

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
НОЯБРЬСКИЙ ИНСТИТУТ НЕФТИ И ГАЗА  
(ФИЛИАЛ) ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»  
(Филиал ТИУ в г. Ноябрьске)**

Кафедра Транспорта и технологий нефтегазового комплекса

**Комплект контрольно-оценочных средств  
по учебной дисциплине**

**ПРОГРАММИРОВАНИЕ И АЛГОРИТМИЗАЦИЯ**  
основной профессиональной образовательной программы  
по направлению подготовки

**15.03.04 Автоматизация технологических процессов и производств**

профиль **Автоматизация технологических процессов и производств в  
нефтяной и газовой промышленности**

Ноябрьск, 2019

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта высшего образования по направлению подготовки 15.03.04 «Автоматизация технологических процессов и производств» и рабочей программы дисциплины «Программирование и алгоритмизация».

Комплект контрольно-оценочных оценочных средств рассмотрен на заседании кафедры ТТНК

Протокол № 9 от «15» мая 2019 г.

Заведующий кафедрой  А.В.Козлов

Разработчик:  
Лаптева С.В., доцент, к.п.н., доцент



**Паспорт комплекта контрольно-оценочных средств  
по учебной дисциплине  
«Программирование и алгоритмизация»**

**1. Контролируемые компетенции**

Компетенции, формируемые в процессе изучения дисциплины для заочной формы обучения: в 3 семестре (5 лет обучения) и во 2 семестре (3,6 года обучения) (таблица 1):

Таблица 1

Код компетенции	Формулировка компетенции
ОПК-3	Способность использовать современные информационные технологии, технику, прикладные программные средства при решении задач профессиональной деятельности
ПК-19	Способность участвовать в работах по моделированию продукции, технологических процессов, производств, средств и систем автоматизации, контроля, диагностики, испытаний и управления процессами, жизненным циклом продукции и ее качеством с использованием современных средств автоматизированного проектирования, по разработке алгоритмического и программного обеспечения средств и систем автоматизации и управления процессами

Формой промежуточной аттестации по учебной дисциплине является экзамен.

**2. Результаты освоения учебной дисциплины, подлежащие проверке**

В процессе изучения дисциплины осуществляется комплексная проверка следующих результатов обучения (таблица 2, 3, 4):

Таблица 2

*Знать:*

Индекс результата	Результаты обучения	Показатели оценки результата
31	Знать: современные информационные технологии получения новых знаний в области использования математического аппарата для решения задач профессиональной деятельности	<b>Истолковывать</b> виды современных информационных технологий для решения профессиональных задач с помощью программирования; <b>Объяснять</b> современные методы решения задач, этапы решения задач
32	Знать: назначение, принцип действия и характери-	<b>Объяснять</b> классификацию моделей систем и процессов; поня-

	<p>стики аналоговых и цифровых электронных схем; методы и средства моделирования технических объектов; методы анализа технологических процессов и оборудования, как объектов автоматизации и управления; классификацию модели систем и процессов, их виды и виды моделирования; принципы и методологию функционального, имитационного и математического моделирования систем и процессов; методы построения моделирующих алгоритмов; методологические основы функционирования, моделирования и синтеза систем автоматического управления (САУ); основные методы анализа САУ во временной и частотных областях, способы синтеза САУ: типовые пакеты прикладных программ анализа динамических систем; методы построения математических моделей, их упрощения; технические и программные средства моделирования; технологию планирования эксперимента; методы статистического моделирования на персональном компьютере</p>	<p>тие алгоритма, методы построения алгоритмов; методологию структурного и объектно-ориентированного подходов к построению различных алгоритмов и решению профессиональных задач;  <b>Истолковывать</b> принципы работы различных систем управления технологическими процессами и систем управления;  <b>Анализировать</b> прикладные программные продукты для выбора инструментария для построения алгоритмов решения вычислительных задач, входящих в состав систем управления;  <b>Объяснять</b> принципы работы ПЭВМ</p>
--	---	--

