

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Сыров Игорь Анатольевич
Должность: Директор
Дата подписания: 25.11.2022 11:14:34
Уникальный программный ключ:
b683afe664d7e9f64175886cf9626a198149ad36

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.17.01

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

Специальность

21.05.05 **Физические процессы горного или нефтегазового производства**
код наименование специальности

Программа

специализация N 2 "Физические процессы нефтегазового производства"

Форма обучения

Заочная

Для поступивших на обучение в
2022 г.

Разработчик (составитель)
д.ф.-м.н., профессор
Хусаинов И. Г.
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	34

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-20. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-20.1. Выстраивает профессиональную деятельность опираясь на основы информационных технологий и программные продукты.	Обучающийся должен: пользоваться знаниями основных технологий и программных продуктов	Не знает способы выстраивания профессиональной деятельности, опираясь на основы информационных технологий и программные продукты.	Плохо знает способы выстраивания профессиональной деятельности, опираясь на основы информационных технологий и программные продукты.	Частично знает способы выстраивания профессиональной деятельности, опираясь на основы информационных технологий и программные продукты.	Хорошо знает способы выстраивания профессиональной деятельности опираясь на основы информационных технологий и программные продукты.	Тест
	ОПК-20.2. Использует современные информационные технологии и программные продукты для решения задач профессиональной деятельности	Обучающийся должен: использовать современные информационные технологии и программные продукты для решения задач профессиональной деятельности	Не умеет использовать современные информационные технологии и программные продукты для решения задач профессиональной деятельности	Плохо умеет использовать современные информационные технологии и программные продукты для решения задач профессиональной деятельности	Частично умеет использовать современные информационные технологии и программные продукты для решения задач профессиональной деятельности	Хорошо умеет использовать современные информационные технологии и программные продукты для решения задач профессиональной деятельности	Тест

	деятельности.	ой деятельности.	деятельности.	деятельности.	деятельности.	деятельности.	
	ОПК-20.3. Применяет методы информационных технологий для решения задач профессиональной деятельности с использованием полученных знаний в области фундаментальных и прикладных наук.	Обучающийся должен: работать методами для применения информационных технологий в решениях задач профессиональной деятельности с использованием полученных знаний в области фундаментальных и прикладных наук.	Не владеет методами применения информационных технологий для решения задач профессиональной деятельности с использованием полученных знаний в области фундаментальных и прикладных наук.	Плохо владеет методами применения информационных технологий для решения задач профессиональной деятельности с использованием полученных знаний в области фундаментальных и прикладных наук.	Частично владеет методами применения информационных технологий для решения задач профессиональной деятельности с использованием полученных знаний в области фундаментальных и прикладных наук.	Хорошо владеет методами применения информационных технологий для решения задач профессиональной деятельности с использованием полученных знаний в области фундаментальных и прикладных наук.	Тест
ОПК-19. Способен участвовать в разработке и реализации образовательных программ в сфере своей профессиональной деятельности, используя	ОПК-19.1. Использует программные продукты общего и специального назначения для моделирования месторождений и технологий в сфере своей профессиональной	Обучающийся должен: пользоваться знаниями программных продуктов общего и специального назначения для моделирования месторождений и технологий в сфере своей	Не знает программные продукты общего и специального назначения для моделирования месторождений и технологий в сфере своей профессиональной	Плохо знает программные продукты общего и специального назначения для моделирования месторождений и технологий в сфере своей профессиональной	Частично знает программные продукты общего и специального назначения для моделирования месторождений и технологий в сфере своей профессиональной	Хорошо знает программные продукты общего и специального назначения для моделирования месторождений и технологий в сфере своей профессиональной	Тест

специальные научные знания	ой деятельности, используя специальные научные знания.	профессиональной деятельности, используя специальные научные знания	деятельности, используя специальные научные знания.	деятельности, используя специальные научные знания.	деятельности, используя специальные научные знания.	деятельности, используя специальные научные знания.	
	ОПК-19.2. Применяет теоретические и методологические основы работы с программными продуктами в сфере своей профессиональной деятельности, используя специальные научные знания.	Обучающийся должен: применять теоретические и методологические основы работы с программными продуктами в сфере своей профессиональной деятельности, используя специальные научные знания.	Не умеет применять теоретические и методологические основы работы с программными продуктами в сфере своей профессиональной деятельности, используя специальные научные знания.	Плохо умеет применять теоретические и методологические основы работы с программными продуктами в сфере своей профессиональной деятельности, используя специальные научные знания.	Частично умеет применять теоретические и методологические основы работы с программными продуктами в сфере своей профессиональной деятельности, используя специальные научные знания.	Хорошо умеет применять теоретические и методологические основы работы с программными продуктами в сфере своей профессиональной деятельности, используя специальные научные знания.	Тест
	ОПК-19.3. Разрабатывает и реализует образовательные программы в сфере своей профессиональной деятельности, используя специальные научные знания.	Обучающийся должен: работать знаниями и умениями для демонстрации базовых знаний и способностью работать в разработке и реализации образовательных	Не владеет методами реализации образовательных программ в сфере своей профессиональной деятельности, используя специальные научные знания.	Плохо владеет методами реализации образовательных программ в сфере своей профессиональной деятельности, используя специальные научные знания.	Частично владеет методами реализации образовательных программ в сфере своей профессиональной деятельности, используя специальные	Хорошо владеет методами реализации образовательных программ в сфере своей профессиональной деятельности, используя специальные научные знания.	Тест

		программ в сфере своей профессиональной деятельности, используя специальные научные знания			научные знания.		
ПК-5. Способен разрабатывать текущие и перспективные программы по оценке ресурсов, подсчету и пересчету запасов	ПК-5.1. Применяет технологии проведения, обработки и интерпретации геолого-геофизических работ в профессиональной деятельности.	Обучающийся должен: работать методами разработки перспективных программ по оценке ресурсов, подсчету и пересчету запасов	Не знает технологии проведения, обработки и интерпретации геолого-геофизических работ в профессиональной деятельности.	Плохо знает технологии проведения, обработки и интерпретации геолого-геофизических работ в профессиональной деятельности.	Частично знает технологии проведения, обработки и интерпретации геолого-геофизических работ в профессиональной деятельности.	Хорошо знает технологии проведения, обработки и интерпретации геолого-геофизических работ в профессиональной деятельности.	Тест
	ПК-5.2. Подготавливает материалы, используемые при разработке программ геологоразведочных работ по подсчету запасов и управлению запасами.	Обучающийся должен: разрабатывать перспективные программы по оценке ресурсов, подсчету и пересчету запасов	Не умеет подготавливать материалы, используемые при разработке программ геологоразведочных работ по подсчету запасов и управлению запасами.	Плохо умеет подготавливать материалы, используемые при разработке программ геологоразведочных работ по подсчету запасов и управлению запасами.	Частично умеет подготавливать материалы, используемые при разработке программ геологоразведочных работ по подсчету запасов и управлению запасами.	Хорошо умеет подготавливать материалы, используемые при разработке программ геологоразведочных работ по подсчету запасов и управлению запасами.	Тест
	ПК-5.3.	Обучающийся	Вообще не	Плохо владеет	Частично	Хорошо владеет	Тест

