отражается на форме.

Компонент TCheckBox организует кнопку независимого переключателя, с помощью которой пользователь может указать свое решение типа да/нет. В программе состояние кнопки связано со значением булевской переменной, которая проверяется с помощью оператора if.

Компонент TRadioGroup организует группу кнопок-зависимых переключателей. При нажатии одной из кнопок группы все остальные кнопки отключаются. В программу передается номер включенной кнопки (0, 1, 2, …), который анализируется с помощью оператора case.

Пример программы разветвляющегося алгоритма

Задание: ввести три числа – *x, y, z*. Вычислить по усмотрению *u = sin(x)* или*u = cos(x)*, или *u = tg(x).* Найти по желанию максимальное из трех чисел: max(*x, y, z*), или min (|*x*|, |*y*|, |*z*|).Создать форму и написать соответствующую программу.

Создайте форму, такую же как в первом задании, скорректировав текст надписей и положение окон TEdit.

Работа с компонентом TCheckBox

Выберите в меню компонентов Standard пиктограмму {} и поместите ее в нужное место формы. С помощью инспектора объектов измените заголовок (Caption) на «maabs». В тексте программы появится переменная CheckBox типа TCheckBox. Теперь в зависимости от того, нажата кнопка или нет, булевская переменная CheckBox.Checked будет принимать значения True или False.

Работа с компонентом TRadioGroup

Выберите в меню компонентов Standard пиктограмму  и поместите ее в нужное место формы. На форме появится окаймленный линией чистый прямоугольник с заголовком RadioGroup1. Замените заголовок (Caption) на *U*(*x*). Для того чтобы разместить на компоненте кнопки, необходимо свойство Columns установить равным единице (кнопки размещаются в одном столбце). Дважды щелкните по правой части свойства Items мышью, появится строчный редактор списка заголовков кнопок. Наберите три строки с именами: в первой строке sin(*x*), во второй – cos(*x*), в третьей – tg(*x*), нажмите ОК. После этого на форме внутри окаймления появzтся три кнопки-переключателя с введенными надписями.

Обратите внимание на то, что в тексте программы появилась переменная RadioGroup типа TRadioGroup. Теперь при нажатии одной из кнопок группы в переменной целого типа RadioGroup1.ItemIndex будет находиться номер нажатой клавиши (отсчитывается от нуля), что используется в тексте приведенной программы.

Создание обработчиков событий FormCreate и Button1Click

Процедуры – обработчики событий FormCreate и Button1Click – создаются аналогично тому, как и в первой теме.

Запустите программу и убедитесь в том, что все ветви алгоритма выполняются правильно. Форма приведена на рис. 5.1.

Рис. 5.1. Форма программы разветляющегося алгоритма

Текст программы приведен ниже.

```
unittema2;
interface

uses
 Windows, Messages, SysUtils, Classes, Graphics, Controls,  Forms, Dialogs, StdCtrls, ExtCtrls;
 type
TForm1 = class(TForm)
CheckBox1: TCheckBox;
RadioGroup1: TRadioGroup;
Memo1: TMemo;
Button1:TButton;
Edit1:TEdit;
Label1: TLabel;
Label2: TLabel;
Edit2:TEdit;
Label3: TLabel:
Edi3: TEdlt;

  procedure FormCreate(Sender: TObject);
  procedure Button1Click(Sender: TObject);
    private
 {Private declarations}
  public
{public declarations}
var
Form1: TForm1;

implementation

{$R.DFM}
procedure TForm1.FormCreate(Sender: TObject);
begin
Edit1.Text:='0,1';
Edit3.Ttext:='0,356';
Memo1.Clear;
Memo1.Lines.Add('Рез–ты  ст.гр. 9383 Валента А.А.');
end;
procedure TForm1.Button1Click(Sender: TObject);
var x, y, z, u, ma : extended;
begin
      // Ввод исходных данных и их вывод в окно Memo1
x:=StrToFloat(Edit1.Text);
      Memo1.Lines.Add('x='+Edit1.Text);
y:=StrToFloat(Edlt2.Text);
      Memo1.Lines.Add('y='+Edit2.Text);
z:=StrToFloat(Edtt3.Text);
      Memo1.Lines.Add('z='+Edit3.Text);
// Проверка номера нажатой кнопки и выбор соответствующей ей функции
case RadioGroup.ItemIndex of
          0: u:=cos(x);
1: u:=sin(x);
          2: u:=sin(x)/cos(x);
  end:
              // ПроверкасостояниякнопкиCheckBox1
     if  CheckBox1.Checked      then
                 begin
                       u:=abs(u);
                       y:=abs(y);
z:=abs(z)
                  end;
                              //   Нахождение максимального из трех чисел
if   u>y    then  ma:=u   else ma: =y;
if   z>ma   then  ma:=z;
```

```
            if  CheckBox1.Checked   then
                Memo1.Lines.Add(' maxabs='+FloatToStrF(ma,ffFixed,8,2))
                else
                Memo1.Lines.Add('max='+FtoatToStrF(ma,ffGeneral,8,2));
end;
end.
```

Индивидуальные задания

Ниже приведены 15 вариантов задач. По указанию преподавателя выберите свое индивидуальное задание. Уточните условие задания, количество, наименование, типы исходных данных. В соответствии с этим установите количество окон Edit, тексты заголовков на форме, размеры шрифтов, а также типы переменных и функции преобразования при вводе и выводе результатов. С помощью инспектора объектов измените цвет формы, шрифт выводимых символов.

1. Известно, что из четырех чисел *a1,a2,a3* и *a4* одно отлично от трех других, равных между собой. Присвоить номер этого числа переменной *n.*

2. По номеру n ($n > 0$) некоторого года определить c – номер его столетия (учесть, что, к примеру, началом XX столетия был 1901, а не 1900 год!).

3. Значения переменных *a, b* и *c* поменять местами так, чтобы оказалось $a \le b \le c$.

4. Дано целое k от 1 до 180. Определить, какая цифра находится в *k*–й позиции последовательности 10111213...9899 , в которой выписаны подряд все двузначные числа.

5. Дано натуральное *k.* Определить *k*-ю цифру в последовательности 110100100010000100000..., в которой выписаны подряд степени 10.

6. В старояпонском календаре был принят 60–летний цикл, состоявший из пяти 12-летних подциклов. Подциклы обозначались названиями цвета: green(зеленый), red (красный), yellow (желтый), white(белый) и black (черный). Внутри каждого подцикла годы носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи (1984 год – *годзеленой крысы* – был началом очередного цикла). Разработать программу, которая вводит номер некоторого года нашей эры и выводит его название по старояпонскому календарю.

7. Если сумма трех попарно различных действительных чисел *x, y, z* меньше единицы, то наименьшее из этих трех чисел заменить полусуммой двух в противном случае заменить меньшее из *x* и *y* полусуммой двул оставшихся значений,

8. Для целого числа *k* от 1 до 99 вывести фразу «мне *k*лет», учитывая при этом, что при некоторых значениях *k* слово «лет» надо заменить на слово «год» или «года».

9. Для натурального числа *k* вывести фразу «мы выпили *k* бутылок пива», согласовывая окончание слова «бутылка» с числом *k*.

10. *Type курс=(С,В,Ю,З); {север, восток, юг, запад}*

Приказ=(вперед, вправо, назад, влево);

Var К1, К2: курс; ПР: приказ;

Корабль сначала шел по курсу К1, а затем его курс был изменен согласно прикаказу ПР1. Определить К2 – новый курс корабля.

11. Type месяц = (январь, февраль, март, апрель, май, июнь, июль, август, сентябрь, октябрь, ноябрь, декабрь);

день=1...31;

Var d1,d2: день;

m1,m2: месяц;

t: boolean;

Переменной *t* присвоить значение 1 если дата *d*1, *m*1предшествует (в рамках года) дате *d*2, *m2,* и значение 0в других случаях.

12. Type нота=(до, ре, ми, фа, соль, ля, си);

интервал=(секунда, терция, кварта, квинта, секста, септима);

var n1, n2: нота;

i: интервал;

Определить *i*-й интервал, образованный нотами *n*1и *n*2 *(n*1<>*n*2*):* секунда – это интервал из двух соседних (по кругу) нот (например, ре и ми, си и до), терция – интервал через ноту (например, фа и ля, си и ре) и т.д.

13. Type единица = (дециметр, километр, метр, миллиметр, сантиметр);

длина=real;

var x: длина;

P: единица;

Значение переменной *x,* означающее некоторую длину в единицах *p,* заменить на величину этой же длины в метрах.

14. Type сезон = (зима, весна, лето, осень);

Var m: месяц; {определение «месяц» см. в 26}

S: сезон;

Определить *S* – сезон, на который приходится месяц *m*.

15. Var k: 1...9;

Вывести значение переменной *k* римскими цифрами.

Лабораторная работа 6

Средства отладки программ в объектно-ориентированном программировании

Цель работы: изучить простейшие средства отладки программ в среде Delphi.

Общие сведения

Операторы организации циклов repeat, whyle, for языка Pascal

Под циклом понимается многократное выполнение одних и тех же операторов при различных значениях промежуточных данных. Число повторений может быть задано в явной или неявной форме.

Для организации повторений в языке Pascal предусмотрены три различных оператора цикла.

Оператор

repeat
<операторы>
until<условие>;

Организует повторение операторов, помещенных между ключевыми словами repeat и until, до тех пор, пока не выполнится <условие> = true, после чего управление передается следующему за циклом оператору.

Оператор

While <условие> do begin
<операторы>
end;

Организует повторение операторов, помещенных между begin и end, до тех пор, пока не выполнится <условие> = false. Следует отметить, что если <условие> = false при первом входе, то <операторы> не выполнятся ни разу, в отличие от repeat, в котором хотя бы один раз они выполнятся.

Оператор

for i:=i1 to i2 do begin
<операторы>
end;

Организует повторение операторов при нарастающем изменении переменной цикла *i* от начального значения *i*1 до конечного *i*2 с шагом «единица». Если *i*2 >*i*1, то <операторы> не выполнятся ни разу. Модификация оператора for i:=i2 downto i1 do begin <операторы> end организует повторения при убывающем изменении *i* на единицу.

Средства отладки программ в Delphi

Практически в каждой вновь написанной программе после запуска обнаруживаются ошибки.

Ошибки первого уровня (ошибки компиляции) связаны с неправильной записью операторов (орфографические, синтаксические). При обнаружении ошибки компилятор Delphi останавливается напротив первого оператора, в котором обнаружена ошибка. В нижней части экрана появляется текстовое окно, содержащее сведения обо всех ошибках, найденных в проекте. Каждая строка этого окна содержит имя файла, в котором найдена ошибка, номер строки с ошибкой и характер ошибки. Для быстрого перехода к интересующей ошибке необходимо дважды щелкнуть на строке с ее описанием. Для получения более полной информации о характере ошибки необходимо обратится к HELP нажатием клавиши F1. Следует обратить внимание на то, что одна ошибка может повлечь за собой другие, которые исчезнут при ее исправлении. Поэтому следует исправлять ошибки последовательно, сверху вниз и после исправления каждой ошибки компилировать программу снова.

Ошибки второго уровня (ошибки выполнения) связаны с ошибками выбранного алгоритма решения или с неправильной программной реализацией алгоритма. Эти ошибки проявляются в том, что результат расчета оказывается неверным либо происходит переполнение, деление на ноль

и др. Поэтому перед использованием отлаженной программы ее надо протестировать, т.е. сделать просчеты при таких комбинациях исходных данных, для которых заранее известен результат. Если тестовые расчеты указывают на ошибку, то для ее поиска следует использовать встроенные средства отладки среды DELPHI.

В простейшем случае для локализации места ошибки рекомендуется поступать следующим образом. В окне редактирования текста установить курсор в строке перед подозрительным участком и нажать клавишу F4 (выполнение до курсора). Выполнение программы будет остановлено на строке, содержащей курсор. Теперь можно увидеть, чему равно значение интересующих переменных. Для этого можно поместить на нужную переменную курсор (на экране будет высвечено ее значение) либо нажать Ctrl-F7 и в появившимся диалоговом окне указать интересующую переменную (с помощью данного окна можно также изменить значение переменной во время выполнения программы). Нажимая клавишу F7 (пошаговое выполнение), можно построчно выполнять программу, контролируя изменение тех или иных переменных и правильность вычислений. Если курсор находится внутри цикла, то после нажатия F4 расчет останавливается после одного выполнения тела цикла. Для продолжения расчетов следует нажать <Run> меню Run.

Пример написания программы циклического алгоритма

Задание: написать и отладить программу, которая выводит таблицу значений функции $S(x)=\sum_{k=0}^{N}(-1)^k\frac{x^k}{k!}$ для x изменяющихся в интервале от $X1$ до $X2$ с шагом h.

Окно диалога представлено на рис. 6.1.