Таблица 3

*Уметь:*

Индекс результата	Результаты обучения	Показатели оценки результата
У1	<p>Уметь: использовать современные информационные технологии, технику, прикладные программные средства при решении задач профессиональной деятельности</p>	<p><b>Применять</b> современные информационные технологии для решения профессиональных задач с помощью программирования;  <b>Применять</b> современные методы решения задач, этапы решения задач</p>
У2	<p>Уметь: самостоятельно разрабатывать математические и физические модели процессов и производственных объектов, выполнять работы по расчету и проектированию средств и систем автоматизации и управления; реализовывать простые алгоритмы имитационного моделирования; использовать основные методы построения математических моделей процессов, систем, их элементов и систем управления; работать с каким либо из основных типов программных систем, предназначенных для математического и имитационного моделирования, планировать модельный эксперимент</p>	<p><b>Применять</b> понятие алгоритма, методы построения алгоритмов; методологию структурного и объектно-ориентированного подходов к построению различных алгоритмов и решению профессиональных задач;  <b>Применять</b> прикладные программные продукты для выбора инструментария для построения алгоритмов решения вычислительных задач, входящих в состав систем управления</p>

	и обрабатывать его результаты на персональном компьютере	
--	--	--

Таблица 4

*Владеть:*

Индекс результата	Результаты обучения	Показатели оценки результата
V1	Владеть: прикладными программными средствами при решении задач профессиональной деятельности; обеспечения систем измерения.	<b>Рациональное и качественное применение</b> современных информационных технологий в решении профессиональных задач с помощью программирования; <b>Построение</b> алгоритмов решения вычислительных задач с использованием современных методов решения задач
V2	Владеть: методами математического анализа и моделирования, теоретического и экспериментального исследования и использовать их для решения конкретных задач; навыками работы с программной системой для математического и имитационного моделирования; методологией постановки задачи по разработке исходного текста программы, приемами разбиения стратегической задачи на последовательность тактических; методами и средствами обработки исходного текста на предмет выявления обнаруживаемых ошибок и получения начального варианта загрузочного модуля	<b>Постановка</b> задачи по разработке исходного текста программы; <b>Разбиение</b> стратегической задачи на последовательность тактических; <b>Обработка</b> исходного текста на предмет выявления обнаруживаемых ошибок и получения начального варианта загрузочного модуля

### 3. Контроль и оценка освоения учебной дисциплины

Таблица 5

№ п/п	Элементы учебной дисциплины (темы/раздела)	Результаты обучения (индекс результата)	Показатели оценки результата	Форма и методы контроля	Макс. балл
1.	Этапы проектирования и жизненный цикл программных продуктов	31, 32, У1, У2, В1, В2	Истолковывать виды современных информационных технологий для решения профессиональных задач с помощью программирования; Объяснять современные методы решения задач, этапы решения задач Объяснять классификацию моделей систем и процессов; понятие алгоритма, методы построения алгоритмов; методологию структурного и объектно-ориентированного подходов к построению различных алгоритмов и решению профессиональных задач;	Опрос	3
2.	Программирование основных алгоритмических структур на языке программирования	31, 32, У1, У2, В1, В2	Истолковывать принципы работы различных систем управления технологическими процессами и систем управления; Анализировать прикладные программные про-	Защита лабораторной работы	16
				Тест	10

3.	Структурный подход к построению программ	31, 32, У1, У2, В1, В2	<p>дукты для выбора инструментария для построения алгоритмов решения вычислительных задач, входящих в состав систем управления;</p> <p>Объяснять принципы работы ПЭВМ</p>	Опрос, тест	13
4.	Типы данных в языке программирования	31, 32, У1, У2, В1, В2	<p>Применять современные информационные технологии для решения профессиональных задач с помощью программирования;</p> <p>Применять современные методы решения задач, этапы решения задач</p> <p>Применять понятие алгоритма, методы построения алгоритмов; методологию структурного и объектно-ориентированного подходов к построению различных алгоритмов и решению профессиональных задач;</p>	Защита лабораторной работы	16
				Опрос, контрольная работа	8
5.	Типы данных, определяемые пользователем	31, 32, У1, У2, В1, В2	<p>Применять прикладные программные продукты для выбора инструментария для построения алгоритмов решения вычислительных задач, входящих в состав систем управления</p>	Тест, контрольная работа	14
1-5	Темы 1-5	31, 32, У1, У2, В1, В2	<p>Рациональное и качественное применение современных информационных технологий в решении профессиональных задач с помощью программирования;</p> <p>Построение алгоритмов решения вычислительных задач с использованием современных методов решения задач</p> <p>Постановка задачи по разработке исходного текста программы;</p> <p>Разбиение стратегической задачи на последовательность тактических;</p> <p>Обработка исходного текста на предмет выявления обнаруживаемых ошибок и получения начального варианта загрузочного модуля</p>	Курсовая работа (статус - защита)	20
ИТОГО:					<b>100</b>

#### 4. Типовые задания для текущего контроля

Типовые задания для текущего контроля представляют собой комплекты заданий, охватывающих пороговый и продвинутой уровень усвоения знаний, умений и навыков согласно тематике изучаемого материала.