	<p>Осуществляет разработки перспективных программ геологоразведочных работ с целью уточнения запасов углеводородов на территории деятельности организации.</p>	<p>должен: иметь понятие о способах разработки перспективных программ по оценке ресурсов, подсчету и пересчету запасов</p>	<p>владеет методами разработки перспективных программ геологоразведочных работ с целью уточнения запасов углеводородов на территории деятельности организации.</p>	<p>методами разработки перспективных программ геологоразведочных работ с целью уточнения запасов углеводородов на территории деятельности организации.</p>	<p>владеет методами разработки перспективных программ геологоразведочных работ с целью уточнения запасов углеводородов на территории деятельности организации.</p>	<p>методами разработки перспективных программ геологоразведочных работ с целью уточнения запасов углеводородов на территории деятельности организации.</p>	
--	--	--	--	--	--	--	--

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

Тестовые задания №1

Перечень вопросов для оценки уровня сформированности компетенции ОПК-20 по индикатору ОПК-20.1:

1. В каком случае константа описана правильно?
А) const n:int=78;
Б) const n=78;
В) const n:int(78);
Г) const n=78:int.
2. Если x – вещественная переменная, а p и n – целые, то какие из следующих операторов правильные?
А) $n=p \% x$;
Б) $p=34 \% p$;
В) $p=n \% 2.5$;
Г) $n=x \% 2$.
3. Чему равно значение выражения $20 \% 6$?
А) 2
Б) 3
В) 4
Г) 6
4. Что выдаст программа

```
#include <iostream.h>
#include <conio.h>
main(){
int p=30;
p=p%7;
cout<<p;
getch();}

```

после запуска?
А) 5
Б) 4
В) 2
Г) сообщение об ошибке
5. Что выдаст программа

```
#include <iostream.h>
#include <conio.h>
main(){
int p=30;
p=p/7;
cout<<p;
getch();}

```

после запуска?
А) 5

Б) 2

В) 4

Г) сообщение об ошибке

6. Типы данных на C++.

Ответ: Тип определяет, какие значения может иметь переменная, какие операции с ней можно производить и сколько байт в памяти она будет занимать.

7. Стандартный тип int.

Ответ: Стандартный тип int определяет целочисленные значения.

8. Стандартный тип double.

Ответ: Стандартный тип double используется для представления чисел с плавающей запятой двойной точности.

9. Стандартный тип void.

Ответ: Стандартный тип void используется, когда результат работы функции не требуется или не имеет значения.

10. Стандартный тип char.

Ответ: Стандартный тип char представляет собой числовое значение ASCII-кода символа.

11. Стандартный тип bool.

Ответ: Стандартный тип bool в языке программирования обычно используется для представления логических значений true (истина) и false (ложь).

12. Стандартный тип float.

Ответ: Тип данных float представляет собой числа с плавающей точкой и используется для представления десятичных чисел.

13. Модификаторы типа signed, unsigned.

Ответ: Модификаторы типа signed и unsigned используются для указания диапазона значений, которые может принимать переменная определенного типа данных.

14. Модификаторы типа short, long.

Ответ: Модификаторы типа short и long используются для изменения размера переменной определенного типа данных.

15. Модификатор const.

Ответ: Модификатор const используется для объявления переменной, значение которой не может быть изменено после ее инициализации.

16. Структура программы на C++.

Ответ: Структура программы на C++: 1) подключение заголовочных файлов; 2) объявление глобальных переменных; 3) определение функций; 4) Основная функция main.

17. Комментарии на C++.

Ответ: Комментарии на C++ используются для добавления пояснений и описания кода.

18. Вывод данных на экран.

Ответ: Вывод данных на экран в программировании осуществляется с помощью команды или функции, которая позволяет вывести определенное значение или текст на экран.

19. Ввод данных с клавиатуры.

Ответ: Ввод данных с клавиатуры в программировании осуществляется с помощью команды или функции, которая позволяет пользователю ввести определенное значение с клавиатуры.

20. Идентификаторы.

Ответ: Идентификатор-это имя, которое идентифицирует уникальный объект.

Перечень вопросов для оценки уровня сформированности компетенции ОПК-20 по индикатору ОПК-20.2:

1. Что будет напечатано программой?

```
int s=0, i=1;
while (i<5) {i=i+1;s=s+i;}
cout<<s;
```

A) 15

Б) 5

В) 10

Г) 14

2. Почему при исполнении следующей программы появляется сообщение об ошибке?

```
#include <iostream.h>
#include <conio.h>
void main(){
int p=2,k,d;
cin>>k;
d=min(p,k);
cout<<d;
getch();
}
```

A) отсутствует return для функции main;

Б) не инициализирована переменная d;

В) нет описания функции min;

Г) неверно описаны переменные;

3. Какие операторы используются для программирования циклов с предусловием в C++?

A) if then

Б) switch

В) do while

Г) while, for

4. Какие операторы используются для программирования циклов с постусловием в C++?

A) break

Б) switch

В) do while

Г) while, for

5. Какой из перечисленных операторов является оператором множественного выбора?
А) if
Б) switch
В) continue
Г) break
6. Описание переменных.
Ответ: Переменная (программирование) – поименованная, либо адресуемая иным способом область памяти, адрес которой можно использовать для осуществления доступа к данным и изменять значение в ходе выполнения программы.
7. Инициализация переменных.
Ответ: Инициализация переменных - это процесс присвоения начального значения переменной при ее объявлении.
8. Унарные операции.
Ответ: Унарные операции представляют собой действия, выполняемые на одном значении, и могут выполняться над различными типами данных, такими как числа, строки и булевы значения.
9. Бинарные операции.
Ответ: Бинарные операции в C++ - это операции, которые принимают два операнда и возвращают результат.
10. Составное присваивание.
Ответ: Операция составного присваивания состоит из простой операции присваивания, скомбинированной с какой-либо другой бинарной операцией.
11. Логические операции.
Ответ: Логическое И: &&; ИЛИ: || ; НЕ: !
12. Приоритеты операций и порядок вычислений операций.
Ответ: В C++ приоритеты операций: операции в скобках; унарные операции; арифметические; логические; присваивания; сравнения; операция НЕ; операции И, ИЛИ.
13. Преобразования типов при вычислении выражений.
Ответ: При выполнении операций производится автоматическое преобразование типов, чтобы привести операнды выражений к общему типу.
14. Оператор ветвления if.
Ответ: Оператор условного ветвления if позволяют принять программе решение, в зависимости от результата проверки условия.
15. Вложенное ветвление.
Ответ: Вложенное ветвление – это ветвление, которое находится внутри другого ветвления.
16. Оператор выбора switch.

Ответ: Оператор выбора switch в C++ позволяет выполнить различные действия в зависимости от значения выражения.

17. Оператор цикла while.

Ответ: Оператор цикла while – цикл с предусловием.

18. Вложенные циклы с помощью while.

Ответ: Вложенные циклы с помощью while в C++ позволяют выполнить повторяющиеся действия внутри другого цикла.

19. Оператор цикла for.

Ответ: Оператор for позволяет повторить выполнение того или иного оператора или составного оператора заданное число раз.