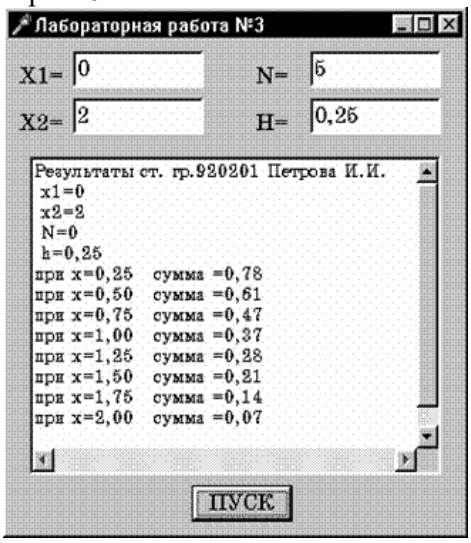


Рис. 6.1. Окно диалога программы

Текст программы приведен ниже.

```
unittema3;
interface
uses
Windows,  Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, ExtCtrls;
type
 TForm1 = class(TForm)
  Memo1: TMemo;
  Button1: TButton;
  Label1: TLabel;
  Label2: TLabel;
  Label3: TLabel;
  Label4: TLabel;
  Edit1: TEdit;
  Edit2: TEdit;
         Edit3: TEdit;
         Edit4: TEdit;
procedure FormCreate(Sender: TObject);
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations}
end;
var Form1: TForm1;
implementation
{$R*.DFM}
procedure TForm1.FormCreate(Sender: TObject);
begin
  Edit1.Text :='0';
```

```
  Edit2.Text :='2';
  Edit3.Text :='5';
  Edit4.Text :='0.25';
Memo1.Clear;
Memo.Lines.Add('Результатыст. гр. 9383 ВалетаА.В.');
end;
procedure TForm1.Button1Cllck(Sender: TObject);
var x1, x2, x, h, a, s : extended;
      N, k, c : integer;
begin
x1:=StrToFloat(Edit1.Text);
   Memo1.Lines.Add('x1='+Edit1.Text);
x2:=StrToFloat(Edit2.Text);
    Memo1.Lines.Add('x2='+Edit2.Text);
N:=StrToInt(Edit3.Text);
    Memo1.lines.Add('N='+Edit3.Text);
h:=StrToFloat(Edit4.Text);
    Memo1.Lines.Add(' h='+Edit4.Text);
      c:=–1;
      x:=x1;
repeat
     a:=1;
     S:=1;
       for k:=1 to N do
           begin
   a:=c*a*x/k;
    s:=s+a;
end;
Memo1.lines.Add('при x=' +FloatToStrF(x,ffFixed,6,2)+ 'сумма =' + FloatToStrF(s,tfFtxed,6,2));
   x:=x+h;
until x>x2;
 end;
 end.
```

После отладки программы составьте тест ($N = 2$, $X1 = 0, X2 = 1$, $h = 0{,}3$), установите курсор на первый оператор (N:=), нажмите клавишу F4. После этого нажимая клавишу F7, выполните пошаговую программу и проследите, как меняются все переменные в процессе выполнения.

Индивидуальные задания

Ниже приведены 15 вариантов задач. По указанию преподавателя выберите свое индивидуальное задание. Уточните условие задания, количество, наименование, типы исходных данных. В соответствии с этим установите количество окон Edit, тексты заголовков на форме, размеры шрифтов, а также типы переменных и функции преобразования при вводе и выводе результатов. С помощью инспектора объектов измените цвет формы, шрифт выводимых символов.

1. Подсчитать k – количество цифр в десятичной записи целогонеотрицательного числа n.

2. Переменной t присвоить значение 1 или 0 в зависимости от того, является ли натуральное число k степенью 3.

3. Дано n вещественных чисел. Вычислить разность между максимальным и минимальным из них.

4. Дана непустая последовательность различных натуральных чисел, за которойследует 0. Определить порядковый номер наименьшего из них.

5. Даны целое $n > 0$ и последовательность из n вещественных чисел, среди которых есть хотя бы одно отрицательное число. Найти значение наибольшего среди отрицательных чисел этой последовательности.

6. Дано n вещественных чисел. Определить, образуют ли они возрастающую последовательность.

7. Дана последовательность из n целых чисел. Определить, со скольких отрицательных чисел она начинается.

8. Определить k – количество трехзначных натуральных чисел, сумма цифр которых равна n ($1 \le n \le 27$). Операции деления (/, div и mod) не использовать.

9. Вывести на экран в возрастающем порядке все трехзначные числа, в десятичной записи которых нет одинаковых цифр (операции деления не использовать).

10. Переменной t присвоить значение 1 или 0 в зависимости от того, можно или нет натуральное число n представить в виде трех полных квадратов.

11. Дано натуральное число n. Выяснить, входит ли цифра 3 в запись числа n.

12. Дано натуральное число n. Найти сумму его цифр

13. Дано целое *n*> 0, за которым следует *n* вещественных чисел. Определить, сколько среди них отрицательных.

14. Дано натуральное число *n*. Переставить местами первую и последнюю цифры числа *n.*

15. Дано натуральное число *n*. Заменить порядок следования цифр числа *n* на оборот.

Практическая работа 7.**Использование стиля программирования**

Цель работы: разработать программу по правилам хорошего стиля программирования. Написать программу с использованием массивов.

Меню – один из распространенных элементов пользовательскогоинтерфейса, который представляет собой список пунктов, объединенных по функциональному признаку, каждый из которых обозначает команду или вложенное меню (подменю).

Главное меню располагается в верхней части формы под ее заголовком и содержит наиболее общие команды приложения. В Delphi главное меню представлено компонентом MainMenu.

Для создания и изменения меню в процессе разработки приложения в среде Delphi предназначен Конструктор меню (MenuDesigner). Запуск Конструктора меню можно выполнить по команде MenuDesigner. контекстного меню компонента MainMenu, а также с помощью двойного щелчка кнопкой мыши на этот компонент. При конструировании меню имеет тот же вид, что и при выполнении приложения.

Наименование пункта меню задается путем присвоения нужного значения его свойству Caption. Кроме того, в Delphi у компонента MainMenu доступны такие свойства, как Checked и Bitmap, определяющие соответственно:

Checked = true/false – наличие/отсутствие отметки √ у пункта меню (для отметки выбора);

Bitmap = рисунок, определяющий наличие картинки перед названием пункта в меню.

Для закрепления процедуры за выбором некоторого пункта меню (событие OnClick) на этапе проектирования приложения следует выбрать этот пункт с помощью клавиатуры или мыши.

Работа с массивами

Массив есть упорядоченный набор однотипных элементов, объединенных под одним именем. Каждый элемент массива обозначается именем, за которым в квадратных скобках следует один или несколько индексов, разделенных запятыми, например: a[1], bb[I], c12[I, J*2], q[1, 1, I*J-1]. В качестве индекса можно использовать любые порядковые типы за исключением LongInt.

Тип массива или сам массив определяются соответственно в разделе типов (Type) или переменных (Var) с помощью ключевого слова Array следующим образом:

Array [описание индексов] of <тип элемента массива>

Примеры описания массивов:

```
Const N=20;    // Задание максимального значения индекса;
Type TVector=array[1..N] of real;  // Описание типа одномерного массива;
Var     a:TVector;          // A – массивтипа Tvector;
Ss:array[1..10] of integer; // Ss – массив из десяти целых чисел;
Y:array[1..5,1..10] of char; //Y – двумерный массив символьного типа.
```

Элементы массивов могут использоваться в выражениях так же, как и обычные переменные, например:

```
F:=2*a[3]+a[ss[I]+1]*3;
A[n]:=1+sqrt(abs(a[n-1]));
```

Компонент TStringGrid

При работе с массивами ввод и вывод информации на экран удобно организовывать в виде таблиц. Компонент TStringGrid предназначен для отображения информации в виде двухмерной таблицы, каждая ячейка которой представляет собой окно однострочного редактора (аналогично окну TEdit). Доступ к информации осуществляется с помощью свойства Cells[ACol, ARow: Integer]: string, где ACol, Arow – индекс элемента двухмерного массива. Свойства ColCount и RowCount устанавливают количество

строк и столбцов в таблице, а свойства FixedCols и FixedRows задают количество строк и столбцов фиксированной зоны. Фиксированная зона выделена другим цветом, и в нее запрещен ввод информации с клавиатуры.

Пример написания программы

Задание: создать программу для определения вектора, где A – квадратная матрица размерностью N×N, а Y, B – одномерные массивы размерностью N. Элементы вектора Y определяются по формуле. Значения N следует вводить в компонент TEdit, A и B – в компонент TStringGrid. Результат после нажатия кнопки типа TButton вывести в компонент TStringGrid.

Настройка компонента TStringGrid

Для установки компонента TStringGrid на форму необходимо на странице Additional меню компонентов щелкнуть мышью по пиктограмме  . После этого щелкните мышью в нужном месте формы. Захватывая кромки компонента, отрегулируйте его размер. В инспекторе объектов значения свойств ColCount и RowCount установите 2 (две строки и два столбца), а FixedCols и FixedRows установите 1 (один столбец и одна строка с фиксированной зоной). Так как компоненты StringGrid2 и StringGrid3 имеют только один столбец, то для них ColCount = 1, RowCount = 2, FixedCols = 0 и FixedRows = 1. По умолчанию в компонент TStringGrid запрещен ввод информации с клавиатуры, поэтому необходимо свойство Options goEditing для компонентов StringGrid1 и StringGrid2 установить в положение True.

Окно диалога приведено на рис. 7.1.

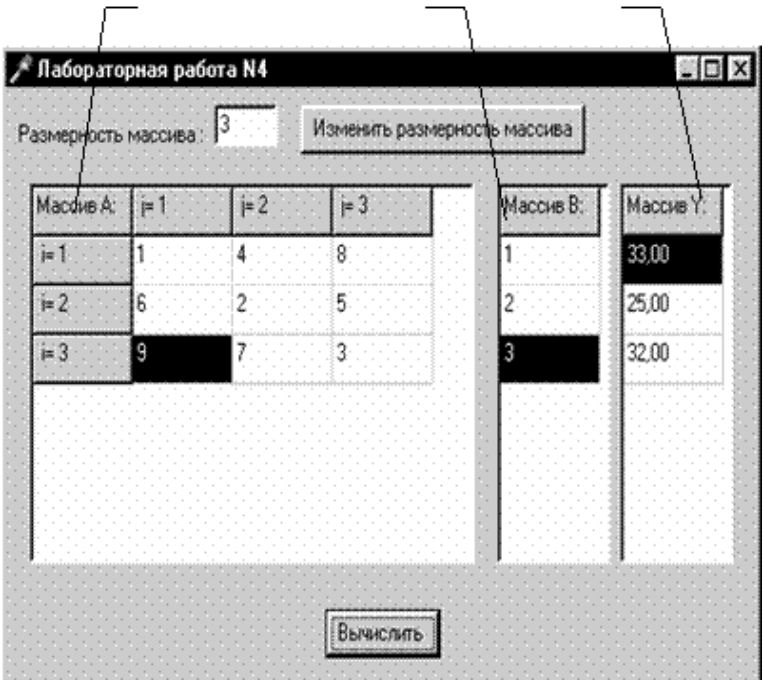


Рис. 7.1. Окно диалога программы

Текст программы приведен ниже.

```
unittem4;
 interface
 uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Grids;

type
  TForm1 = class(TForm)
```

```
    Label1: TLabel;
    Edit1: TEdit;
    Button1: TButton;
    Button2: TButton;
    StringGrid1: TStringGrid;
    StringGrid2: TStringGrid;
    StringGrid3: TStringGrid;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
const
 Nmax=10;              // Максимальнаяразмерностьмассива
Type
 Mas2 = array[1..Nmax,1..Nmax] ofextended;    // ОбъявлениетипадвухмерногомассиваразмерностьюNmax
 Mas1 = array[1..Nmax] ofextended;                // Объявление типа одномерного массива размерностью Nmax
var
 Form1: TForm1;
 A : Mas2;               // Объявление двухмерного массива
 B,Y : Mas1;             // Объявление одномерных массивов
 N,i,j : integer;

implementation

{$R *.DFM}

procedure TForm1.FormCreate(Sender: TObject);
begin
    N:=3;          // Размерностьмассива
      Edit1.Text:=FloatToStr(N);
 {Задание числа строк и столбцов в таблицах}
  StringGrid1.ColCount:=N+1;
   StringGrid1.RowCount:=N+1;
   StringGrid2.RowCount:=N+1;
   StringGrid3.RowCount:=N+1;

    {Ввод в левую верхнюю ячейку таблицы названия массива}
  StringGrid1.Cells[0,0]:='Массив A:';
   StringGrid2.Cells[0,0]:='Массив B:';
   StringGrid3.Cells[0,0]:='Массив Y:';
    {Заполнение верхнего и левого столбцов поясняющими подписями}
  for i:=1 to N do begin
    StringGrid1.Cells[0,i]:=' i= '+IntToStr(i);
    StringGrid1.Cells[i,0]:=' j= '+IntToStr(i);
                end;
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
  N:=StrToInt(Edit1.Text);
 {Задание числа строк и столбцов в таблицах}
  StringGrid1.ColCount:=N+1;
   StringGrid1.RowCount:=N+1;
   StringGrid2.RowCount:=N+1;
   StringGrid3.RowCount:=N+1;
 {Заполнение верхнего и левого столбцов поясняющими подписями}
  for i:=1 to N do begin
    StringGrid1.Cells[0,i]:=' i= '+IntToStr(i);
    StringGrid1.Cells[i,0]:=' j= '+IntToStr(i);
                end;
end;

procedure TForm1.Button2Click(Sender: TObject);
 vars: extended;
begin
  {Заполнение массива A элементами из таблицы StringGrid1}
 for i:=1 to N do
   for j:=1 to N do
     A[i,j]:=StrToFloat(StringGrid1.Cells[j,i]);
  {Заполнение массива B элементами из таблицы StringGrid2}
 for i:=1 to N do
     B[i]:=StrToFloat(StringGrid2.Cells[0,i]);
   {Умножение массива A на массив B}
 for i:=1 to N do begin
```

```
    s:=0;
  for j:=1 to N do s:=s+A[i,j]*B[j];
      Y[i]:=s;
       {Вывод результата в таблицу StringGrid3}
      StringGrid3.Cells[0,i]:=FloatToStrf(y[i],fffixed,6,2);
            end;
end;
end.
```

Индивидуальные задания

Ниже приведены 15 вариантов задач. По указанию преподавателя выберите свое индивидуальное задание. Уточните условие задания, количество, наименование, типы исходных данных. В соответствии с этим установите количество визуальных компонентов в форме.

