

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Сыров Игорь Анатольевич
Должность: Директор
Дата подписания: 21.08.2025 20:19:08
Уникальный программный ключ:
b683afe664d7e9f64175886cf9626a196149ad36

СТЕРЛИТАМАКСКИЙ ФИЛИАЛ
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«УФИМСКИЙ УНИВЕРСИТЕТ НАУКИ И ТЕХНОЛОГИЙ»

Факультет Математики и информационных технологий
Кафедра Прикладной информатики и программирования

Оценочные материалы по дисциплине (модулю)

дисциплина **Программирование**

Блок Б1, обязательная часть, Б1.О.19

цикл дисциплины и его часть (обязательная часть или часть, формируемая участниками образовательных отношений)

Направление

02.03.03 Математическое обеспечение и администрирование информационных систем

код наименование направления

Программа

Сетевое программирование и администрирование информационных систем

Форма обучения

Очная

Для поступивших на обучение в
2020 г.

Разработчик (составитель)
к.ф.-м.н., доцент
Кильдибаева С. Р.
ученая степень, должность, ФИО

1. Перечень компетенций, индикаторов достижения компетенций и описание показателей и критериев оценивания результатов обучения по дисциплине (модулю).....	3
2. Оценочные средства, необходимые для оценки результатов обучения по дисциплине (модулю).....	8
3. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине (модулю), описание шкал оценивания	66

1. Перечень компетенций, индикаторов достижения компетенций и описание показателей и критериев оценивания результатов обучения по дисциплине (модулю)

Формируемая компетенция (с указанием кода)	Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине (модулю)	Показатели и критерии оценивания результатов обучения по дисциплине (модулю)				Вид оценочного средства
			1	2	3	4	
			неуд.	удовл.	хорошо	отлично	
ОПК-3. Способен применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	ОПК-3.1. Знает основные положения и концепции прикладного и системного программирования, архитектуры компьютеров и сетей (в том числе и глобальных), современные языки программирования, технологии создания и эксплуатации программных продуктов и программных комплексов.	Обучающийся должен: Обучающийся должен знать: основные алгоритмические конструкции и основные операторы языков программирования C# и C++; основные структуры данных, способы их создания и обработки; основы структурного, модульного и визуального	Не может раскрыть основное содержание учебного материала; незнание или непонимание большей или наиболее важной части учебного материала; незнание структуры программы на языках Pascal, C++	Знание содержания основных понятий и методов программирования; отвечает только на конкретный вопрос, соединяет знания из разных разделов курса только при наводящих вопросах; показывает знание основного материала в объеме, необходимом	Показывает полное знание программного материала, основной и дополнительной литературы; демонстрирует понимание приобретенных знаний и умений для будущей профессиональной деятельности	Демонстрирует системное и глубокое знание программного материала, свободное владение материалом из различных разделов курса, твердое знание лекционного материала, обязательной и рекомендованной литературы.	Компьютерные тесты

		<p>программирован ия; основные этапы разработки алгоритмов и программ; современные языки программирован ия и пакеты программ в области программирован ия; современные способы и средства работы с информацией для изучения основных конструкций современных языков программирован ия; общие принципы и навыки практического применения объектно- ориентированно го программирован</p>		<p>для предстоящей профессиональн ой деятельности.</p>			
--	--	---	--	--	--	--	--

		ия.					
ОПК-3.2. Умеет использовать их в профессиональной деятельности.	Обучающийся должен : уметь выбирать структуры данных, необходимые для решения поставленной задачи; составлять алгоритмы обработки данных; разрабатывать программы для ЭВМ на языках программирования C# и C++, проводить их отладку и тестирование; ориентироваться в постановках задач, при решении поставленных задач обоснованно строить алгоритмы,	Не способен аргументировано и последовательно излагать материал, допускает грубые ошибки в ответах; не может самостоятельно выполнить программную реализацию даже простых задач	Неполно или непоследовательно раскрывает содержание материала, но показывает общее понимание вопроса и демонстрирует умения, достаточные для дальнейшего усвоения материала; не может применить теорию в новой ситуации; может представить работоспособную программу для решения задачи и объяснить решение; с трудом применяются некоторые формы	Демонстрирует умение анализировать материал, однако не все выводы носят аргументированный и доказательный характер; может провести анализ ситуации, сравнение, обобщение и т.д., но не всегда делает это самостоятельно; умеет применять полученные знания на практике, получает верные, но не всегда эффективные программные решения.	Показывает умение иллюстрировать теоретические положения конкретными примерами, применять их в новой ситуации; демонстрирует умение разрабатывать эффективные алгоритмы и программные решения (предлагает при этом несколько вариантов решения и выбирает лучшее); демонстрирует различные формы мыслительной деятельности: анализ, синтез, сравнение,	Аудиторные и домашние контрольные работы	

		реализовывать их на языках программирования C# и C++.		мыслительной деятельности: анализ, синтез, сравнение, обобщение и т.д.		обобщение и т.д.	
ОПК-3.3. Имеет практические навыки разработки программного обеспечения.	Обучающийся должен: владеть методами использования в профессиональной деятельности языков программирования C# и C++; современными методами практического программирования конкретных задач в определенной языковой среде; умениями и навыками использования библиотек объектов (классов) для решения практических	Допускает ошибки в определении понятий, при использовании терминологии, которые не исправлены после нескольких наводящих вопросов; не владеет навыками составления алгоритмов, не может объяснить предложенное решение.	Не в полной мере демонстрирует способность применять теоретические знания для анализа практических ситуаций; с трудом может соотнести теорию и практические примеры из учебных материалов (при этом примеры не всегда правильные); редко использует при ответе термины, подменяет одни понятия	При изложении материала допускает небольшие пробелы, не искажающие содержание ответа; имеются незначительные ошибки в формулировке понятий, присутствуют неточности при описании отдельных алгоритмов, однако они могут быть исправлены студентом при их обнаружении.	Демонстрирует способность творчески применять знание теории к решению профессиональных задач; уверенно ориентируется в проблемных ситуациях; владеет аргументацией, грамотной, лаконичной, доступной и понятной речью.	Лабораторные работы	

		задач; современными методами и приемами эффективной обработки информации средствами языков программирован ия.		другими, не всегда понимая разницы.			
--	--	---	--	---	--	--	--

2. Оценочные средства, необходимые для оценки результатов обучения по дисциплине (модулю)

Задачи для проведения практических занятий, домашних и аудиторных контрольных работ, а также задания к лабораторным работам представлены в следующих учебно-методических материалах:

1. Дейл Н. Программирование на С++ [Электронный ресурс]: учебник / Дейл Н., Уимз Ч., Хедингтон М. – Электрон. дан. – М.: ДМК Пресс, 2007. – 672 с. – Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=1219
2. Дмитриев В.Л. Теория и практика решения задач по программированию. – Уфа: РИЦ БашГУ, 2010. – 264 с. [Электронный ресурс] – Режим доступа: <https://sspaedu.bibliotech.ru/Reader/Book/2013102812425175582000005954> (28.08.2018)

Ниже приведены примеры типовых заданий к контрольным работам, типовые задания к лабораторным работам, задания повышенной сложности и материалы для проведения тестирования (база тестов программы MD Test).

Тестовые задания

*Тестовые материалы (тест №1) по разделу
«Основные приёмы программирования на языке С++. Структуры данных.»*

1. Структура программы на языке С++ и типы данных.

1.1.* *Выбрать один вариант ответа*

Набор символов, используемых в языке программирования:

- а) алфавит
- б) зарезервированные символы
- в) лексемы
- г) таблица соответствия

1.2.* *Выбрать правильные варианты ответа*

Алфавит С++ включает:

- а) латинские буквы от а до z (строчные и заглавные)
- б) русские буквы от а до я (строчные и заглавные)
- в) знаки арифметических действий (сложение, вычитание, умножение, деление)
- г) цифры от 0 до 9

1.3.* *Выбрать правильные варианты ответа*

Переменные, используемые в программе:

- а) могут быть объявлены без указания их типа (тип должен быть указан позднее)
- б) могут быть объявлены в любой части программы по мере необходимости
- в) могут содержать в себе знак подчеркивания «_»
- г) могут содержать в себе символ «пробел»

1.4.* *Выбрать правильные варианты ответа*

Зарезервированные слова:

- а) должны быть объявлены в разделе описания
- б) не могут использоваться в качестве идентификаторов имен переменных
- в) могут использоваться в качестве идентификаторов имен переменных
- г) в окне редактора С++ выделяются цветом

1.5.* *Выбрать правильный вариант ответа*

Для доступа к глобальной переменной из блока, внутри которого описана локальная переменная, имя которой совпадает с глобальной переменной, используется операция:

- а) тернарная операция «?:»
- б) операция разрешения области действия «::»
- в) в одной программе не может быть локальной и глобальной переменных с одинаковыми именами
- г) доступ к глобальной переменной в таком случае нельзя осуществить

1.6.* *Выбрать правильные варианты ответа*

Какие фрагменты кода программы не содержат ошибок:

- а) `int a=10;`
- б) `float a=b=c=10;`
- в) `int a,b,c;`
`a=b=c=12;`
- г) `char s[20];`

1.7.* *Выбрать правильные варианты ответа*

Укажите правильные (допустимые) имена переменных:

- а) `F_Name`
- б) `45a`
- в) `Count2`
- г) `void`

1.8.* *Выбрать правильные варианты ответа*

Укажите правильные (допустимые) имена переменных:

- а) `Code_1`
- б) `Сумма`
- в) `123`
- г) `Sum`

1.9.*** *Выбрать правильный вариант ответа*

Какой объем памяти требуется при работе с переменной для типа, описанного ниже:

```
struct Work
{ char st1[15];
  char st2[15];
  int a; };
```

- а) 34 байт
- б) 5 байт
- в) 32 байта
- г) объем требуемой памяти для этого случая заранее определить нельзя

1.10.** *Выбрать правильный вариант ответа*

Какой объем памяти требуется при работе с переменной для типа, описанного ниже:

```
short int a[25];
```

- а) 25 байт
- б) 2 байта
- в) 50 байт
- г) объем требуемой памяти для этого случая заранее определить нельзя

1.11.* *Выбрать правильный вариант ответа*

Укажите высказывание, в котором содержится ошибка:

- а) переменная типа `short int` занимает в памяти 2 байта
- б) значение 3.15 можно записать в переменную типа `float`
- в) в одной программе не может быть описано двух переменных с одинаковыми именами
- г) в C++ в именах переменных строчные и заглавные буквы различаются

1.12.* *Выбрать правильный вариант ответа*

Какой объем памяти требуется при работе с переменной для типа, описанного ниже:

```
char A[10][2];
```

- а) 22 байт
- б) 10 байт
- в) 20 байт
- г) объем требуемой памяти для этого случая заранее определить нельзя

1.13.** *Выбрать правильный вариант ответа*

Что произойдет после выполнения следующей строки кода программы:

```
cout<<(int)'a';
```

- а) будет выдано сообщение об ошибке
- б) будет выдано число, соответствующее коду символа «а»
- в) будет отображен сам символ «а»
- г) данная строка кода программы будет проигнорирована

1.14.** *Выбрать правильный вариант ответа*

Что выведет на экран следующий фрагмент кода программы:

```
int a=5;
void main( ) { int a=10, c;
                c=a+90;
                a=c+::a;
                cout<<a; }
```

- а) 105;
- б) 5;
- в) 200;
- г) сообщение об ошибке

1.15.** *Выбрать правильный вариант ответа*

Что выведет на экран следующий фрагмент кода программы:

```
int a=5;
void main( ) { int a=10, c=20;
                {int a;
                 a=3*c;}
                cout<<a; }
```

- а) 60;
- б) 10;
- в) 5;
- г) сообщение об ошибке

1.16.** *Выбрать правильный вариант ответа*

Что выведет на экран следующий фрагмент кода программы:

```
int a=5;
void main( ) { int a=10, c=20;
                {int a;
                 a=3*c;}
                cout<<::a; }
```

- а) 5;
- б) 10;
- в) 60;
- г) сообщение об ошибке

1.17.* *Задание на упорядочение*

Расположите типы данных в порядке возрастания объема памяти, необходимой для хранения одной переменной указанного типа:

- 1: char
- 2: short int
- 3: float
- 4: double

1.18.** *Задание на упорядочение*

Расположите в порядке возрастания значений переменных А, В, С, D строки кода программы, если начальные значения переменных равны 25:

- 1: A-=20;
- 2: B++;
- 3: C+=10;
- 4: D*=2;

1.19.* *Задание на соответствие*

Соответствие между способами объявления и типами переменных:

Символьный	char
Целый	int

Вещественный	float
--------------	-------

1.20.* *Задание на соответствие*

Определить, к какому типу относится переменная (или константа):

int	3455
float	3.125
long int	156734
char	A
bool	true

1.21.** *Задание на соответствие*

Определить, к какому типу относится переменная (или константа):

int	3455
long int	56348
float	1.67 E+32
double	2.34 E+150
bool	true

1.22.* *Задание на соответствие*

Привести в соответствие описания и зарезервированные слова:

Подключение заголовочных файлов	include
Указывает на то, что функция не возвращает значения	void
Функция, которая обозначает точку начала выполнения программы на C++	main
Оператор выбора вариантов (множественное ветвление)	switch
Позволяет определить размер памяти, которая соответствует идентификатору или типу	sizeof

2. Оператор ветвления. Множественное ветвление.

2.1.* *Выбрать правильные варианты ответа*

Какие из приведенных ниже операций отношения можно использовать в C++:

- а) <, <=, >, >=
- б) =, !=
- в) :=, <>
- г) and, or

2.2.* *Выбрать один вариант ответа*

Чему будет равно значение переменной «а» после выполнения фрагмента кода программы:

```
int a, b, c;
void main()
{ a=3; b=10; c=15;
  if (c % b > 4) a = (b+c) % 5; }
```

- а) 3
- б) 0
- в) 5
- г) 4

2.3.*** *Выбрать один вариант ответа*

Чему будет равно значение переменной «а» после выполнения фрагмента кода программы:

```
int a, b, c;
void main()
{ b =10; c =15;
  a =b + c % 4;
  if ( ( a<c) && (a % c>b+2) ) a =b*c; else a =b;
  if (a>c) a =c-b; else a=3; }
```

- а) 10

- б) 150
- в) 3
- г) 5

2.4.** *Выбрать один вариант ответа*

Каков результат работы следующей программы:

```
int a, b;
unsigned char c;
void main( )
{ a =10; b =5;
  if ( (a<b) && (c>0) ) c =10; else c =a*b;
  c*=a;
  cout << c; }
```

- а) 500
- б) сообщение об ошибке (диапазон типа unsigned char меньше диапазона типа int)
- в) 244
- г) сообщение об ошибке, так как значение переменной «с» не задано

2.5.** *Выбрать один вариант ответа*

Каков результат работы следующей программы:

```
int a =10, b =5;
unsigned char c;
void main( )
{ if ( (a<b) || (c= =0) ) c =10; else c =a*b;
  c*=a;
  cout << c; }
```

- а) 100
- б) сообщение об ошибке (диапазон типа unsigned char меньше диапазона типа int)
- в) 244
- г) сообщение об ошибке, так как значение переменной «с» не задано

2.6.* *Выбрать правильные варианты ответа*

Каков результат работы следующей программы, в зависимости от введенного значения переменной «а»:

```
int a, b =10;
void main( )
{ cin >> a;
  if (a>b) cout << a; else cout << (2*b); }
```

- а) 20 (если для «а» введено значение 10)
- б) 11 (если для «а» введено значение 11)
- в) 10 (если для «а» введено значение 10)
- г) 20 (если для «а» введено значение 30)

2.7.* *Выбрать один вариант ответа*

Каков результат работы фрагмента программы (переменная «а» типа int):

```
a =2;
switch (a)
{ case 1: cout << "Привет!"; break;
  case 2: cout << "Пока!"; break;
  case 5: cout << "Как дела?"; }
```

- а) Как дела?
- б) Привет!
- в) Пока!
- г) Привет!Пока!Как дела?

2.8.* *Выбрать один вариант ответа*

Чему будет равно значение переменной «а» типа int после выполнения фрагмента программы:

```
a =200;
b =55;
if (a>b) a =35; else a =117;
```

- а) 200
- б) 35
- в) 117

г) 55

2.9.* *Выбрать один вариант ответа*

Каков результат работы фрагмента программы (переменная «а» типа int):

```
a =2;
a>2 ? cout<<"Да" : cout<<"Нет";
```

- а) Да
- б) Нет
- в) в приведенном коде содержатся ошибки
- г) нет правильного ответа

2.10.** *Задание на упорядочение*

В каком порядке следует записать строки кода программы, чтобы результатом ее работы было значение, равное «-1» (переменные «а» и «b» типа int)?

- 1: b =0;
- 2: a = ++b;
- 3: b =10;
- 4: if (a>b) b =a; else b =-a;
- 5: cout << b;

2.11.* *Задание на упорядочение*

В каком порядке следует записать строки кода программы, чтобы результатом ее работы было «Да»?

- 1: a =3; b =5;
- 2: if (b>a) b =a;
- 3: if (b= =a) cout<<"Да"; else cout<<"Нет";

2.12.* *Задание на соответствие*

Установите соответствие для логических элементов:

Логическое отрицание («НЕ»)	!
Логическое умножение («И»)	&&
Логическое сложение («ИЛИ»)	
	= =

2.13.* *Задание на соответствие*

Установите соответствие для операций отношения:

Больше или равно	>=
Не равно	!=
Равно	= =
Меньше	<
Больше	>
	<>

3. Циклы в C++.