20. Вложенные циклы for.

Ответ: Вложенные циклы с помощью for в C++ также позволяют выполнить повторяющиеся действия внутри другого цикла.

Перечень вопросов для оценки уровня сформированности компетенции ОПК-20 по индикатору ОПК-20.3:

1. Какой из перечисленных операторов является простым оператором условия?

A) if

Б) switch

В) goto

Г) break

2. Какой из перечисленных операторов является оператором немедленного перехода?

А) if

Б) switch

В) goto

Г) break

3. Какой из перечисленных операторов является оператором немедленной передачи управления в начало следующей итерации цикла?

А) if

Б) switch

В) continue

Г) break

4. Какой из перечисленных операторов является оператором выхода из цикла?

А) if

Б) switch

В) continue

Г) break

5. Какой из перечисленных типов данных является пустым?

А) char

Б) int

В) void

Г) double

6. Оператор цикла `dowhile`.
Ответ: Оператор цикла `dowhile` в C++ выполняет определенные действия, пока указанное условие истинно.
7. Вложенные циклы `do while`.
Ответ: Вложенные циклы с помощью `dowhile` в C++ позволяют выполнять повторяющиеся действия внутри другого цикла.
8. Оператор `break`.
Ответ: `break` — это оператор, предназначенный для досрочного выхода из цикла.
9. Оператор `continue`.
Ответ: Оператор `continue` осуществляет принудительный переход к следующему шагу цикла, пропуская любой код, оставшийся невыполненным.
10. Массивы.
Ответ: Массив – это структура данных, состоящая из набора элементов (значений или переменных), каждый из которых идентифицируется по крайней мере одним индексом или ключом массива.
11. Одномерные массивы.
Ответ: Одномерный массив - это структура данных, которая представляет собой последовательность элементов одного типа, расположенных последовательно в памяти.
12. Многомерные массивы.
Ответ: Многомерный массив – это структура данных, которая позволяет хранить элементы в виде таблицы с несколькими измерениями.
13. Функции для работы с массивами.
Ответ: В C++ для работы с массивами доступны следующие функции: `sizeof()`; `copy()`; `sort()`; `reverse()` и др.
14. Строки на C++.
Ответ: Строка — последовательность символов.
15. Описание строковой переменной.
Ответ: Строковая переменная в программировании является типом данных, который используется для хранения текстовой информации.
16. Использование символьного массива при работе со строками.
Ответ: При работе со строками можно использовать символьный массив для хранения и обработки символов строки.
17. Операции со строками.
Ответ: Операции со строками включают различные действия, которые можно выполнять над строковыми данными.
18. Основные строковые функции.
Ответ: Основные строковые функции: сцепления ; функция получения длины строки; получения подстроки; поиска подстроки.

19. Функция пользователя.

Ответ: Функция пользователя – это механизм, позволяющий программисту создавать функции, которые могут быть вызваны во время выполнения программы.

20. Описание и определение функций пользователя.

Ответ: Функции пользователя описываются перед функцией main().

Тестовые задания №2

Перечень вопросов для оценки уровня сформированности компетенции ОПК-19 по индикатору ОПК-19.1:

1. Выберите вариант ответа, который соответствует циклу с предусловием

- a. if () / else
- b. do/while ()
- c. **while ()**
- d. for(; ;)

2. Операция присвоения в C++:

- a. !=
- b. **=**
- c. :=
- d. ==

3. Часть кода программы означает, что:

```
for (int i=0; i<n; i++) {...}
```

- a. цикл будет выполнять неограниченное число итераций
- b. **цикл будет выполнять n итераций**
- c. цикл не выполнит ни одной итерации
- d. цикл описан некорректно

4. Где правильно описана переменная

- a. **int x;**
- b. INT x;
- c. Int x;
- d. INT X

5. Оператор cin>> используется

- a. для вывода данных
- b. **для ввода данных**
- c. Для описания переменных

6. Передача параметров функции по значению.

Ответ: Передача параметра по значению означает, что вызывающая функция копирует в память, доступную вызываемой, (обычно стек) непосредственное значение.

7. Передача параметров функции по ссылке.

Ответ: Передача параметров функции по ссылке означает, что при вызове функции вместо создания копии значения параметра передается сама ссылка на объект или переменную.

8. Передача массивов функции в качестве параметров.

Ответ: Передача массивов в качестве параметров функции позволяет функции работать с элементами массива и изменять их значения.

9. Рекурсивные функции.

Ответ: Рекурсивная функция — это функция, которая вызывает сама себя.

10. Доступ к элементам массива.

Ответ: Чтобы получить доступ к элементу массива, необходимо указать индекс этого элемента в квадратных скобках после имени массива.

11. Тип данных указатель.

Ответ: Указатель — переменная, содержащая адрес объекта.

12. Описание типизированных указателей.

Ответ: Типизированные указатели - это указатели на определенный тип данных в языках программирования, которые обеспечивают компилятору информацию о типе данных, на который они указывают.

13. Описание не типизированных указателей.

Ответ: Не типизированные указатели – это указатели, которые могут указывать на объекты любого типа данных.

14. Оператор выбора.

Ответ: Оператор выбора (переключатель, селектор) предназначен для выбора одного варианта из множества возможных вариантов разветвления программы.

15. Примеры использования оператора выбора. Досрочный выход из оператора выбора.

Ответ: Оператор выбора, такой как условный оператор "if" или оператор "switch", позволяет программе принимать решения на основе значения условия

16. Примеры использования цикла.

Ответ: Цикл "for" для выполнения определенного числа итераций:
`for(i=1; i<5; i++){cout<<i<<"\n";}` Этот цикл будет выводить числа от 1 до 5.

17. Понятие заикливания.

Ответ: Заикливание - ситуация, при которой выполнение цикла никогда не заканчивается.

18. Операции допустимые над целыми числами (в том числе остаток от деления).

Ответ: Операции, которые допустимы над целыми числами, включают: +, -, *, /, %.

19. Операции инкремента и декремента (++ , --). Отличие постинкремента (декремента) от преинкремента (декремента).

Ответ: Операции инкремента (++) и декремента (--) являются унарными операторами, которые позволяют увеличивать или уменьшать значение переменной на 1 соответственно.

20. Получение целой части от деления.

Ответ: Чтобы определить целую часть от деления, необходимо выполнить следующую операцию: / разделить одно целое число на другое.

Перечень вопросов для оценки уровня сформированности компетенции ОПК-19 по индикатору ОПК-19.2:

1. Основные целочисленные стандартные типы в C++

A) void

Б) char

В) bool, char, int, wchar_t

Г) double, float

2. Основные вещественные стандартные типы в C++

A) void

Б) char

В) bool, char, int, wchar_t

Г) double, float

3. Какое из перечисленных зарезервированных слов-модификаторов типов является обозначением знакового типа?

A) short

Б) long

В) signed

Г) unsigned

4. Управляющий символ \n осуществляет:

А) новая строка, перевод строки

Б) знак вопроса

В) новая страница, перевод страницы

Г) пробел

5. Какой из ниже перечисленных идентификаторов в языке C++ не употребляются как служебное слово?