Во всех заданиях скалярные переменные следует вводить с помощью компонента TEdit с соответствующим пояснением в виде компонента TLabel. Скалярный результат выводить в виде компонента TLabel. Массивы представлять на форме в виде компонентов TStringGrid, в которых 0-й столбец и 0-ую строку использовать для отображения индексов массивов. Вычисления выполнять после нажатия соответствующих кнопок визуального компонента MainMenu.

1. Задана матрица размером $N \times M$. Получить массив B, присвоив его k-му элементу значение 0, если все элементы k-го столбца матрицы нулевые, и значение 1 в противном случае.
2. Задана матрица размером $N \times M$. Получить массив B, присвоив его k-му элементу значение 1, если элементы k-й строки матрицы упорядочены по убыванию, и значение 0 в противном случае.
3. Задана матрица размером $N \times M$. Получить массив B, присвоив его k-му значение 1, если k-я строка матрицы симметрична, и значение 0 в противном случае.
4. Задана матрица размером $N \times M$. Определить k – количество «особых» элементов матрицы, считая элемент «особым», если он больше суммы остальных элементов своего столбца.
5. Задана матрица размером $N \times M$. Определить k – количество «особых» элементов матрицы, считая элемент «особым», если в его строке слева от него находятся элементы, меньшие его, а справа – большие.
6. Задана символьная матрица размером $N \times M$. Определить k – количество различных элементов матрицы (т.е. повторяющиеся элементы считать один раз).
7. Дана матрица размером N×M. Упорядочить ее строки по неубыванию их первых элементов.
8. Дана матрица размером $N \times M$. Упорядочить ее строки по неубыванию суммы их элементов.
9. Дана матрица размером $N \times M$. Упорядочить ее строки по неубыванию их наибольших элементов.
10. Определить, является ли заданная квадратная матрица n-го порядка симметричной относительно побочной диагонали.
11. Для матрицы размером $N \times M$ вывести на экран все ее седловые точки. Элемент матрицы называется седловой точкой, если он является наименьшим в своей строке и одновременно наибольшим в своем столбце, или наоборот.
12. В матрице n-го порядка переставить строки так, чтобы на главной диагонали матрицы были расположены элементы, наибольшие по абсолютной величине.
13. В матрице n-го порядка найти максимальный среди элементов, лежащих ниже побочной диагонали, и минимальный среди элементов, лежащих выше главной диагонали.
14. В матрице размером $N \times M$ поменять местами строку, содержащую элемент с наибольшим значением, со строкой, содержащей элемент с наименьшим значением.
15. Из матрицы n-го порядка получить матрицу порядка $n - 1$ путем удаления из исходной матрицы строки и столбца, на пересечении которых расположен элемент с наибольшим по модулю значением.

Практическая работа 8. **Методы оптимальной обработки текстовой информации**

Цель работы: изучить правила работы с компонентами TListBox и TComboBox.

Короткие строки имеют фиксированное количество символов. Строка ShortString может содержать 255 символов. Строка String[N] может содержать N символов, но не более 255. Первый байт этих переменных содержит длину строки.

Длинная строка типа String

При работе с этим типом данных память выделяется по мере необходимости (динамически) и может занимать всю доступную программе память. Вначале компилятор выделяет для переменной 4 байта, в которых размещается номер ячейки памяти, начиная с которой будет располагаться символьная строка. На этапе выполнения программа определяет необходимую длину цепочки символов и обращается к ядру операционной системы с требованием выделить необходимую па-

мять.

Широкая строка типа WideString введена для обеспечения совместимости с компонентами, основанными на OLE-технологии. От типа String отличается только тем, что для представления каждого символа используется не один, а два байта.

Нуль-терминальная строка типа PChar представляет собой цепочку символов, ограниченную символом #0. Максимальная длина строки ограничена только доступной программе памятью. Нуль-терминальные строки широко используются при обращениях к API-функциям Windows (API – Application Program Interface – интерфейс прикладных программ).

Представление строки в виде массива символов

Строка может быть описана как массив символов. Если массив имеет нулевую границу, он совместим с типом PChar.

```
Var
  MasS : array[1…100] ofChar;
```

В отличие от нуль-терминальной строки здесь длина имеет фиксированное значение и не может меняться в процессе выполнения программы.

Компонент TListBox

Компонент TListBox представляет собой список, элементы которого выбираются при помощи клавиатуры или мыши. Список элементов задается свойством Items, методы Add, Delete и Insert которого используются для добавления, удаления и вставки строк. Объект Items (TString) хранит строки, находящиеся в списке. Для определения номера выделенного элемента используется свойство ItemIndex.

Компонент TComboBox

Комбинированный список TComboBox представляет собой комбинацию списка TListBox и редактора TEdit, поэтому практически все свойства заимствованы у этих компонентов. Для работы с окном редактирования используется свойство Text как в TEdit, а для работы со списком выбора – свойство Items как в TListBox. Существует пять модификаций компонента, определяемых его свойством Style. В модификации csSimple список всегда раскрыт, в остальных он раскрывается после нажатия кнопки справа от редактора.

Компонент TBitBtn

Компонент TBitBtn расположен на странице Additonal палитры компонентов и представляет собой разновидность стандартной кнопки TBotton. Его отличительная особенность – наличие растрового изображения на поверхности кнопки, которое определяется свойством Clyph. Кроме того, имеется свойство Kind, которое задает одну из 11 стандартных разновидностей кнопок. Нажатие любой из них, кроме bkCustom и bkHelp, закрывает модальное окно и возвращает в программу результат mr*** (например, bkOk – mrOk). Кнопка bkClose закрывает главное окно и завершает работу программы.

Обработка событий

Обо всех происходящих в системе событиях, таких как создание формы, нажатие кнопки мыши или клавиатуры и т.д., ядро Windows информирует окна путем посылки соответствующих сообщений. Среда Delphi позволяет принимать и обрабатывать большинство таких сообщений. Каждый компонент содержит обработчики сообщений на странице Events инспектора объектов.

Для создания обработчика события необходимо раскрыть список компонентов в верхней части окна инспектора объектов и выбрать необходимый компонент. Затем на странице Events нажатием левой клавиши мыши выбрать обработчик и дважды щелкнуть по его левой (белой) части. В ответ Delphi активизирует окно текста программы и покажет заготовку процедуры обработки выбранного события.

Каждый компонент имеет свой набор обработчиков событий. Наиболее часто применяемые события представлены в табл. 8.1.

Таблица 8.1

Часто применяемые события в Delphi

Событие	Описание события
OnActivate	Форма получает это событие при активации
OnCreate	Возникает при создании формы (компонент TForm). В обработчике данного события следует задавать действия, которые должны происходить в момент создания формы, например установка начальных значений
OnKeyPress	Возникает при нажатии кнопки на клавиатуре. Параметр Key имеет тип Char и содержит ASCII–

	код нажатой клавиши (клавиша Enter клавиатуры имеет код #13, клавиша Esc – #27 и т.д.). Обычно это событие используется в том случае, когда необходима реакция на нажатие одной из клавиш
OnKeyDown	Возникает при нажатии клавиши на клавиатуре. Обработчик этого события получает информацию о нажатой клавише и состоянии клавиш Shift, Alt и Ctlr, а также о нажатой кнопке мыши. Информация о клавише передается параметром Key, который имеет тип Word
OnKeyUp	Является парным событием для OnKeyDown и возникает при отпускании ранее нажатой клавиши
OnClick	Возникает при нажатии кнопки мыши в области компонента

Окончание табл. 8.1

Событие	Описание события
OnDblClick	Возникает при двойном нажатии кнопки мыши в области компонента

Пример написания программы

Задание: написать программу подсчета числа слов в произвольной строке. В качестве разделителя может быть любое число пробелов. Для ввода строк и работы с ними использовать TComboBox. Ввод строки заканчивать нажатием клавиши Enter. Для выхода из программы использовать кнопку Close.

Окно диалога будет иметь вид (рис. 8.1).

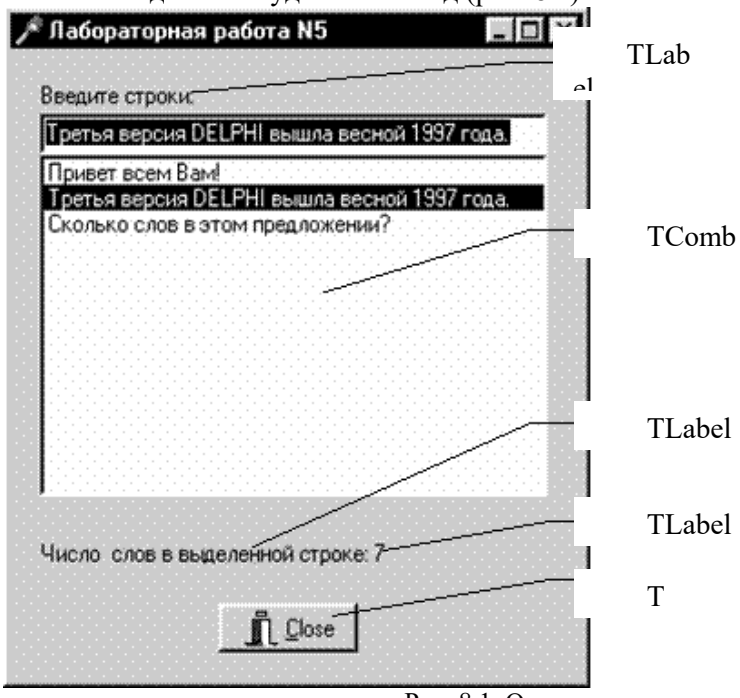


Рис. 8.1. Окно диалога программы

Текст программы.

```
      unit tema6;
interface
uses  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons;
type
  TForm1 = class(TForm)
    Label2: TLabel;
    Label3: TLabel;
    BitBtn1: TBitBtn;
    ComboBox1: TComboBox;
    Label1: TLabel;
    procedure FormActivate(Sender: TObject);
    procedure ComboBox1KeyPress(Sender: TObject; var Key: Char);
    procedure ComboBox1Click(Sender: TObject);
private    { Private declarations }
Public     { Public declarations }
  end;
```

```
var
  Form1: TForm1;
implementation
 {$R *.DFM}
// Обработка события активизации формы
procedure TForm1.FormActivate(Sender: TObject);
begin
        ComboBox1.SetFocus;                  // Передачафокуса ComboBox1
end;
    // Обработка события нажатия левой клавиши мыши
procedure TForm1.ComboBox1KeyPress(Sender: TObject; var Key: Char);
begin
if key=#13 then begin                  // Еслинажатаклавиша Enter, то...
ComboBox1.Items.Add(ComboBox1.Text);
// Строка из окна редактирования
// заносится в список выбора
ComboBox1.Text:='';              // Очистка окна редактирования
      end;  end;
procedure TForm1.ComboBox1Click(Sender: TObject);
var st : string;
    n,i,nst,ind: integer;
begin
  n:=0;                                  // Содержит число слов
     ind:=0;
  nst:=ComboBox1.ItemIndex;
  // Определение номера выбранной строки
  st:=ComboBox1.Items[nst];
   // Занесение выбранной строки в переменную st
    for i:=1 to Length(st) do begin
   // Просмотр всех символов строки st
     caseindof
        0 : if st[i]<>' ' then begin
   // Если встретился символ после пробела
              ind:=1;
                n:=n+1;
    // Число слов увеличивается на единицу
                           end;
       1 : if st[i]=' ' then ind:=0;
    // Если встретился пробел после символов
                        end;
      end;
   Label3.Caption:=IntToStr(n);
    // Вывод числа слов в Label3
end; end.
```

Индивидуальные задания

Во всех заданиях (табл.8.2) исходные данные следует вводить с помощью компонента TEdit в компонент TListBox либо с помощью свойства Text в свойство Items компонента TComboBox. Ввод строки заканчивать нажатием клавиши Enter. Для выхода из программы использовать кнопку Close. Для расчетов вводить несколько различных строк.

Таблица 8.2

Индивидуальные задания

Вариант	Задание
1	Дана строка, состоящая из групп нулей и единиц. Каждая группа отделяется от другой одним или несколькими пробелами. Найти количество групп с пятью символами
2	Дана строка, состоящая из групп нулей и единиц. Найти и вывести на экран самую короткую группу
3	Дана строка, состоящая из групп нулей и единиц. Подсчитать количество символов в самой длинной группе
4	Дана строка, состоящая из групп нулей и единиц. Найти и вывести на экран группы с четным количеством символов
5	Дана строка, состоящая из групп нулей и единиц. Подсчитать количество единиц в группах с нечетным количеством символов
6	Дана строка, состоящая из букв, цифр, запятых, точек, знаков “+” и “–“. Выделить подстроку, которая соответствует записи целого числа (т.е. начинается со знака “+” или “-” и внутри подстроки нет букв, запятых и точек)
7	Дана строка символов, состоящая из букв, цифр, запятых, точек, знаков “+” и “-”. Выделить подстроку, которая соответствует записи вещественного числа с фиксирован-

Вариант	Задание
	ной точкой
8	Дана строка символов, состоящая из букв, цифр, запятых, точек, знаков “+” и “–”. Выделить подстроку, которая соответствует записи вещественного числа с плавающей точкой
9	Дана строка символов, состоящая из произвольных десятичных цифр, разделенных пробелами. Вывести на экран числа этой строки в порядке возрастания их значений

Вариант	Задание
10	Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Заменить буквы латинского алфавита на соответствующие им буквы русского алфавита
11	Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Вывести на экран слова этого текста в порядке, соответствующем латинскому алфавиту
12	Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Поменять местами первую и последнюю буквы каждого слова
13	Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Вывести на экран порядковый номер слова минимальной длины и количество символов в этом слове
14	Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. В каждом слове заменить первую букву на прописную
15	Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Удалить первые *k* слов из строки, сдвинув на их место последующие слова строки

Практическая работа 9. **Оптимальное построение структур данных**

Цель работы: изучить правила работы с компонентами TOpenDialog и TSaveDialog. Написать программу с использованием файлов и данных типа запись с оптимальной структурой данных.