Текущий контроль представлен заданиями для контрольных работ, тестами и вопросами для самоконтроля (опрос).

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
НОЯБРЬСКИЙ ИНСТИТУТ НЕФТИ И ГАЗА  
(ФИЛИАЛ) ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»  
(Филиал ТИУ в г. Ноябрьск)**

Кафедра экономики, менеджмента и естественнонаучных дисциплин

**Примерная тематика курсовых работ  
по дисциплине «Программирование и алгоритмизация»**

1. Сортировка методом пузырька (обменом). Сравнительный анализ методов сортировки.
2. Сортировка выбором. Сравнительный анализ методов сортировки.
3. Сортировка вставками. Сравнительный анализ методов сортировки.
4. Метод быстрой сортировки. Сравнение итерации и рекурсии.
5. Последовательный и бинарный поиск в массиве.
6. Рекурсивный алгоритм бинарного поиска. Сравнение итерации и рекурсии.
7. Алгоритм Евклида (нахождение НОД): итерационный и рекурсивный. Алгоритм нахождения НОК. Сравнение итерации и рекурсии.
8. Сортировка подсчетом. Множество как тип данных.
9. Генерирование перестановок.
10. Переборные алгоритмы. Задача о 8-ми ферзях.
11. Алгоритм приближенного решения уравнения методом бинарного деления.
12. Рекурсия в задаче о ханойских башнях.
13. Алгоритмы моделирования многочленов на базе массивов.
14. Матрица как двумерный массив: процедуры ввода, вывода и заполнения матрицы.
15. Алгоритмы основных действий с матрицами.
16. Алгоритм построения магических квадратов.
17. Алгоритмы со строками: проверка правильности расстановки скобок в строке.
18. Алгоритм приближенного вычисления бесконечной суммы (см. цикл).
19. Алгоритм поиска простых чисел (Решето Эратосфена).
20. Алгоритмы основных процедур и функций (подпрограммы) для работы с массивами.
21. Алгоритмы длинной арифметики: сложение длинных чисел.
22. Рекурсивный алгоритм вычисления натуральной степени числа.

## Критерии оценки курсовой работы

Вид работы по выполнению КР	Критерий	Макс. кол-во баллов за критерий	Нарушение критерия
Содержание курсовой работы	Актуальность темы (или необходимость решения данной задачи)	5	
	Методы исследования, применяемые в работе	5	
	Полнота раскрытия темы	20	
Оформление пояснительной записки	Соответствие содержания пояснительной записки цели и задачам курсовой работы	10	
	Нормоконтроль	5	
Разработка и отладка программного приложения	Работоспособность	10	Работа не допускается к защите
	Соответствие цели и задачам курсовой работы	10	
	Использование нескольких языков программирования (или систем разработки приложений)	10	
Защита курсовой работы	Своевременное предоставление курсовой работы на проверку и защиту (согласно графику работы над КР)	5	Минус 15-30 баллов
	Доклад и презентация на защите	5	
	Дополнительные вопросы на защите	5	
	Оценка руководителя	5	
<b>ИТОГО</b>	<i>До 100 баллов – 5 бонусных баллов</i>	95	

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
НОЯБРЬСКИЙ ИНСТИТУТ НЕФТИ И ГАЗА  
(ФИЛИАЛ) ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»  
(Филиал ТИУ в г. Ноябрьск)**

Кафедра Экономики, менеджмента и естественнонаучных дисциплин

**Вопросы для самоконтроля по темам (опрос)  
по дисциплине «Программирование и алгоритмизация»**

**Тема 1. Этапы проектирования и жизненный цикл программных продуктов.**

Постановка задачи. Хорошо и плохо поставленные задачи. Модель задачи. Адекватность модели. Тестирование

**Вопросы для самоконтроля**

1. Постановка задачи. Хорошо и плохо поставленные задачи.
2. Модель задачи. Адекватность модели. Тестирование

**Тема 3. Структурный подход к построению программ**

Функциональная декомпозиция. Объявление и описание функций. Глобальные и локальные переменные. Передача параметров в функции по значению и по ссылке. Программирование рекурсивных алгоритмов. Концепция модульного программирования. Реализация рекурсии.