A) else

Б) char

В) break

Г) begin

6. Получение вещественного результата деления целого числа на целое.

Ответ: Для получения вещественного результата деления целого числа на целое, можно выполнить явное преобразование одного из операндов в вещественное число.

7. Функция main – отличие от других функций.

Ответ: Функция main ничем не отличается от других функций за исключением того, что возвращаемое значение этой функции используется операционной системой как код возврата.

8. Инструкция return.

Ответ: Инструкция `return` прекращает исполнение функции, даже если в теле функции остались другие инструкции.

9. Инструкция вызова функции.

Ответ: Инструкция вызова функции - это способ вызвать функцию в программе, передавая аргументы.

10. Понятие модуля и заголовочного файла.

Ответ: Модуль представляет собой независимую и самодостаточную единицу программного кода, которая выполняет какую-либо конкретную функцию или предоставляет определенные сервисы.

11. Инициализация массивов.

Ответ: Инициализация — это процесс создания массива и заполнения его значениями.

12. Методы сортировки данных: метод отбора.

Ответ: Метод отбора является простым алгоритмом сортировки, который состоит в выборе наименьшего (или наибольшего) элемента из списка и его перемещении в начало (или конец) списка.

13. Методы сортировки данных: метод «пузыря».

Ответ: Метод «пузыря» является одним из наиболее простых алгоритмов сортировки. Он работает по принципу сравнения и обмена соседних элементов с целью постепенно сдвигать большие элементы в конец списка.

14. Динамические массивы.

Ответ: Динамические массивы данных — это структура, которая позволяет прямо в процессе написания кода автоматически менять размер массива и освобождать ненужные ячейки.

15. Динамические переменные.

Ответ: Динамическая переменная — переменная в программе, место в оперативной памяти под которую выделяется во время выполнения программы.

16. Ссылки.

Ответ: Ссылки в программировании - это адреса, которые указывают на определенные объекты или данные в памяти компьютера.

17. Структуры.

Ответ: Структуры — это встроенная структура данных, объединяющая данные разных типов под одним именем.

18. Динамические структуры данных. Односвязный список.

Ответ: Динамические структуры данных - это структуры данных, которые могут изменять свой размер и форму во время выполнения программы.

19. Операции со строками.

Ответ: Операции со строками обычно включают набор действий, которые можно выполнить над строками, такие как создание, объединение (конкатенация), сравнение, извлечение подстроки, поиск и замена символов и многое другое.

20. Основные строковые функции.

Ответ: Строковые функции: - len(), str(), - upper(), lower().

Перечень вопросов для оценки уровня сформированности компетенции ОПК-19 по индикатору ОПК-19.3:

1. Какой из перечисленных ниже циклов является бесконечным?

A) for (i=1;i>23;i++) cout<<"цикл ";

Б) for (i=0;i>=0;i++) cout<<"цикл ";

B) for (i=10;i>6;i++) cout<<"цикл ";

Г) for (i=5;i>15;i++) cout<<"цикл ".

2. Поразрядные операции можно использовать с переменными типа:

A) float;

B) double;

B) long double;

Г) short int.

3. Укажите порядок присваивания переменным числового значения в выражении вида:

a=b=c=10;

A) a,b,c;

Б) c,b,a;

B) b,c,a;

Г) c,a,b.

4. По Вашему мнению, с точки зрения C++ верен ли приведенный код:

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
double A;
```

```
void main ()
```

```
{ int A;
```

```
A=5;
```

```
::A = 2.5;
```

```
}
```

A) код содержит синтаксическую ошибку;

Б) код верный;

B) это код не на C/C++;

Г) код неверный.

5. Какие операции можно выполнять над массивами в целом, если их типы идентичны?

A) Сравнение;

Б) Присваивание;

B) Сложение;

Г) Сравнение, сложение, вычитание.

6. Структура данных в C++.

Ответ: Структура данных в C++ - это способ организации и хранения данных в программе.

7. Указатель на функцию в C++.

Ответ: Указатель на функцию в C++ - это переменная, которая содержит адрес функции.

8. Объявление и использование ссылки на константу в C++.
Ответ: Ссылка на константу объявляется с использованием ключевого слова `const` перед типом данных.
9. Перегрузка функций в C++.
Ответ: Перегрузка функций в C++ - это возможность создания нескольких функций с одним и тем же именем, но разными параметрами.
10. Объявление константы в C++.
Ответ: Константа объявляется с использованием ключевого слова `const`.
11. Статическая переменная в функции.
Ответ: Статическая переменная в функции - это переменная, которая сохраняет свое значение между вызовами функции.
12. Основные типы данных, поддерживаемые в C++.
Ответ: Основные типы данных в C++ включают целочисленные типы (int, long, short), типы с плавающей запятой (float, double), символьный тип (char), логический тип (bool).
13. Оператор, используемый для вычисления остатка от деления в C++.
Ответ: В C++ оператор % используется для вычисления остатка от деления одного числа на другое.
14. "Рекурсивная функция" (recursivefunction) в C++.
Ответ: Рекурсивная функция - это функция, которая вызывает саму себя.
15. Константный указатель в C++.
Ответ: Константный указатель объявляется с использованием `const` перед указателем.
16. Автоматическая (локальная) переменная.
Ответ: Автоматическая (локальная) переменная - это переменная, объявленная внутри функции и существующая только внутри этой функции.
17. Объявление и использование ссылки на указатель в C++.
Ответ: Ссылка на указатель объявляется с использованием `&` перед указателем.
18. Дружественные функции в C++.
Ответ: Дружественные функции - это функции, которые имеют доступ к закрытым членам класса.
19. Множественное наследование в C++.
Ответ: Множественное наследование в C++ позволяет классу наследовать от нескольких базовых классов.
20. Виртуальный деструктор.
Ответ: Виртуальный деструктор используется в C++ для правильного удаления объектов при использовании полиморфизма.

Тестовые задания №3

Перечень вопросов для оценки уровня сформированности компетенции ПК-5 по индикатору ПК-5.1:

1. В результате выполнения фрагмента программы

```
int y= 5, x= 0;  
while (x < 6){ y = y + x; x = x + 2;}
```

переменная y примет значение
А) 5;
Б) 6;
В) 11;
Г) 9.
2. Значение переменной k после выполнения фрагмента программы

```
int k = 8, i=7;  
if ((i % 3 = 1) && (i % 5 = 2)) k = i; else k=10;
```

равно
А) 0;
Б) 8;
В) 10;
Г) 7.
3. Тело цикла в программе

```
int s=5,i;  
for(i=1; i<10;i++) s=s+i;
```

выполнится
А) 5 раз;
Б) 10 раза;
В) 9 раза;
Г) бесконечное число раз.
4. Тело цикла в программе

```
int m = 56, n= 56;  
while (m <> n)  
if (m > n) m = m - n; else n = n - m;
```

выполнится
А) 0 раз;
Б) 4 раза;
В) 6 раз;
Г) бесконечно много раз.
5. В фрагмент программы