Общие сведения

Запись – это структура данных, объединяющая элементы одного или различных типов, называемые полями. Записи удобны для создания структурированных баз данных с разнотипными элементами, например:

```
Type            {Объявление типа запись}
TStudent = record
 Fio: string[20];              {Поле ФИО}
   Group: integer;               {Поле номера студ. группы}
 Ocn: array[1..3] of integer;          {Полемассиваоценок}
end;
Var
   Student: TStudent;   {Объявление переменной типа запись}
```

Доступ к каждому полю осуществляется указанием имени записи и поля, разделенных точкой, например:

```
        Student.Fio:= ‘Иванов А.И.’;
{Внесение данных в поля записи}
        Student. Group:=720603;
        . . .
```

Доступ к полям можно также осуществлять при помощи оператора with:

```
With Student do
      Begin
```

```
            Fio:= 'ИвановА.И.';
            Group:=720603;
        End;
```

Работа с файлами

Файл – это именованная область данных на внешнем физическом носителе. В Object Pascal различают три вида файлов в зависимости от способа их организации и доступа к элементам: текстовые, типизированные и нетипизированные.

Текстовой файл – это файл, состоящий из строк. Примером текстового файла может служить файл исходного текста программы в DELPHI (расширение *.pas). Для работы с текстовым файлом должна быть описана соответствующая файловая переменная: Var F: TextFile;.

Типизированные файлы имеют строго заданную их описанием структуру, когда все элементы имеют фиксированный и одинаковый размер. Это свойство типизированных файлов позволяет получить доступ к любому компоненту файла по его порядковому номеру. Элементами такого файла являются, как правило, записи. В описании файловой переменной указывается ее тип: Var F: TStudent;.

Нетипизированный файл – это файл, в котором данные не имеют определенного типа и рассматриваются как последовательность байт. Файловая переменная объявляется: Var F: File;.

Порядок работы с файлами следующий:

```
    . . .
    AssignFile(F, 'Filename.txt');
// Связывание файловой переменной F
// с именем дискового файла "Filename.txt"
    Rewrite(F);
// Создание нового или открытие (Reset(F);)
// уже существующего файла
    . . .
    Read(F, Stud);
// Чтение данных из файла или
// запись (Write(F, Stud)) в файл
    . . .
    CloseFile(F);          // Закрытие файла
```

Подпрограммы работы с файлами

AssignFile(var F; FileName: string) – связывает файловую переменную F и файл с именем FileName.

Reset(var F[: File; RecSize: word]) – открывает существующий файл. При открытии нетипизированного файла RecSize задает размер элемента файла.

Rewrite(var F[: File; RecSize: word]) – создает и открывает новый файл.

Append(var F: TextFile) – открывает текстовой файл для дописывания текста в конец файла.

Read(F,v1[,v2,…vn]) – чтение значений переменных начиная с текущей позиции для типизированных файлов и строк для текстовых.

Write(F,v1[,v2,…vn]) – запись значений переменных начиная с текущей позиции для типизированных файлов и строк для текстовых.

CloseFile(F) – закрывает ранее открытый файл.

Rename(var F; NewName: string) – переименовывает неоткрытый файл любого типа.

Erase(var F) – удаляет неоткрытый файл любого типа.

Seek(var F; NumRec: Longint) – для нетекстового файла устанавливает указатель на элемент с номером NumRec.

SetTextBuf(var F: TextFile; var Buf [; Size: word]) – для текстового файла устанавливает новый буфер ввода-вывода объема Size.

Flush(var F: TextFile) – немедленная запись в файл содержимого буфера ввода–вывода.

Truncate(var F) – урезает файл, начиная с текущей позиции.
LoResult: integer – код результата последней операции ввода–вывода.
FilePos(var F): longint – для нетекстовых файлов возвращает номер текущей позиции. Отсчет ведется от нуля.
FileSize(var F): longint – для нетекстовых файлов возвращает количество компонентов в файле.
Eoln(var F: TextFile): boolean – возвращает True, если достигнут конец строки.
Eof(var F)): boolean – возвращает True, если достигнут конец файла.
SeekEoln(var F: TextFile): boolean – возвращает True, если пройден последний значимый символ в строке или файле, отличный от пробела или знака табуляции.
SeekEof(var F: TextFile): boolean – то же, что и SeekEoln, но для всего файла.
BlockRead(var F: File; var Buf; Count: word[; Result: word]), BlockWrite(var F: File; var Buf; Count: word[; Result: word]) – соответственно процедуры чтения и записи переменной Buf с количеством Count блоков.
Компоненты TOpenDialog и TSaveDialog
Компоненты TOpenDialog и TSaveDialog находятся на странице DIALOGS. Все компоненты этой страницы являются невизуальными, т.е. не видны в момент работы программы. Поэтому их можно разместить в любом удобном месте формы. Оба рассматриваемых компонента имеют идентичные свойства и отличаются только внешним видом. После вызова компонента появляется диалоговое окно, с помощью которого выбирается имя программы и путь к ней. В случае успешного завершения диалога имя выбранного файла и маршрут поиска содержатся в свойстве FileName. Для фильтрации файлов, отображаемых в окне просмотра, используется свойство Filter, а для задания расширения файла, в случае, если оно не задано пользователем, – свойство DefaultExt. Если необходимо изменить заголовок диалогового окна, используется свойство Title.
Пример написания программы
Задание: написать программу, вводящую в файл или читающую из файла ведомость абитуриентов, сдавших вступительные экзамены. Каждая запись должна содержать фамилию, а также оценки по физике, математике и сочинению. Вывести список абитуриентов, отсортированный в порядке уменьшения их среднего балла, и записать эту информацию в текстовой файл.
Настройка компонентов TOpenDialog и TSaveDialog
Для установки компонентов TOpenDialog и TSaveDialog на форму необходимо на странице Dialogs меню компонентов щелкнуть мышью соответственно по пиктограммам  или  и поставить их в любое свободное место формы. Установка фильтра производится следующим образом. Выбрав соответствующий компонент, дважды щелкнуть по правой части свойства Filter инспектора объектов. Появится окно Filter Editor, в левой части которого записывается текст, характеризующий соответствующий фильтр, а в правой части – маску. Для OpenDialod1 установим значения маски как показано на рис. 9.1. Формат *.dat означает, что будут видны все файлы с расширением .dat, а формат *.* – что будут видны все файлы (с любым именем и с любым расширением).

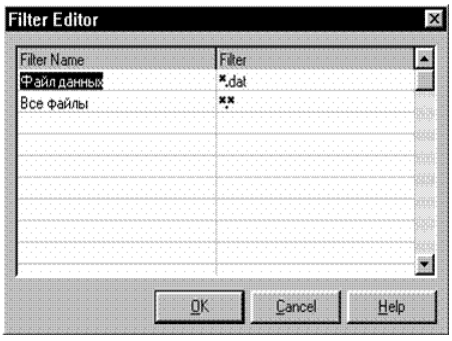


Рис. 9.1. Установка масок для OpenDialod

Для того чтобы файл автоматически записывался с расширением .dat, в свойстве DefaultExt запишем требуемое расширение – .dat. Аналогичным образом настроим SaveDialog1 для текстового файла (расширение .txt).

Работа с программой

После запуска программы на выполнение появится диалоговое окно программы. Кнопка «Ввести запись» видна не будет. Необходимо создать новый файл записей, нажав на кнопку «Создать» или открыть ранее созданный, нажав кнопку «Открыть». После этого станет видна кнопка «Ввести запись» и можно будет вводить записи. При нажатии на кнопку «Сортировка» будет проведена сортировка ведомости по убыванию среднего балла и диалоговое окно примет вид как на рис. 9.2. Затем при нажатии на кнопку «Сохранить» будет создан текстовой файл, содержащий отсортированную ведомость. Файл записей закрывается одновременно с программой при нажатии на кнопку «Close».

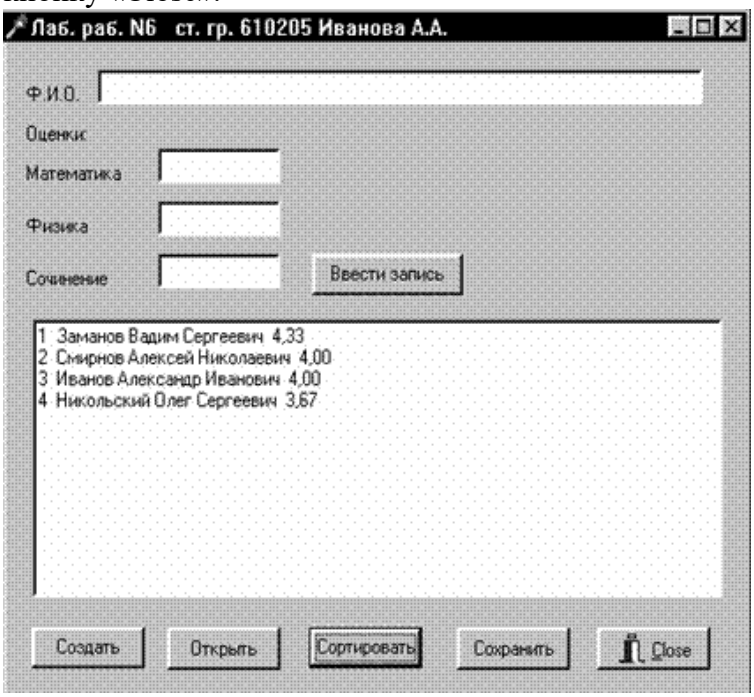


Рис.9.2. Диалоговое окно программы

Текст программы.

```
unittema7;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Memo1: TMemo;
Button1: TButton;
Button3: TButton;
Splitter1: TSplitter;
Button5: TButton;
```

```
    BitBtn1: TBitBtn;
    SaveDialog1: TSaveDialog;
    Button2: TButton;
    OpenDialog1: TOpenDialog;
    Button4: TButton;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
private
    { Private declarations }
public
    { Public declarations }
  end;
Type
  TStudent = record
    FIO: string[40];                    // ПолеФИО
    otc: array[1..3] of word;           // Полемассиваоценок
    sball : extended;                   // Полесреднегобалла
    end;
Var
  Fz  : file of  Tstudent;              // Файлтипазапись
  Ft  : TextFile;                       // Текстовойфайл
  Stud : array[1..100] of Tstudent;   // Массивзаписей
  nzap : integer;                       // Номерзаписи
  FileNameZ, FileNameT : string;   // Имяфайла
var
  Form1: TForm1;
implementation
{$R *.DFM}
procedure TForm1.FormCreate(Sender: TObject);
begin
    Edit1.Text:='';
    Edit2.Text:='';
    Edit3.Text:='';
    Edit4.Text:='';
    Memo1.Clear;
    Button1.Hide;     // Сделать невидимой кнопку «Ввести запись»
    nzap:=0;
end;
procedure TForm1.Button1Click(Sender: TObject);       // Ввестиновуюзапись
begin
    nzap:=nzap+1;
  with stud[nzap] do begin
          FIO:=Edit1.Text;
          otc[1]:=StrToInt(Edit2.Text);
          otc[2]:=StrToInt(Edit3.Text);
          otc[3]:=StrToInt(Edit4.Text);
          sball:=(otc[1]+otc[2]+otc[3])/3;
  Memo1.Lines.Add(fio+'  '+IntToStr(otc[1])+'  '+ IntToStr(otc[2])+'   '+IntToStr(otc[3]));
                        end;
     Write(fz,Stud[nzap]);                                  // Записьвфайл
  Edit1.Text:='';
    Edit2.Text:='';
    Edit3.Text:='';
    Edit4.Text:='';
end;
procedure TForm1.Button2Click(Sender: TObject);
// Создание нового файла записей
begin
    OpenDialog1.Title :='Создать новый файл';
// Изменение заголовка окна диалога
 if OpenDialog1.Execute then
// Выполнение стандартного диалога выбора имени файла
    begin
     FileNameZ:= OpenDialog1.FileName;
// Возвращение имени дискового файла
     AssignFile(Fz, FileNameZ);
// Связывание файловой переменной Fz с именем файла
      Rewrite(Fz);
// Создание нового файла
    end;
     Button1.Show;
// Сделать видимой кнопку «Ввести запись»
end;
procedure TForm1.Button3Click(Sender: TObject);
```

```
// Открытьсуществующийфайл
begin
 ifOpenDialog1.Executethen
// Выполнение стандартного диалога выбора имени файла
   begin
    FileNameZ:= OpenDialog1.FileName;
// Возвращение имени дискового файла
    AssignFile(Fz, FileNameZ);
// Связывание файловой переменной Fz с именем файла
     Reset(Fz);
// Открытие существующего файла
  end;
 while not eof(fz) do begin
   nzap:=nzap+1;
     Read(fz,stud[nzap]);              // Чтениезаписиизфайла
        with stud[nzap] do
     Memo1.Lines.Add(fio+'  '+IntToStr(otc[1])+'  '+IntToStr(otc[2])+'   '+IntToStr(otc[3]));
                          end;
     Button1.Show;       // Сделать видимой кнопку «Ввести запись»
end;
procedure TForm1.Button4Click(Sender: TObject);
// Сортировка записей
 vari,j : word;
      st : TStudent;
begin
 for i:=1 to nzap–1 do
// Сортировка массива записей
 forj:=i+1 tonzapdo
  if Stud[i].sball < Stud[j].sball then begin
        st:=Stud[i];
        Stud[i]:=Stud[j];
        Stud[j]:=st;
        end;
      Memo1.Clear;
  for i:=1 to nzap do
// Вывод в окно Memo1 отсортированных записей
        with stud[i] do
    Memo1.Lines.Add(IntToStr(i)+' '+fio+' '+FloatToStrf(sball,fffixed,4,2));
end;
procedure TForm1.Button5Click(Sender: TObject);
// Сохранение результатов сортировки в текстовом файле
var i:word;
begin
  if SaveDialog1.Execute then
// Выполнение стандартного диалога выбора имени файла
   begin
  FileNameT:= SaveDialog1.FileName;
// Возвращениеименидисковогофайла
   AssignFile(Ft, FileNameT);
// Связывание файловой переменной Ft с именем файла
      Rewrite(Ft);
// Открытие нового текстового файла
    end;
  for i:=1 to nzap do
        with stud[i] do  Writeln(Ft,i:4,'. ',fio,sball:8:2);
// Запись в текстовой файл
  CloseFile(Ft);
// Закрытие текстового файла
end;
procedure TForm1.BitBtn1Click(Sender: TObject);
begin
 CloseFile(fz);
// Закрытие файла записей при нажатии на кнопку «Close»
end;
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
 CloseFile(fz);
```

// Закрытие файла записей при нажатии на кнопку
end; end.

