ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ИНКЛЮЗИВНОГО ВЫСШЕГО ОБРАЗОВАНИЯ
**«МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО ЭКОНОМИЧЕСКИЙ УНИВЕРСИТЕТ»**

КАФЕДРА ЦИФРОВЫХ ТЕХНОЛОГИЙ

УТВЕРЖДАЮ
Проректор по учебно-методической работе
_____ Е.С. Сахарчук
«27» 04 2022г.

# ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Алгоритмизация и программирование

наименование дисциплины

09.03.03 "Прикладная информатика"

шифр и наименование направления подготовки

Прикладная информатика в биоинформационных технологиях

направленность (профиль)

Москва 2022

Разработчик:

МГГЭУ, ассистент кафедры цифровых технологий
должность — место работы, занимаемая

_____ Печерский Д.К. 14.03 2022 г
подпись Ф.И.О. Дата

Фонд оценочных средств рассмотрен и одобрен на заседании кафедры цифровых технологий

(протокол № 4 от « 24 » 03 2022 г.)

на заседании Учебно-методического совета МГГЭУ

(протокол № 1 от « 27 » 04 2022 г.)

Согласовано:

Представитель работодателя
или объединения работодателей _____/ Демидов Л.Н./
к.т.н., доцент АО «Микропроцессорные системы»
(должность, место работы)
« 29 » 03 2022 г.

Начальник учебно-методического управления
_____И.Г. Дмитриева
« 27 » 04 2022 г.

Начальник методического отдела
_____Д.Е. Гапеенок
« 27 » 04 2022 г.

Декан факультета
_____ Е.В. Петрунина
« 27 » 04 2022 г.

**Содержание**

## 1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине «Алгоритмизация и программирование»

Оценочные средства составляются в соответствии с рабочей программой дисциплины и представляют собой совокупность контрольно-измерительных материалов (типовые задачи (задания), контрольные работы, тесты и др.), предназначенных для измерения уровня достижения обучающимися установленных результатов обучения.

Оценочные средства используются при проведении текущего контроля успеваемости и промежуточной аттестации.

Таблица 1 - Перечень компетенций, формируемых в процессе освоения дисциплины

| **Код компетенции** | **Наименование результата обучения** |
|---|---|
| ОПК-7 | Способен разрабатывать алгоритмы и программы, пригодные для практического применения. |
| | ОПК-7.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.<br>ОПК-7.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнеспроцессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.<br>ОПК-7.3. Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач. |

Конечными результатами освоения дисциплины являются сформированные когнитивные дескрипторы «знать», «уметь», «владеть», расписанные по отдельным компетенциям. Формирование дескрипторов происходит в течение всего семестра по этапам в рамках контактной работы, включающей различные виды занятий и самостоятельной работы, с применением различных форм и методов обучения (табл.2).

Таблица 2 - Формирование компетенций в процессе изучения дисциплины:

| Код компетенции | Уровень освоения компетенций | Индикаторы достижения компетенций | Вид учебных занятий[1], работы, формы и методы обучения, способствующие формированию и развитию компетенций[2] | Контролируемые разделы и темы дисциплины[3] | Оценочные средства, используемые для оценки уровня сформированности компетенции[4] |
|---|---|---|---|---|---|
| *ОПК-7* | | *Знает* | | | |
| | Недостаточный уровень | ОПК-7. Студент не способен самостоятельно выделять главные положения в изученном материале дисциплины. Не знает язык программирования, операционные системы и оболочки, современную программную среду разработки программного обеспечения -MS VisualStudio 2017. Показывает отсутствие знаний основ алгоритмизации и программирования, | Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации, подготовка и сдача зачета | 1. Разработка алгоритмов решения задач.<br>2. Основы программирования на языке С++.<br>3. Указатели.<br>4. Массивы и строки.<br>5. Функции<br>6. Структуры.<br>7. Объекты и классы.<br>8. Перегрузка операций.<br>9. Наследование.<br>10. Указатели. Управление памятью.<br>11. Виртуальные функции.<br>12. Потоки и файлы. | Текущий контроль – устный опрос, тестирование. |

[1] Лекционные занятия, практические занятия, лабораторные занятия, самостоятельная работа…

[2] Необходимо указать активные и интерактивные методы обучения (например, интерактивная лекция, работа в малых группах, методы мозгового штурма и т.д.), способствующие развитию у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств.