3.1.* *Выбрать один вариант ответа*

В чем заключается основное отличие циклов while и do...while?:

- а) никаких отличий нет
- б) тело цикла do...while в любом случае выполнится хотя бы один раз
- в) тело цикла while в любом случае выполнится хотя бы один раз
- г) цикл while имеет ограниченное применение

3.2.* *Выбрать правильные варианты ответа*

Из следующих утверждений выберите верные:

- а) переменная, используемая в цикле for, должна быть целого типа
- б) переменная, используемая в цикле for, может быть типа float
- в) переменная, используемая в цикле while, может быть вещественного типа
- г) тело цикла for в любом случае выполнится хотя бы один раз

3.3.* *Выбрать один вариант ответа*

Алгоритм, в котором команда или серия команд повторяется, называется:

- а) линейным
- б) разветвляющимся

- в) циклическим
- г) замкнутым

3.4.* *Выбрать один вариант ответа*

Когда закончится выполнение фрагмента кода программы:

```
while (getch() !=27);
```

- а) никогда
- б) после того, как будет нажата любая клавиша на клавиатуре
- в) сразу же, так как тело цикла пустое
- г) после того, как пользователь нажмет клавишу «Esc»

3.5.* *Выбрать один вариант ответа*

Циклом с «постусловием» в C++ является цикл:

- а) while
- б) for
- в) do...while
- г) такого цикла в C++ нет

3.6.* *Выбрать один вариант ответа*

Циклом с «предусловием» в C++ является цикл:

- а) while
- б) for
- в) do...while
- г) такого цикла в C++ нет

3.7.* *Выбрать один вариант ответа*

Циклом с «параметром» в C++ является цикл:

- а) while
- б) for
- в) do...while
- г) такого цикла в C++ нет

3.8.** *Выбрать правильные варианты ответа*

Извлечь корень квадратный из числа «x» можно следующим образом (x>0):

- а) `sqrt(x);`
- б) `fabs(x);`
- в) `sqrt(x);`
- г) `exp(log(x)/2);`

3.9.** *Выбрать правильные варианты ответа*

Возвести число «x» в квадрат можно следующим образом (x>0):

- а) `sqrt(x);`
- б) `(int) x;`
- в) `sqrt(x);`
- г) `exp(2*log(x));`

3.10.* *Выбрать один вариант ответа*

Для завершения выполнения цикла досрочно используется оператор:

- а) `halt`
- б) `continue`
- в) `break`
- г) завершить цикл досрочно нельзя

3.11.** *Выбрать один вариант ответа*

Каков результат работы фрагмента программы (переменная «a» типа int):

```
a =20;
for (int i =1; i<=10; i++)
    { a -=3;
      if (i>7) continue;
      a +=2; }
cout << a;
```

- а) 4
- б) 10

- в) 2
- г) 6

3.12.** *Выбрать один вариант ответа*

Каков результат работы фрагмента программы (переменная «a» типа int):

```
a =20;
for (int i =1; i<=10; i++) { a -=3;
                               if (i>8) break;
                               a +=2; }
cout << a;
```

- а) 4
- б) 9
- в) 10
- г) 6

3.13.* *Выбрать один вариант ответа*

Каков результат работы фрагмента программы (переменная «a» типа int):

```
a =0;
for (int i =1; i<= 5; a += i++);
cout << a;
```

- а) 1
- б) 10
- в) 5
- г) 15

3.14.** *Выбрать правильные варианты ответа*

В каких случаях значение переменной «a» типа int примет значение, большее 20:

- а) for (a =0, i =1; i<=10; a += i++);
- б) a =0; do { a +=2; } while (a<=20);
- в) for (a =10, i =10; i>=1; a +=1, i--);
- г) a =5; while (a>30) a+=2;

(Правильные ответы: а), б)

3.15.** *Выбрать один вариант ответа*

Каков результат работы фрагмента программы:

```
for (float i =0; i<= 1.01; cout<<i<<endl, i+=0.1);
```

- а) фрагмент кода содержит ошибку, т.к. переменная i описана как вещественная
- б) так использовать цикл с параметром нельзя
- в) будут выведены числа от 0 до 1 с шагом 0.1
- г) нет правильного ответа

3.16.** *Задание на упорядочение*

В каком порядке следует записать строки фрагмента кода программы, чтобы результатом ее работы была сумма чисел от 1 до 10 (переменная «s»)?

- 1: s =0; i =1;
- 2: while (i<=10) {
- 3: s += i;
- 4: i++;
- 5: }

3.17.** *Задание на упорядочение*

Записать строки кода программы в порядке возрастания значений переменной «s»

- 1: s =0; for (k =1; k<=5; s++, k++);
- 2: s =0; for (k =1; k<=5; s += k++);
- 3: s = -20; while (s<=25) s += 2;

3.18.** *Задание на упорядочение*

Записать строки кода программы в порядке возрастания значений переменной «s»

- 1: s = -10; do s += 2; while (s>50);
- 2: s =0; for (k =1; k<=5; s++, k++);
- 3: s =0; for (k =1; k<=5; s += k++);
- 4: s = -20; while (s<=25) s += 2.14;

3.19.* *Задание на соответствие*

Установите соответствие:

Возвращает символ по коду ASCII	(char)
Возвращает код ASCII указанного символа	(int)
Находит остаток от деления	%
Операция приведения типа к вещественному	(float)
	true

3.20.* *Задание на соответствие*

Установите соответствие для следующих функций:

Возвращает абсолютное значение вещественного аргумента	fabs()
Возвращает экспоненту аргумента	exp()
Возвращает квадрат аргумента	sqr()
Возвращает десятичный логарифм аргумента	log10()
	log ()
	abs()

3.21.* *Задание на соответствие*

Установите соответствие между видами циклов и их особенностями:

Нельзя изменять значение переменной цикла внутри тела цикла	Нет правильного ответа
Тело цикла в любом случае выполнится хотя бы один раз	do...while
Цикл с «предусловием»	while
	for

3.22.* *Задание на соответствие*

Установите соответствие между видами циклов и их особенностями:

Нельзя изменять значение переменной цикла внутри тела цикла	Нет правильного ответа
Тело цикла в любом случае выполнится хотя бы один раз	do...while
Цикл с «предусловием»	while
Можно использовать вещественные числа в качестве значений переменной цикла	while, for, do...while
	while, do...while
	for

4. Массивы и указатели.

4.1.* *Выбрать правильные варианты ответа*

Массивы могут быть:

- а) одномерными
- б) многомерными
- в) бесконечными
- г) неопределенными

4.2.* *Выбрать один вариант ответа*

Множество однотипных элементов, объединённых общим именем и занимающих в памяти компьютера определённую область:

- а) массив
- б) запись
- в) переменная
- г) константа

4.3.* *Выбрать один вариант ответа*

В качестве элементов массива можно использовать:

- а) только тип int
- б) только тип char
- в) любой тип данных
- г) только типы int, long int и float

4.4.* *Выбрать один вариант ответа*

Что означает следующий фрагмент кода программы:

```
float a[10][5];
```

- а) переменная «а» представляет собой одномерный массив
- б) переменная «а» представляет собой двумерный массив
- в) некорректное описание двумерного массива
- г) объявление массива открытого типа

4.5.* *Выбрать один вариант ответа*

Содержит ли следующий фрагмент кода программы ошибки:

```
int a[3][10];  
int b[5]={1, 3, 5, 7, -10};
```

- а) ошибка в описании одномерного массива
- б) ошибка в описании двумерного массива
- в) ошибок нет
- г) ошибки в описании и одномерного, и двумерного массивов

4.6.** *Выбрать правильные варианты ответа*

Задан следующий фрагмент кода программы:

```
int a[15];  
float b[5][5];
```

Какие операции из перечисленных ниже не приводят к ошибкам:

- а) `b[3][4] = a[5];`
- б) `for (int k = 0; k < 20; a[k] = k++);`
- в) `a[7] = b[2][5];`
- г) `for (int k = 0; k < 4; b[k][k] = a[k++]);`

4.7.** *Выбрать правильные варианты ответа*

Задан следующий фрагмент кода программы:

```
int a[15];  
float b[10];
```

Какие операции из перечисленных ниже являются допустимыми:

- а) `b[3] = a[5];`
- б) `if (a[10] > 5) b[5] = 2.37;`
- в) `b[7] = a[20];`
- г) `for (int k = 0; k < 20; b[k] = a[k++]);`

4.8.** *Выбрать правильные варианты ответа*

Задан следующий фрагмент кода программы:

```
float a[10], b[10];
```

Какие операции из перечисленных ниже являются допустимыми:

- а) `b[3] = (int) a[5] % 4;`
- б) `if (a[2] > 8) b[5] = 2.37;`
- в) `b[7] = a[7] % 12;`
- г) `for (int k = 20; k > 3; b[k] = a[k - -]);`

4.9.* *Выбрать один вариант ответа*

В одном и том же массиве могут содержаться:

- а) только однотипные элементы
- б) любой набор элементов
- в) нет правильного ответа
- г) элементы разных типов, в зависимости от описания массива

4.10.** *Выбрать один вариант ответа*

Каков результат работы следующей программы:

```
#include <iostream>  
#include <conio>  
#include <windows>  
void main()  
{ int a[20], k, s = 0;  
  randomize();  
  for (k = 0; k < 20; k++)  
    { a[k] = random(20);
```

```

        if (a[k] % 2 == 0) s += a[k]; }
    cout<<"s="<<s;
    getch(); }

```

- а) подсчитывается сумма всех элементов массива
- б) подсчитывается сумма элементов массива, содержащих четные числа
- в) подсчитывается сумма элементов массива с четными индексами
- г) подсчитывается сумма элементов массива, содержащих нечетные числа

4.11.* *Выбрать один вариант ответа*

Что происходит в процессе выполнения следующего фрагмента кода программы:

```

for (k =0; k<20; k++) { cout << "элемент "<< (k+1)<< "= ";
                        cin >> a[k]; }

```

- а) вывод элементов массива на экран
- б) выдается сообщение об ошибке
- в) ввод элементов массива с клавиатуры
- г) элементам массива присваиваются значения от 1 до 20

4.12.* *Выбрать один вариант ответа*

Что происходит в процессе выполнения следующего фрагмента кода программы:

```

for (k =0; k<4; k++)
    { for (m =0; m<5; m++) cout << a[k][m] << " ";
      cout<<endl; }

```

- а) вывод элементов двумерного массива на экран в виде матрицы 4×5
- б) выдается сообщение об ошибке
- в) вывод элементов двумерного массива на экран в одну строку
- г) вывод элементов одномерного массива на экран в виде матрицы

4.13.** *Выбрать один вариант ответа*

Каков результат работы фрагмента следующей программы:

```

float a[10] = { 1.3, 3.5, 1.1, 4.6, 5.6, 3.2, 2.8, 3.3, 1.0, 1.0};
float *p, s =0.;
int i;
p=&a[0];
for(i =0; i<10; i++) cout << *p++ <<" ";

```

- а) подсчитывается сумма всех элементов массива
- б) осуществляется вывод элементов массива
- в) выводятся адреса элементов массива
- г) фрагмент кода содержит ошибки

4.14.** *Выбрать один вариант ответа*

Каков результат работы фрагмента следующей программы:

```

float a[10] = { 1.3, 3.5, 1.1, 4.6, 5.6, 3.2, 2.8, 3.3, 1.0, 1.0};
float *p, s=0.;
int i=0;
p=&a[0];
s=*p;
while (i<5) { s+=*(p+=2);
              i++; }
cout<<s<<endl;

```

- а) подсчитывается сумма всех элементов массива
- б) программа содержит ошибку
- в) суммируются элементы, расположенные на нечетных позициях
- г) суммируются элементы, расположенные на четных позициях

4.15.* *Выбрать один вариант ответа*

Каков результат работы фрагмента следующей программы:

```

float a =10;
float *p =&a;
cout << *p;

```

- а) выдается значение переменной «a»
- б) программа содержит ошибку
- в) выдается адрес переменной «a»

г) на экран ничего не выводится, т.к. указатель инициализируется при объявлении, что не допустимо

4.16.* *Выбрать один вариант ответа*

Каков результат работы фрагмента следующей программы:

```
float a =10;
float *p =&a;
cout << p;
```

- а) выдается значение переменной «а»
- б) программа содержит ошибку
- в) выдается адрес переменной «а»
- г) на экран ничего не выводится, т.к. указатель инициализируется при объявлении, что не допустимо

4.17.* *Выбрать один вариант ответа*

Каков результат работы фрагмента следующей программы:

```
float *p;
*p =10;
cout<<*p<<endl;
```

- а) выдается значение «10»
- б) в программе содержится ошибка
- в) выдается адрес, по которому хранится значение «10»
- г) выдается «0», т.к. указатель был инициализирован неверно

4.18.* *Выбрать один вариант ответа*

Каков результат работы фрагмента следующей программы:

```
int i,n;
int *p= new int [n];
cin>>n;
for (i =0; i<n; i++) cin >> p[i];
```

- а) осуществляется ввод элементов динамического массива
- б) осуществляется ввод элементов статического массива
- в) осуществляется ввод адресов элементов
- г) программа выдает сообщение об ошибке

4.19.* *Выбрать правильные варианты ответа*

Какие из перечисленных ниже операций допустимы при работе с отдельными элементами массива типа double:

- а) операции приведения типов
- б) операции отношения (>=, <=, !=, <, >, ==)
- в) %
- г) операции изменения типа элементов

4.20.*** *Выбрать один вариант ответа*

Что происходит в процессе выполнения следующего фрагмента кода программы:

```
for (k =0; k<4; k++)
for (m =0; m<4; m++)
if (a[k]>a[m]) { s =a[k];
a[k] =a[m];
a[m] =s; }
```

- а) сортировка элементов массива по возрастанию
- б) замена каждого следующего элемента массива предыдущим
- в) изменение порядка следования элементов массива на противоположный
- г) сортировка элементов массива по убыванию

4.21.*** *Выбрать один вариант ответа*

Что происходит в процессе выполнения следующего фрагмента кода программы:

```
for (k =0; k<4; k++)
for (m =0; m<3; m++)
if (a[m]>a[m+1]) { s =a[m];
a[m] =a[m+1];
a[m+1] =s; }
```

- а) сортировка элементов массива по возрастанию

- б) замена каждого следующего элемента массива предыдущим
- в) изменение порядка следования элементов массива на противоположный
- г) сортировка элементов массива по убыванию

4.22.* *Выбрать один вариант ответа*

Что происходит в процессе выполнения следующего фрагмента кода программы:

```
s = a[0];
for (k = 0; k < 10; k++)
    if (a[k] > s) s = a[k];
```

- а) выдается сообщение об ошибке
- б) подсчет суммы элементов массива
- в) поиск минимального элемента массива
- г) поиск максимального элемента массива

4.23.* *Выбрать один вариант ответа*

Что происходит в процессе выполнения следующего фрагмента кода программы:

```
s = 0;
for (k = 0; k < 30; k++)
    if (a[k] == 20) s++;
```

- а) выдается сообщение об ошибке
- б) считается, сколько раз в массиве встречается число «20»
- в) в элементы массива записывается число «20»
- г) переменная «s» после выполнения цикла примет значение, равное «1»

4.24.* *Задание на упорядочение*

Требуется поменять местами элементы с номерами 3 и 8 некоторого одномерного массива. Записать строки кода программы в требуемом для этого порядке:

- 1: s = a[3];
- 2: a[3] = a[8];
- 3: a[8] = s;

4.25.* *Задание на соответствие*

Установите соответствие:

Одномерный массив целых чисел из 20 элементов	int a[20];
Двумерный массив целых чисел из 20 элементов	int b[4][5];
Одномерный массив символов	char c[50];
Двумерный массив вещественных чисел	double d[8][8];
	int a[20][2];

4.26.* *Задание на соответствие*

Установите соответствие:

Одномерный массив целых чисел	int a[20];
Двумерный массив целых чисел	int b[6][5];
Двумерный массив целых чисел из 20 элементов	int b[4][5];
Двумерный массив вещественных чисел	float b[2][10];

5. Строки. Структуры.