**Вопросы для самоконтроля**

1. Функциональная декомпозиция.
2. Объявление и описание функций.
3. Глобальные и локальные переменные.
4. Передача параметров в функции по значению и по ссылке.
5. Программирование рекурсивных алгоритмов.
6. Концепция модульного программирования.
7. Реализация рекурсии.

**Тема 4. Типы данных в языке программирования**

Концепция типа. Классификация типов данных. Массивы: одномерные и многомерные. Операции с массивами, передача массивов как параметров функций. Алгоритмы сортировки и поиска в массивах. Строка как массив символов. Библиотечные функции обработки строк. Применение файлов для ввода-вывода массивов. Виды файлов. Объявление файловой переменной. Функции работы с файлами.

### Вопросы для самоконтроля

1. Концепция типа.
2. Классификация типов данных.
3. Массивы: одномерные и многомерные.
4. Операции с массивами, передача массивов как параметров функций.
5. Алгоритмы сортировки и поиска в массивах.
6. Строка как массив символов.
7. Библиотечные функции обработки строк.
8. Применение файлов для ввода-вывода массивов.
9. Виды файлов.
10. Объявление файловой переменной.
11. Функции работы с файлами.

### Критерии оценки:

Оценка (в баллах)	Описание оценки
2	Ответ <b>полный</b> : даны все понятия и охарактеризованы все процессы вопроса; дан ответ на дополнительный вопрос (при необходимости)
1	Ответ <b>неполный</b> : даны только основные понятия и неполностью охарактеризованы процессы вопроса; дан ответ на дополнительный вопрос без пояснения (при необходимости)
0	Ответ <b>отсутствует</b> или даны не все основные понятия и неполностью охарактеризованы процессы вопроса

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
НОЯБРЬСКИЙ ИНСТИТУТ НЕФТИ И ГАЗА  
(ФИЛИАЛ) ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»  
(Филиал ТИУ в г. Ноябрьск)**

Кафедра экономики, менеджмента и естественнонаучных дисциплин

**Типовые задания контрольных работ  
по дисциплине «Программирование и алгоритмизация»**

**Тема 4.**

**Упражнения:**

1. Удалить в массиве все числа, которые повторяются более двух раз.
2. Введите одномерный целочисленный массив. Найдите наибольший нечетный элемент. Далее трижды осуществите циклический сдвиг влево элементов, стоящих справа от найденного максимума, и один раз сдвиг элементов вправо, стоящих слева от найденного максимума.
3. Найдите сумму отрицательных элементов массива.
4. Найдите произведение элементов массива с нечетными номерами.
5. Найдите сумму элементов массива между двумя первыми нулями. Если двух нулей нет в массиве, то выведите ноль.
6. Найдите наибольший элемент массива.
7. Найдите наименьший четный элемент массива. Если такого нет, то выведите первый элемент.
8. Преобразовать массив так, чтобы сначала шли нулевые элементы, а затем все остальные.
9. Найдите сумму номеров минимального и максимального элементов.
10. Найдите минимальный по модулю элемент массива.
11. По целому  $n$  и  $n$  положительным целым числам определите, можно ли из них образовать подмножество, сумма элементов которого делится на  $n$  без остатка. Если можно, то найти любое из таких подмножеств.
12. Найти подмножество данного множества чисел такое, что сумма его элементов равна заданному числу.
13. Напишите программу бинарного поиска по ключу в массиве целых чисел.

**Тема 5.**

1. Создать класс Tiles (кафель), который будет содержать поля с открытым доступом: brand, size\_h, size\_w, price и метод класса getData(). В главной

функции объявить пару объектов класса и внести данные в поля. Затем отобразить их, вызвав метод `getData()`.