[3] Наименование темы (раздела) берется из рабочей программы дисциплины.

[4] Оценочное средство должно выбираться с учетом запланированных результатов освоения дисциплины, например:

«Знать» – собеседование, коллоквиум, тест…

«Уметь», «Владеть» – индивидуальный или групповой проект, кейс-задача, деловая (ролевая) игра, портфолио…

| | | | | | |
|---|---|---|---|---|---|
| | | разработки алгоритмов различной структуры, основы построения программ на языке высокого уровня С++. | | | |
| | Базовый уровень | ОПК-7.1. Студент усвоил основное содержание материала дисциплины, но имеет пробелы в усвоении материала. Имеет несистематизированные знания о языках программирования, операционные системы и оболочки, современные программные среды разработки программного обеспечения- MS VisualStudio 2017. Показывает слабое знание основ алгоритмизации и программирования, разработки алгоритмов различной структуры, основы построения программ на языке высокого уровня С++.. | Лекционные и практические занятия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации, подготовка и сдача зачета | 1. Разработка алгоритмов решения задач.<br>2. Основы программирования на языке С++.<br>3. Указатели.<br>4. Массивы и строки.<br>5. Функции<br>6. Структуры.<br>7. Объекты и классы.<br>8. Перегрузка операций.<br>9. Наследование.<br>10. Указатели. Управление памятью.<br>11. Виртуальные функции.<br>12. Потоки и файлы. | Текущий контроль – устный опрос, тестирование. |

| | | | | | |
|---|---|---|---|---|---|
| | Средний уровень | ОПК-7.1. Студент способен самостоятельно выделять главные положения в изученном материале.<br>Знает основы программирования, операционные системы и оболочки, современную программную среду разработки программного обеспечения. -MS VisualStudio 2017.<br>Показывает глубокое знание основ алгоритмизации и программирования, разработки алгоритмов различной структуры, основы построения программ на языке высокого уровня С++. | Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации, подготовка и сдача зачета | 1. Разработка алгоритмов решения задач.<br>2. Основы программирования на языке С++.<br>3. Указатели.<br>4. Массивы и строки.<br>5. Функции<br>6. Структуры.<br>7. Объекты и классы.<br>8. Перегрузка операций.<br>9. Наследование.<br>10. Указатели. Управление памятью.<br>11. Виртуальные функции.<br>12. Потоки и файлы. | Текущий контроль – устный опрос, тестирование. |

| | | | | | |
|---|---|---|---|---|---|
| | Высокий уровень | ОПК-7.1. Студент знает, понимает, выделяет главные положения в изученном материале и способен дать краткую характеристику основным идеям проработанного материала дисциплины. Знает основы программирования, операционные системы и оболочки, современную программную среду разработки программного обеспечения. -MS VisualStudio 2017. Показывает глубокое знание основ алгоритмизации и программирования, разработки алгоритмов различной структуры, основы построения программ на языке высокого уровня С++. | Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации, подготовка сдаче зачета | 1. Разработка алгоритмов решения задач.<br>2. Основы программирования на языке С++.<br>3. Указатели.<br>4. Массивы и строки.<br>5. Функции<br>6. Структуры.<br>7. Объекты и классы.<br>8. Перегрузка операций.<br>9. Наследование.<br>10. Указатели. Управление памятью.<br>11. Виртуальные функции.<br>12. Потоки и файлы. | Текущий – контроль опрос, устный тестирование |
| | | *Умеет* | | | |