```
int k = 0;  
for(i=0; i<n;i++) if ( __ ) k++;
```

определяющий число положительных элементов в массиве `int a[n]`, нужно вписать логическое выражение
А) `i <= n`;
Б) `a[i] < a[1]`;
В) `a[i] > 0`;
Г) `a[i] <= 0`.

6. Объявление и использование макроса с параметрами.
Ответ: Макрос с параметрами объявляется с использованием `#define` и параметры указываются в скобках.
7. Исключение "undefinedbehavior" в C++.
Ответ: "Undefinedbehavior" (неопределенное поведение) - это состояние, при котором программа нарушает стандарт C++. Результат ее выполнения становится непредсказуемым.
8. Объявление и использование перегрузки операторов в C++.
Ответ: Перегрузка операторов в C++ позволяет определить новое поведение операторов для пользовательских типов данных. Это достигается путем создания специальных функций-членов класса.
9. Объявление и использование члена класса (pointertomember) в C++.
Ответ: Указатель на член класса (pointertomember) в C++ позволяет работать с членами класса через указатели. Его объявление зависит от типа члена (переменной, функции) и его класса.
10. "Ссылка на функцию" (functionreference) в C++ и его использование.
Ответ: Ссылка на функцию в C++ - это альтернативное имя для функции. Она объявляется с использованием &. Она позволяет вызывать функцию по её имени или по имени ссылки.
11. Объявление и использование пространства имен в C++.
Ответ: Пространство имен (namespace) в C++ используется для организации кода и избегания конфликтов имен. Для объявления пространства имен используется ключевое слово `namespace`.
12. Анонимное пространство имен в C++.
Ответ: Анонимное пространство имен (unnamednamespace). В C++ она используется для создания пространства имен без имени, которое видно только в пределах текущего файла.
13. Создание и использование псевдонима типа (typedef) в C++.
Ответ: Псевдоним типа (typedef) в C++ используется для создания альтернативных имен типам данных.
14. Шаблоны классов в C++.
Ответ: Шаблоны классов в C++ позволяют создавать обобщенные классы, которые могут работать с разными типами данных.
15. Основные контейнеры STL в C++.
Ответ: STL (Standard Template Library) в C++ предоставляет разнообразные контейнеры для хранения и управления данными.
16. Типы данных для использования в качестве аргументов функции в C++.
Ответ: В C++ аргументы функции могут быть любыми типами данных, включая встроенные типы (int, double, char), пользовательские классы и структуры, указатели, ссылки и константы. Это позволяет функциям работать с различными типами данных.

17. "Возвращаемое значение" (returnvalue) функции в C++ и его влияние на тип функции.
Ответ: Возвращаемое значение функции в C++ - это значение, которое функция возвращает после выполнения. Оно определяет тип функции и может быть использовано в других выражениях или передано другим функциям.
18. Итератор в C++.
Ответ: Итератор в C++ - это объект, позволяющий перебирать элементы в контейнере, например, в векторе, списке или множестве.
19. Передача аргументов в функцию как массив (или указатель на массив) в C++.
Ответ: В C++ аргументы могут быть переданы в функцию как массивы или указатели на массивы. Это позволяет функции работать с последовательностями данных.
20. Манипуляторы, используемые с потоками ввода/вывода.
Ответ: С манипуляторами можно форматировать вывод в потоки ввода/вывода в C++.

Перечень вопросов для оценки уровня сформированности компетенции ПК-5 по индикатору ПК-5.2:

1. Какое логическое выражение нужно вписать во фрагмент программы, определяющий индекс последнего элемента массива `int p[n]`, равного `x`?
- ```
int m, x;
cin >> x;
for(i=0; i<n; i++) if (__) m=i;
```
- а) `p[i] = x;`  
б) `p[i] > x;`  
**в) p[i]==x;**  
г) `i <= n.`
2. Задан одномерный массив `int x[n]`. Фрагмент алгоритма
- ```
t = x[0];
for(i=1; i<n; i++) if ( t < x[i] ) t = x[i];
```
- определяет:
- А) максимальный элемент массива;**
Б) сумму элементов массива, больших `t`;
В) количество элементов массива, больших `t`;
Г) индекс последнего из элементов массива, больших `t`.
3. К операции для работы с указателями относится
- А) `&&`;
Б) new;
В) `||`;
Г) `@`.
4. Что будет напечатано программой?
- ```
#include <iostream.h>
#include <conio.h>
main(){
int p=2;
cout << "p+2";
```