Индивидуальные задания

Ниже приведены 15 вариантов задач (табл. 9.1). По указанию преподавателя выберите свое индивидуальное задание. Уточните условие задания, количество, наименование, типы исходных данных.

В программе предусмотреть сохранение вводимых данных в файле и возможность чтения из ранее сохраненного файла.

Результаты выводить в окно просмотра и в текстовой файл.

Таблица 9.1
Индивидуальные задания

Вариант	Задание
1	В магазине формируется список лиц, записавшихся на покупку товара повышенного спроса. Каждая запись этого списка содержит: порядковый номер, ФИО, домашний адрес покупателя и дату постановки на учет. Удалить из списка все повторные записи, проверяя ФИО и домашний адрес
2	Список товаров, имеющихся на складе, включает наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Вывести в алфавитном порядке список товаров, хранящихся больше месяца, стоимость которых превышает 1 млн. руб.
3	Для получения места в общежитии формируется список студентов, который включает ФИО студента, группу, средний балл, доход на члена семьи. Общежитие в первую очередь предоставляется тем, у кого доход на члена семьи меньше двух минимальных зарплат, затем остальным в порядке уменьшения среднего балла. Вывести список очередности предоставления мест в общежитии
Вариант	Задание
4	В справочной автовокзала хранится расписание движения автобусов. Для каждого рейса указаны его номер, тип автобуса, пункт назначения, время отправления и прибытия. Вывести информацию о рейсах, которыми можно воспользоваться для прибытия в пункт назначения раньше заданного времени
5	На междугородной АТС информация о разговорах содержит дату разговора, код и название города, время разговора, тариф, номер телефона в этом городе и номер телефона абонента. Вывести по каждому городу общее время разговоров с ним и сумму
6	Информация о сотрудниках фирмы включает: ФИО, табельный номер, количество проработанных часов за месяц, почасовой тариф. Рабочее время свыше 144 часов считается сверхурочным и оплачивается в двойном размере. Вывести размер заработной платы сотрудника фирмы за вычетом подоходного налога, который составляет 12% суммы заработка
7	Информация об участниках спортивных соревнований содержит наименование страны, название команды, ФИО игрока, игровой номер, возраст, рост, вес. Вывести информацию о самой молодой, рослой и легкой команде
8	Для книг, хранящихся в библиотеке, задаются регистрационный номер книги, автор, название, год издания, издательство, количество страниц. Вывести список книг с фамилиями авторов в алфавитном порядке, изданных после заданного года
9	Различные цехи завода выпускают продукцию нескольких наименований. Сведения о выпущенной продукции включают наименование, количество, номер цеха. Для заданного цеха необходимо вывести количество выпущенных изделий по каждому наименованию в порядке убывания количества
10	Информация о сотрудниках предприятия содержит ФИО, номер отдела, должность, дату начала работы. Вывести списки сотрудников по отделам в порядке убывания стажа
11	Ведомость абитуриентов, сдавших вступительные экзамены в университет, содержит ФИО, адрес, оценки.

Вариант	Задание
	Определить количество абитуриентов, проживающих в г. Минске и сдавших экзамены со средним баллом не ниже 4.5, вывести их фамилии в алфавитном порядке

Вариант	Задание
12	В справочной аэропорта хранится расписание вылета самолетов на следующие сутки. Для рейса указаны: номер рейса, тип самолета, пункт назначения, время вылета. Вывести номера рейсов, типы самолетов и времена вылета для заданного пункта в порядке возрастания времени вылета
13	У администратора железнодорожных касс хранится информация о свободных местах в поездах дальнего следования на ближайшую неделю в следующем виде: дата выезда, пункт назначения, время отправления, число свободных мест. Оргкомитет международной конференции обращается к администратору с просьбой зарезервировать *m* мест до города *N* на *k*–й день недели с временем отправления поезда не позднее *t* часов вечера. Вывести время отправления или сообщение о невозможности выполнить заказ в полном объеме.
14	Ведомость абитуриентов, сдавших вступительные экзамены в университет, содержит ФИО абитуриента, оценки. Определить средний балл по университету и вывести список абитуриентов, средний балл которых выше среднего балла по университету. Первыми в списке должны идти студенты, сдавшие все экзамены на 5
15	Разработать программу формирования ведомости об успеваемости студентов. Каждая запись этой ведомости должна содержать: номер группы, ФИО студента, оценки за последнюю сессию. Вывести списки студентов по группам. В каждой группе ФИО студентов должны быть расположены в порядке убывания среднего балла

*Практическая работа 10.***Структурное программирование с использованием подпрограмм и модулей**

Цель работы: изучить возможности Delphi для написания подпрограмм и создания модулей. Составить и отладить программу, использующую внешний модуль UNIT с подпрограммой.

Индивидуальные задания

По указанию преподавателя выберите вариант задачи из заданий, приведенных в четвёртой работе. Предусмотрите возможность выбора функции, для которой будет рассчитываться таблица значений. Функции поместите в отдельный модуль. Вызывать выбранную функцию должна процедура, использующая в качестве входного параметра имя соответствующей функции.

***Практическая работа 11*. Программирование с использованием средств** графической информации

Цель работы: изучить возможности построения графиков с помощью компонента отображения графической информации TChart. Написать и отладить программу построения на экране графика заданной функции.

Индивидуальные задания

Постройте графики функций для соответствующих вариантов из четвертой лабораторной работы. Таблицу данных получить, изменяя параметр X с шагом h. Ввод исходных данных организовать через окна TEdit. Самостоятельно выбрать удобные параметры настройки.

Практическая работа 12. Использование OLE- и COM-технологий

программирования

Цель работы: познакомиться на конкретных примерах с технологиями программирования OLE и COM.

Краткие теоретические сведения

На протяжении многих лет программисты стремятся придумать способы, позволяющие использовать уже написанные коды. Например, если в программе нужна работа с таблицами, следует использовать MicrosoftExcel. Технология, позволяющая вызвать из вашей программы Excel или даже встроить это приложение в свою систему, называется OLE.

Еще одним шагом в эволюции программирования стала технология COM. По сути, именно она лежит в основе технологии OLE и именно с помощью нее реализуется сложное взаимодействие между приложениями, написанными разными программистами и на разных языках. На этом занятии предлагается рассмотреть использование COM-серверов Delphi для работы с Word и Excel.

Программа 1 (OLE-технологии)

Шаг 1. Создайте новую форму.

Шаг 2. Расположите на ней компоненты: OLEContainer (закладка System) и кнопку (Button).

Шаг 3. Создайте файл Excel. Занесите в него свои ФИО. Сохраниете его ‘H:\l.xls’.

Шаг 4. Создайте процедуру обработки сообщения о нажатии на кнопку:

```
procedure TForml.ButtonlClick(Sender: TObject); begin
OLEContainerl.CreateLinkToFile('H:\1.xls',FALSE); end;
```

Шаг 5. Скомпилируйте приложение. Нажмите на кнопку. Убедитесь, что в OLEContainer будет загружен созданный вами файл 1 .xls.

Шаг 6. Щелкните 2 раза на OLEConatainer, и вы увидите, что будет запущен MicrosoftExcel с загруженным в него вашим файлом. Внесите в файл изменения. Сохраните файл и закройте Excel. Нажмите в своем приложении кнопку и убедитесь, что в OLEContainer отобразились изменения файла (рис. 12.1).

Шаг 7. Измените процедуру обработки сообщения о нажатии на кнопку следующим образом:

```
procedure TForml.ButtonlClick(Sender: TObject);
begin
   OLEContainerl. CreateObj ectFromFile (' H::\l .xls ' , false) ;
end;
```

Шаг 8. Добавьте еще одну кнопку на форму и напишите для нее обработчик события нажатия на нее:

```
procedure TForml.Button2Click(Sender: TObject);
 begin
   OLEContainerl.Close;
end;
```

Шаг9. Скомпилируйте проект. Нажмите на кнопку «Загрузить файл». В OLEContainer'e отобразится содержимое файла.

Щелкните 2 раза на OLEContainer. Вы получите результат, показанный на рис. 12.2.

Excel будет встроен в ваше приложение, и в нем будет открыт ваш файл. Для того чтобы выйти из режима редактирования, нажмите кнопку «Закрыть файл».

Шаг 10. Теперь снова поменяйте процедуру для первой кнопки:

```
 procedure TForml.ButtonlClick(Sender: TObject);
 begin
    OLEContainerl.InsertObjectDialog;
 end;
```

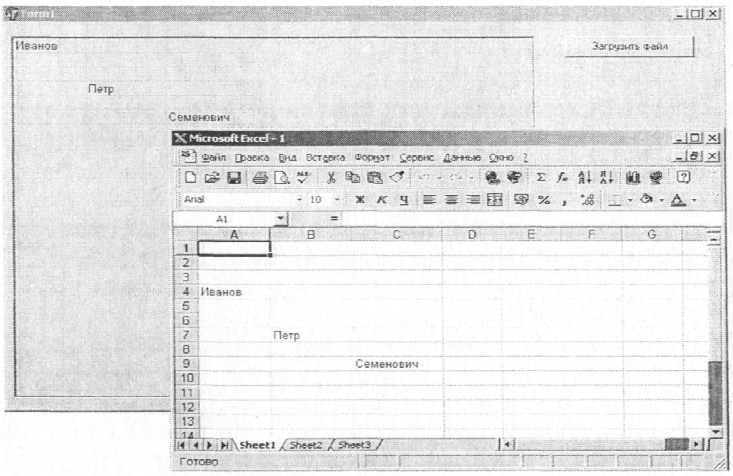


Рис. 12.1. Иллюстрация шага 6

Шаг 11. Скомпилируйте проект. Теперь при нажатии на кнопку «Загрузить файл» будет появляться стандартная форма (рис. 12.3), в которой показаны все программы, которые вы можете встраивать или связывать со своей программой.

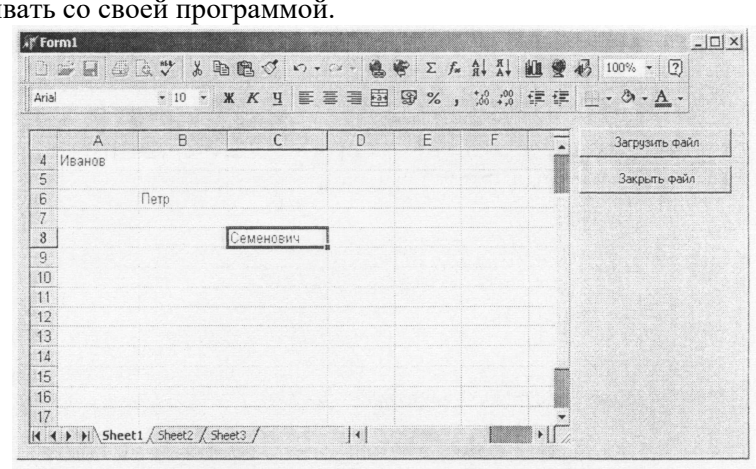


Рис. 12.2. Иллюстрация шага 9

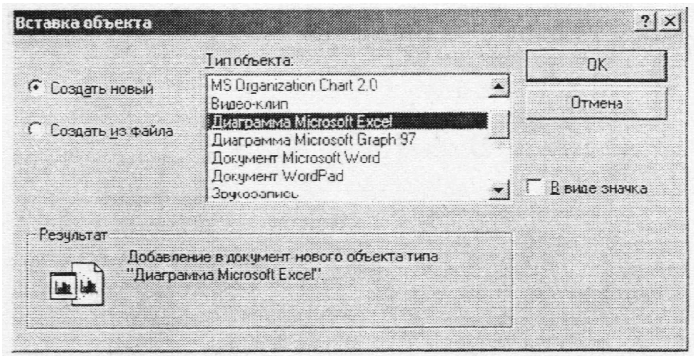


Рис. 12.3. Иллюстрация шага 10

Можно создать новый объект этих программ или использовать объекты из файла, также есть возможность связывать объект или внедрять в приложение (рис. 12.4).

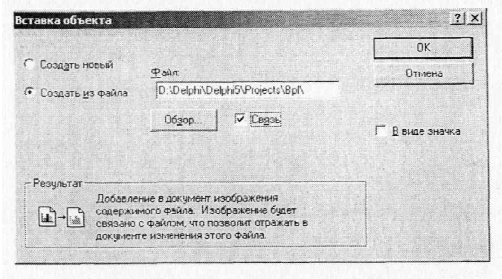


Рис. 12.4. Иллюстрация шага 11

Шаг 12.Поэкспериментируйте с различными приложениями.

Программа 2 (COM сервера Delphi)

Рассмотрим также, как можно использовать COM-технологии для внесения изменений в документы других программ.

Следующая программа предназначена для записи данных из вашей программы в документ Excel и считывания информации из документа.