2. Определить класс `Children`, который содержит такие поля (члены класса): закрытые – имя ребенка, фамилию и возраст, публичные – методы ввода данных и отображения их на экран. Объявить два объекта класса, внести данные и показать их.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
НОЯБРЬСКИЙ ИНСТИТУТ НЕФТИ И ГАЗА  
(ФИЛИАЛ) ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»  
(Филиал ТИУ в г. Ноябрьск)**

Кафедра экономики, менеджмента и естественнонаучных дисциплин

**Фонд тестовых заданий  
по дисциплине «Программирование и алгоритмизация»**

**Тема 2.**

**1. Структура объявления переменных в C++**

- [=]; <идент. 2>, ...;
- [==]; <идент. 2>, ...;
- [:=], <идент. 2>, ...;
- [=], <идент. 2>, ...;

**2. Простые типы данных в C++.**

- целые – int, вещественные – float или real, символьные – char
- целые – int, вещественные – float или double, символьные – string
- целые – bool, вещественные – float или double, символьные – string
- целые – int, вещественные – float или double, символьные – char

**3. Какие служебные символы используются для обозначения начала и конца блока кода?**

- begin end
- ( )
- < >
- { }

**4. Какая из следующих записей - правильный комментарий в C++?**

- /\* Комментарий \*/
- {комментарий}
- \*\* Комментарий \*\*
- /\* комментарий \*/

**5. Укажите правильную форму записи цикла do while**

- - // форма записи оператора цикла do while:
  - do // начало цикла do while
  - {
  - /\*блок операторов\*/;
  - }
  - while {/\*условие выполнения цикла\*/} // конец цикла do while

```
// форма записи оператора цикла do while:  
do // начало цикла do while  
{  
/*блок операторов*/;  
}  
while (/*условие выполнения цикла*/) // конец цикла do while
```

- // форма записи оператора цикла do while:  
do // начало цикла do while  
{  
/\*блок операторов\*/;  
}  
while (/\*условие выполнения цикла\*/); // конец цикла do while

## 6. Общий формат оператора множественного выбора - switch

- switch (switch\_expression)  
{  
case constant1, case constant2: statement1; [break;]  
case constantN: statementN; [break;]  
[default: statement N+1;]  
}

- switch (switch\_expression)  
{  
case constant1: statement1; [break;]  
case constant2: statement2; [break;]  
case constantN: statementN; [break;]  
[default: statement N+1;]  
}

- switch (switch\_expression)  
{  
case constant1: statement1; [break;]  
case constant2: statement2; [break;]  
case constantN: statementN; [break;]  
[else: statement N+1;]  
}

## 7. Тело любого цикла выполняется до тех пор, пока его условие ...

- ложно
- истинно
- у цикла нет условия

## 8. Какой служебный знак ставится после оператора case ?

- ;
- :

- .
- -

**9. Какой из перечисленных типов данных не является типом данных в C++?**

- double
- int
- float
- real

**10. Какие среды программирования (IDE) предназначены для разработки программных средств?**

- MVS, NetBeans, QT Creator, RAD Studio, Dev-C++
- MVS, Code::Blocks, QT Creator, RAD Studio, MathCAD
- MVS, Code::Blocks, QT Creator, AutoCAD, Eclipse

**Критерий оценки:**

Общая сумма баллов – 10 баллов.

За каждый правильный ответ – 1 балл.

### Тема 3.

**1. Какую функцию должны содержать все программы на C++?**

- main()
- system()
- program()
- start()

**2. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на результирующем языке, называется:**

- интерпретатор
- сканер
- компилятор
- транслятор

**3. Цикл с постусловием?**

- while
- do while
- for

**4. Какими знаками заканчивается большинство строк кода в Си++?**

- ; (точка с запятой)
- , (запятая)
- : (двоеточие)
- . (точка)

**5. Какому зарезервированному слову программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?**

- default
- all
- contingency
- other

**6. Укажите правильное определение функции main в соответствии со спецификацией стандарта ANSI**

- void main(void)
- void main()
- int main(void)
- int main()

**7. Чему будет равна переменная a, после выполнения этого кода int a; for(a = 0; a < 10; a++) {}?**

- 1
- 10
- 9

**8. Какое значение, по умолчанию, возвращает программа операционной системе в случае успешного завершения?**

- -1
- 1
- 0
- Программа не возвращает значение.