```
getch();
}
```

А) сообщение об ошибке;

Б) 11;

**В) p+2;**

Г) 9+2.

5. Какое значение будет иметь переменная z после выполнения операторов  
z=0; x=1; y= -1;

if (x>0) if (y>0) z=1; else z=2; else z=6;

А) сообщение об ошибке;

**Б) 2;**

В) 0;

Г) 6.

6. Открытие и закрытие файла в C++.

**Ответ:** Для открытия и закрытия файлов в C++ используются классы `'fstream'` и `'ofstream'`. Для открытия файла на чтение или запись используется метод `'open()'`, а для закрытия - метод `'close()'`.

7. "Аргументы по умолчанию" (defaultarguments) в C++ и их объявление.

**Ответ:** Аргументы по умолчанию в C++ - это аргументы функции, которые имеют значения по умолчанию и могут быть опущены при вызове функции.

8. Константные аргументы функции в C++ и их работа.

**Ответ:** Константные аргументы функции объявляются с использованием ключевого слова `const` и они не могут быть изменены внутри функции.

9. Объявление и использование макроса в C++.

**Ответ:** Макрос объявляется с использованием `'#define'` и представляет собой текстовую подстановку в коде. Макросы заменяются компилятором на указанный текст при компиляции.

10. Стек и куча (heap) в C++.

**Ответ:** Стек и куча - это области памяти, используемые для хранения данных в C++. Стек предназначен для хранения локальных переменных и управляется автоматически. Куча используется для динамического выделения памяти с использованием операторов `'new'` и `'delete'`.

11. Выделение памяти в куче в C++.

**Ответ:** Память выделяется в куче с использованием оператора `'new'`.

12. Многозадачность (multithreading) в C++.

**Ответ:** Многозадачность - это способность программы выполнять несколько потоков параллельно.

13. Мьютекс (mutex) в C++.

**Ответ:** Мьютекс (mutex) в C++ используется для обеспечения синхронизации между потоками.

14. Исключения (exceptions) в C++.

**Ответ: Исключения в C++ - это механизм обработки ошибок, который позволяет перехватывать и обрабатывать исключительные ситуации.**

15. Создание и сброс исключения в C++.

**Ответ: Исключение создается с использованием оператора `throw`. Исключение можно перехватить и обработать с использованием блока `try-catch`.**

16. Атрибуты (attributes) в C++ и их использование.

**Ответ: Атрибуты в C++ - это специальные аннотации, которые могут быть применены к функциям, классам и переменным. Они предоставляют дополнительную информацию о сущности.**

17. Полиморфизм в C++.

**Ответ: Полиморфизм в C++ - это возможность объектов разных классов обладать общим интерфейсом и реализовывать его по-разному.**

18. Абстрактный класс в C++.

**Ответ: Абстрактный класс - это класс, который содержит одну или несколько чистых виртуальных функций.**

19. "Friendclass" в C++ и его использование.

**Ответ: "Friendclass" в C++ - это класс, который имеет доступ к закрытым членам другого класса. Он объявляется внутри класса с использованием ключевого слова friend и может использоваться для выполнения операций, которые требуют доступа к данным класса, но не являются его членами.**

20. Преимущества и недостатки ссылок на объекты в C++.

**Ответ: Ссылки в C++ обеспечивают удобный и эффективный способ работы с объектами, так как они позволяют создавать альтернативные имена для существующих данных, уменьшая необходимость копирования объектов в памяти.**

### **Перечень вопросов для оценки уровня сформированности компетенции ПК-5 по индикатору ПК-5.3:**

1. Этап, являющийся заключительным при решении задач на ЭВМ, – ...

А) построение математической модели;

Б) анализ результатов;

**В) тестирование и отладка;**

Г) программирование;

Д) разработка алгоритма.

2. На этапе тестирования и отладки при проектировании задачи на ЭВМ происходит ...

А) получение результата;

**Б) обнаружение ошибок;**

В) запись алгоритма на языке программирования;

Г) формализация задачи;

Д) составление алгоритма решения задачи.

3. Ошибки, при решении задачи на ЭВМ, которые не обнаруживаются транслятором, – ...

**А) логические;**

Б) синтаксические;

- В) динамические;
- Г) логические и синтаксические.

4. Вычислительный эксперимент – это...

А) метод анализа обобщения суждений и предложений с помощью экспертов;

**Б) метод изучения объектов или процессов с помощью математического моделирования;**

В) один из методов экспертных оценок, при помощи которого осуществляется быстрый поиск решений, среди которых выбирается наилучшее;

Г) метод решения задачи анализа или синтеза сложной системы на основе использования ее компьютерной модели.

5. Что относится к первому этапу вычислительного эксперимента?

А) Анализ результатов расчетов, сравнение (если это возможно) с натурным экспериментом;

Б) Создание программы, реализующей вычислительный алгоритм;

**В) Построение математической модели (составление уравнений, описывающих исследуемое явление);**

Г) Выбор численных методов расчета (построение дискретной модели, аппроксимирующей исходную математическую задачу, построение разностной схемы, разработка вычислительного алгоритма и т. д.).

6. Различие между операторами пре-инкремента и пост-инкремента в C++.

**Ответ: Оператор pre-increment возвращает новое значение переменной после инкремента, а оператор post-increment возвращает исходное значение до инкремента.**

7. Указатели в C++.

**Ответ: Указатель - это переменная, которая хранит в памяти местоположение другой переменной в качестве своего значения.**

8. Влияние выбора сетки на точность моделирования физических процессов.

**Ответ: Выбор сетки имеет существенное влияние на точность моделирования физических процессов. Сетка должна быть достаточно плотной и сбалансированной, чтобы адекватно описывать пространственное разрешение процесса.**

9. Принципы, лежащие в основе разработки физических моделей.

**Ответ: Разработка физических моделей основана на основных принципах сохранения массы, энергии и импульса, а также на применении математических уравнений. Эти принципы отражают законы физики для описания процесса.**

10. Вычислительные методы, применяемые для моделирования аэродинамики.

**Ответ: Для моделирования аэродинамики широко применяются методы вычислительной гидродинамики, такие как метод конечных объемов, метод конечных элементов и метод конечных разностей.**

11. Типы нелинейности, возникающие при моделировании физических процессов.

**Ответ: Нелинейная зависимость между входными и выходными параметрами, нелинейные материальные свойства, нелинейные граничные условия или же нелинейные дифференциальные уравнения, описывающие физический процесс.**

12. Методы, используемые для моделирования динамики механических систем.  
**Ответ:** Для моделирования динамики механических систем широко применяются методы символьного анализа, методы численной интеграции.
13. Аппроксимация и его применение для моделировании физических процессов.  
**Ответ:** Аппроксимация в моделировании физических процессов означает приближенное представление сложных функций, уравнений или данных более простыми или понятными формами.
14. Различие между эмпирическим и фундаментальным моделированием в физике.  
**Ответ:** Эмпирическое моделирование в физике базируется на экспериментальных данных и наблюдениях. Фундаментальное моделирование строится на фундаментальных законах физики и математических уравнениях.
15. Моделирование физических процессов в разработке новых технологий и материалов.  
**Ответ:** Моделирование физических процессов играет важную роль в разработке новых технологий и материалов, позволяя проводить виртуальные эксперименты, оптимизировать параметры и условия процесса, и изучать влияние различных факторов на систему.
16. Методы, используемые для моделирования электромагнитных полей.  
**Ответ:** Для моделирования электромагнитных полей широко применяются методы конечных элементов, методы конечных разностей и методы конечных объемов.
17. Факторы, приводящие к неверным результатам при моделировании физических процессов.  
**Ответ:** Ошибки в формулировке модели или параметрах, недостаточность или неточность входных данных, неточности численных методов или недостаток учета нелинейных эффектов или неучтенных физических явлений в модели.
18. Факторы, влияющие на выбор численного метода для моделирования физических процессов.  
**Ответ:** При выборе численного метода следует учитывать тип физического процесса, его особенности и условия, доступные ресурсы (вычислительные мощности, время), требуемую точность результата.
19. Моделирование физических процессов в аэрокосмической и автомобильной промышленности.  
**Ответ:** Моделирование физических процессов широко применяется в аэрокосмической и автомобильной промышленности для проектирования и оптимизации конструкций, аэродинамики, теплопередачи, акустики и других аспектов.
20. Методы, используемые для моделирования распространения звука и акустических волн.

**Ответ: Для моделирования распространения звука и акустических волн широко используются методы конечных элементов, методы конечных разностей и методы интегральных уравнений.**

# Список экзаменационных вопросов

## Перечень вопросов для оценки уровня сформированности компетенции ОПК-20:

1. Типы данных на C++. Стандартный тип int. Стандартный тип double.  