5.1.* *Выбрать правильные варианты ответа*

Какие из приведенных ниже операций можно использовать в C++ при работе с символьными массивами:

- а) <, <=, >, >=
- б) =, !=, +
- в) <>
- г) mod, div

5.2.* *Выбрать один вариант ответа*

Что происходит в процессе выполнения следующего фрагмента кода программы:

```
int i=0,ss=0;
char s[10];
gets(s);
while (i<strlen(s))
```

```
if (s[i++]='a') ss++;  
cout<<ss;
```

- а) выдается сообщение об ошибке
- б) на экран выводится строка текста
- в) подсчитывается число символов в строке
- г) подсчитывается, сколько раз в строке встречается символ «а»

5.3.* *Выбрать один вариант ответа*

Что делает фрагмента кода программы:

```
int i=0,ss=0;  
char s[100];  
while ( (s[i]=getch())!='\n' && i<100 ) putchar(s[i++]);
```

- а) вводит строку посимвольно и отображает печатаемые символы на экране
- б) вводит строку до тех пор, пока не будет введено 100 символов
- в) вводит строку, но не отображает ее на экране
- г) код программы содержит ошибку

5.4.* *Выбрать один вариант ответа*

Длина строки в C++:

- а) может лежать в диапазоне от 0 до 255 символов
- б) не ограничена
- в) зависит только от объема свободной памяти
- г) не может превышать 128 символов

5.5.** *Выбрать правильные варианты ответа*

В каких из приведенных ниже случаев допущены ошибки:

- а) char s[200];
- б) char a[25], b[500];
- в) char a[];
- г) char s[3][];

5.6.** *Выбрать правильные варианты ответа*

В каких перечисленных случаях строка будет отображена на экране корректно:

- а) char a[]={ '1', '2', '3' };
- б) char b[]={ '1', '2', '3', '\0' };
- в) char c[4]={ '1', '2', '3', '\0' };
- г) char a[3]={ '1', '2', '3' };

5.7.* *Выбрать правильные варианты ответа*

Каждый символ:

- а) занимает 2 байта памяти
- б) занимает 1 байт памяти
- в) имеет числовой код в соответствии с таблицей кодов ASCII
- г) представляет собой составной тип данных

5.8.* *Выбрать правильные варианты ответа*

Для преобразования данных типа char в целое число (т.е. для получения кода ASCII символа):

- а) можно использовать явное преобразование типов (int)
- б) потребуется описать достаточно длинную функцию перевода
- в) потребуется использовать массив, содержащий алфавит
- г) можно ничего не использовать – C++ сам умеет преобразовывать типы

5.9.* *Выбрать один вариант ответа*

В языке C++ тип «структура» определяется путем указания служебного слова:

- а) struct
- б) sizeof
- в) record
- г) continue

5.10.* *Выбрать один вариант ответа*

Структурированный комбинированный тип данных, состоящий из фиксированного числа компонент (полей) разного типа:

- а) структура (struct)

- б) массив
- в) строка (char)
- г) нет правильного ответа

5.11.** *Выбрать один вариант ответа*

В чем заключается ошибка в приведенном ниже фрагменте кода программы:

```
struct Work {char Name1[15];
             char Name2[15];
             int Year; }
```

Work A;

- а) ошибка в данном фрагменте кода отсутствует
- б) не поставлена «точка с запятой» после закрывающейся фигурной скобки описания структуры
- в) не указано слово void до перечисления полей (перед полем Name1)
- г) первая строка фрагмента должна быть записана как Work struct

5.12.* *Выбрать один вариант ответа*

Есть ли ошибка в приведенном ниже фрагменте кода программы:

```
struct Work { char Name1[15];
             char Name2[15];
             int Marks[30]; };
```

Work A;

- а) не указано слово struct до перечисления полей (перед полем Name1)
- б) массив не может быть одним из полей записи
- в) ошибка в данном фрагменте кода отсутствует
- г) лишняя «точка с запятой» после закрывающейся фигурной скобки описания структуры

5.13.* *Выбрать один вариант ответа*

Как осуществляется в программе доступ к полям структуры (struct):

- а) указывается идентификатор записи и через точку – нужное поле
- б) простым прямым обращением к имени поля
- в) указывается идентификатор записи и в скобках – нужное поле
- г) нет правильного ответа

5.14.** *Выбрать один вариант ответа*

Есть ли ошибка в приведенном ниже фрагменте кода программы:

```
struct Work { char Name1[15];
             char Name2[15];
             int Marks[30]; }
```

Work A;

int Name2;

- а) лишняя «точка с запятой» после закрывающейся фигурной скобки описания структуры
- б) массив не может быть одним из полей записи
- в) ошибка в данном фрагменте кода отсутствует
- г) переменная Name2 использована и в качестве поля структуры, и как отдельная переменная типа int, что недопустимо

5.15.** *Выбрать один вариант ответа*

Что будет являться результатом выполнения фрагмента кода программы:

```
struct Work { char Name1[15];
             char Name2[15];
             float Year; };
```

Work A[10];

cout << (sizeof(A));

- а) сообщение об ошибке
- б) 0, так как поля записи еще не заполнены
- в) 34
- г) 340

5.16.* *Задание на соответствие*

Установите соответствие между описанием и функцией (или процедурой):

Определяет текущую длину строкового выражения	strlen(S);
Считывает символ с клавиатуры (но не выводит его)	getch();

Отображает указанный символ на экране	putch(S[k]);
Считывает строку с клавиатуры	gets(S);
	puts(S);

6. Функции. Перегрузка функций.

6.1.* *Выбрать правильные варианты ответа*

Какие строки нельзя включать в состав одной и той же программы:

- а) void sum(int, int);
- б) int sum(int, int);
- в) int sum(int);
- г) void sum(int, float);

6.2.* *Выбрать один вариант ответа*

Для чего служит описанная ниже функция:

```
int sum(int a, int b)
{ if (a>b) return a; else return b; }
```

- а) возвращает минимальное из двух чисел
- б) ничего не возвращает
- в) возвращает максимальное из двух чисел
- г) возвращает два числа

6.3.** *Выбрать один вариант ответа*

Для чего служит описанная ниже функция:

```
int sum(int a[], int n)
{ int s =0, *p = &a[0], i =0;
  while (i++ < n) s += *p++;
  return s; }
```

- а) подсчитывает сумму элементов одномерного массива
- б) подсчитывает сумму адресов элементов
- в) ничего не выполняет, так как не указан размер передаваемого массива
- г) приведенный фрагмент кода содержит ошибку

6.4.** *Выбрать один вариант ответа*

В чем заключается ошибка в приведенном ниже фрагменте кода программы:

```
int sum(int a[], int n)
{ int s=0, i=0;
  while (i<n)
    {if (a[i]%2==0) s+=a[i];
     i++;}
  return s; }
```

- а) в строке «int sum(int a[], int n)» не указан размер массива
- б) операцию «%» нельзя применять к элементам массива
- в) цикл внутри функции будет выполняться бесконечное число раз
- г) ошибок в приведенном фрагменте кода нет

6.5.** *Выбрать один вариант ответа*

Какие действия выполняет функция:

```
int fun(char *s)
{ int n =0;
  while (*s++) n++;
  return n; }
```

- а) никакие действия выполнены не будут, т.к. в коде содержатся ошибки
- б) функция определяет длину строки
- в) функция подсчитывает сумму кодов символов строки
- г) функция возвращает «0»

6.6.** *Выбрать один вариант ответа*

В чем заключается смысл перегрузки функций в C++?

- а) можно использовать одинаковые имена функций, но при этом функции должны отличаться количеством или типом формальных параметров

- б) можно использовать одинаковые имена функций, при этом функции могут иметь абсолютно одинаковый набор формальных параметров
- в) перегрузка функций в С++ не допускается
- г) можно перегружать только функции, уже описанные в языке С++

6.7.** *Выбрать один вариант ответа*

Какие действия выполняет функция:

```
double fun (int a, int b)
{ return pow(a*a+b*b,0.5);}
```

- а) никакие действия выполнены не будут, т.к. в коде содержатся ошибки
- б) функция находит сумму квадратов двух чисел
- в) функция находит модуль вектора, заданного на плоскости
- г) функция возвращает «0»

6.8.** *Выбрать один вариант ответа*

Можно ли перегружать встроенные функции С++?:

- а) можно перегружать любые
- б) нельзя
- в) можно, но только математические
- г) можно, но только не математические

Тестовые материалы (тест №2) по разделу «Объектно-ориентированное программирование на С++»

1. *Выбрать один вариант ответа*

В определении класса члены класса с ключевым словом *private* доступны:

- любой функции программы;
- методам этого класса и дружественным функциям этого класса;
- только методам этого класса;
- только открытым членам класса;
- нет верного ответа.

2. *Выбрать один вариант ответа*

Правильный конструктор класса TRectangle:

- `void TRectangle () {}`
- `~ TRectangle () {}`
- `TRectangle () {}`
- `int TRectangle () {}`

3. *Выбрать правильные варианты ответа*

Следующие операции не могут быть перегружены:

- разрешение области видимости (::);
- тернарная операция (?:);
- `sizeof`;
- ++

4. *Выбрать один вариант ответа*

Конструкторы с одинаковыми именами в языке С++:

- недопустимы;
- должны различаться типом возвращаемого значения;
- должны различаться количеством или типом параметров;

- не нужны;
- нет верного ответа.

5. Выбрать один вариант ответа

Для перегрузки префиксной операции ++ в классе X используется оператор:

- X operator ++ ()
- X operator ++ (int)
- int operator ++ ()
- int operator ++ (int)
- нет верного ответа.

6. Выбрать один вариант ответа

Количество параметров для перегруженной операции a + b равно:

- 0;
- 1;
- 2;
- 3;
- нет верного ответа.

7. Выбрать один вариант ответа

Если в базовом классе переменная расположена в секции *private*, а в производном классе имеет атрибут наследования *private*, то:

- она недоступна в производном классе;
- она переходит в производный класс в секцию *private*;
- она переходит в производный класс в секцию *protected*;
- она переходит в производный класс в секцию *public*;
- нет верного ответа.

8. Задание на упорядочение

Расположите строки кода программы таким образом, чтобы в результате получился простой класс, описывающий дробь. При этом необходимо, чтобы числитель и знаменатель дроби нельзя было менять непосредственно из программы путем прямого обращения к ним. Функция *Reduce()* должна выступать как функция-утилита класса. Непосредственный доступ из основной программы должен быть только к функциям, выполняющим ввод и вывод значений дроби.

```
class TDrob
{
    int a,b;           // числитель и знаменатель дроби
    void Out();        // вывод значения дроби
    bool Set(int,int); // установка значений числителя и знаменателя
public:
    void Reduce();    // сокращение дроби
};
TDrob(int=1,int=1); // конструктор
```

9. Выбрать один вариант ответа

Чистая виртуальная функция – это виртуальная функция, которая:

- делает свой класс абстрактным;
- не возвращает результата;
- не имеет параметров;

- имеет пустое тело функции;
- нет верного ответа.

10. Выбрать один вариант ответа

Признаком виртуальной функции является:

- ключевое слово *abstract*;
- ключевое слово *virtual*;
- ключевое слово *void*;
- ключевое слово *friend*;
- нет верного ответа.

11. Выбрать один вариант ответа

В приведенном описании класса

```
class Vector
{ public:
  Vector();
  Vector(int, int);
  void SetXY(int,int);
  void Out();
private:
  int x,y;
  double Modul;
  double ABS(); };
```

- имеется ошибка, связанная с определением конструкторов класса;
- используется спецификатор доступа *private*, что в данном классе недопустимо;
- ошибок нет;
- имеются две функции с одинаковыми именами (*Vector*), что недопустимо.

12. Выбрать один вариант ответа

Если деструктор не определен в классе явным образом, то:

- компилятор автоматически создает пустой деструктор;
- будет выдано сообщение об ошибке;
- такой класс будет являться абстрактным;
- такой класс является вложенным.

13. Выбрать правильные варианты ответа

Укажите верные высказывания относительно механизма перегрузки операций в C++:

- операция присваивания = может быть использована с любым классом без явной перегрузки;
- существуют операции, которые перегружать нельзя;
- старшинство операций не может быть изменено перегрузкой, однако с помощью скобок можно принудительно изменять последовательность оценки перегруженных операций в выражениях;
- можно создавать новые операции, а также перегружать их.

14. Задание на упорядочение

Расположить классы в порядке «от простого к сложному» в плане использования механизма наследования:

Окружность
Точка
Сфера

15. Задание на соответствие

Установите соответствие между высказываниями и соответствующими им ключевыми словами:

Используется при описании виртуальной функции	<i>friend</i>
Используется при описании дружественной функции	<i>virtual</i>
Может быть использован для определения адреса объекта или для сцепленного вызова функций	<i>void</i>
Если функция не возвращает никакого значения, то она описывается именно с этим словом	<i>this</i>
	<i>main</i>

Перечень вопросов к экзамену

I семестр

1. Алфавит, синтаксис и семантика языка C++. Понятие алгоритма.
2. Структура программы на языке C++. Характеристика процедурных языков программирования.
3. Типы данных. Описание констант и переменных стандартных типов. Приведение типов.
4. Полная и неполная форма ветвления.
5. Структура множественного выбора switch. Оператор безусловного перехода goto.
6. Перечисления. Объединения. Смеси.
7. Структуры повторения: цикл с предусловием. Операторы break и continue.
8. Структуры повторения: цикл с постусловием. Операторы break и continue.
9. Структура повторения for. Операторы break и continue.
10. Функции в C++. Описание функции и ее вызов. Локальные переменные. Формальные и фактические параметры. Прототипы функций.
11. Функции в C++. Перегрузка функций. Примеры программного кода.
12. Ссылки и ссылочные параметры. Вызов функций по ссылке с аргументами-указателями. Аргументы по умолчанию.
13. Функции с переменным количеством параметров.
14. Указатели на функцию как параметр.
15. Шаблоны функций. Встраиваемые (inline) функции. Унарная операция разрешения области действия (::).
16. Рекурсия. Рекурсивные алгоритмы. Итерационные алгоритмы.
17. Адреса и указатели. Объявление указателей. Операции над указателями. Указатели, объявленные как const.
18. Статические массивы в C++. Сечения массива и указатели.
19. Использование указателей при работе с массивами. Динамическое выделение памяти. Передача массивов в функции.
20. Символьный тип в C++. Библиотека обработки символов ctype.
21. Строки в C++. Функции gets и puts стандартной библиотеки stdio. Функции работы со строками библиотек cstring и stdlib.
22. Тип struct (структура) в C++. Сравнительный анализ с типом record в Turbo Pascal.
23. Динамические структуры. Списки.
24. Динамические структуры. Стеки. Деки. Очереди.
25. Файловый ввод-вывод в C++. Использование средств библиотеки stdio.

26. Файловый ввод-вывод в C++. Использование средств библиотеки `fstream`. Файлы последовательного доступа. Файлы произвольного доступа.
27. Текстовый вывод (на экран) в C++.
28. Манипуляторы потоков.

2 семестр

1. Объектное программирование. Основные понятия. Объекты и классы.
2. Классы. Определение класса. Методы класса. Спецификаторы доступа к членам класса.
3. Классы. Область действия класса и доступ к членам класса.
4. Классы. Статические компоненты класса.
5. Управление доступом к членам класса. Инициализация объектов класса: конструкторы.
6. Использование конструкторов с аргументами по умолчанию. Использование деструкторов.
7. Классы. Константные объекты и функции-члены. Дружественные функции и дружественные классы.
8. Использование указателя `this`. Конструктор копирования.
9. Вложенные классы.
10. Наследование: базовые классы и производные классы. Защищенные члены класса. Типы наследования (открытое, защищенное, закрытое).
11. Использование конструкторов и деструкторов в производных классах.
12. Множественное наследование.
13. Виртуальные функции. Полиморфизм и позднее связывание.
14. Чисто виртуальные функции. Абстрактные базовые классы и конкретные классы.
15. Организация ввода-вывода между программой и объектом `string`.
16. Шаблоны классов. Шаблоны и наследование. Шаблоны и друзья.
17. Шаблоны классов. Частичная специализация шаблона.
18. Основы обработки исключений в C++. Генерация исключений. Обработка неуспешного выполнения `new`.
19. Перегрузка операций. Механизм перегрузки операций.
20. Перегрузка операций. Перегрузка операций "поместить в поток" и "взять из потока".
21. Перегрузка операций. Перегрузка унарных операций.
22. Перегрузка операций. Перегрузка бинарных операций.
23. Перегрузка операций. Перегрузка операции присваивания.
24. Перегрузка операций. Перегрузка операции индексирования.
25. Перегрузка операций `++` и `--` (в префиксной и постфиксной форме).
26. Перегрузка операций. Перегрузка операции вызова функции.

3 семестр

1. Понятие объектно-ориентированного программирования. Определение объекта. Поля и методы.
2. Инкапсуляция в объектно-ориентированном программировании. Спецификаторы

доступа *public* и *private*.

3. Инициализация объектов: конструкторы. Использование деструкторов.
4. Наследование в объектно-ориентированном программировании.
5. Полиморфизм в объектно-ориентированном программировании.
6. Визуальное программирование. События и сообщения. Механизмы передачи и обработки сообщений в объектно-ориентированных средах.
7. Компоненты панели элементов «Стандартные элементы управления». Краткое описание свойств и примеры работы.
8. Компоненты панели элементов «Контейнеры». Краткое описание свойств и примеры работы.
9. Компоненты панели элементов «Меню и панели инструментов». Краткое описание свойств и примеры работы.
10. Компоненты панели элементов «Компоненты». Краткое описание свойств и примеры работы.
11. Компоненты панели элементов «Диалоговые окна». Краткое описание свойств и примеры работы.
12. Библиотеки объектов. Интерфейсные объекты: управляющие элементы, окна и диалоги.
13. Основы интерфейса пользователя. Составляющие пользовательского интерфейса: приложение (класс Application), класс Form, класс «Буфер обмена» (Clipboard).
14. Класс InstalledFormCollection. Получение списка доступных приложению шрифтов, добавление в коллекцию указателей мыши.
15. Обработка событий и исключительных ситуаций. Стандартные диалоги.
16. Работа с графикой. Принцип отрисовки изображений в Windows.
17. Добавление формы в проект и ее удаление из проекта. Настойка свойств проекта. Редактирование файла проекта. Возможность назначения главной формы приложения.
18. Динамическое создание элементов управления и других компонент, используемых в приложении.
19. Создание собственных классов и объектов. Практическое использование классов.
20. Класс List. Создание собственных классов, организация групп объектов на основе List.
21. Класс List. Добавление списков объектов. Изменение свойств отдельных объектов, добавление и удаление объектов.

Контрольные работы

Типовые варианты аудиторной контрольной работы №1

I вариант

1. Вычислить значения выражения $\frac{(a-b)^2 + c^4}{\ln(|a+b+c|-2)} + \left(\frac{3ab}{\sqrt{1-c^2}}\right)^{\frac{3}{4}}$, учитывая все возможные значения вводимых переменных a, b, c .

2. Написать программу, которая в данной последовательности действительных чисел a_1, \dots, a_n позволяет выяснить, имеются ли среди соседних элементов в этой последовательности взаимно обратные числа.
3. В одномерном массиве заданного размера N найти элемент, меньше всего отличающийся от \sqrt{S} , где S – сумма элементов данного массива.