**9. Какой оператор не допускает перехода от одного константного выражения к другому?**

- break;
- Stop;
- end;
- точка с запятой

**10. Чтобы подключить заголовочный файл в программу на C++, например iostream необходимо написать:**

- include #iostream,h;
- #include <> с iostream внутри скобок
- #include <>; с iostream.h внутри скобок
- include (iostreamh)

**11. Что будет напечатано?**

```
int main()
{
    for (int i = 0; i < 4; ++i)
    {
        switch (i)
        {
            case 0 : std::cout << "0";
```

```

0         case 1 : std::cout << "1"; continue;
          case 2 : std::cout << "2"; break;
          default : std::cout << "D"; break;
1         }
          std::cout << ".";
2         }
          return 0;
3         }
4
5

```

- 0.1.2.
- 01.2.D.
- 011.2.D
- Ошибка компиляции в строке 10
- 0112.D.

**12. Укажите объектно-ориентированный язык программирования**

- Все варианты ответов
- C++
- Eiffel
- Java

**13. Выберите правильный вариант объявления константной переменной в C++, где type - тип данных в C++ variable - имя переменной value - константное значение**

- const variable = value;
- const type variable = value;
- const type variable := value;

**14. Каков результат работы следующего фрагмента кода?**

```

int x = 0;

switch(x)
{

case 1: cout << "Один";

case 0: cout << "Ноль";

0         case 2: cout << "Привет мир";
1         }

```

2

- Привет мир
- НульПривет мир
- Один
- Нуль

**15. В приведённом коде измените или добавьте один символ чтобы код напечатал 20 звёздочек - \*.**

- ```
int i, N = 20;
for(i = 0; i < N; i--)
    printf("*");
```
- ```
int i, N = 20;
for(i = 19; i < N; i--)
    printf("*");
```
  - ```
int i, N = 20;
for(i = 20; i < N; i--)
    printf("*");
```
  - ```
int i, N = 20;
for(i = 0; i < N; N--)
    printf("*");
```
  - ```
int i, N = 40;
for(i = 0; i < N; i--)
    printf("*");
```

**16. Укажите правильное приведение типа данных!**

- a(char);
- (char)a;
- char:a;
- to(char, a);

**17. Какой из следующих операторов - оператор сравнения двух переменных?**

- ==
- :=
- equal
- =

**18. Какой из ниже перечисленных операторов, не является циклом в C++?**

- do while
- repeat until
- for
- while

**19. До каких пор будут выполняться операторы в теле цикла while (x < 100)?**

- Пока x меньше или равен ста
- Пока x строго меньше ста
- Пока x равен ста
- Пока x больше ста

**20. Цикл с предусловием?**

- while
- do while
- for

**Критерий оценки:**

Общая сумма баллов – 10 баллов.

За каждый правильный ответ – 0,2 балла.

## **Тема 5.**

**1. Класс - это:**

- любой тип данных, определяемый пользователем
- \* тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки
- структура, для которой в программе имеются функции работы с нею

**2. Членами класса могут быть**

- \* как переменные, так и функции, могут быть объявлены как private и как public
- только переменные, объявленные как private
- только функции, объявленные как private
- только переменные и функции, объявленные как private
- только переменные и функции, объявленные как public

**3. Что называется конструктором?**

- \* метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса
- метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)
- метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса
- метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса

**4. Объект - это**

- переменная, содержащая указатель на класс
- \* экземпляр класса
- класс, который содержит в себе данные и методы их обработки

**5. Отметьте правильные утверждения**

- \* конструкторы класса не наследуются
- конструкторов класса может быть несколько, их синтаксис определяется программистом

- \* конструкторов класса может быть несколько, но их синтаксис должен подчиняться правилам перегрузки функций

- конструктор возвращает указатель на объект

- \* конструктор не возвращает значение

### **6. Что называется деструктором?**

- метод, который уничтожает объект

- метод, который удаляет объект

- \* метод, который освобождает память, занимаемую объектом

- системная функция, которая освобождает память, занимаемую объектом

### **7. Выберите правильные утверждения**

- \* у конструктора могут быть параметры

- конструктор наследуется, но должен быть перегружен

- конструктор должен явно вызываться всегда перед объявлением объекта

- \* конструктор вызывается автоматически при объявлении объекта

- объявление каждого класса должно содержать свой конструктор

- \* если конструктор не создан, компилятор создаст его автоматически

### **8. Выберите правильные утверждения**

- деструктор - это метод класса, применяемый для удаления объекта

- \* деструктор - это метод класса, применяемый для освобождения памяти, занимаемой объектом