1. Найдите на закладке Servers компонент ExcelApplication и поместите его на форму нового приложения. Установите свойство этого компонента AutoQuit=true.

2. Создайте файл H:\2.xls\.

3. Поместите на форму одну кнопку. Дайте ей название «Записать данные». Напишите для нее обработчик события нажатия на кнопку:

```
procedure TForml.ButtonlClick(Sender: TObject);
var Filenamel:01eVariant;
begin
ExcelApplicationl.Connect;
//Запись в существующий файл
Filenamel: = 'H:\2.xls ' ;
ExcelApplicationl.Workbooks.Open(Filenamel, EmptyParam,
EmptyParam, EmptyParam, EmptyParam, EmptyParam,
EmptyParam, EmptyParam, EmptyParam, EmptyParam,
EmptyParam, EmptyParam, false, 0);
//Установкацветазаливкиячейки Excel
ExcelApplicationl.Range[Edit2.Text,Edit2.Text].Interior.Colorlndex:=5;
//ЗанесениеинформациивопределеннуюячейкутаблицыExcelApplicationl.Range[Edit2.Text,Edit2.Tex
t].Value:= Editl.Text;
ExcelApplicationl.Disconnect;
end;
```

4. Добавьте на форму два компонента Label и Edit.

5. Скомпилируйте проект. Внесите текст для вставки в Excel и номер ячейки (он состоит из буквы – номер столбца, и цифры – номер строки) (рис. 12.5).

Рис. 12.5. Иллюстрация шага 5

После нажатия на кнопку сохраните файл. Потом откройте файл и убедитесь, что данные записаны в нужную ячейку.

6. Добавьте в предыдущую процедуру для наглядности следующую строку:

```
.....
ExcelApplicationl.Visible[0]:=TRUE;
//Установка цвета заливки ячейки Excel
.....
```

7. Скомпилируйте проект еще раз. Теперь Excel будет запущен не в фоновом режиме, как это было ранее.При программировании работы с Excel и Word рекомендуем написать макрос в самом Excel или Word, а потом перенести код макроса в свою программу с учетом небольших особенностей синтаксиса выбранного языка.

8. Измените название элементов на форме.

9. Измените процедуру нажатия на кнопку следующим образом:

```
procedure TFcrml.ButtonlClick(Sender: TObject);
var FileNamel: OleVariant;
begin
ExcelApplicationl.Connect;
Filename1: = 'D:\2.xls[1];
ExcelApplicationl.Workbooks.Open(Filenamel, EmptyParam,
EmptyParam, EmptyParam, EmptyParam, EmptyParam,
```

```
    EmptyParam,
    EmptyParam, EmptyParam, EmptyParam, EmptyParam,
    EmptyParam, false, 0);
    Editl.Text:=ExcelApplication1.Range[Edit2.Text, EmptyParam].Text;
    ExcelAppiicationl.Disconnect;
    end;
```

10. Создайте программу для записи информации в документ Word. Для этого измените форму проекта, как показано на рис. 12.6 (добавьте компоненты WordApplication и WordFont) и перепишите обработчик нажатия на кнопку:

```
procedure TForml.ButtonlClick(Sender: TObject);
begin
WordApplicationl.Connect;
//создание нового документа
WordApplicationl.Documents.Add(EmptyParam,EmptyParam);
WordApplicationl.Visible:=true;
//Установкашрифта
WordFontl .ConnectTo (WordApplicationl. Selection. Font);
WordFontl.Bold:=3;
WordFontl.Size:=17;
//Вставкатекста
WordApplicationl.Selection.InsertAfter(Editl.Text);
WordApplicationl.Disconnect;
end;
```

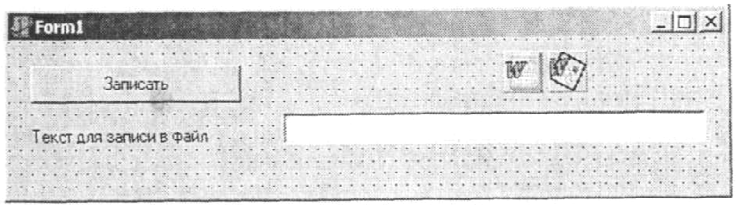


Рис. 12.6. Форма с компонентами WordApplication и WordFont

Индивидуальные задания

Ниже приведены 15 вариантов задач. По указанию преподавателя выберите свое индивидуальное задание. Уточните условие задания, количество, наименование, типы исходных данных.

Во всех заданиях скалярные переменные вводить с помощью компонента TEdit с соответствующим пояснением в виде компонента TLabel. Скалярный результат нужно вывести в виде компонента ExcelApplication в файл табличного процессора Excel или WordApplication в файл Word.

1. Дан массив из k символов. Вывести в файл табличного процессора сначала все цифры, входящие в него, а затем все остальные символы, сохраняя при этом взаимное расположение символов в каждой из этих двух групп.

2. Дан массив, содержащий от 1 до k символов, за которым следует точка. Напечатать этот текст в файл Word в обратном порядке.

3. Дан непустой массив из цифр. Вывести в файл Excel цифру, наиболее часто встречающуюся в этом массиве.

4. Отсортировать элементы массива X по возрастанию. Результат напечатать в файл Excel.

5. Элементы массива X расположить в обратном порядке. Результат напечатать в файл Excel.

6. Элементы массива X циклически сдвинуть на k позиций влево. Результат напечатать в файл Excel.

7. Элементы массива X циклически сдвинуть на k позиций вправо. Результат напечатать в файл Excel.

8. Преобразовать массив X по следующему правилу: все отрицательные элементы массива перенести в начало, а все остальные – в конец, сохраняя исходное взаимное расположение как среди отрицательных, так и среди остальных элементов. Результат напечатать в файл Excel.

9. Элементы каждого из массивов X и Y упорядочены по неубыванию. Объединить элементы этих двух массивов в один массив Z так, чтобы они снова оказались упорядоченными по неубыванию. Результат напечатать в файл Excel.

10. Дан массив из 4 символов. Определить, симметричен ли он, т.е. читается ли он одинаково слева направо и справа налево. Если симметричен, то вывести его в файл Word.

11. Дано два массива. Найти наименьшее среди тех элементов первого массива, которые не входят во второй массив. Напечатать эти наименьшие числа в файл Word.

12. Определить количество инверсий в этом массиве *X* (т.е. таких пар элементов, в которых большее число находится слева от меньшего: *x{i}>x{j}* при *i<j*). Напечатать эти пары в файл.

13. Дан массив из строчных латинских букв. Вывести в файл Word этот массив в алфавитном порядке все буквы, которые входят в этот текст по одному разу.

14. Вывести в файл Excel заданный массив из *k* символов, удалив из него повторные вхождения каждого символа.

15. Определить, сколько различных символов входит в заданный текст (текст вводится в файл Word и открывается посредством соответствующего компонента), содержащий не более *k* символов и оканчивающийся точкой (в сам текст точка не входит).

МДК 02.02

Практическая работа № 1. Тема: "Программирование ветвлений".

1. Даны целые числа m, n. Если числа не равны, то заменить каждое из них одним и тем же числом, равным большему из исходных, а если равны, то заменить числа нулями.

2. Подсчитать количество отрицательных среди чисел a, b, c.

3. Подсчитать количество целых среди чисел a, b, c.

4. Определить, делителем каких чисел a, b, c является число k.

5. Услуги телефонной сети оплачиваются по следующему правилу: за разговоры до A минут в месяц – B руб., а разговоры сверх установленной нормы оплачиваютсяиз расчета C руб. за минуту. Написать программу, вычисляющую плату за пользование телефоном для введенного времени разговоров за месяц.

6. Грузовой автомобиль выехал из одного города в другой со скоростью v1 км/ч. Через t ч в этом же направлении выехал легковой автомобиль со скоростью v2 км/ч. Составить программу, определяющую, догонит ли легковой автомобиль грузовой через t1 ч после своего выезда.

7. Перераспределить значения переменных x и y так, чтобы в x оказалось большее из этих значений, а в y – меньшее.

8. Определить правильность даты, введенной с клавиатуры (число – от 1 до 31, месяц – от 1 до 12). Если введены некорректные данные, то сообщить об этом.

9. Составить программу, определяющую результат гадания на ромашке – «любит – не любит», взяв за исходное данное количество лепестков n.

10. Написать программу – модель анализа пожарного датчика в помещении, которая выводит сообщение «Пожароопасная ситуация», если температура в комнате превысила 60° C.

11. Рис расфасован в два пакета. Масса первого – t кг, второго – n кг. Составить программу, определяющую: а) какой пакет тяжелее – первый или второй; б) массу более тяжелого пакета.

12. Написать программу, которая анализирует данные о возрасте и относит человека к одной из четырех групп: дошкольник, ученик, работник, пенсионер. Возраст вводится с клавиатуры.

13. Составить программу, определяющую, пройдет ли график функции $y = a \cdot x^2 + b \cdot x + c$ через заданную точку с координатами (m, n).

14. К финалу конкурса лучшего по профессии «Специалист электронного офиса» были допущены трое: Иванов, Петров, Сидоров. Соревнования проходили в три тура. Иванов в первом туре набрал m1 баллов, во втором – n1, в третьем – p1. Петров – m2, n2, p2 соот-

ветственно; Сидоров – m3, n3, p3. Составить программу, определяющую, сколько баллов набрал победитель.

15. Написать программу нахождения суммы большего и меньшего из трех чисел,

Практическая работа № 2. Тема: "Оператор выбора".

1. Написать программу, которая по номеру дня недели (целому числу от 1 до 7) выдает в качестве результата количество уроков в вашем классе в этот день.

2. Написать программу, позволяющую по последней цифре числа определить последнюю цифру его квадрата.

3. Составить программу, которая по заданным году и номеру месяца определяет количество дней в этом месяце.

4. Для каждой введенной цифры (0 – 9) вывести соответствующее ей название на английском языке (0 – zero, 1 – one, 1 – two,...).

5. Составить программу, которая по данному числу (1 – 12) выводит название соответствующего ему месяца.

6. Составить программу, позволяющую получить словесное описание школьных отметок (1 – «плохо», 2 – «неудовлетворительно», 3 – «удовлетворительно», 4 – «хорошо», 5 – «отличное).

7. Пусть элементами круга являются радиус (первый элемент), диаметр (второй элемент) и длина окружности (третий элемент). Составить программу, которая по номеру элемента запрашивала бы его соответствующее значение и вычисляла бы площадь круга.

8. Пусть элементами прямоугольного равнобедренного треугольника являются: 1) катет a; 2) гипотенуза b, 3) высота h, опущенная из вершины прямого угла на гипотенузу; 4) площадь S. Составить программу, которая по заданному номеру и значению соответствующего элемента вычисляла бы значение всех остальных элементов треугольника.

9. Написать программу, которая по номеру месяца выдает название следующего за ним месяца (при t = 1 получаем февраль, 4 – май и т.д.).

10. Написать программу, которая бы по введенному номеру времени года (1 – зима, 2 – весна, 3 – лето, 4 – осень) выдавала соответствующие этому времени года месяцы, количество дней в каждом из месяцев.

11. В старояпонском календаре был принят 12-летний цикл. Годы внутри цикла носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. Написать программу, которая вводит номер некоторого года и печатает его название по старояпонскому календарю. (Справка: 1996 г. – год Крысы – начало очередного цикла.)

12. Для целого числа k; от 1 до 99 напечатать фразу «Мне k лет», учитывая при этом, что при некоторых значениях k слово «лет» надо заменить на слово «год» или «года». Например, 11 лет, 22 года, 51 год.

13. Написать программу, которая бы по введенному номеру единицы измерения (1 – дециметр, 2 – километр, 3 – метр, 4 – миллиметр, 5 – сантиметр) и длине отрезка L выдавала бы соответствующее значение длины отрезка в метрах.

14. Написать программу, которая по вводимому числу от 1 до 11 (номеру класса) выдает соответствующее сообщение «Привет, k–классник». Например, если k = 1, «Привет, первоклассник»; если k = 4, «Привет, четвероклассник».

15. Написать программу, которая по введенному числу от 1 до 12 (номеру месяца) выдает все приходящиеся на этот месяц праздничные дни (например, если введено число 1, то должно получиться 1 января – Новый год, 7 января – Рождество).

16. Дано натуральное число N. Если оно делится на 4, вывести на экран ответ N=4k (где k – соответствующее частное); если остаток от деления на 4 равен 1, то N=4k+1; если остаток от деления на 4 равен 2, то N=4k+2; если остаток от деления на 4 равен 3, то N=4k+3. Например, 12 = 4*3, 22 = 4*5 + 2.

17. Имеется пронумерованный список деталей: 1) шуруп, 2) гайка, 3) винт, 4) гвоздь, 5) болт. Составить программу, которая по номеру детали выводит на экран ее название.

Практическая работа № 3. Тема: "Операторы цикла".