**Ответ:** Тип определяет, какие значения может иметь переменная, какие операции с ней можно производить и сколько байт в памяти она будет занимать. Стандартный тип int определяет целочисленные значения. Стандартный тип double используется для представления чисел с плавающей запятой двойной точности.
2. Стандартный тип void. Стандартный тип char. Стандартный тип bool.  
**Ответ:** Стандартный тип void используется, когда результат работы функции не требуется или не имеет значения. Стандартный тип char представляет собой числовое значение ASCII-кода символа. Стандартный тип bool в языке программирования обычно используется для представления логических значений true (истина) и false (ложь).
3. Стандартный тип float. Модификаторы типа signed, unsigned. Модификаторы типа short, long.  
**Ответ:** Тип данных float представляет собой числа с плавающей точкой и используется для представления десятичных чисел. Модификаторы типа signed и unsigned используются для указания диапазона значений, которые может принимать переменная определенного типа данных. Модификаторы типа short и long используются для изменения размера переменной определенного типа данных.
4. Модификатор const. Структура программы на C++. Комментарии на C++.  
**Ответ:** Модификатор const используется для объявления переменной, значение которой не может быть изменено после ее инициализации. Структура программы на C++: 1) подключение заголовочных файлов; 2) объявление глобальных переменных; 3) определение функций; 4) Основная функция main. Комментарии на C++ используются для добавления пояснений и описания кода.
5. Вывод данных на экран. Ввод данных с клавиатуры.  
**Ответ:** Вывод данных на экран в программировании осуществляется с помощью команды или функции, которая позволяет вывести определенное значение или текст на экран. Ввод данных с клавиатуры в программировании осуществляется с помощью команды или функции, которая позволяет пользователю ввести определенное значение с клавиатуры.
6. Унарные операции. Бинарные операции.  
**Ответ:** Унарные операции представляют собой действия, выполняемые на одном значении, и могут выполняться над различными типами данных, такими как числа, строки и булевы значения. Бинарные операции в C++ - это операции, которые принимают два операнда и возвращают результат.
7. Логические операции. Приоритеты операций и порядок вычислений операций. Преобразования типов при вычислении выражений.  
**Ответ:** Логическое И: &&; ИЛИ: || ; НЕ: ! В C++ приоритеты операций: операции в скобках; унарные операции; арифметические; логические; присваивания;

сравнения; операция НЕ; операции И, ИЛИ. При выполнении операций производится автоматическое преобразование типов, чтобы привести операнды выражений к общему типу.

8. Оператор ветвления if. Вложенное ветвление. Оператор выбора switch. Типы константы выбора.

**Ответ:** Оператор условного ветвления if позволяют принять программе решение, в зависимости от результата проверки условия. Вложенное ветвление – это ветвление, которое находится внутри другого ветвления. Оператор выбора switch в C++ позволяет выполнить различные действия в зависимости от значения выражения.

9. Оператор цикла while. Вложенные циклы с помощью while. Оператор цикла for.

**Ответ:** Оператор цикла while – цикл с предусловием. Вложенные циклы с помощью while в C++ позволяют выполнить повторяющиеся действия внутри другого цикла. Оператор for позволяет повторить выполнение того или иного оператора или составного оператора заданное число раз.

10. Оператор цикла dowhile. Оператор break. Оператор continue.

**Ответ:** Оператор цикла dowhile в C++ выполняет определенные действия, пока указанное условие истинно. break — это оператор, предназначенный для досрочного выхода из цикла. Оператор continue осуществляет принудительный переход к следующему шагу цикла, пропуская любой код, оставшийся невыполненным.

11. Массивы. Одномерные массивы.

**Ответ:** Массив – это структура данных, состоящая из набора элементов (значений или переменных), каждый из которых идентифицируется по крайней мере одним индексом или ключом массива. Одномерный массив - это структура данных, которая представляет собой последовательность элементов одного типа, расположенных последовательно в памяти.

12. Многомерные массивы. Функции для работы с массивами. Строки на C++.

**Ответ:** Многомерный массив – это структура данных, которая позволяет хранить элементы в виде таблицы с несколькими измерениями. В C++ для работы с массивами доступны следующие функции: sizeof(); copy(); sort(); reverse() и др. Строка — последовательность символов.

#### **Перечень вопросов для оценки уровня сформированности компетенции ОПК-19:**

13. Передача параметров функции по значению. Передача параметров функции по ссылке.

**Ответ:** Передача параметра по значению означает что вызывающая функция копирует в память, доступную вызываемой, (обычно стек) непосредственное значение. Передача параметров функции по ссылке означает, что при вызове функции вместо создания копии значения параметра передается сама ссылка на объект или переменную.

14. Передача массивов функции в качестве параметров. Рекурсивные функции.

**Ответ:** Передача массивов в качестве параметров функции позволяет функции работать с элементами массива и изменять их значения. Рекурсивная функция — это функция, которая вызывает сама себя.

15. Тип данных указатель. Описание типизированных указателей. Описание не типизированных указателей.  
**Ответ:** Указатель — переменная, содержащая адрес объекта. Типизированные указатели - это указатели на определенный тип данных в языках программирования, которые обеспечивают компилятору информацию о типе данных, на который они указывают. Не типизированные указатели – это указатели, которые могут указывать на объекты любого типа данных.
16. Оператор выбора. Примеры использования цикла. Понятие заикливания.  
**Ответ:** Оператор выбора (переключатель, селектор) предназначен для выбора одного варианта из множества возможных вариантов разветвления программы. Цикл "for" для выполнения определенного числа итераций: `for(i=1; i<5; i++) {cout<<i<<"\n";}` Этот цикл будет выводить числа от 1 до 5. Заикливание - ситуация, при которой выполнение цикла никогда не заканчивается.
17. Функция main – отличие от других функций. Инструкция return.  
**Ответ:** Функция main ничем не отличается от других функций за исключением того, что возвращаемое значение этой функции используется операционной системой как код возврата. Инструкция return прекращает исполнение функции, даже если в теле функции остались другие инструкции.
18. Инструкция вызова функции. Понятие модуля и заголовочного файла.  
**Ответ:** Инструкция вызова функции - это способ вызвать функцию в программе, передавая аргументы. Модуль представляет собой независимую и самодостаточную единицу программного кода, которая выполняет какую-либо конкретную функцию или предоставляет определенные сервисы.
19. Инициализация массивов. Динамические массивы. Динамические переменные.  
**Ответ:** Инициализация — это процесс создания массива и заполнения его значениями. Динамические массивы данных — это структура, которая позволяет прямо в процессе написания кода автоматически менять размер массива и освобождать ненужные ячейки. Динамическая переменная — переменная в программе, место в оперативной памяти под которую выделяется во время выполнения программы.
20. Ссылки. Структуры. Динамические структуры данных.  
**Ответ:** Ссылки в программировании - это адреса, которые указывают на определенные объекты или данные в памяти компьютера. Структуры — это встроенная структура данных, объединяющая данные разных типов под одним именем. Динамические структуры данных - это структуры данных, которые могут изменять свой размер и форму во время выполнения программы.
21. Структура данных в C++. Указатель на функцию в C++. Объявление и использование ссылки на константу в C++.  
**Ответ:** Структура данных в C++ - это способ организации и хранения данных в программе. Указатель на функцию в C++ - это переменная, которая содержит адрес функции. Ссылка на константу объявляется с использованием ключевого слова `const` перед типом данных.
22. Перегрузка функций в C++. Объявление константы в C++. Статическая переменная в функции.

**Ответ: Перегрузка функций в C++ - это возможность создания нескольких функций с одним и тем же именем, но разными параметрами. Константа объявляется с использованием ключевого слова `const`. Статическая переменная в функции - это переменная, которая сохраняет свое значение между вызовами функции.**

23. Основные типы данных, поддерживаемые в C++. "Рекурсивная функция" (recursive function) в C++. Константный указатель в C++.

**Ответ: Основные типы данных в C++ включают целочисленные типы (int, long, short), типы с плавающей запятой (float, double), символьный тип (char), логический тип (bool). Рекурсивная функция - это функция, которая вызывает саму себя. Константный указатель объявляется с использованием `const` перед указателем.**

24. Автоматическая (локальная) переменная. Дружественные функции в C++. Виртуальный деструктор.

**Ответ: Автоматическая (локальная) переменная - это переменная, объявленная внутри функции и существующая только внутри этой функции. Дружественные функции - это функции, которые имеют доступ к закрытым членам класса. Виртуальный деструктор используется в C++ для правильного удаления объектов при использовании полиморфизма.**

#### **Перечень вопросов для оценки уровня сформированности компетенции ПК-5:**

25. Объявление и использование макроса с параметрами. Исключение "undefined behavior" в C++.