- деструктор - это отдельная функция головной программы, применяемая для освобождения памяти, занимаемой объектом

- \* деструктор не наследуется

- деструктор наследуется, но должен быть перегружен

### **9. Что называется наследованием?**

- \* это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы

- это механизм переопределения методов базового класса

- это механизм, посредством которого производный класс получает все поля базового класса

- это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить

### **10. Выберите правильное объявление производного класса**

- class MoreDetails:: Details;

- class MoreDetails: public class Details;

- \* class MoreDetails: public Details;

- class MoreDetails: class(Details);

### **11. Выберите правильные утверждения:**

- если элементы класса объявлены как private, то они доступны только наследникам класса, но не внешним функциям

- \* если элементы класса объявлены как `private`, то они недоступны ни наследникам класса, ни внешним функциям
- если элементы объявлены как `public`, то они доступны наследникам класса, но не внешним функциям
- \* если элементы объявлены как `public`, то они доступны и наследникам класса, и внешним функциям

## **12. Возможность и способ обращения производного класса к элементам базового определяется**

- ключами доступа: `private`, `public`, `protected` в теле производного класса
- только ключом доступа `protected` в заголовке объявления производного класса
- \* ключами доступа: `private`, `public`, `protected` в заголовке объявления производного класса
- ключами доступа: `private`, `public`, `protected` в теле базового класса

## **13. Выберите правильные соответствия между спецификатором базового класса, ключом доступа в объявлении производного класса и правами доступа производного класса к элементам базового**

- ключ доступа - `public`; в базовом классе: `private`; права доступа в производном классе - `protected`
- \* ключ доступа - любой; в базовом классе: `private`; права доступа в производном классе - нет прав
- \* ключ доступа - `protected` или `public` ; в базовом классе: `protected`; права доступа в производном классе - `protected`
- ключ доступа - `private`; в базовом классе: `public`; права доступа в производном классе - `public`
- \* ключ доступа – любой; в базовом классе: `public`; права доступа в производном классе – такие же, как ключ доступа

## **14. Дружественная функция - это**

- функция другого класса, среди аргументов которой есть элементы данного класса
- \* функция, объявленная в классе с атрибутом `friend`, но не являющаяся членом класса;
- функция, являющаяся членом класса и объявленная с атрибутом `friend`;
- функция, которая в другом классе объявлена как дружественная данному

## **15. Выберите правильные утверждения:**

- \* одна функция может быть дружественной нескольким классам
- дружественная функция не может быть обычной функцией, а только методом другого класса
- \* дружественная функция объявляется внутри класса, к элементам которого ей нужен доступ
- дружественная функция не может быть методом другого класса

## **16. Шаблон функции - это...**

- \* определение функции, в которой типу обрабатываемых данных присвоено условное обозначение
- прототип функции, в котором вместо имен параметров указан условный тип
- определение функции, в котором указаны возможные варианты типов обрабатываемых параметров
- определение функции, в котором в прототипе указан условный тип, а в определении указаны варианты типов обрабатываемых параметров

### **17. Выберите правильные утверждения:**

- \* по умолчанию члены класса имеют атрибут private
- по умолчанию члены класса имеют атрибут public;
- члены класса имеют доступ только к элементам public;
- \* элементы класса с атрибутом private доступны только членам класса

### **18. Переопределение операций имеет вид:**

- имя\_класса, ключевое слово operation, символ операции
- \* имя\_класса, ключевое слово operator, символ операции, в круглых скобках могут быть указаны аргументы
- имя\_класса, ключевое слово operator, список аргументов
- имя\_класса, два двоеточия, ключевое слово operator, символ операции

### **19. Для доступа к элементам объекта используются:**

- \* при обращении через имя объекта – точка, при обращении через указатель – операция «->»
- при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «точка»
- при обращении через имя объекта – точка, при обращении через указатель – два двоеточия
- при обращении через имя объекта – два двоеточия, при обращении через указатель – операция «->»

### **20. Полиморфизм – это:**

- \* средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов
- средство, позволяющее в одном классе использовать методы с одинаковыми именами;
- средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий
- средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов.