II вариант

1. Вычислить значения выражения $\ln|a^b| + \exp(a+b) - \sqrt{c-ab}$, учитывая все возможные значения вводимых переменных a, b, c .
2. Составить программу нахождения первого члена последовательности $a_n = 1 + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^n}$, который не принадлежит отрезку $[a, b]$. Числа a и b запрашиваются в начале работы с программой. (Указание: данная последовательность является сходящейся и ее сумма при $n \rightarrow \infty$ дает 2, т.о., все значения $[a, b]$ заключены между 1 и 2).
3. В заданном массиве $A(N \times N)$ поменять местами главную диагональ и первый столбец.

III вариант

1. Вычислить значения выражения $\frac{1}{\arcsin a} + \ln(b+ac) - \frac{a^2 + b^2 + c^2}{abc}$, учитывая все возможные значения вводимых переменных a, b, c .
2. Найти сумму простых чисел в промежутке $[1, N]$. Число N вводится в ходе работы программы.
3. В двумерном массиве $A(N \times M)$, не содержащем повторяющиеся элементы, найти элементы, наиболее близкие друг другу по значению, а также их положение в массиве (номер строки и столбца). Указание: воспользоваться правилом нахождения максимального элемента.

Типовые варианты аудиторной контрольной работы №2

I вариант

Дана следующая информация о книгах библиотеки: автор, название книги, издательство, год выпуска, количество экземпляров книги в библиотеке. Написать программу, позволяющую осуществлять следующие операции: добавление и удаление информации о книге, просмотр информации о книгах библиотеки, сортировку списка книг по году издания (автору, количеству экземпляров). Данные должны записываться в файл, и считываться из него.

II вариант

Обработать информацию о результатах сессии в некоторой группе студентов, считая, что количество экзаменов равно четырем. Для каждого студента подсчитать средний балл по всем экзаменам и составить список, включающий сведения о задолжниках. Исходными данными считать: фамилия и инициалы студента, номер зачетной книжки, список сдаваемых предметов и оценки по каждому из них. Предусмотреть возможность

изменения информации (т.к., например, студент мог пересдать экзамен и тем самым ликвидировать задолженность). Данные должны записываться в файл, и считываться из него.

Типовые варианты аудиторной контрольной работы №3

I вариант

Создать класс студент, имеющий имя (указатель на строку), курс и идентификационный номер. Определить конструктор, деструктор и функцию печати. Создать public-производный класс – студент-дипломник, имеющий тему диплома. Определить конструктор по умолчанию, деструктор, функцию печати. Определить функции переназначения названия диплома и идентификационного номера. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

II вариант

Создать класс автомобиль, имеющий марку (указатель на строку), число цилиндров, мощность. Определить конструктор, деструктор и функцию печати. Создать public-производный класс – грузовики, имеющий грузоподъемность кузова. Определить конструктор с разным числом параметров, деструктор, функцию печати. Определить функции переназначения марки и грузоподъемности. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

III вариант

Создайте класс точка, которая имеет координаты. Создать производные классы – окружность и эллипс. Определить функции печати, конструкторы, деструкторы, вычисление площади. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

IV вариант

Создать класс точка, имеющая координаты. Определить конструкторы, деструктор и функцию печати. Создать public-производный класс – цветная точка, имеющий цвет точки. Определить конструктор по умолчанию, деструктор, функцию печати. Определить функции переназначения цвета и координат точки, вывода точки на экран. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Типовые варианты аудиторной контрольной работы №4

I вариант

Создать абстрактный класс *Square_Eq* для решения квадратного уравнения. Определить в нем чисто виртуальный метод *Calculate* для вычисления корней и виртуальный метод *ShowResult*, внутри которого вызывается *Calculate*. Определить два производных класса, которые находят корни разными способами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

II вариант

Создать класс *автомобиль*, имеющий марку (указатель на строку), число цилиндров, мощность. Определить конструктор, деструктор и функцию печати. Создать *public*-производный класс – *грузовики*, имеющий грузоподъемность кузова. Определить конструктор с разным числом параметров, деструктор, функцию печати. Определить

функции переназначения марки и грузоподъемности как виртуальные. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

III вариант

Создать класс *человек*, имеющий имя (указатель на строку), возраст, вес. Определить конструктор, деструктор и функцию печати. Создать *public*-производный класс – *школьник*, имеющий класс (год обучения). Определить конструктор по умолчанию, деструктор, функцию печати. Определить функции переназначения возраста и класса как виртуальные. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

IV вариант

Создать класс *точка* и производные от него – *окружность* и *эллипс*. Определить конструкторы, деструктор и функцию печати. Определить функции переустановки центра окружности и эллипса как виртуальные. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Типовые варианты аудиторной контрольной работы №5

Выполнить задания контрольной работы, используя компоненты C#. Окно программы, выполняющей требуемые действия, должно содержать оптимальное количество элементов, их расположение должно удовлетворять требованиям эргономичности.

I вариант

1. Преобразовать одномерный массив, содержащий N элементов, по правилу: наименьший элемент массива \rightarrow наибольший из оставшихся элементов \rightarrow наименьший из оставшихся элементов (исключаем первые два, только что записанные), и т.д. Например, если массив содержал элементы $(10, -5, 17, 0, -4, 5)$, то после преобразования этот массив примет вид: $(-5, 17, -4, 10, 0, 5)$.
2. Во введенной строке символов найти пары, состоящие из одинаковых символов и заменить второй из них цифрой «1».

II вариант

1. Найти сумму простых чисел в одномерном массиве $[1, N]$.
2. Определить наличие слов в заданном тексте, содержащих сочетание символов, задаваемое с клавиатуры.

III вариант

1. В одномерном массиве размерности N найти самую длинную последовательность, образованную элементами, расположенными по возрастанию.
2. В заданном тексте определить, имеются ли слова-перевертыши и если да, то вывести их.

IV вариант

1. Дана матрица A размером $N \times M$. Найти сумму минимальных элементов всех столбцов матрицы.
2. Пусть имеется некоторая строка текста. Поменять местами слова с номерами k и n . Если какому-либо номеру не соответствует слово, вывести соответствующее сообщение.

Типовой вариант домашней контрольной работы №1

I вариант

1. Задана последовательность, содержащая только цифры от 0 до 9. Произвести сжатие последовательности по следующему правилу: если подряд идут несколько одинаковых цифр, то заменить их одним числом, первая цифра которого – сама повторяющаяся цифра, а другие образуют число повторений этой цифры. Например, последовательность цифр: 1,6,6,6,6,3,9,9,9,9,9,9,9,9,9,9,2,3,4,5,5,5,3 после сжатия приобретает вид: 1,64,3,911,2,3,4,54,3, т.к. цифра 6 встречается подряд 4 раза, цифра 9 встречается 11 раз, цифра 5 – встречается 4 раза. Определить коэффициент сжатия последовательности, принимая во внимание необходимый объем памяти для ее размещения в развернутом и сжатом состояниях.
2. Дана следующая информация о книгах библиотеки: автор, название книги, издательство, год выпуска, количество экземпляров книги в библиотеке. Написать программу, позволяющую осуществлять следующие операции: добавление и удаление информации о книге, поиск книг по году выпуска (автору, издательству, названию), просмотр информации о книгах библиотеки.

II вариант

1. Имеется последовательность 011212201220200112..., которая строится по следующему правилу: сначала пишется 0, затем повторяются следующие действия: уже написанную часть приписывают справа с заменой 0 на 1, 1 на 2, 2 на 0, т.е. по шагам имеем: $0 \rightarrow 01 \rightarrow 0112 \rightarrow 01121220 \rightarrow \dots$. Написать программу, которая по введенному N ($0 \leq N \leq 10000$) определяет, какое число стоит на k -й позиции в последовательности.
2. На складе медикаментов решили провести проверку лекарственных препаратов на истечение срока годности. Информация о препаратах заносится в список, состоящий из следующих полей: название лекарства, производитель, дата выпуска, срок годности, стоимость. Организовать ввод сведений о лекарственных препаратах, а также сформировать список препаратов с истекающим сроком годности. Предусмотреть поиск по названию лекарственного препарата и по годам выпуска.

Типовые варианты домашней контрольной работы №2

I вариант

Описать класс *TEdit*, демонстрирующий основные принципы работы компонента *textBox* в среде визуального программирования *Visual Studio (C#)*. Реализовать перемещение по тексту компонента (уже реализовано в рассмотренном выше примере), удаление символов в поле компонента, режим вставки или замены (в зависимости от установленного режима, при редактировании текста в поле компонента текст или раздвигается, или заменяется вновь вводимым), изменение других параметров компонента (ширина, высота, цвет текста, и т.д.).

II вариант

Описать класс *TMemo* (за основу можно взять приведенный пример класса *Edit*) – набор строк, демонстрирующий основные принципы работы компонента *textBox* в среде визуального программирования *Visual Studio (C#)*. Реализовать перемещение по тексту (с помощью стрелок управления курсором); удаление, изменение, добавление символов в любую строку; изменение других параметров компонента (ширина, высота, цвет текста, и т.д.). При этом набираемый текст, при достижении правой границы компонента, должен автоматически переноситься на следующую строку. Если же достигнута нижняя граница компонента, то должна быть реализована прокрутка строк компонента.

III вариант

Описать класс «Прямоугольник» со сторонами, параллельными осям координат (задается указанием координат начала и конца одной из его диагоналей). Для прямоугольника должны быть заданы стили отображения – цвет и тип границы, толщина границы, цвет и тип заливки. Предусмотреть методы: изменение размеров прямоугольника; перемещение прямоугольника по экрану; построение наименьшего прямоугольника, содержащего два заданных прямоугольника (реализовать в виде дружественной функции); получение прямоугольника, являющегося пересечением двух заданных прямоугольников.

Типовые варианты домашней контрольной работы №3

Выполнить задание, используя компоненты C#. Результатом выполнения задания должно быть приложение, удовлетворяющее требованиям функциональности и эргономичности.

I вариант

Написать простейший текстовый редактор на основе компонента *textBox*. Предусмотреть возможности открытия существующего файла и сохранения текста в окне редактора. Программа должна предусматривать изменение параметров текста в окне редактора (размер и цвет выделенного в окне редактора шрифта).

II вариант

Написать программу – простейший калькулятор (с одним полем ввода), выполняющий следующие действия: сложение, вычитание, умножение, деление, извлечение корня, вычисление логарифма (десятичного и натурального). Предусмотреть возможность промежуточного хранения двух результатов, которые могут быть использованы в дальнейших вычислениях.

III вариант

Составить программу – словарь терминов. Предусмотреть возможность добавления нового слова в словарь и изменение уже существующего. Программа должна содержать систему поиска слова. Данные программы (база слов и описаний) должны храниться в одном файле. Слова в базе хранятся в алфавитном порядке.

IV вариант

Написать программу, выполняющую построение графика функции (например, парабола). Коэффициенты функциональной зависимости задаются. Предусмотреть изменение цвета и

толщины графика функции. Построить систему координат с соответствующими подписями по осям.

V вариант

Написать программу – простейший графический редактор. Инструментарий редактора: карандаш, возможность рисовать простейшие геометрические фигуры (линия, прямоугольник). Предусмотреть возможность выбора цвета рисования.

VI вариант

Написать программу – простейший графический редактор. Инструментарий редактора: карандаш, ластик. Предусмотреть возможность выбора цвета рисования. Реализовать цветной ластик – стирает только то, что нарисовано указанным пользователем цветом.

Лабораторные работы

Типовые задания для выполнения лабораторных работ

Лабораторная работа №1. Программирование ветвлений и циклов.

1. Ввести с клавиатуры значения переменных a и b , вычислить и вывести на экран значение функции. В заданиях считать $x = \frac{a \cdot b + a}{b}$.

№	Функция	Тип данных
1	$y = a^x + b \cdot \operatorname{arctg}(a) - \ln \left \frac{b}{a} \right $	Вещественный
2	$y = \ln(a^b - 5) + e^{a+b} - \cos(bx)$	Целый
3	$y = e^a - \cos(x) + \sin(x) ^{a+b} - \ln(a)$	Вещественный
4	$y = e^{3 \cdot \left \sin\left(\frac{b}{x}\right) - 1 \right } - a^x$	Вещественный
5	$y = \frac{(e^{ax} + e^{-ax})}{(2 \cdot \sin(ax + b))}$	Целый
6	$y = \lg ax - bx + \frac{\cos(x)}{x}$	Вещественный
7	$y = \sin(x - a) + b^x - b \frac{x}{\cos(x - b)}$	Вещественный
8	$y = \frac{e^{a+b} - e^{a-b}}{x} + \lg(x - a)$	Вещественный
9	$y = a \cdot \cos(b^{ a-b }) + \sin(a) + \frac{e^{a-b}}{a+b-1}$	Вещественный
10	$y = \arcsin(x) - a^x + e^{a+b}$	Вещественный
11	$y = \ln(x) + \sin\left(\frac{x}{a}\right) + \frac{1}{\cos(x - 2 + b)}$	Вещественный

12	$y = e^x - \frac{a+b+\sin(x)}{x-b^a} + b^{\cos(a)}$	Целый
13	$y = \frac{1}{a \cdot (e^{ax} + e^{-ax})} + \frac{b}{x}$	Вещественный
14	$y = \lg ax+bx + \frac{\cos(x)+4}{1-x}$	Целый
15	$y = \sin(\sqrt{x}-a) + b^x - b \frac{x}{\sqrt{\cos(x-b)}}$	Вещественный

2. Найти сумму ряда y с точностью до члена ряда, меньшего 0.0001 для заданного значения x и сравнить эту сумму с функцией контроля. Для вычисления последующего значения члена ряда использовать рекуррентную формулу.

№	Разложение функции в ряд	Значение x	Функция контроля
1.	$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$	любое	e^x
2.	$1 + \frac{x \cdot \ln 3}{1!} + \frac{x^2 \cdot \ln^2 3}{2!} + \dots + \frac{x^n \cdot \ln^n 3}{n!}$	любое	3^x
3.	$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}$	любое	$\cos x$
4.	$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$	любое	$\sin x$
5.	$x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots + (-1)^{n-1} \frac{x^n}{n}$	$-1 < x < 1$	$\ln(x+1)$
6.	$x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2n+1}}{(2n+1)!}$	любое	$\frac{e^x - e^{-x}}{2}$
7.	$1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{2n}}{(2n)!}$	любое	$\frac{e^x + e^{-x}}{2}$
8.	$x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots + (-1)^n \frac{x^{2n+1}}{2n+1}$	$-1 < x < 1$	$\operatorname{arctg} x$
9.	$\prod_{i=1}^n \left(1 - \frac{4x^2}{(2i-1)^2 \pi^2} \right)$	0.5	$\cos x$
10.	$2 \left(x + \frac{x^3}{3} + \frac{x^5}{5} + \dots + \frac{x^{2n+1}}{2n+1} \right)$	$-1 < x < 1$	$\ln \left(\frac{1+x}{1-x} \right)$
11.	$2 \left(\frac{x-1}{x+1} + \frac{1}{3} \left(\frac{x-1}{x+1} \right)^3 + \dots + \frac{1}{2n+1} \left(\frac{x-1}{x+1} \right)^{2n+1} \right)$	$x > 0$	$\ln x$
12.	$x + \frac{1}{2} \frac{x^3}{3} + \frac{1 \cdot 3}{2 \cdot 4} \frac{x^5}{5} + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6} \frac{x^7}{7} + \dots$	$-1 < x < 1$	$\arcsin x$
13.	$1 - x + x^2 - x^3 + \dots + (-1)^n x^n$	$-1 < x < 1$	$\frac{1}{1+x}$

14.	$1 + \frac{x}{2} - \frac{x^2}{2 \cdot 4} + \frac{1 \cdot 3}{2 \cdot 4 \cdot 6} x^3 - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6 \cdot 8} x^4 + \dots$	$-1 \leq x \leq 1$	$\sqrt{1+x}$
15.	$\frac{x}{1 \cdot 2} + \frac{x^2}{2 \cdot 3} + \frac{x^3}{3 \cdot 4} + \dots + \frac{x^n}{n \cdot (n+1)}$	$-1 \leq x < 0,$ $0 < x < 1$	$1 + \frac{1-x}{x} \ln(1-x)$

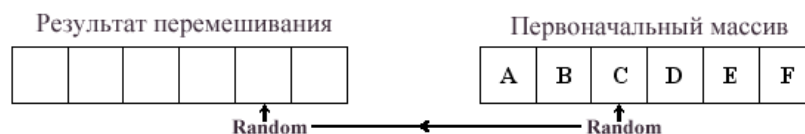
Лабораторная работа №2. Работа со статическими и динамическими массивами.

Одномерные массивы. Написать программу в соответствии с вариантами заданий, приведенными ниже.