**Ответ: Макрос с параметрами объявляется с использованием `#define` и параметры указываются в скобках. "Undefined behavior" (неопределенное поведение) - это состояние, при котором программа нарушает стандарт C++. Результат ее выполнения становится непредсказуемым.**

26. Объявление и использование перегрузки операторов в C++. Объявление и использование члена класса (pointertomember) в C++.

**Ответ: Перегрузка операторов в C++ позволяет определить новое поведение операторов для пользовательских типов данных. Это достигается путем создания специальных функций-членов класса. Указатель на член класса (pointertomember) в C++ позволяет работать с членами класса через указатели. Его объявление зависит от типа члена (переменной, функции) и его класса.**

27. Объявление и использование пространства имен в C++. Анонимное пространство имен в C++.

**Ответ: Пространство имен (namespace) в C++ используется для организации кода и избегания конфликтов имен. Для объявления пространства имен используется ключевое слово `namespace`. Анонимное пространство имен (unnamed namespace). В C++ она используется для создания пространства имен без имени, которое видно только в пределах текущего файла.**

28. Создание и использование псевдонима типа (typedef) в C++. Шаблоны классов в C++. Основные контейнеры STL в C++.

**Ответ: Псевдоним типа (typedef) в C++ используется для создания альтернативных имен типам данных. Шаблоны классов в C++ позволяют создавать обобщенные классы, которые могут работать с разными типами**

данных. STL (Standard Template Library) в C++ предоставляет разнообразные контейнеры для хранения и управления данными.

29. Открытие и закрытие файла в C++. "Аргументы по умолчанию" (default arguments) в C++ и их объявление.  
**Ответ:** Для открытия и закрытия файлов в C++ используются классы `fstream` и `ofstream`. Для открытия файла на чтение или запись используется метод `open()`, а для закрытия - метод `close()`. Аргументы по умолчанию в C++ - это аргументы функции, которые имеют значения по умолчанию и могут быть опущены при вызове функции.
30. Константные аргументы функции в C++ и их работа. Объявление и использование макроса в C++.  
**Ответ:** Константные аргументы функции объявляются с использованием ключевого слова `const` и они не могут быть изменены внутри функции. Макрос объявляется с использованием `#define` и представляет собой текстовую подстановку в коде. Макросы заменяются компилятором на указанный текст при компиляции.
31. Стек и куча (heap) в C++. Выделение памяти в куче и её освобождение в C++.  
**Ответ:** Стек и куча - это области памяти, используемые для хранения данных в C++. Стек предназначен для хранения локальных переменных и управляется автоматически. Куча используется для динамического выделения памяти с использованием операторов `new` и `delete`. Память выделяется в куче с использованием оператора `new`.
32. Многозадачность (multithreading) в C++. Исключения (exceptions) в C++. Создание и сброс исключения в C++.  
**Ответ:** Многозадачность - это способность программы выполнять несколько потоков параллельно. Исключения в C++ - это механизм обработки ошибок, который позволяет перехватывать и обрабатывать исключительные ситуации. Исключение создается с использованием оператора `throw`. Исключение можно перехватить и обработать с использованием блока `try-catch`.
33. Атрибуты (attributes) в C++ и их использование. Полиморфизм в C++.  
**Ответ:** Атрибуты в C++ - это специальные аннотации, которые могут быть применены к функциям, классам и переменным. Они предоставляют дополнительную информацию о сущности. Полиморфизм в C++ - это возможность объектов разных классов обладать общим интерфейсом и реализовывать его по-разному.
34. Различие между операторами пре-инкремента и пост-инкремента в C++. Указатели в C++.  
**Ответ:** Оператор `pre-increment` возвращает новое значение переменной после инкремента, а оператор `post-increment` возвращает исходное значение до инкремента. Указатель - это переменная, которая хранит в памяти местоположение другой переменной в качестве своего значения.
35. Вычислительные методы, применяемые для моделирования аэродинамики. Методы, используемые для моделирования динамики механических систем.  
**Ответ:** Для моделирования аэродинамики широко применяются методы вычислительной гидродинамики, такие как метод конечных объемов, метод

**конечных элементов и метод конечных разностей. Для моделирования динамики механических систем широко применяются методы символьного анализа, методы численной интеграции.**

36. **Аппроксимация и его применение для моделирования физических процессов.**

Различие между эмпирическим и фундаментальным моделированием в физике.

**Ответ: Аппроксимация в моделировании физических процессов означает приближенное представление сложных функций, уравнений или данных более простыми или понятными формами. Эмпирическое моделирование в физике базируется на экспериментальных данных и наблюдениях. Фундаментальное моделирование строится на фундаментальных законах физики и математических уравнениях.**

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

Рейтинг-план дисциплины:

| Виды учебной деятельности студентов                                              | Балл за конкретное задание | Число заданий за семестр | Баллы       |              |
|----------------------------------------------------------------------------------|----------------------------|--------------------------|-------------|--------------|
|                                                                                  |                            |                          | Минимальный | Максимальный |
| <b>Модуль 1</b>                                                                  |                            |                          |             |              |
| <b>Текущий контроль</b>                                                          |                            |                          | <b>0</b>    | <b>20</b>    |
| 1. Тестирование                                                                  | 10                         | 2                        | 0           | 20           |
| <b>Рубежный контроль</b>                                                         |                            |                          |             | <b>15</b>    |
| 1. Тестирование                                                                  | 15                         | 1                        | 0           | 15           |
| <b>итого</b>                                                                     |                            |                          | <b>0</b>    | <b>35</b>    |
| <b>Модуль 2</b>                                                                  |                            |                          |             |              |
| <b>Текущий контроль</b>                                                          |                            |                          | <b>0</b>    | <b>20</b>    |
| 1. Тестирование                                                                  | 10                         | 2                        | 0           | 20           |
| <b>Рубежный контроль</b>                                                         |                            |                          | <b>0</b>    | <b>15</b>    |
| 1. Тестирование                                                                  | 15                         | 1                        | 0           | 15           |
| <b>итого</b>                                                                     |                            |                          | <b>0</b>    | <b>35</b>    |
| <b>Поощрительные баллы</b>                                                       |                            |                          |             |              |
| 1. Выполнение дополнительных заданий (из перечня заданий для практических работ) | 2                          | 5                        | 0           | 10           |
| <b>Посещаемость (баллы вычитаются из общей суммы набранных баллов)</b>           |                            |                          |             |              |
| Посещение лекционных занятий                                                     |                            |                          | 0           | -6           |
| Посещение практических занятий                                                   |                            |                          | 0           | -10          |
| <b>Итоговый контроль</b>                                                         |                            |                          |             |              |
| Экзамен                                                                          |                            |                          | 0           | 30           |
| <b>итого</b>                                                                     |                            |                          | <b>0</b>    | <b>110</b>   |

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:

Рейтинговый балл =  $k \times$  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.