### **21. Полиморфизм реализован через механизмы:**

- \* перегрузки функций, виртуальных функций, шаблонов
- перегрузки функций, наследования методов, шаблонов;
- наследования методов, виртуальных функций, шаблонов
- перегрузки функций, наследования, виртуальных функций.

### **22. Виртуальными называются функции:**

- \* функции базового класса, которые могут быть переопределены в производном классе
- функции базового класса, которые не используются в производном классе;
- функции базового класса, которые не могут быть переопределены в базовом классе;
- функции производного класса, переопределенные относительно базового класса

**23. Выберите правильный вариант выделения динамической памяти под переменную X типа float:**

- \* float \*ptr = new float; X = \*ptr;
- float & ptr = new float; X = & ptr;
- float \* ptr = &X; X = new float;

**24. Полиморфизм в объектно-ориентированном программировании реализуется:**

- \* через механизмы перегрузки (функций и операций), виртуальные функции и шаблоны
- через механизмы перегрузки (функций и операций) и шаблоны;
- через виртуальные функции и шаблоны;
- через механизмы перегрузки (функций и операций) и виртуальные функции

**25. Дано определение класса**

```
class monstr {
int health, armo;
monstr(int he, int arm);
public:
monstr(int he=50, int arm=10);
int color;
}
```

**Критерий оценки:**

Общая сумма баллов – 10.

За каждый правильный ответ – 0,4 балла.

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
НОЯБРЬСКИЙ ИНСТИТУТ НЕФТИ И ГАЗА  
(ФИЛИАЛ) ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»  
(Филиал ТИУ в г. Ноябрьск)**

Кафедра Экономики, менеджмента и естественнонаучных дисциплин

**Перечень вопросов к экзамену  
по дисциплине «Программирование и алгоритмизация»**

1. Постановка задачи. Хорошо и плохо поставленные задачи.
2. Модель задачи. Адекватность модели. Тестирование
3. Состав языка.
4. Основные типы данных, объявление переменных.
5. Операции и выражения.
6. Базовые конструкции: оператор «выражение», операторы ветвления, операторы цикла, операторы передачи управления.
7. Функциональная декомпозиция.
8. Объявление и описание функций.
9. Глобальные и локальные переменные.
10. Передача параметров в функции по значению и по ссылке.
11. Программирование рекурсивных алгоритмов.
12. Концепция модульного программирования.
13. Реализация рекурсии.
14. Концепция типа.
15. Классификация типов данных.
16. Массивы: одномерные и многомерные.
17. Операции с массивами, передача массивов как параметров функций.
18. Алгоритмы сортировки и поиска в массивах.
19. Строка как массив символов.
20. Библиотечные функции обработки строк.
21. Применение файлов для ввода-вывода массивов.
22. Виды файлов.
23. Объявление файловой переменной.
24. Функции работы с файлами.
25. Структуры (struct/record): объявление, инициализация, доступ к полям.
26. Массивы структур. Иерархия структур.
27. Передача структур как параметров функций.
28. Описание класса.
29. Объявления переменных (объектов) типа данного класса.
30. Конструкторы и деструкторы класса.
31. Объектно-ориентированное программирование.
32. Реализация в языках программирования.

**Критерии оценки:****Шкала оценивания на экзамене**

| <b>Оценка</b>       | <b>Описание</b>                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| «удовлетворительно» | выставляется в случае, если обучающийся набрал от 61 до 75 баллов в соответствии с балльно-рейтинговой системой оценки, т.е. может сформулировать все основные понятия и определения по дисциплине                                                                                                                                                                                                                                                       |
| «хорошо»            | выставляется в случае, если обучающийся набрал от 76 до 90 баллов в соответствии с балльно-рейтинговой системой оценки, т.е. может сформулировать все основные понятия и определения по дисциплине и способен четко изложить ее суть, выводы, ответить на вопросы                                                                                                                                                                                        |
| «отлично»           | выставляется в случае, если обучающийся набрал от 91 до 100 баллов в соответствии с балльно-рейтинговой системой оценки, т.е. может сформулировать все основные понятия и определения по дисциплине. Кроме этого обучающийся, претендующий на отличную оценку, должен продемонстрировать аналитическое, нестандартное мышление, креативность и находчивость в ответах на дополнительные, усложненные вопросы преподавателя в рамках изучаемой дисциплины |