1. Задана последовательность целых чисел. Сформировать на основе нее массив, содержащий номера отрицательных элементов последовательности. Если таких элементов нет, – вывести сообщение об этом.
2. Задана последовательность действительных чисел. Подсчитать количество чисел последовательности, расположенных на отрезке, концами которого являются два последних элемента массива.
3. Задана последовательность целых чисел, содержащая повторяющиеся элементы. Сформировать из нее два массива, один из которых содержит номера позиций минимальных элементов последовательности, а другой – номера позиций ее максимальных элементов.
4. В одномерном массиве заданного размера N найти элемент, меньше всего отличающийся от \sqrt{S} , где S – сумма элементов данного массива. Вывести номер элемента и его значение (если таких элементов несколько – для каждого из них) Предусмотреть, что сумма S может принимать отрицательные значения.
5. В одномерном массиве заданного размера N определить номера позиций, соседние элементы которых равны между собой. Если таких позиций нет, вывести соответствующее сообщение.
6. Написать программу, которая в данной последовательности действительных чисел a_1, a_2, \dots, a_n позволяет выяснить, имеются ли среди соседних элементов в этой последовательности взаимно обратные числа.
7. Написать программу, которая в данной последовательности целых положительных чисел a_1, a_2, \dots, a_n находит числа, встречающиеся более двух раз. Вывести номер числа и его значение. Если таких чисел нет – вывести соответствующее сообщение.
8. Задана последовательность, содержащая натуральные числа. Найти сумму максимального элемента (среди четных чисел) и минимального элемента (среди нечетных чисел) заданной последовательности.
9. В одномерном массиве размерности N найти самую длинную последовательность, образованную элементами, расположенными по убыванию.
10. В одномерном массиве целых чисел подсчитать количество раз встречаемости каждого из элементов.
11. В одномерном массиве проверить наличие пифагоровых троек (элементы массива, образующие пифагоровы тройки, могут находиться в разных частях массива) и вывести их, если имеются.
12. Задана последовательность действительных чисел. Преобразовать последовательность таким образом, чтобы вначале в ней шли в порядке

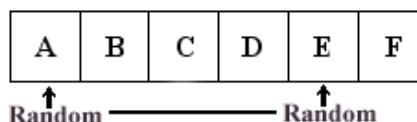
возрастания отрицательные числа, а затем все остальные. Причем при формировании такой последовательности отрицательные элементы меняются местами с соответствующими занимаемой ими позицией положительными элементами. Например, если исходная последовательность: 1.3, -2.7, 2.1, 5, 3, 7, 5.2, -1.1, 4, -1.2, то после преобразования она примет вид: -2.7, -1.2, -1.1, 5, 3, 7, 5.2, 2.1, 4, 1.3. Однако возможен и другой вид преобразованной последовательности (все зависит от используемого при реализации программы алгоритма получения конечного результата): -2.7, -1.2, -1.1, 5, 3, 7, 5.2, 1.3, 4, 2.1.

13. Организовать перемешивание элементов одномерного массива так, чтобы процесс был наиболее коротким, и каждый элемент массива перемещался не более одного раза в новую позицию, согласно представленному рисунку.



14. Организовать перемешивание элементов одномерного массива так, чтобы процесс был наиболее коротким, и каждый элемент массива перемещался не более одного раза в новую позицию, согласно представленному рисунку.

Обмен позициями элементов исходного массива, содержащего четное количество элементов



15. В одномерном массиве заданы координаты N - точек в пространстве путем перечисления вначале всех x - координат, затем y - координат, и потом z - координат. Определить координаты центра и радиус сферы, вмещающей все указанные в массиве точки.

Двумерные массивы. Написать программу в соответствии с вариантами заданий, приведенными ниже.

1. Дана матрица A размером $N \times M$. Найти номер столбца, элементы которого образуют возрастающую последовательность, или вывести сообщение, что такого столбца нет.
2. Дана матрица A размером $N \times M$. Найти сумму минимальных элементов всех столбцов матрицы.
3. Дана матрица A размером $N \times M$. Вычислить суммы элементов каждого столбца. Определить наибольшее значение этих сумм и номер соответствующего столбца.
4. Даны матрица A размером $N \times M$ и вектор B размером N . Записать на главную диагональ элементы вектора, а в вектор – элементы главной диагонали.
5. В заданной матрице A размером $N \times M$ найти максимальный элемент и удалить строку и столбец, на пересечении которых стоит данный элемент.
6. В матрице размером $N \times M$ поменять местами главную диагональ и первый столбец.
7. В матрице размером $N \times M$ найти наибольший элемент главной диагонали и заменить нулями все элементы матрицы, кроме элементов, находящихся в строке и столбце, содержащих этот наибольший элемент.

8. В заданной матрице A размером $N \times M$ найти максимальный и минимальный элементы. Поменять местами столбец и строку, содержащие минимальный элемент, соответственно со столбцом и строкой, содержащими максимальный элемент.
9. Для матрицы размером $N \times M$ вывести положение всех ее седловых точек. Элемент матрицы называется седловой точкой, если его значение является минимальным в строке, но максимальным в столбце.
10. В заданной матрице A размером $N \times M$ найти максимальный элемент и переставить столбец и строку, на пересечении которых находится данный элемент. Недостающие элементы строки (столбца) заменить нулями, лишние элементы столбца (строки) отбросить.
11. В трехмерном пространстве заданы координаты N точек. Составить программу, определяющую пару точек, расстояние между которыми максимально.
12. Из заданного одномерного массива A размером N получить квадратную матрицу B размером $M \times M$ таким образом, чтобы ее размер был минимально необходимым для вмещения всех элементов массива A . Недостающие элементы заменить нулями (не отбрасывая элементы матрицы A).
13. В заданной матрице A размером $N \times M$ разместить все положительные элементы в верхней области матрицы (заполняя ими матрицу по строкам слева направо), а неположительные – в нижней области. Нулевые элементы оставить на месте.
14. Пусть задана матрица A размером $N \times M$. Получить из нее матрицу B размером $K \times 3$ по правилу: первый столбец состоит из положительных элементов, второй – из отрицательных, третий – из нулей. Недостающие элементы в первом и втором столбце заменить нулями, в третьем – единицами. Ни один из элементов матрицы A не должен быть отброшен.
15. Пусть дан двумерный массив A размером $N \times M$. Выделить из него массив B размером $K \times L$ и найти в этом массиве элемент, больше всего повторяющийся в оставшейся части массива A , или сообщить о том, что такого элемента нет. Позиция (i, j) первого элемента массива B в массиве A задается в процессе работы программы. В случае если при этом заданные K или L выходят за пределы массива A , то они уменьшаются до необходимого значения.

Лабораторная работа №3. Работа со строками.

Написать программу в соответствии с вариантами заданий, приведенными ниже. Предусмотреть все возможные варианты исходных данных.

1. Во введенной строке текста найти пару слов, идущих подряд, которые имеют общую максимальную длину. Если между словами в строке один пробел, он дает вклад в общую длину двух слов, если пробелов более чем один, то все, кроме одного, отбрасываются.
2. Каждое слово текста преобразовать таким образом, чтобы оно читалось справа налево.
3. Раздвинуть заданный текст, вставив заданную последовательность после i -го символа каждого слова. Если номер i превышает длину слова, то такое слово не изменяется.

4. В заданном тексте определить, имеются ли слова-перевертыши и если да, то вывести их.
5. Написать свою функцию $Str_Insert(St1, St2, N)$ для вставки строки $St2$ в строку $St1$, начиная с позиции N .
6. Написать свою функцию $Str_Delete(St, N, M)$ для удаления N символов из строки St , начиная с позиции M .
7. В строку через пробел вводятся слова. Отсортировать введенные слова по длине в порядке возрастания. Результат вывести в виде столбца.
8. Пусть имеется некоторая строка текста. Поменять местами слова с номерами k и n . Если какому-либо номеру не соответствует слово, вывести соответствующее сообщение. Организовать работу программы для двух способов реализации: а) с использованием массива; б) без использования массива.
9. В заданном тексте, содержащем повторяющиеся слова, заменить указанное слово на другое. Заменяемое и заменяющее слова вводятся с клавиатуры.
10. Во введенной строке текста удалить одинаковые слова, идущие друг за другом.
11. Во введенном тексте подсчитать, сколько раз встречается каждый из символов. Вывести результат работы программы в виде двух колонок, в каждой из которых указан символ и количество раз его встречаемости в тексте. Предложить способ, отличающийся от приведенного в примере 1.
12. Во введенном тексте подсчитать, сколько раз встречаются повторяющиеся символы. Вывести результат работы программы в виде двух колонок, в каждой из которых указан символ и количество раз его использования в тексте.
13. Пусть имеется строка символов (текстовая строка). Подсчитать, сколько раз встречается каждая буква в тексте и на этой основе составить таблицу относительной частоты встречаемости каждой из букв в тексте. Затем, используя таблицу встречаемости, в произвольно введенной строке текста, содержащей вместо некоторых одинаковых букв символы "*", "?", "#", "!", попытаться восстановить недостающие буквы.
14. Задан некоторый текст, в котором слова содержат заглавные и прописные буквы. Изменить текст так, чтобы слова содержали только прописные буквы (заменить заглавные буквы на прописные). Причем если слово начинает новое предложение, оно должно начинаться с заглавной буквы. Считать признаком конца предложения символы "?", "!", ".".
15. Заданы две строки, содержащие текст. Проверить, можно ли из букв второй строки составить текст, содержащийся в первой строке. Можно использовать не более того количества разных букв, которое имеется во второй строке.

Лабораторная работа №4. Функции. Структуры (struct) в C++.

В соответствии с вариантом заполнить массив из структур, написать и выполнить программу, используя функции.

Вариант 1

1. Описать структуру с именем STUDENT, содержащую следующие поля:
 - фамилия и инициалы;
 - номер группы;
 - успеваемость (массив из пяти элементов).

2. Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию номера группы;
 - вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4.0;
 - если таких студентов нет, вывести соответствующее сообщение.

Вариант 2

1. Описать структуру с именем STUDENT, содержащую следующие поля:
- фамилия и инициалы;
 - номер группы;
 - успеваемость (массив из пяти элементов).
2. Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по возрастанию среднего балла;
 - вывод на дисплей фамилий и номеров групп для всех студентов, имеющих оценки 4 и 5;
 - если таких студентов нет, вывести соответствующее сообщение.

Вариант 3

1. Описать структуру с именем STUDENT, содержащую следующие поля:
- фамилия и инициалы;
 - номер группы;
 - успеваемость (массив из пяти элементов).
2. Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT; записи должны быть упорядочены по алфавиту;
 - вывод на дисплей фамилий и номеров групп для всех студентов, имеющих хотя бы одну оценку 2;
 - если таких студентов нет, вывести соответствующее сообщение.

Вариант 4

1. Описать структуру с именем AEROFLOT, содержащую следующие поля:
- название пункта назначения рейса;
 - номер рейса;
 - тип самолета.
2. Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в массив, состоящий из семи элементов типа AEROFLOT; записи должны быть упорядочены по возрастанию номера рейса;
 - вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры;
 - если таких рейсов нет, выдать на дисплей соответствующее сообщение.

Вариант 5

1. Описать структуру с именем AEROFLOT, содержащую следующие поля:
- название пункта назначения рейса;
 - номер рейса;
 - тип самолета.
2. Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в массив, состоящий из семи элементов типа AEROFLOT; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;

- вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры;
- если таких рейсов нет, выдать на дисплей соответствующее сообщение.

Вариант 6

1. Описать структуру с именем WORKER, содержащую следующие поля:
 - фамилия и инициалы работника;
 - название занимаемой должности;
 - год поступления на работу.
2. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в массив, состоящий из десяти структур типа WORKER; записи должны быть размещены по алфавиту.
 - вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры;
 - если таких работников нет, вывести на дисплей соответствующее сообщение.

Вариант 7

1. Описать структуру с именем TRAIN, содержащую следующие поля:
 - название пункта назначения;
 - номер поезда;
 - время отправления.
2. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в массив, состоящий из восьми элементов типа TRAIN; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;
 - вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени;
 - если таких поездов нет, выдать на дисплей соответствующее сообщение.

Вариант 8

1. Описать структуру с именем TRAIN, содержащую следующие поля:
 - название пункта назначения;
 - номер поезда;
 - время отправления.
2. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в массив, состоящий из шести элементов типа TRAIN; записи должны быть упорядочены по времени отправления поезда;
 - вывод на экран информации о поездах, направляющихся в пункт, название которого введено с клавиатуры;
 - если таких поездов нет, выдать на дисплей соответствующее сообщение.

Вариант 9

1. Описать структуру с именем TRAIN, содержащую следующие поля:
 - название пункта назначения;
 - номер поезда;
 - время отправления.
2. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в массив, состоящий из восьми элементов типа TRAIN; записи должны быть упорядочены по номерам поездов;
 - вывод на экран информации о поезде, номер которого введен с клавиатуры;
 - если таких поездов нет, выдать на дисплей соответствующее сообщение.

Вариант 10

1. Описать структуру с именем MARSH, содержащую следующие поля:
 - название начального пункта маршрута;
 - название конечного пункта маршрута;
 - номер маршрута.
2. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в массив, состоящий из восьми элементов типа MARSH; записи должны быть упорядочены по номерам маршрутов;
 - вывод на экран информации о маршруте, номер которого введен с клавиатуры;
 - если таких маршрутов нет, выдать на дисплей соответствующее сообщение.

Вариант 11

1. Описать структуру с именем MARSH, содержащую следующие поля:
 - название начального пункта маршрута;
 - название конечного пункта маршрута;
 - номер маршрута.
2. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в массив, состоящий из восьми элементов типа MARSH; записи должны быть упорядочены по номерам маршрутов;
 - вывод на экран информации о маршрутах, которые начинаются или кончаются в пункте, название которого введено с клавиатуры;
 - если таких маршрутов нет, выдать на дисплей соответствующее сообщение.

Вариант 12

1. Описать структуру с именем NOTE, содержащую следующие поля:
 - фамилия, имя;
 - номер телефона;
 - день рождения (массив из трех чисел).
2. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в массив, состоящий из восьми элементов типа NOTE; записи должны быть упорядочены по датам дней рождения;
 - вывод на экран информации о человеке, номер телефона которого введен с клавиатуры;
 - если такого нет, выдать на дисплей соответствующее сообщение.

Вариант 13

1. Описать структуру с именем NOTE, содержащую следующие поля:
 - фамилия, имя;
 - номер телефона;
 - день рождения (массив из трех чисел).
2. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в массив, состоящий из восьми элементов типа NOTE; записи должны быть размещены по алфавиту;
 - вывод на экран информации о людях, чьи дни рождения приходятся на месяц, значение которого введено с клавиатуры;
 - если таких нет, выдать на дисплей соответствующее сообщение.

Вариант 14

1. Описать структуру с именем NOTE, содержащую следующие поля:
 - фамилия, имя;
 - номер телефона;
 - день рождения (массив из трех чисел).

2. Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа NOTE; записи должны быть упорядочены по трем первым цифрам номера телефона;
 - вывод на экран информации о человеке, чья фамилия введена с клавиатуры;
 - если такого нет, выдать на дисплей соответствующее сообщение.

Вариант 15

1. Описать структуру с именем ZNAK, содержащую следующие поля:
- фамилия, имя;
 - знак Зодиака;
 - день рождения (массив из трех чисел).
2. Написать программу, выполняющую следующие действия:
- ввод с клавиатуры данных в массив, состоящий из восьми элементов типа ZNAK; записи должны быть упорядочены по датам дней рождения;
 - вывод на экран информации о человеке, чья фамилия введена с клавиатуры;
 - если такого нет, выдать на дисплей соответствующее сообщение.

Лабораторная работа №5. Файлы в C++.

Вариант 1

Для хранения данных о цветных планшетных сканерах описать структуру вида:

```
struct scan_info
{ char model[25]; // наименование модели
  int price; // цена
  double x_size; // горизонтальный размер области сканир-ия
  double y_size; // вертикальный размер области сканирования
  int optr; // оптическое разрешение
  int grey; // число градаций серого
};
```

Написать функцию, которая записывает в бинарный файл данные о сканере из приведенной структуры. Структура файла: в первых двух байтах размещается значение типа int, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах.

Написать функцию, которая извлекает из этого файла данные о сканере в структуру типа scan_info. Обязательный параметр — номер требуемой записи. Функция должна возвращать нулевое значение, если чтение прошло успешно, и – 1 в противном случае.

Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) — 6–8 записей и выводящей на дисплей данные о запрошенной записи.

Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 2

Для хранения данных о планшетных сканерах описать структуру вида, описанного в варианте 1.

Написать функцию, которая записывает в бинарный файл данные о сканере из приведенной структуры. Структура файла: в первых двух байтах размещается значение типа int, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах.

Написать функцию, которая сортирует записи в описанном выше бинарном файле по одной из следующих характеристик: цена либо число градаций серого. Обязательный параметр – признак, задающий критерий сортировки.

Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) из не менее восьми записей и осуществляющий его сортировку.

Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 3

Для хранения данных о планшетных сканерах описать структуру вида, описанного в варианте 1.

Написать функцию, которая записывает в бинарный файл данные о сканере из приведенной структуры. Структура файла: в первых четырех байтах размещается значение типа long, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах.

Написать функцию, которая сортирует записи в описанном выше бинарном файле по наименованию модели сканера.

Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) из не менее восьми записей и осуществляющий его сортировку.

Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 4

Для хранения данных о планшетных сканерах описать структуру вида, описанного в варианте 1.

Написать функцию, которая динамически выделяет память под массив структур (не меньше шести элементов), заполняет его данными в режиме диалога и записывает массив в бинарный файл. Структура файла: в первых двух байтах размещается значение типа int, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах,

Написать функцию, которая извлекает данные о сканере из описанного выше бинарного файла в структуру типа scan_info. Обязательный параметр — номер требуемой записи. Функция должна возвращать нулевое значение, если чтение прошло успешно, и –1 в противном случае.

Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) из не менее восьми записей и осуществляющий вывод на дисплей данных о требуемой записи.

Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 5

Для хранения данных о планшетных сканерах описать структуру вида, описанного в варианте 1.

Написать функцию, которая записывает данные о сканере из приведенной структуры в требуемую позицию в бинарном файле. Структура файла: в первых двух байтах размещается значение типа `int`, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах. Запись может осуществляться в любую позицию, причем если между вводимой записью и последней (или началом файла) имеются пропуски, они заполняются нулями.

Написать функцию, которая «уплотняет» описанный выше бинарный файл путем удаления из него записей, содержащих все нули.

Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) из не менее шести записей и осуществляющий его уплотнение.

Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 6

Для хранения данных о планшетных сканерах описать структуру вида, описанного в варианте 1.

Написать функцию, которая динамически выделяет память под массив структур (не меньше шести элементов), заполняет его данными в режиме диалога и записывает массив в бинарный файл. Структура файла: в первых двух байтах размещается значение типа `int`, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах.

Написать функцию, которая запрашивает данные о сканере в режиме диалога и замещает записи в бинарном файле по заданному номеру. Обязательный параметр — номер замещаемой записи. Функция должна возвращать нулевое значение, если запись прошла успешно, и `-1` в противном случае.

Привести пример программы, создающей файл с данными о сканерах (данные вводятся с клавиатуры) из не менее восьми записей и осуществляющий вставку новых данных о сканере.

Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 7

Для хранения данных о планшетных сканерах описать структуру вида, описанного в варианте 1.

Написать функцию, которая записывает в бинарный файл данные о сканере из приведенной структуры. Структура файла: в первых двух байтах размещается значение типа `int`, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах.

Написать функцию, которая вводит данные о сканере с клавиатуры в структуру типа `scan-info`, и если данные об этом сканере отсутствуют в файле, помещает содержимое структуры в конец файла; в противном случае выдает соответствующее сообщение.

Привести пример программы, создающей файл с данными о сканерах (данные вводятся из текстового файла) — 6–8 записей и дополняющей файл записями о 2–3 сканерах, вводимых с клавиатуры.

Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 8

Для хранения данных о планшетных сканерах описать структуру вида, описанного в варианте 1,

Написать функцию, которая записывает в бинарный файл данные о сканере из приведенной структуры. Структура файла: в первых двух байтах размещается значение типа `int`, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах.

Написать функцию, которая вводит данные о сканере с клавиатуры в структуру типа `scan-info` и помещает ее содержимое на место первой записи в файле. Файл должен существовать. При этом, запись ранее занимавшая первую позицию, помещается на вторую, вторая запись на третью, и т. д.

Привести пример программы, создающей файл с данными о сканерах (данные вводятся из текстового файла) — 6–8 записей и дополняющей этот файл 1-2 новыми записями, вводимыми с клавиатуры.

Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 9

Для хранения данных о планшетных сканерах описать структуру вида, описанного в варианте 1.

Написать функцию, которая запрашивает количество сканеров, информация о которых будет вводиться, динамически выделяет память под массив структур соответствующего размера и заполняет его данными в режиме диалога (с клавиатуры). При этом имя сканера может содержать пробелы.

Написать функцию, которая записывает данный массив в создаваемый бинарный файл. Если цена сканера меньше 200, то данные об этом сканере в файл не записываются. Информация об остальных сканерах помещается в бинарный файл, причем сначала пишутся данные о всех сканерах, имя которых начинается с заглавной буквы, а затем — с прописной.

Структура файла: в первых четырех байтах размещается значение типа `long`, определяющее количество сделанных в файл записей; далее без пропусков размещаются записи о сканерах.

Привести пример программы, создающей файл с данными о сканерах и осуществляющий вывод на дисплей данных о требуемой записи (либо всех, либо по номеру).

Все необходимые данные для функций должны передаваться им в качестве параметров. Использование глобальных переменных в функциях не допускается.

Вариант 10

Для хранения данных о ноутбуках описать структуру вида (при необходимости дополнив ее):

```

struct NOTEBOOK{
    char model[21]; // наименование
    struct size
    { // габаритные размеры
        float x;
        float y;
        float z;
    };
    float w; // вес
    int price; // цена
}

```

Написать функцию, которая читает данные о ноутбуках из файла note.txt (см. вар. 20) в структуру приведенного вида. Написать функцию, которая записывает содержимое структуры в конец бинарного файла. Структура бинарного файла: первые два байта (целое) — число записей в файле; далее записи в формате структуры NOTEBOOK.

Написать программу, в которой на основе разработанных функций осуществляется чтение данных только для тех ноутбуков, частота процессора которых больше 120 МГц, и запись в бинарный файл по убыванию цены.

Вариант 11

Для хранения данных о ноутбуках описать структуру вида, описанного в варианте 10.

Написать функцию, которая читает данные о ноутбуках из файла note.txt (см. вар. 20) в структуру приведенного вида. Написать функцию, которая записывает содержимое структуры в конец бинарного файла. Структура бинарного файла: первые два байта (целое) — число записей в файле; далее записи в формате структуры NOTEBOOK.

Написать программу, в которой на основе разработанных функций осуществляется чтение данных только для тех ноутбуков, объем HDD которых меньше 1 Гбайт, и запись считанных данных в бинарный файл в алфавитном порядке по наименованию.

Вариант 12

Для хранения данных о ноутбуках описать структуру вида, описанного в варианте 10.

Написать функцию, которая читает данные о ноутбуках из файла note.txt (см. вар. 20) в структуру приведенного вида. Написать функцию, которая записывает содержимое структуры в конец бинарного файла. Структура бинарного файла: первые два байта (целое) — число записей в файле; далее записи в формате структуры NOTEBOOK.

Написать программу, в которой на основе разработанных функций осуществляется запись в двоичный файл данных только о тех ноутбуках, целое количество которых в одном кубическом метре не превышает 285 штук.

Вариант 13

Для хранения данных о ноутбуках описать структуру вида, описанного в варианте 10.

Написать функцию, которая читает данные о ноутбуках из файла note.txt (см. вар. 20) в структуру приведенного вида. Написать функцию, которая записывает содержимое структуры в конец бинарного файла. Структура бинарного файла: первые два байта (целое) — число записей в файле; далее записи в формате структуры NOTEBOOK.

Написать программу, в которой на основе разработанных функций осуществляется запись в двоичный файл данных только о тех ноутбуках, максимальный объем ОЗУ которых не менее 40 Мбайт, отсортированных по объему.

Вариант 14

Для хранения данных о ноутбуках описать структуру вида, описанного в варианте 10.

Написать функцию, которая читает данные о ноутбуках из файла note.txt (см. вар. 20) в структуру приведенного вида. Написать функцию, которая записывает содержимое структуры в конец бинарного файла. Структура бинарного файла: первые два байта — целое число записей в файле; далее записи в формате структуры NOTEBOOK.

Написать программу, в которой на основе разработанных функций осуществляется запись в двоичный файл данных только о тех ноутбуках, диагональ дисплея которых больше одиннадцати дюймов.

Вариант 15

Для хранения данных о ноутбуках описать структуру вида (при необходимости дополнив ее):

```
struct NOTEBOOK
{
    struct disp_res
    {
        // разрешающая способность дисплея
        int x: // по горизонтали
        int y: // по вертикали
    };
    int f: // частота регенерации
    float d; // размер диагонали дисплея
    int price; // цена
    char model[21]; // наименование
};
```

Написать функцию, которая читает данные о ноутбуках из файла note.txt (см. вар. 20) в структуру приведенного вида. Написать функцию, которая записывает содержимое структуры в конец бинарного файла. Структура бинарного файла:

первые два байта — целое число записей в файле; далее записи в формате структуры NOTEBOOK.

Написать программу, в которой на основе разработанных функций осуществляется запись в двоичный файл данных только о тех ноутбуках, вес которых менее 7 кг, отсортированных в порядке возрастания цены.

Лабораторная работа №6. Объекты и классы. Введение.

Вариант 1

Построить класс для описания плоской геометрической фигуры: круг. Предусмотреть методы для создания объекта, перемещения на плоскости, изменения размеров. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 2

Построить класс для описания плоской геометрической фигуры: квадрат. Предусмотреть методы для создания объекта, перемещения на плоскости, изменения размеров. Написать

программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 3

Построить класс для описания плоской геометрической фигуры: прямоугольник. Предусмотреть методы для создания объекта, перемещения на плоскости, изменения размеров. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 4

Построить описание класса, содержащего информацию в почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 5

Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 6

Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 7

Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменения размеров, построения наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 8

Составить описание класса для определения одномерных массивов целых чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы массива, возможность задания произвольных границ индексов при создании объекта, вывода на экран элемента массива по заданному индексу, вывода на экран всего массива. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 9

Составить описание класса для определения одномерных массивов строк фиксированной длины. Предусмотреть возможность обращения к отдельным строкам массива по индексам, контроль выхода за пределы массива, вывод на экран элемента массива по заданному индексу и всего массива. Написать программу, демонстрирующую работу с

этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 10

Составить описание класса многочленов от одной переменной, задаваемых степенью многочлена и массивом коэффициентов. Предусмотреть методы для вычисления значения многочлена для заданного аргумента, операции сложения и вычитания многочленов с получением нового объекта-многочлена, вывод на экран описания многочлена. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 11

Составить описание класса одномерных массивов строк, каждая строка задается длиной и указателем на выделенную для нее память. Предусмотреть контроль выхода за пределы массивов, возможность обращения к отдельным строкам массива по индексам, а также вывод на экран элемента массива и всего массива. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 12

Составить описание класса, обеспечивающего представление матрицы произвольного размера с возможностью изменения числа строк и столбцов, вывода на экран подматрицы любого размера и всей матрицы. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 13

Написать класс для эффективной работы со строками, позволяющий форматировать и сравнивать строки, хранить в строках числовые значения и извлекать их. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 14

Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (например, по автору или по году издания), добавления книг в библиотеку, удаления книг из нее, сортировки книг по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Вариант 15

Описать класс «записная книжка». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа может содержать меню, позволяющее осуществить проверку всех методов класса.

Лабораторная работа №7. Производные классы. Наследование.

При выполнении заданий необходимо написать программу, демонстрирующую работу с каждым конкретным классом. Программа должна содержать пункты меню, позволяющие осуществить проверку всех методов класса.

Вариант 1

Создать класс *студент*, имеющий имя (указатель на строку), курс и идентификационный номер. Определить конструктор, деструктор и функцию печати. Создать public-производный класс – *студент-дипломник*, имеющий тему диплома. Определить конструктор по умолчанию, деструктор, функцию печати. Определить функции переназначения названия диплома и идентификационного номера. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 2

Создать класс *автомобиль*, имеющий марку (указатель на строку), число цилиндров, мощность. Определить конструктор, деструктор и функцию печати. Создать public-производный класс – *грузовики*, имеющий грузоподъемность кузова. Определить конструктор с разным числом параметров, деструктор, функцию печати. Определить функции переназначения марки и грузоподъемности. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 3

Создать класс *карта*, имеющая ранг и масть. Карту можно перевернуть и открыть. Создать класс – *колода карт*, содержащий карты. Создать два производных класса от колоды карт, в одном карты могут доставаться только по порядку, в другом – вытаскиваться произвольно. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 4

Создайте класс *точка*, которая имеет координаты. Класс *эллипсов*, и класс *окружностей*. Определить иерархию типов. Определить функции печати, конструкторы, деструкторы, вычисление площади. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 5

Создать класс *четырехугольников, квадратов и прямоугольников*. Создать из них иерархию. Определить функции печати, конструкторы и деструкторы, вычисление площади и периметра. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 6

Создать абстрактный класс *CVehicle*. На его основе реализовать классы *CPlane*, *CCar* и *CShip*. Классы должны иметь возможность задавать и получать координаты, параметры средств передвижения (цена, скорость, год выпуска). Для самолета должна быть определена высота, для самолета и корабля – количество пассажиров. Для корабля – порт приписки.

Написать программу, создающую список объектов этих классов в динамической памяти. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 7

Создать класс *человек*, имеющий имя (указатель на строку), возраст, вес. Определить конструктор, деструктор и функцию печати. Создать public-производный класс – *школьник*, имеющий класс (год обучения). Определить конструктор по умолчанию, деструктор, функцию печати. Определить функции переназначения возраста и класса. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 8

Создать класс *окно*, имеющий координаты верхнего левого и нижнего правого угла, цвет фона (указатель на строку). Определить конструктор, деструктор и функцию печати. Создать public-производный класс – *окно с меню*, имеющий строку меню. Определить конструктор с разным числом параметров, деструктор, функцию печати. Определить функции переназначения цвета фона и строки меню. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 9

Создать класс *точка*, имеющая координаты. Определить конструкторы, деструктор и функцию печати. Создать public-производный класс – *цветная точка*, имеющий цвет точки. Определить конструктор по умолчанию, деструктор, функцию печати. Определить функции переназначения цвета и координат точки, вывода точки на экран. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 10

Создать класс *человек*, имеющий имя (указатель на строку), возраст, вес. Определить конструктор, деструктор и функцию печати. Создать public-производный класс – *совершеннолетний*, имеющий номер паспорта. Определить конструктор по умолчанию, функцию печати. Определить функции переназначения возраста и номера паспорта. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 11

Создать *колоду карт*. Конструкторы колоды должны инициализировать колоду упорядочено и случайным образом. Создать производный класс от колоды – *пасьянс*, в котором выбираются по три карты и, если две крайние одного цвета, то их выбрасывают. Всю колоду проходят три раза. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 12

Создать класс *точка* и производные от него – *окружность* и *эллипс*. Определить конструкторы, деструктор и функцию печати. Определить функции переустановки центра окружности и эллипса. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 13

Создать класс *CFile*, инкапсулирующий в себе такие функции работы с файлами, как Open, Close, Seek, Read, Write, GetPosition и GetLength. На базе этого класса создать производный класс *CMyDataFile* – файл, содержащий в себе данные некоторого определенного типа MyData, а также заголовок, облегчающий доступ к этому файлу. Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществлять проверку всех методов класса.

Вариант 14

Создать класс *CPoint* – точка. На его основе создать классы *CColoredPoint* и *CLine*. На основе класса *CLine* создать класс *CColoredLine* и класс *CPolyLine* – многоугольник. Все классы должны иметь методы для установки и получения значений всех координат, а также изменения цвета и получения текущего цвета.

Написать демонстрационную программу, в которой будет использоваться список объектов этих классов в динамической памяти.

Вариант 15

Создать иерархию классов для моделирования взаимодействия «*Магазин-склад-Заказчик*». В распоряжении магазина имеется:

- склад с товарами различных наименований, стоимости и веса;
- гараж с машинами различной грузоподъемности, вместительности и потребляемой мощности.

В магазин поступают заказы на поставку определенного количества товаров различных наименований в заданный район (задается как расстояние между магазином и районом). Магазин должен выбрать наиболее подходящее транспортное средство (учитывать, что машины могут быть заняты другими заказчиками) и определить счет за поставку товара.

Лабораторная работа №8. Полиморфизм. Виртуальные функции.

При выполнении заданий необходимо написать программу, демонстрирующую работу с каждым конкретным классом. Программа должна содержать пункты меню, позволяющие осуществить проверку всех методов класса.

Вариант 1

Создать класс *студент*, имеющий имя (указатель на строку), курс и идентификационный номер. Определить конструктор, деструктор и функцию печати. Создать public-производный класс – *студент-дипломник*, имеющий тему диплома. Определить конструктор по умолчанию, деструктор, функцию печати. Определить функции переназначения названия диплома и идентификационного номера как виртуальные. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 2

Создать класс *автомобиль*, имеющий марку (указатель на строку), число цилиндров, мощность. Определить конструктор, деструктор и функцию печати. Создать public-производный класс – *грузовики*, имеющий грузоподъемность кузова. Определить конструктор с разным числом параметров, деструктор, функцию печати. Определить функции переназначения марки и грузоподъемности как виртуальные. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 3

Создать класс *карта*, имеющая ранг и масть. Карту можно перевернуть и открыть. Создать класс – *колода карт*, содержащий карты. Создать два производных класса от колоды карт, в одном карты могут доставаться только по порядку, в другом – вытаскиваться произвольно. Создать данные методы как виртуальные. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 4

Создайте класс *точка*, которая имеет координаты. Класс *эллипсов*, и класс *окружностей*. Определить иерархию типов. Определить функции печати, конструкторы, деструкторы, вычисление площади. Функция вычисления площади должна быть виртуальной. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 5

Создать класс *четырёхугольников, квадратов и прямоугольников*. Создать из них иерархию. Определить функции печати, конструкторы и деструкторы, вычисление площади и периметра. Функции вычисления площади и периметра должна быть виртуальными. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 6

Создать абстрактный класс *CVehicle*. На его основе реализовать классы *CPlane*, *CCar* и *CShip*. Классы должны иметь возможность задавать и получать координаты, параметры средств передвижения (цена, скорость, год выпуска). Для самолета должна быть определена высота, для самолета и корабля – количество пассажиров. Для корабля – порт приписки.

Написать программу, создающую список объектов этих классов в динамической памяти. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 7

Создать класс *человек*, имеющий имя (указатель на строку), возраст, вес. Определить конструктор, деструктор и функцию печати. Создать public-производный класс – *школьник*, имеющий класс (год обучения). Определить конструктор по умолчанию, деструктор, функцию печати. Определить функции переназначения возраста и класса как виртуальные. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 8

Создать класс *окно*, имеющий координаты верхнего левого и нижнего правого угла, цвет фона (указатель на строку). Определить конструктор, деструктор и функцию печати. Создать public-производный класс – *окно с меню*, имеющий строку меню. Определить конструктор с разным числом параметров, деструктор, функцию печати. Определить функции переназначения цвета фона и строки меню как виртуальные. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 9

Создать класс *точка*, имеющая координаты. Определить конструкторы, деструктор и функцию печати. Создать public-производный класс – *цветная точка*, имеющий цвет точки. Определить конструктор по умолчанию, деструктор, функцию печати. Определить функции переназначения цвета и координат точки, вывода точки на экран как виртуальные. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Вариант 10

Создать класс *человек*, имеющий имя (указатель на строку), возраст, вес. Определить конструктор, деструктор и функцию печати. Создать public-производный класс – *совершеннолетний*, имеющий номер паспорта. Определить конструктор по умолчанию, функцию печати. Определить функции переназначения возраста и номера паспорта как виртуальные. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Выполнить задания лабораторной работы №13, используя при этом абстрактные классы и чисто виртуальные функции.