1. Даны положительные числа A и B (A > B). На отрезке длины A размещено максимально возможное количество отрезков длины B (без наложений). Не используя операции умножения и деления, найти длину незанятой части отрезка A.
2. Даны положительные числа A и B (A > B). На отрезке длины A размещено максимально возможное количество отрезков длины B (без наложений). Не используя операции умножения и деления, найти количество отрезков B, размещенных на отрезке A.
3. Даны целые положительные числа N и K. Используя только операции сложения и вычитания, найти частное от деления нацело N на K, а также остаток от этого деления.
4. Дано целое число N (> 0), являющееся некоторой степенью числа 2: . Найти целое число K – показатель этой степени.
5. Дано целое число N (> 0). Найти двойной факториал N: $N!! = N \cdot (N-2) \cdot (N-4) \ldots$ (последний сомножитель равен 2, если N – четное, и 1, если N – нечетное). Чтобы избежать целочисленного переполнения, вычислять это произведение с помощью вещественной переменной и вывести его как вещественное число.
6. Дано целое число N (> 0). Найти наименьшее целое положительное число K, квадрат которого превосходит N: $K^2 > N$. Функцию извлечения квадратного корня не использовать.
7. Дано целое число N (> 0). Найти наибольшее целое число K, квадрат которого не превосходит N: $K^2 \le N$. Функцию извлечения квадратного корня не использовать.
8. Дано целое число N (> 1). Найти наименьшее целое число K, при котором выполняется неравенство $3^K > N$.
9. Дано целое число N (> 1). Найти наибольшее целое число K, при котором выполняется неравенство $3^K < N$.
10. Дано целое число N (> 1). Вывести наименьшее из целых чисел K, для которых сумма 1 + 2 + … + K будет больше или равна N, и саму эту сумму.
11. Дано целое число N (> 1). Вывести наибольшее из целых чисел K, для которых сумма 1 + 2 + … + K будет меньше или равна N, и саму эту сумму.
12. Начальный вклад в банке равен 1000 руб. Через каждый месяц размер вклада увеличивается на P процентов от имеющейся суммы (P – вещественное число, 0 < P < 25). По данному P определить, через сколько месяцев размер вклада превысит 1100 руб., и вывести найденное количество месяцев K (целое число) и итоговый размер вклада S (вещественное число).
13. Спортсмен-лыжник начал тренировки, пробежав в первый день 10 км. Каждый следующий день он увеличивал длину пробега на P процентов от пробега предыдущего дня (P – вещественное, 0 < P < 50). По данному P определить, после какого дня суммарный пробег лыжника за все дни превысит 200 км, и вывести найденное количество дней K (целое) и суммарный пробег S (вещественное число).
14. Дано целое число N (> 0). Используя операции деления нацело и взятия остатка от деления, вывести все его цифры, начиная с самой правой (разряда единиц).
15. Дано целое число N (> 0). Используя операции деления нацело и взятия остатка от деления, найти количество и сумму его цифр.
16. Дано целое число N (> 0). Используя операции деления нацело и взятия остатка от деления, найти число, полученное при прочтении числа N справа налево.
17. Дано целое число N (> 0). С помощью операций деления нацело и взятия остатка от деления определить, имеется ли в записи числа N цифра «2». Если имеется, то вывести True, если нет – вывести False.
18. Дано целое число N (> 0). С помощью операций деления нацело и взятия остатка от деления определить, имеются ли в записи числа N нечетные цифры. Если имеются, то вывести True,

если нет – вывести False.

19. Дано целое число N (> 1). Если оно является простым, то есть не имеет положительных делителей, кроме 1 и самого себя, то вывести True, иначе вывести False.

20. Даны целые положительные числа A и B. Найти их наибольший общий делитель (НОД), используя алгоритм Евклида: НОД(A, B) = НОД(B, A mod B), если B ≠ 0; НОД(A, 0) = A, где «mod» обозначает операцию взятия остатка от деления.

Практическая работа № **4.** "Программирование массивов".

1. В одномерном массиве, состоящем из nвещественных элементов, вычислить: 1) сумму отрицательных элементов массива; 2) произведение элементов массива, расположенных между максимальным и минимальным элементами. Упорядочить элементы массива по возрастанию.

2. В одномерном массиве, состоящем из nвещественных элементов, вычислить: 1) сумму положительных элементов массива; 2)произведение элементов массива, расположенных между максимальным по модулю и минимальным, но модулю элементами. Упорядочить элементы массива по убыванию.

3. В одномерном массиве, состоящем из nцелых элементов, вычислить: 1) произведение элементов массива с четными номерами; 2) сумму элементов массива, расположенных между первым и последним нулевыми элементами. Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом — все отрицательные (элементы, равные 0, считать положительными).

4. В одномерном массиве, состоящем из nвещественных элементов, вычислить: 1) сумму элементов массива с нечетными номерами; 2) сумму элементов массива, расположенных между первым и последним отрицательными элементами. Сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями.

5. В одномерном массиве, состоящем из nвещественных элементов, вычислить: 1) минимальный элемент массива; 2) сумму элементов массива, расположенных между первым и последним положительными элементами. Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом — все остальные.

6. В одномерном массиве, состоящем из nцелых элементов, вычислить: 1) номер максимального элемента массива; 2) произведение элементов массива, расположенных между первым и вторым нулевыми элементами. Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине — элементы, стоявшие в четных позициях.

7. В одномерном массиве, состоящем из nвещественных элементов, вычислить: 1) номер минимального элемента массива; 2) сумму элементов массива, расположенных между первым и вторым отрицательными элементами. Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом — все остальные.

8. В одномерном массиве, состоящем из nвещественных элементов, вычислить: 1) максимальный по модулю элемент массива; 2) сумму элементов массива, расположенных между первым и вторым положительными элементами. Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.

9. В одномерном массиве, состоящем из nцелых элементов, вычислить: 1) минимальный по модулю элемент массива; 2) сумму модулей элементов массива, расположенных после первого элемента, равного нулю. Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине — элементы, стоявшие в нечетных позициях.

10. В одномерном массиве, состоящем из nвещественных элементов, вычислить: 1) номер минимального по модулю элемента массива; 2) сумму модулей элементов массива, расположенных после первого отрицательного элемента. Сжать массив, удалив из него все элементы, величина которых находится в интервале [a,b]. Освободившиеся в конце массива элементы заполнить нулями.

11. В одномерном массиве, состоящем из nвещественных элементов, вычислить: 1) номер максимального по модулю элемента массива; 2) сумму элементов массива, расположенных после первого положительного элемента. Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале [a,b], а потом — все остальные.

12. В одномерном массиве, состоящем из n вещественных элементов, вычислить: 1) количество элементов массива, лежащих в диапазоне от A до B; 2) сумму элементов массива, расположен-

ных после максимального элемента. Упорядочить элементы массива по убыванию модулей элементов.

13. В одномерном массиве, состоящем из nвещественных элементов, вычислить: 1) количество элементов массива, равных 0; 2) сумму элементов массива, расположенных после минимального элемента. Упорядочить элементы массива по возрастанию модулей элементов.

14. В одномерном массиве, состоящем из nвещественных элементов, вычислить: 1) количество элементов массива, больших C; 2) произведение элементов массива, расположенных после максимального по модулю элемента. Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом — все положительные (элементы, равные 0, считать положительными).

Практическая работа № **5.** "Рекурсия".

1. Описать рекурсивный алгоритм, осуществляющий перевод натурального числа из одной системы счисления в другую.

2. Задана последовательность натуральных чисел. Требуется определить количество пар взаимно простых чисел в этой последовательности.

3. Дано натуральное число. Составить рекурсивную процедуру вывода на экран цифр заданного натурального числа, определить количество и сумму его цифр.

4. Определить цифровой корень заданного натурального числа. Например, для числа 123456 цифровой корень равен 3, т.к. 1+2+3+4+5+6=21, 2+1=3.

5. Описать рекурсивную логическую функцию Simm(s,i,j), проверяющую, является ли симметричной часть строки s, начинающаяся i-м и кончающаяся j-м ее элементами.

6. Описать рекурсивную функцию DigitCount(S) целого типа, которая находит количество цифр в строке S, не используя оператор цикла. С помощью этой функции найти количество цифр в каждой из пяти данных строк.

7. Дано натуральное число n. Вычислить

8. $1 \times 2 + 2 \times 3 + 3 \times 4 + \ldots + (n-1) \times n.$

9. Найти значение следующего выражения:

10. $$\cfrac{1}{1+\cfrac{1}{3+\cfrac{1}{5+\cfrac{1}{\ldots}}}}$$

$$\ldots$$

$$101+\frac{1}{103}$$

11. Дано действительное число x=0. Вычислить

$$\cfrac{x}{x^2+\cfrac{2}{x^2+\cfrac{4}{x^2+\cfrac{8}{\ldots}}}}$$

12. $x^2+\frac{256}{x^2}$

13. Найти сумму ряда:

14. $1+\frac{1}{2!}+\frac{1}{3!}+\frac{1}{4!}+\ldots$

15. Вычислите, используя рекурсию при n=5, 10 (n – количество членов последовательности:

а) $\sqrt{1+2\sqrt{1+3\sqrt{1+4\sqrt{1+5\sqrt{1+\ldots}}}}}$;

б) $\sqrt{8-\sqrt{8+\sqrt{8-8-\sqrt{8+\ldots}}}}$;

в) $\sqrt{11-2\sqrt{11+2\sqrt{11-2\sqrt{11-2\sqrt{11+\ldots.}}}}}$;

г) $\sqrt{23-2\sqrt{23+2\sqrt{23+2\sqrt{23-2\sqrt{23+\ldots}}}}}$;

д) $\sqrt{6+2\sqrt{7+3\sqrt{8+4\sqrt{9+\ldots}}}}$;

е) $\sqrt{1+(n+1)\sqrt{1+(n+2)\sqrt{1+(n+3)\sqrt{1+\ldots}}}}$.

16.

17. Реализовать средствами языка C# алгоритм «Ханойские башни»: «Имеется три стержня A, B, C. На стержне A имеется n дисков радиуса 1, 2,..., n таким образом, что диск радиуса i является i-м сверху. Требуется переместить все диски на стержень B, сохраняя их порядок расположения (диск с большим радиусом находится ниже)»

МДК 02.03

Практическая работа. «Построение простейших математических моделей.»
Построить математические модели следующих задач.

Задача 1

Оператор связи оказывает 2 вида услуг:

1. Предоставление одной линии телефонной сети общего пользования (ТСОП) и трёх линий цифровой связи (ЦС);
2. Предоставление одной линии ЦС и двух линий ТСОП.

Стоимость услуг указана в табл. 1:

Таблица 1

	ТСОП	ЦС	Цена
Услуга 1	1	3	750
Услуга 2	2	1	600

Сети связи и эксплуатируемое оборудование накладывает следующие ограничения на количество используемых линий связи:

ТСОП ≤ 300

ЦС ≤ 120

ТСОП+2*ЦС ≤ 380

Определить оптимальное соотношение услуг 1 и 2, которые оператор должен предоставлять для получения максимальной выручки.

Задача 2Для изготовления продукции двух видов А и В предприятие расходует ресурсы, а от реализации выпущенной продукции получает доход. Информация о нормах затрат ресурсов на одно изделие, запасах расходуемых ресурсов и цен изделий приведена в таблице.

задача3.

Фирма собирается организовать в следующем квартале выпуск новых изделий А и В. Она располагает необходимым сырьем и оборудованием и может привлечь рабочих на условиях почасовой оплаты. Для оплаты их труда фирма может получить в банке кредит на три месяца под 40% годовых. Информация о нормах затрат ресурсов, наличных объемах сырья и оборудования, а также ценах изделий А и В приведена в таблице. Целью организации выпуска продукции является получение максимальной суммарной прибыли.

Практическая работа. «Сведение произвольной задачи линейного программирования к основной задаче линейного программирования»

Привести задачи к основной задаче линейного программирования.

Задача 1.Для изготовления продукции двух видов А и В предприятие расходует ресурсы, а от реализации выпущенной продукции получает доход. Информация о нормах затрат ресурсов на одно изделие, запасах расходуемых ресурсов и цен изделий приведена в таблице.

Задача2.

Фирма собирается организовать в следующем квартале выпуск новых изделий А и В. Она располагает необходимым сырьем и оборудованием и может привлечь рабочих на условиях почасовой оплаты. Для оплаты их труда фирма может получить в банке кредит на три месяца под 40% годовых. Информация о нормах затрат ресурсов, наличных объемах сырья и оборудования, а также ценах изделий А и В приведена в таблице. Целью организации выпуска продукции является получение максимальной суммарной прибыли.

Практическая работа. «Решение простейших однокритериальных задач графическим методом».

Решить графическим методом задачу программирования.

Задача 1..Для изготовления продукции двух видов А и В предприятие расходует ресурсы, а от реализации выпущенной продукции получает доход. Информация о нормах затрат ресурсов на одно изделие, запасах расходуемых ресурсов и цен изделий приведена в таблице.

Задача2.

Фирма собирается организовать в следующем квартале выпуск новых изделий А и В. Она располагает необходимым сырьем и оборудованием и может привлечь рабочих на условиях почасовой оплаты. Для оплаты их труда фирма может получить в банке кредит на три месяца под 40% годовых. Информация о нормах затрат ресурсов, наличных объемах сырья и оборудования, а также ценах изделий А и В приведена в таблице. Целью организации выпуска продукции является получение максимальной суммарной прибыли.

.

Решить графическим методом задачу программирования.

Задача 1..Для изготовления продукции двух видов А и В предприятие расходует

ресурсы, а от реализации выпущенной продукции получает доход. Информация о нормах затрат ресурсов на одно изделие, запасах расходуемых ресурсов и цен изделий приведена в таблице.

Задача2.

Фирма собирается организовать в следующем квартале выпуск новых изделий А и В. Она располагает необходимым сырьем и оборудованием и может привлечь рабочих на условиях почасовой оплаты. Для оплаты их труда фирма может получить в банке кредит на три месяца под 40% годовых. Информация о нормах затрат ресурсов, наличных объемах сырья и оборудования, а также ценах изделий А и В приведена в таблице. Целью организации выпуска продукции является получение максимальной суммарной прибыли.

Практическая работа«Решение простейших однокритериальных задач симплекс-методом».

Задача 1..Для изготовления продукции двух видов А и В предприятие расходует ресурсы, а от реализации выпущенной продукции получает доход. Информация о нормах затрат ресурсов на одно изделие, запасах расходуемых ресурсов и цен изделий приведена в таблице.

Задача2.

Фирма собирается организовать в следующем квартале выпуск новых изделий А и В. Она располагает необходимым сырьем и оборудованием и может привлечь рабочих на условиях почасовой оплаты. Для оплаты их труда фирма может получить в банке кредит на три месяца под 40% годовых. Информация о нормах затрат ресурсов, наличных объемах сырья и оборудования, а также ценах изделий А и В приведена в таблице. Целью организации выпуска продукции является получение максимальной суммарной прибыли.