При выполнении заданий необходимо написать программу, демонстрирующую работу с каждым конкретным классом. Программа должна содержать пункты меню, позволяющие осуществить проверку всех методов класса.

Лабораторная работа №10. Перегрузка операций.

Вариант 1

Создать класс **Vector** (вектор целых чисел), переопределить для него операции:

- а) «+=» – сложение векторов с сохранением значения в одном из них;
- б) «+» – сложение векторов с сохранением значения в новом векторе;
- в) «-» – разность векторов;
- г) «[]» – доступ к заданному элементу;
- д) «>>» – ввод элементов вектора с консоли;
- е) «<<» – вывод элементов вектора на экран или в файл.

Вариант 2

Разработайте класс **Polynomials** (полиномы). Внутренним представлением класса **Polynomials** является массив членов полинома. Каждый член содержит коэффициент и показатель степени. Член $2x^4$ имеет коэффициент 2 и показатель степени 4. Разработайте полный класс, содержащий соответствующие функции конструктора и деструктора. Класс должен обеспечивать путем использования перегруженных операций следующие возможности:

- а) перегрузить операцию сложения (+), чтобы складывать два объекта класса **Polynomials**;
- б) перегрузить операцию вычитания (-), чтобы вычитать два объекта класса **Polynomials**;
- в) перегрузить операцию присваивания (=), чтобы присваивать один объект класса **Polynomials** другому;
- г) перегрузить операцию умножения (*), чтобы перемножать два объекта класса **Polynomials**;
- е) перегрузить операцию сложения с присваиванием (+=), операцию вычитания с присваиванием (-=), операцию умножения с присваиванием (*=).

Вариант 3

Рассмотрите класс **Complex**. Класс позволяет работать с операциями над так называемыми *комплексными числами*. Они представляются в виде **realPart + imaginaryPart*i**, где **i** имеет значение корня квадратного от -1.

- а) измените этот класс так, чтобы иметь возможность вводить и выводить комплексные числа посредством перегруженных операций >> и << соответственно (вы должны заменить функцию печати класса);
- б) перегрузите операцию умножения так, чтобы иметь возможность перемножать два числа как в алгебре;
- в) перегрузите операции = и != так, чтобы иметь возможность сравнивать два числа.

Вариант 4

Создать класс **Matrix** (матрица целых чисел), используя класс **Vector** (вектор целых чисел) с переопределенными для него операциями. Определить для этого класса следующие операции:

- а) «>>» – ввод матрицы с консоли;

- б) «<<<» – вывод матрицы на экран или в файл;
- в) «+» – сложение матриц;
- г) «-» – вычитание матриц;
- д) «*» – умножение матриц;
- е) «[[[]]]» – доступ к элементу.

Вариант 5

Создайте класс **rationalNumber** (дроби) со следующими возможностями:

- а) создайте конструктор, который предотвращает равенство нулю знаменателя дроби, сокращает или упрощает дроби, если они не в сокращенной форме, и исключает отрицательные знаменатели;
- б) перегрузите операции сложения, вычитания, умножения и деления для этого класса;
- в) перегрузите операции отношения и проверки на равенство для этого класса.

Вариант 6

Создать класс **Matrix** (матрица целых чисел) и определить для него операции, указанные в варианте 1, не используя класс **Vector** (вектор целых чисел).

Вариант 7

Используя класс **Matrix** (матрица целых чисел) и определенные для него операции, создать класс **Sistem** (Система линейных уравнений) и определить для него операции:

- а) решение системы методом Гаусса;
- б) решение системы методом Крамера.

Вариант 8

Создать класс **Stroka C** (строка Си), определить для него операции:

- а) «+» – слияния строк;
- б) «= =» – сравнения строк на равенство;
- в) «=» – копирования строки;
- г) «- -» – удаления из строки подстроки;
- д) «/» – удаления из строки всех символов, содержащихся в строке-делителе;
- е) функцию *at (char*substr, char*dest, int i)*, возвращающую номер позиции *i*-того вхождения строки *substr* в строку *dest*.

Вариант 9

Создать класс **Function** (функция), определить для него операции:

- а) «>>>» – ввод функции в виде текстовой строки с консоли;
- б) «<<<» – вывод функции в виде текстовой строки на экран;
- в) «()» – вычисление значения функции в точке по дереву-формуле;
- г) «++» – получение дерева-функции по функции, заданной в виде текстовой строки;
- д) «- -» – получение представления функции в виде текстовой строки по ее дереву-формуле;
- е) функцию *dif* получения дерева-производной дерева-функции.

Вариант 10

Создать класс **Matrix** (матрица целых чисел), используя класс **Vector** (вектор целых чисел) с переопределенными для него операциями. Определить для этого класса следующие операции:

- а) функцию *det* вычисления определителя матрицы;

- б) функцию *ort* проверки ортогональности строк матрицы;
- в) функцию *transp* получения транспонированной матрицы;
- г) функцию *revers* получения обратной матрицы;
- д) функцию *self_value* получения собственных значений матрицы;
- е) функцию *self_vector* получения собственных векторов матрицы.

Вариант 11

Создать класс **Список целых чисел** и переопределить для него операции:

- а) «[]» – взятие элемента списка с заданным номером;
- б) «←» – удаление элемента с заданным значением;
- в) «=» – сравнение на равенство двух списков;
- г) «<», «>» – сравнение длин списков;
- д) «*» – слияние двух упорядоченных списков в один, не нарушая упорядоченности.

Вариант 12

Создать класс **Стек целых чисел**, производный от класса **Список целых чисел**, переопределить для него операции:

- а) «+=» – добавление к стеку элемента или другого списка;
- б) «←» – удаление верхнего элемента стека;
- в) «>>» – ввод элементов стека с консоли;
- г) «<<» – вывод элементов стека на экран или в файл.

Вариант 13

Переопределить операции, указанные в варианте 11, для класса **Очередь целых чисел**, производного от класса **Список целых чисел**.

Вариант 14

Создать класс вектор, содержащий ссылку на *unsigned char*, размерность вектора. Класс имеет конструктор с двумя параметрами, конструктор копирования и деструктор. Определить оператор +, -, *, - как дружественные функции, =, +=, -=, *=, [] – как функции-члены. Определить операторы =, +, -, *, +=, -=, *= с числом типа *unsigned char*, операторы ++ и --. Определить функцию печати. Перегрузить операторы вывода и ввода в поток.

Вариант 15

Создать класс вектор, содержащий ссылку на *long double*, размерность вектора. Класс имеет конструктор с одним параметром, конструктор копирования и деструктор. Определить оператор +, -, *, - как дружественные функции, =, +=, -=, *=, [] – как функции-члены. Определить операторы =, +, -, *, +=, -=, *= с целым числом операторы ++ и --. Определить операторы =, +, -, *, +=, -=, *= с числом типа *long double*, операторы ++ и --. Определить функцию печати. Перегрузить операторы вывода и ввода в поток.

Используя компоненты панели элементов, выполнить решение математических задач согласно индивидуальным вариантам. Расположение элементов на форме приложения должно удовлетворять требованиям эргономичности.

1. Написать программу для вычисления корней биквадратного уравнения $ax^4+bx^2+c=0$ с произвольными коэффициентами.
2. Написать программу для решения системы из двух линейных уравнений с двумя неизвестными.

3. Написать программу, которая приближенно вычисляет интеграл $\int_a^b (x^2 + 2x) \ln x dx$.

Исходными данными считаются а и b.

4. Написать программу, которая вычисляет высоту, на которую подлетит подброшенное вертикально вверх тело с известной начальной скоростью.
5. Написать программу для вычисления первой космической скорости для произвольной планеты. Исходными данными считаются радиус и масса планеты.
6. Написать программу, которая вычисляет определитель матрицы 3×3 .
7. Написать программу, которая находит дальность выстрела из орудия, если известны начальная скорость, масса снаряда и угол наклона к горизонту.
8. Написать программу, которая находит наибольший общий делитель двух целых чисел.
9. Написать программу, которая проверяет число, введенное пользователем на простоту.
10. Исходными данными являются десятичное целое число и основание системы счисления от 2 до 9. Написать программу перевода числа в указанную систему счисления.
11. Написать программу приближенного расчета косинуса по формуле $\cos(x) = \prod_{N=1}^{\infty} (1 - \frac{4x^2}{(2N-1)^2 \pi^2})$.
12. Написать программу приближенного расчета синуса по формуле $\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^N \frac{x^{2N+1}}{(2N+1)!} + \dots$.
13. Написать программу приближенного расчета экспоненты по формуле $e^x = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \dots + \frac{x^N}{N!} + \dots$.
14. Написать программу для расчета дальности и времени полета при горизонтальном бросании тела.
15. Написать программу, демонстрирующую зависимости между давлением, объемом и температурой в уравнении состояния идеального газа.

Разработать программу-приложение, создавая все необходимые элементы приложения (кроме основного окна) динамически. Необходимые для работы с объектами классов дополнительные элементы управления выбрать самостоятельно.

1. Построить класс для описания плоской геометрической фигуры: круг. Предусмотреть методы для создания объекта, перемещения на плоскости, изменения размеров.
2. Построить класс для описания плоской геометрической фигуры: прямоугольник со сторонами, параллельными осям координат. Предусмотреть методы для создания объекта, перемещения на плоскости, изменения размеров.
3. Построить класс для описания плоской геометрической фигуры: прямоугольник (стороны могут быть не параллельны осям координат). Предусмотреть методы для создания объекта, перемещения на плоскости, изменения размеров.
4. Построить класс для описания плоской геометрической фигуры: треугольник. Предусмотреть методы для создания объекта, перемещения на плоскости, изменения размеров.
5. Построить класс для описания плоской геометрической фигуры: ломаная линия. Для хранения узлов линии использовать массив. Предусмотреть методы для создания объекта, изменения узлов ломаной, перемещения на плоскости.
6. Построить класс для описания плоской геометрической фигуры: многоугольник. Для хранения вершин многоугольника использовать массив. Предусмотреть методы для создания объекта, изменения координат вершин, перемещения на плоскости.
7. Построить класс для описания объектов типа «Строка текста». Предусмотреть методы для создания объекта, изменений параметров текста, перемещения текста.
8. Построить описание класса, содержащего информацию в почтовом адресе организации. Предусмотреть возможность отдельного изменения составных частей адреса, создания и уничтожения объектов этого класса.
9. Описать класс «Химический элемент», включающий поля: название элемента, массовое и зарядовое числа, краткое описание, электронная формула элемента (например, для хлора: $1s^2-2s^2-2p^6-3s^2-3p^5$), валентность. Класс должен содержать следующие методы: добавление и удаление элемента, изменение описания элемента. В программе необходимо реализовать поиск по названию элемента, вывод полной информации обо всех имеющихся в списке элементах.
10. Описать класс «Библиотека», в полях которого хранится следующая информация о книгах: автор, название, издательство, год издания, количество страниц, количество экземпляров книги. Класс должен содержать следующие методы: добавление и удаление книги, изменение количества экземпляров книги. В программе необходимо реализовать поиск по автору или названию, вывод информации обо всех книгах библиотеки.
11. Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.
12. Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами.

13. Составить описание класса многочленов от одной переменной, задаваемых степенью многочлена и массивом коэффициентов. Предусмотреть методы для вычисления значения многочлена для заданного аргумента, операции сложения и вычитания многочленов с получением нового объекта-многочлена.
14. Составить описание класса, обеспечивающего представление матрицы произвольного размера с возможностью изменения числа строк и столбцов, вывода подматрицы любого размера и всей матрицы.
15. Написать класс для эффективной работы со строками, позволяющий форматировать и сравнивать строки, хранить в строках числовые значения и извлекать их.

Лабораторная работа №13. Разработка приложений прикладного характера.

Разработать программу-приложение на основе предложенных ниже вариантов заданий. Каждая программа должна содержать возможность настройки и автоматического сохранения ее параметров, а также окно «О программе».

1. Реализовать программу-тест, в которой будет предусмотрено любое количество вопросов и четыре варианта ответов. Причем можно выбрать несколько вариантов ответов, или все (но нельзя ничего не выбирать). До тех пор, пока не выбран ни один из вариантов ответов, переход к следующему вопросу невозможен. Данные о вопросах хранятся в файле. Программа должна в конце теста создавать файл с выбранными вариантами ответов.
2. Реализовать программу-тест, в которой будет предусмотрено любое количество вопросов и четыре варианта ответов. Причем можно выбрать только один вариант ответа. Можно пропускать текущий вопрос и переходить к следующему. Время теста ограничено – информация о том, на сколько рассчитан тест, вводится в файл, содержащий текст теста (если время заканчивается, то тест прекращается и считается, что на вопросы, до которых не дошел тестируемый, были даны неверные ответы). По окончании теста, если осталось время и есть пропущенные вопросы, предлагается ответить на них. Данные о вопросах теста хранятся в файле. Программа должна в конце теста создавать файл со следующей информацией: ФИО отвечающего, время теста, какие варианты ответов были выбраны и напротив каждого из них – правильные варианты ответов, общее количество вопросов, количество правильных ответов, оценка. В настройках программы предусмотреть возможность отключить тест на время, а также выбрать, какую информацию выводить в файл после окончания теста.
3. Для программы, указанной в варианте 1, создать отдельную программу проверки, которая будет сравнивать варианты ответов по заданному ключу, а затем выставит оценку и укажет, на какие вопросы были даны неверные ответы (какие именно варианты были выбраны, и какие варианты были правильными).
4. Написать программу-тест, в которой будет предусмотрено любое количество вопросов и различное количество вариантов ответов на каждый из них. Остальные условия для программы – такие же, как и в варианте 1 или варианте 2.
5. Составить программу-архиватор монохромных изображений. Программа должна предлагать несколько вариантов сжатия. Например, таких.

Вариант 1. Так как монохромное изображение можно представить в виде двумерного массива черных и белых точек, то очевидно, можно заменить черные точки единицами, а белые – нулями. Тогда,

группируя их по 7, можно, используя таблицу кодировки, представить эти группы символами и таким образом сжать изображение.

Вариант 2. Рассмотрим строку, содержащую белые и черные точки. Тогда можно поступить с ней следующим образом. Указать номер первой черной точки, затем через пробел - расстояние до первой белой точки, затем – расстояние до черной точки, и т.д., до конца строки. Например, строка, содержащая два черных отрезка длиной 12 и 17 единиц, находящихся на расстоянии 6 единиц друг от друга, если первый находится на расстоянии 13 единиц от левого края рисунка, будет иметь вид: 13 12 6 17. Таким образом, мы можем сжать изображение. Очевидно, что такой способ больше подходит при сжатии изображений, содержащих протяженные белые или черные области.

6. Разработать программу-кодировщик текстового документа (например, перекодирование текста в графику) и декодировщик к ней. Предусмотреть не менее двух шагов кодирования, т.е. получаем из текста некоторый код на первом шаге, и производим его дополнительное кодирование по какому-либо правилу. Реализовать программу в виде текстового редактора, позволяющего просмотреть содержимое документа до кодирования и после. Совместить в одной программе функции кодировщика и декодировщика.
7. Написать программу-словарь толкования смысла некоторых слов (терминов). Предусмотреть возможность поиска слова, вывода слов по алфавиту. Если в процессе поиска найдено несколько слов или выражений, удовлетворяющих условию запроса, то должна быть реализована возможность их поочередного просмотра. Реализовать возможность выбрать группу слов (терминов), которую можно также редактировать и просматривать, а затем сохранить описания всех выбранных слов в виде отдельного файла с каким-либо указанным именем.
8. Написать программу, содержащую коллекции ссылок на внешние ресурсы (файлы), содержащие информацию, например, по какому-либо разделу информатики, или имеющих отношение к ней – пользователь может добавить или удалить ссылку на такой ресурс, просмотреть ресурс с помощью определенной для него программы. Предусмотреть возможность тестирования базы данных такой программы на наличие записей, указывающих на несуществующий ресурс с выводом информации о том, какие ресурсы отсутствуют. Пользователь может разрешить удаление ссылок на такие ресурсы.
9. Разработать программу, позволяющую в одной и той же системе координат строить несколько графиков функций (вид графика и его параметры настраиваются) в заданных интервалах $[a_i, b_i]$ – свой для каждого графика функции.
10. Пусть имеется поле размером $N \times M$, состоящее из ячеек. Размеры поля задаются в настройках программы. Первоначально поле содержит ячейки только одного цвета A . Каждая из ячеек может также быть закрашена цветом B . Игра заключается в следующем. Если на некоторую ячейку наведена мышка и нажата левая ее кнопка, то, если ячейка имела цвет A , ее цвет становится B , и наоборот (если ячейка имела цвет B , то он заменяется на цвет A). По тому же правилу изменяются цвета ячеек, находящихся слева и справа от данной ячейки, а также находящиеся сверху и снизу от нее. Игра завершается, когда все ячейки поля перекрашиваются в цвет B . В настройках программы реализовать выбор размеров поля программы, а также установку цветов A и B . Рассмотреть случай, когда ячейка, на которую наведена мышка и нажата ее левая кнопка, не меняет своего цвета, а ячейки левее и правее нее, а также выше и ниже ее, изменяют свой цвет. Таким образом, в настройках

программы необходимо организовать выбор типа игры (первый, или второй – в зависимости от того, меняют свое состояние 5 ячеек, или 4). По возможности реализовать подсчет очков, набранных игроком, основываясь на подсчете количества шагов, совершенных им до завершения игры и предусмотреть сохранение результатов в таблице рекордов.

11. Разработать текстовый редактор на основе компонента TRichEdit. Предусмотреть возможность настройки параметров вводимого и редактируемого текста (название, размер, начертание, цвет шрифта). Добавить возможность форматирования текста (настройка отступов и выступов) по примеру текстового редактора Microsoft Word. В строке статуса предусмотреть вывод информации о количестве строк и слов в тексте, а также информации о текущем положении курсора. Программа должна содержать основное (верхнее) меню и панель инструментов.
12. Разработать игру, развивающую зрительную память. Пусть имеется поле размером $N \times M$, состоящее из квадратных ячеек (всего $N \times M$ ячеек). Размеры ячеек выбираются в процессе реализации программы. Каждая ячейка содержит определенный рисунок. В момент начала новой игры каждой ячейке случайно назначается некоторый рисунок, причем количество рисунков каждого типа задается в настройках программы. Затем игроку открывается все поле, содержащее распределенные по нему рисунки, которое после просмотра и запоминания можно скрыть, выбрав соответствующий пункт меню, или снова просмотреть. После того, как рисунки скрыты, игрок должен по памяти открыть вначале все рисунки какого-либо одного типа. Если рисунки данного типа открыты не все, и игрок выбирает ячейку с рисунком другого типа, то все ячейки, содержащие рисунки последнего не открытого типа, снова скрываются (ячейки, содержащие типы рисунков, открытых полностью, остаются). Если игрок на каком-либо шаге выбирает просмотр всего поля, то после этого все поле снова скрывается, включая уже открытые ячейки. Игра завершается, когда все ячейки открыты. По возможности реализовать подсчет очков, набранных игроком (например, количество ходов, за которое удалось открыть все ячейки) и их сохранение в таблице рекордов.

Примерный перечень дополнительных задач повышенной сложности

1. Заполнить одномерный массив целыми числами от 1 до N за один проход так, чтобы их положение выбиралось случайно. Числа не должны повторяться. Используя генератор случайных чисел, реализовать алгоритм, позволяющий исключить из рассмотрения уже заполненные в массиве позиции.
2. Заданы две строки, содержащие текст. Проверить, можно ли из букв второй строки составить текст, содержащийся в первой строке. Можно использовать не более того количества разных букв, которое имеется во второй строке.
3. В матрице A размером $N \times M$, не содержащей повторяющиеся элементы, найти элементы, наиболее близкие друг к другу по значению. Указать их положение (номер строки и столбца). Например, для матрицы

$$\begin{pmatrix} 1 & 23 & 17 & 3 \\ 15 & 16 & 12 & 6 \\ 56 & 5 & 4 & 43 \end{pmatrix}$$

- наиболее близкими друг к другу по значению являются пары элементов 3 и 4, 4 и 5, 5 и 6, 16 и 17 (т.к. элементы во всех этих парах отличаются по значению друг от друга на 1).
4. Во время аттестации вуза N студентам предлагалось написать M тестов по различным предметам. По результатам всех тестов для каждого из студентов был вычислен средний балл, причем, если студент не присутствовал более чем на 20% тестах, его результат аннулировался (присутствие или отсутствие каждого из студентов на тесте реализовать с использованием генератора случайных чисел). Средний балл каждого студента подсчитывается по следующей формуле:
$$a = \frac{1}{M} \cdot \sum_{i=1}^M S_i k_i$$
, где S_i – суммарный балл по одному из M тестов, k_i – некоторый коэффициент ($0,5 \leq k_i \leq 2$) пропорциональности. Информация о студенте включает: фамилию, имя, отчество, группу, количество посещенных тестов, количество баллов по каждому из тестов. На основе таких сведений вывести информацию о студенте с самым высоким и самым низким средним баллом, сведения о проценте студентов, попавших в данное исследование (те, кто писал не менее 80% тестов). Сформировать список студентов, не вошедших в данное исследование, т.е. пропустивших более 20% тестов.
 5. Написать программу, осуществляющую поиск выхода из лабиринта размером $N \times M$ (задается с клавиатуры). Стены лабиринта располагаются с использованием генератора случайных чисел и обозначаются символом "#". После того, как лабиринт построен, указываются координаты объекта в лабиринте (если в точке с указанными координатами стена – программа просит указать другие координаты) и выводится сам объект в виде символа "0". Затем программа проверяет, можно ли из указанной точки пройти к одному из выходов лабиринта (выходов может вообще не быть). Если такой путь есть, он отображается символами "+", иначе – выводится сообщение о невозможности выйти из лабиринта.
 6. Написать программу, позволяющую провести сжатие текстового файла, а также распаковку сжатого программой файла. Алгоритм сжатия предложить самостоятельно. Программа должна сообщать размеры файла до сжатия и после него, а также коэффициент сжатия.
 7. Построить модель идеального движения двух шаров в отсутствии силы тяжести в некотором замкнутом прямоугольном объеме. Удары шаров считать абсолютно упругими. Отражение от стен происходит по закону отражения. Продолжать процесс до столкновения. Начальные параметры – скорости шаров, их положение и направление движения (учесть, что для каждого из шаров они будут различны). Подсчитать количество соударений, которое испытает каждая стена.
 8. Дружественными числами называются такие два различных числа A и B , что A есть сумма натуральных делителей числа B (исключая само число B), а B есть сумма натуральных делителей числа A (исключая само число A). Например: 220 и 284 – первая пара различных дружественных чисел (делители числа 284: 1, 2, 4, 71, 142; делители числа 220: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110; тогда 220 есть сумма делителей числа 284, так как $220=1+2+4+71+142$, а число 284 есть сумма делителей числа 220, так как $284=1+2+4+5+10+11+20+22+44+55+110$). Найти все пары дружественных чисел в диапазоне от a до b , или сообщить, что в указанном диапазоне таких чисел нет.
 9. Дан одномерный массив, содержащий n целых положительных элементов, причем значение каждого из элементов этого массива не менее 1 и не более n . Найти элементы массива, которые встречаются в нем ровно по 1 разу. Предложить эффективный способ решения задачи.

10. Многие современные текстовые редакторы позволяют проверять правописание слов в набранном тексте. Требуется написать программу, осуществляющую проверку правильности написания некоторого слова на основе его сравнения с имеющимся словарем. Предполагается, что в проверяемом слове одна или несколько букв записаны с ошибкой, но никакая буква не пропущена.
11. В сказочной стране по поводу празднования совершеннолетия принцессы решено было устроить грандиозный бал, на который пригласили всех жителей страны. Тем не менее, желающие попасть на бал должны были соблюдать одно простое условие: на бал можно приходить лишь парами. Так как жителей в стране очень много, то на входе пары решено было нумеровать целыми числами, в том числе и отрицательными. Происходило это следующим образом. Пара совместным решением выбирала любое понравившееся ей целое число (по модулю не превышающее 10^9), и, если оно оказывалось не занятым другой парой, то каждый член этой пары получал выбранное число и наклеивал его на свою одежду так, чтобы его было видно всем. Уже после начала бала служба безопасности короля выяснила, что на бал попал один житель без пары. Так как было неизвестно, какой номер он мог получить, то решено было проверить наличие всех пар. Ситуация осложняется тем, что на балу пары уже разбились, потому жители, относящиеся к одной паре, могли находиться в разных местах огромного зала. Не прерывая торжество, служба безопасности решила вписывать номера встреченных на балу жителей в специально отведенный для этого журнал, а уже потом сравнить, какие из записанных чисел имеют пары, и, таким образом, выявить номер нарушителя. Чтобы не посчитать несколько раз одного и того же жителя, каждому уже учтенному жителю дополнительно наклеивалась на одежду красная звездочка. Помогите службе безопасности выяснить, кто из жителей проник на бал без пары.
12. Извлечь корень кубический из заданного вещественного числа с указанной точностью, используя для этого только 4 арифметические операции (сложение, вычитание, деление, умножение).
13. Описать класс «Связанный вектор» в трехмерном пространстве (связанный вектор в трехмерном пространстве определяется шестью числами – координатами точек начала и конца вектора). Класс должен содержать конструктор с параметрами по умолчанию, а также следующие методы: определение модуля вектора, вывод текущих координат вектора, изменение координат начала или конца вектора, сложение и вычитание векторов, скалярное и векторное произведение векторов.
14. Описать класс «Треугольник», который определяет эту геометрическую фигуру посредством задания длин трех его сторон. Класс должен содержать конструктор с параметрами по умолчанию, а также следующие методы: вывод информации о длинах сторон треугольника; изменение длин сторон; определение углов, высот и площади треугольника. Предусмотреть проверку существования треугольника с заданными сторонами (при задании в конструкторе и при изменении сторон посредством соответствующего метода).
15. Составить описание класса для вывода текста в рамке в заданной точке экрана. Класс должен содержать конструктор, а также следующие методы: вывод текста в указанную точку экрана; установка цвета текста, рамки и фона; изменение текста; изменение расположения текста. Для изменения расположения текста оформить также дружественную функцию (при этом поля-координаты текста должны быть описаны со спецификатором доступа *private*).
16. Описать класс «Химический элемент», включающий поля: название элемента, массовое и зарядовое числа, краткое описание, электронная формула элемента (например, для хлора: $1s^2-2s^2-2p^6-3s^2-3p^5$), валентность. Класс должен содержать

конструкторы и деструктор, а также следующие методы: добавление и удаление элемента, изменение описания элемента, поиск по названию элемента, вывод полной информации обо всех элементах.

3. Методические материалы, определяющие процедуры оценивания результатов обучения по дисциплине (модулю), описание шкал оценивания

Рейтинг-план 1 семестра

Виды учебной деятельности студентов	Балл за конкретное задание	Число заданий	Баллы	
			Минимальный	Максимальный
Модуль 1				
Текущий контроль			0	20
1) Аудиторная работа	1	2	0	2
2) Выполнение домашнего задания	2	3	0	6
3) Лабораторные работы	4	3	0	12
Рубежный контроль			0	15
Контрольная работа №1	3	3	0	9
Домашняя контрольная работа №1	3	2	0	6
Модуль 2				
Текущий контроль			0	20
1) Аудиторная работа	1	4	0	4
2) Выполнение домашнего задания	2	2	0	4
3) Лабораторные работы	6	2	0	12
Рубежный контроль			0	15
Контрольная работа №2	5	1	0	5
Тест №1	10	1	0	10
		Итого:	0	70
Поощрительные баллы			0	10
Участие в олимпиаде			0	6
Решение задач	2	2	0	4

повышенной сложности				
Посещаемость (баллы вычитаются из общей суммы набранных баллов)				
1. Посещение лекционных занятий			0	-6
2. Посещение практических и лабораторных занятий			0	-10
Итоговый контроль				
Экзамен			0	30
ИТОГО:			0	110

Рейтинг-план 2 семестра

Виды учебной деятельности студентов	Балл за конкретное задание	Число заданий	Баллы	
			Минимальный	Максимальный
Модуль 3				
Текущий контроль			0	20
1) Аудиторная работа	1	2	0	2
2) Выполнение домашнего задания	2	3	0	6
3) Лабораторные работы	4	3	0	12
Рубежный контроль			0	15
Контрольная работа №3	7	1	0	7
Домашняя контрольная работа №2	8	1	0	8
Модуль 4				
Текущий контроль			0	20
1) Аудиторная работа	1	4	0	4
2) Выполнение домашнего задания	2	2	0	4
3) Лабораторные работы	6	2	0	12
Рубежный контроль			0	15
Контрольная работа №4	8	1	0	8
Тест №2	7	1	0	7
		Итого:	0	70
Поощрительные баллы			0	10
Участие в олимпиаде			0	6
Решение задач повышенной сложности	2	2	0	4
Посещаемость (баллы вычитаются из общей суммы набранных баллов)				
1. Посещение лекционных занятий			0	-6
2. Посещение практических и лабораторных занятий			0	-10
Итоговый контроль				

Экзамен			0	30
ИТОГО:			0	110

Рейтинг-план 3 семестра

Виды учебной деятельности студентов	Балл за конкретное задание	Число заданий	Баллы	
			Минимальный	Максимальный
Модуль 5				
Текущий контроль			0	25
1) Аудиторная работа	1	4	0	4
2) Выполнение домашнего задания	2	3	0	6
3) Лабораторные работы	5	3	0	15
Рубежный контроль			0	25
Контрольная работа №5	7/8	2	0	25
Модуль 6				
Текущий контроль			0	25
1) Аудиторная работа	1	5	0	5
2) Выполнение домашнего задания	2	2	0	4
3) Лабораторные работы	8	2	0	16
Рубежный контроль			0	25
Домашняя контрольная работа №3	15	1	0	25
		Итого:	0	100
Поощрительные баллы			0	10
Участие в олимпиаде			0	6
Решение задач повышенной сложности	2	2	0	4
Посещаемость (баллы вычитаются из общей суммы набранных баллов)				
1. Посещение лекционных занятий			0	-6
2. Посещение практических и лабораторных занятий			0	-10
Итоговый контроль				
Экзамен			0	30
ИТОГО:			0	110

Рейтинг-план 4 семестра

Виды учебной деятельности студентов	Балл за конкретное задание	Число заданий	Баллы	
			Минимальный	Максимальный
Модуль 5				
Текущий контроль			0	20
1) Аудиторная работа	1	2	0	2
2) Выполнение домашнего задания	2	3	0	6
3) Лабораторные работы	4	3	0	12
Рубежный контроль			0	15
Контрольная работа №5	7/8	2	0	15
Модуль 6				
Текущий контроль			0	20
1) Аудиторная работа	1	4	0	4
2) Выполнение домашнего задания	2	2	0	4
3) Лабораторные работы	6	2	0	12
Рубежный контроль			0	15
Домашняя контрольная работа №3	15	1	0	15
		Итого:	0	70
Поощрительные баллы			0	10
Участие в олимпиаде			0	6
Решение задач повышенной сложности	2	2	0	4
Посещаемость (баллы вычитаются из общей суммы набранных баллов)				
1. Посещение лекционных занятий			0	-6
2. Посещение практических и лабораторных занятий			0	-10
Итоговый контроль				
Экзамен			0	30
ИТОГО:			0	110

Объем и уровень сформированности компетенций целиком или на различных этапах у обучающихся оцениваются по результатам текущего контроля количественной оценкой, выраженной в рейтинговых баллах. Оценке подлежит каждое контрольное мероприятие.

При оценивании сформированности компетенций применяется четырехуровневая шкала «неудовлетворительно», «удовлетворительно», «хорошо», «отлично».

Максимальный балл по каждому виду оценочного средства определяется в рейтинг-плане и выражает полное (100%) освоение компетенции.

Уровень сформированности компетенции «хорошо» устанавливается в случае, когда объем выполненных заданий соответствующего оценочного средства составляет 80 - 100%; «удовлетворительно» – выполнено 40 - 80%; «неудовлетворительно» – выполнено 0 - 40%

Рейтинговый балл за выполнение части или полного объема заданий соответствующего оценочного средства выставляется по формуле:

$$\text{Рейтинговый балл} = k \times \text{Максимальный балл}$$

где $k = 0,2$ при уровне освоения «неудовлетворительно», $k = 0,6$ при уровне освоения «удовлетворительно», $k = 0,8$ при уровне освоения «хорошо» и $k = 1$ при уровне освоения «отлично».

Оценка на этапе промежуточной аттестации выставляется согласно Положению о модульно-рейтинговой системе обучения и оценки успеваемости студентов БашГУ:

На экзамене выставляется оценка:

- отлично – при накоплении от 80 до 110 рейтинговых баллов (включая 10 поощрительных баллов),
- хорошо – при накоплении от 60 до 79 рейтинговых баллов,
- удовлетворительно – при накоплении от 45 до 59 рейтинговых баллов,
- неудовлетворительно – при накоплении менее 45 рейтинговых баллов.

Результаты обучения по дисциплине (модулю) у обучающихся оцениваются по итогам текущего контроля количественной оценкой, выраженной в рейтинговых баллах. Оценке подлежит каждое контрольное мероприятие.

При оценивании сформированности компетенций применяется четырехуровневая шкала «неудовлетворительно», «удовлетворительно», «хорошо», «отлично».

Максимальный балл по каждому виду оценочного средства определяется в рейтинг-плане и выражает полное (100%) освоение компетенции.

Уровень сформированности компетенции «хорошо» устанавливается в случае, когда объем выполненных заданий соответствующего оценочного средства составляет 80-100%; «удовлетворительно» – выполнено 40-80%; «неудовлетворительно» – выполнено 0-40%

Рейтинговый балл за выполнение части или полного объема заданий соответствующего оценочного средства выставляется по формуле:

$$\text{Рейтинговый балл} = k \times \text{Максимальный балл},$$

где $k = 0,2$ при уровне освоения «неудовлетворительно», $k = 0,4$ при уровне освоения «удовлетворительно», $k = 0,8$ при уровне освоения «хорошо» и $k = 1$ при уровне освоения «отлично».

Оценка на этапе промежуточной аттестации выставляется согласно Положению о модульно-рейтинговой системе обучения и оценки успеваемости студентов УУНиТ:

На экзамене и дифференцированном зачете выставляется оценка:

- отлично - при накоплении от 80 до 110 рейтинговых баллов (включая 10 поощрительных

баллов),

- хорошо - при накоплении от 60 до 79 рейтинговых баллов,
- удовлетворительно - при накоплении от 45 до 59 рейтинговых баллов,
- неудовлетворительно - при накоплении менее 45 рейтинговых баллов.

При получении на экзамене оценок «отлично», «хорошо», «удовлетворительно», на зачёте оценки «зачтено» считается, что результаты обучения по дисциплине (модулю) достигнуты и компетенции на этапе изучения дисциплины (модуля) сформированы.