Практическая работа. «Решение простейших однокритериальных задач симплекс-методом с искусственным базисом»

Задача1.

Решить задачу ЛП, найдя начальный опорный план методом искусственного базиса. Определим максимальное значение целевой функции $F(X) = 3x_3 - 2x_4 - x_5$ при следующих условиях:

$2x_1 + x_2 + x_3 + x_4 + 3x_5=5$

$3x_1 + 2x_3 - x_4 + 6x_5=7$

$x_1 - x_3 + 2x_4 + x_5=2$

Задача 2.

Задача 2. Решить задачу линейного программирования симплекс-методом.

$$f = 2X_1 + X_2 - 2X_3 \rightarrow \min$$

$$\begin{cases} X_1 + X_2 - X_3 \geq 8; \\ X_1 - X_2 + 2X_3 \geq 2; \\ -2X_1 - 8X_2 + 3X_3 \geq 1; \\ X_i \geq 0 (i = 1, 2, 3). \end{cases}$$

Задача 3. Предприятие производит 3 вида продукции: А1, А2, А3, используя сырьё двух типов. Известны затраты сырья каждого типа на единицу продукции, запасы сырья на планируемый период, а также прибыль от единицы продукции каждого вида.

Сырьё	Затраты сырья на единицу продукции			Запас сырья
	А1	А2	А3	
I	3,5	7	4,2	1400
II	4	5	8	2000
Прибыль от ед. прод.	1	3	3	

1. Сколько изделий каждого вида необходимо произвести, чтобы получить максимум прибыли?
2. Определить статус каждого вида сырья и его удельную ценность.
3. Определить максимальный интервал изменения запасов каждого вида сырья, в пределах которого структура оптимального плана, т.е. номенклатура выпуска, не изменится.

4. Определить количество выпускаемой продукции и прибыль от выпуска при увеличении запаса одного из дефицитных видов сырья до максимально возможной (в пределах данной номенклатуры выпуска) величины.
5. Определить интервалы изменения прибыли от единицы продукции каждого вида, при которых полученный оптимальный план не изменится.

Практическая работа. «Нахождение начального решения транспортной задачи методом северо-западного угла.»

Задача 1.Из трех холодильников A_i, i=1..3, вмещающих мороженную рыбу в количествах a_i т, необходимо последнюю доставить в пять магазинов B_j, j=1..5 в количествах b_i т. Стоимости перевозки 1т рыбы из холодильника A_i в магазин B_i заданы в виде матрицы C_{ij}, 3x5. Написать математическую модель задачи и спланировать перевозки так, чтобы их общая стоимость была минимальной.

$$a = (15, 25, 10),$$
$$b = (2, 20, 18)$$
$$C = \begin{pmatrix} 2 & 5 & 7 \\ 8 & 12 & 2 \\ 1 & 3 & 8 \end{pmatrix}$$

Построить начальное распределение транспортной задачи методом северо-западного угла.

Задание 2. Найти первоначальный опорный план .

a_i / b_j	500	400	300
200	1	5	6
300	2	6	7
500	3	7	8

Практическая работа. «Нахождение начального решения транспортной задачи методом минимальной стоимости.»

Задача 1.Из трех холодильников A_i, i=1..3, вмещающих мороженную рыбу в количествах a_i т, необходимо последнюю доставить в пять магазинов B_j, j=1..5 в количествах b_i т. Стоимости перевозки 1т рыбы из холодильника A_i в магазин B_i заданы в виде матрицы C_{ij}, 3x5. Написать математическую модель задачи и спланировать перевозки так, чтобы их общая стоимость была минимальной.

Построить начальное распределение транспортной задачи методом минимальной стоимости.

$$a = (15, 25, 10),$$
$$b = (2, 20, 18)$$
$$C = \begin{pmatrix} 2 & 5 & 7 \\ 8 & 12 & 2 \\ 1 & 3 & 8 \end{pmatrix}$$

Задание 2. Найти первоначальный опорный план .

a_i / b_j	500	400	300
200	1	5	6
300	2	6	7
500	3	7	8

Практическая работа. «Решение транспортной задачи методом потенциалов»

Задача 1.Из трех холодильников A_i, i=1..3, вмещающих мороженную рыбу в количествах a_i т, необходимо последнюю доставить в пять магазинов B_j, j=1..5 в количествах b_i т. Стоимости перевозки 1т рыбы из холодильника A_i в магазин B_j заданы в виде матрицы C_{ij}, 3x5.
Написать математическую модель задачи и спланировать перевозки так, чтобы их общая стоимость была минимальной.

Построить начальное распределение транспортной задачи методом северо-западного угла.

$$a = (15, 25, 10),$$

$$b = (2, 20, 18)$$

$$C = \begin{pmatrix} 2 & 5 & 7 \\ 8 & 12 & 2 \\ 1 & 3 & 8 \end{pmatrix}$$

Задание 2.

a_i / b_j	500	400	300
200	1	5	6
300	2	6	7
500	3	7	8

Практическая работа. Графический метод решения задач нелинейного программирования

Решить задачу нелинейного программирования

$$f = 3x_1^2 + 2x_2^2 - 3x_1 + 1$$
$$x_1^2 + x_2^2 = 4$$
$$x_1, x_2 \geq 0$$

Практическая работа. Метод множителей Лагранжа.

Решить методом множителей Лагранжа:

$$f = 3x_1^2 + 2x_2^2 - 3x_1 + 1$$
$$x_1^2 + x_2^2 = 4$$
$$x_1, x_2 \geq 0$$

Практическая работа. «Исследование характеристик случайного потока требований в телекоммуникационной системе».

Реальный поток вызовов **X(t)**, поступающий на коммутатор АТС за интервал вре-

мени (0, t), аппроксимируется пуассоновским законом распределения вероятностей появления **i** вызовов

$$P\{X(t)=i,\lambda t\}=P(i,\lambda t)=\frac{(\lambda t)^i}{i!}e^{-\lambda t}$$

здесь i = (0, 1, 2, …, ∞); λt – параметр закона распределения; λ – интенсивность (плотность) потока, т.е. среднее число вызовов в единицу времени t.

Такой поток называется простейшим потоком требований (ППТ). Он обладает свойствами стационарности, ординарности, отсутствия последействия.

Задание

1. Вычислить вероятности P(i, λt) для i = (0, 1, 2, …, 10) и заданного параметра λt**,** результаты свести в таблицу,построить в масштабе график закона распределения вероятностей.
2. Определить наивероятнейшее число вызовов за интервал (0, t).
3. Вычислить и построить график функции распределения вероятностей

$$F(i,\lambda t)=P\{X(t)<i,\lambda t\}=\sum_{k=0}^{i-1}P(k,\lambda t)$$

для i = (1, 2, …, 10).

4. Вычислить вероятности того, что за интервал времени (0, t) поступит:
   - точно «k» вызовов;
   - менее «k» вызовов;
   - более «k» вызовов.
5. Вычислить среднее значение всех вызовов, мощность разброса и среднеквадратическое отклонение (m_i, D_i, σ_i) за интервал времени t
6. Вычислить вероятность того, что за время t число вызовов будет лежать в интервале значений (k-1 <i ≤ k + 2)

Описание задачи 2.

К оператору поступает случайный поток телеграмм. Обработка оператором одной телеграммы занимает ровно h минут. Плотность распределения вероятностей случайного интервала Z между соседними требованиями определяется соотношением [1. с.80, 82]

$$W_z(t)=\begin{cases}0, t\le 0\\ at, 0<t\le b\\ 0, t>b\end{cases}$$

Данный поток требований можно охарактеризовать точечным случайным процессом, у которого все интервалы Z распределены одинаково (может быт за исключением первого интервала). Такие потоки требований называются потоками Пальма

Задание

1. Построить график функции $W_z(t)$ с соблюдением масштаба, определив значение “α” из условия нормировки.
2. Вычислить:
   - среднее значение интервала Z, мощность разброса и среднеквадратическое отклонение случайной величины Z от своего среднего значения (m_z, D_z, σ_z),
   - интенсивность потока телеграмм λ;
   - вероятность того, что к приходу очередной телеграммы оператор будет свободен и требование будет обслужено, то есть $\mathbf{P_{об} = P\{Z>h\}}$;
   - вероятность отказа $P_{отк}$ в обслуживании очередной телеграммы, так как оператор будет занят, то есть $P_{отк} =$ **P{Z<h}**;
   - среднее число $n_{ср}$ поступивших к оператору телеграмм за время работы $t_{раб}$;
   - среднее число телеграмм, которым было отказано в обслуживании за время $t_{раб}$.

ТАБЛИЦА ЗАДАНИЙ

Группа	λt	k	b, мин	h, мин	$t_{раб}$, час

1	3+0,1№	4	10+№	5	2
2	2,5+0,1№	5	10+0,5№	4	1,5
3	2,8+0,1№	4	20+№	10	1,5
4	2,5+0,15 №	3	8+№	3	1

Вычисления проводить с точностью до 4-х знаков после запятой.

Практическая работа. «Исследование характеристик случайного потока освобождений каналов в телекоммуникационной системе».

В аппаратном зале обработки пакетов сообщений имеется n идентичных процессоров (каналов обслуживания). Экспериментально установлено, что функцию распределения времени обслуживания T_j одним каналом можно аппроксимировать показательным законом

$$F_{Tj}(t) = 1 - e^{-\nu_j t}, t \geq 0$$

Где $\nu_j = 1/T_{jcp}$ – интенсивность обслуживания (интенсивность освобождения канала), T_{jcp} – среднее время обслуживания одним каналом.

Многоканальная СМО работает так, что освободившийся канал тут же занимается новым требованием. В этом случае вероятность того, что за промежуток времени (0, t) произойдет i освобождений каналов описывается законом распределения Пуассона.

$$P_i(t) = \frac{(\nu t)^i}{i!} e^{-\nu t}, t \geq 0,$$

а функция распределения времени обслуживания T задается выражением

$$F_T(t) = 1 - e^{-\nu t}, t \geq 0,$$

где $\nu = n\nu_j$ – интенсивность обслуживания n-канальной системой, если все каналы идентичны.

Задание

1. Определить плотность вероятности времени T обслуживания n-канальной СМО, построить график $W_T(t)$, $t \geq 0$ для интервала значений t от 0 до $3/\nu$ в удобном масштабе.
2. Определить математическое ожидание m_T, дисперсию D_T, и среднеквадратическое отклонение σ_T, дать их толкование.
3. Определить вероятности того, что за время t в системе:
   - не освободится ни один канал;
   - освободится хотя бы один канал;
   - освободится точно один, два, пять каналов;
   - освободится более одного канала.
4. По графику плотности вероятности $W_T(t)$ геометрически определить вероятности **P{i=0,** νt **}**, **P{i ≥1,** νt **}**, как соответствующие площади под кривой $W_T(t)$. Сравните результаты, полученные геометрически с результатами, полученными аналитически в п.3 задания.
5. Определить среднее число освобождений L, то есть число обслуженных пакетов за время работы $t_{раб.}$
6. В случайный момент времени в n-канальную СМО поступает требование (пакет сообщений). Найти закон распределения времени ожидания $T_{ож}$, которое требованию придется ждать, и вычислить вероятность того, что требование будет принято к обслуживанию, то есть освободится какой-либо канал не позднее t_1.

ТАБЛИЦА ЗАДАНИЙ

Группа	ν_i , 1/мин	n	t, сек	$t_{раб}$,сек	t_1, сек
1	100 – 3№	25	4	30	5
2	130 - 4№	20	3,5	40	4
3	80 – 2,5№	30	5	40	6
4	150 – 4,5№	20	3	30	3

Практическая работа. Прогнозирование временных рядов с использованием скользящей средней

Задание.

Приведены данные продаж на конец недели. Данные записывались в течение 18 недель.

1. Постройте график и определите скользящее среднее
2. Выполните экспоненциальное сглаживание с константой сглаживания α =0,1
3. Найдите оптимальную константу экспоненциального сглаживания.
4. Выполните экспоненциальное сглаживание с полученной константой сглаживания α.
5. Какой из построенных наборов сглаженных данных подходит наилучшим образом для определения асимптотического ряда?

неделя	
1	33
2	35
3	37
4	40
5	38
6	43
7	44
8	45
9	42
10	47
11	48
12	49
13	53
14	56
15	60
16	66

17 7
2
18 7
3

Практическая работа. Прогнозирование временных рядов с использованием тренда.

Приведены данные продаж на конец недели(см.пред.занятие). Данные записывались в течение 18 недель.

Постройте линейную модель и выполните прогноз на 2 недели.

Практическая работа. Элементы теории игр

Постановка задачи

Предприятие может выпускать несколько видов продукции A1, A2, A3, …, получая при этом прибыль. Величина прибыли определяется состоянием спроса («природой» рынка), который может находится в одном из нескольких возможных состояний: B1, B2, B3, …

Зависимость величины прибыли от вида выпускаемой продукции и состояния рынка представлена в платежных матрицах.

	B1	B2	B3	B4	B5
A1	196	428	464	212	320
A2	476	386	335	203	479
A3	307	228	424	377	250
A4	93	143	195	163	179
A5	220	309	363	156	289
A6	330	339	362	420	468
A7	447	221	281	482	181
A8	282	316	229	385	243
A9	230	360	490	408	447

Рассмотрите таблицу как матричную игру «предприятие (игрок А) против «природы» рынка (игрок В)». Для заданной платежной матрицы:

1. Найдите нижнюю и верхнюю цену игры;
2. Определите оптимальные смешанные стратегии игроков с помощью сведения игры к задаче линейного программирования;
3. Интерпретируйте полученные результаты применительно к рассматриваемой экономической задаче.