| | | | | | |
|---|---|---|---|---|---|
| | Базовый уровень | ОПК-7.2. Студент умеет непоследовательно применять языки программирования и, современную программную среду разработки программного обеспечения. -MS VisualStudio 2017. Умеет разрабатывать алгоритмы прикладных задач различных классов различной структуры, на языке высокого уровня С++. | Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации, подготовка и сдача зачета | 1. Разработка алгоритмов решения задач.<br>2. Основы программирования на языке С++.<br>3. Указатели.<br>4. Массивы и строки.<br>5. Функции<br>6. Структуры.<br>7. Объекты и классы.<br>8. Перегрузка операций.<br>9. Наследование.<br>10. Указатели. Управление памятью.<br>11. Виртуальные функции.<br>12. Потоки и файлы. | Текущий контроль – устный опрос, тестирование. |
| | Средний уровень | ОПК-7.2. Студент умеет применять языки программирования и, современную программную среду разработки программного обеспечения. -MS VisualStudio 2017. Умеет разрабатывать алгоритмы прикладных задач различных классов структуры, на языке высокого уровня С++. | Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации, подготовка и сдача зачета | 1. Разработка алгоритмов решения задач.<br>2. Основы программирования на языке С++.<br>3. Указатели.<br>4. Массивы и строки.<br>5. Функции<br>6. Структуры.<br>7. Объекты и классы.<br>8. Перегрузка операций.<br>9. Наследование.<br>10. Указатели. Управление памятью.<br>11. Виртуальные функции.<br>12. Потоки и файлы. | Текущий контроль – устный опрос, тестирование. |

| | | | | | |
|---|---|---|---|---|---|
| | Высокий уровень | ОПК-7.2. Студент умеет применять языки программирования и, современную программную среду разработки программного обеспечения. -MS VisualStudio 2017. Умеет разрабатывать алгоритмы прикладных задач различных классов различной структуры, на языке высокого уровня С++. | Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации, подготовка и сдача зачета | 1. Разработка алгоритмов решения задач.<br>2. Основы программирования на языке С++.<br>3. Указатели.<br>4. Массивы и строки.<br>5. Функции<br>6. Структуры.<br>7. Объекты и классы.<br>8. Перегрузка операций.<br>9. Наследование.<br>10. Указатели. Управление памятью.<br>11. Виртуальные функции.<br>12. Потоки и файлы. | Текущий контроль – устный опрос, тестирование. |
| | | *Владеет* | | | |
| | Базовый уровень | ОПК-7.3. Студент владеет основными навыками программирования на языке высокого уровня С++, испытывает затруднения при отладке и тестировании прототипов программнотехнических задач в среде разработки программного обеспечения. -MS VisualStudio 2017. | Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации, подготовка и сдача зачета | 1. Разработка алгоритмов решения задач.<br>2. Основы программирования на языке С++.<br>3. Указатели.<br>4. Массивы и строки.<br>5. Функции<br>6. Структуры.<br>7. Объекты и классы.<br>8. Перегрузка операций.<br>9. Наследование.<br>10. Указатели. Управление памятью.<br>11. Виртуальные функции.<br>12. Потоки и файлы. | Текущий контроль – устный опрос, тестирование. |

| | | | | | |
|---|---|---|---|---|---|
| | Средний уровень | ОПК-7.3. Студент владеет навыками программирования на языке высокого уровня С++. Допускает ошибки при отладке и тестировании задач в среде разработки программного обеспечения. -MS VisualStudio 2017. | Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации, подготовка и сдача зачета | 1. Разработка алгоритмов решения задач.<br>2. Основы программирования на языке С++.<br>3. Указатели.<br>4. Массивы и строки.<br>5. Функции<br>6. Структуры.<br>7. Объекты и классы.<br>8. Перегрузка операций.<br>9. Наследование.<br>10. Указатели. Управление памятью.<br>11. Виртуальные функции.<br>12. Потоки и файлы. | Текущий контроль – устный опрос, тестирование. |
| | Высокий уровень | ОПК-7.3. Студент владеет навыками программирования на языке высокого уровня С++, отладки и тестирования задач в среде разработки программного обеспечения. -MS VisualStudio 2017. | Лекционные и практические занятия, работа в малых группах, интерактивная лекция, дискуссия, самостоятельная работа обучающихся, подготовка и сдача промежуточной аттестации, подготовка и сдача зачета | 1. Разработка алгоритмов решения задач.<br>2. Основы программирования на языке С++.<br>3. Указатели.<br>4. Массивы и строки.<br>5. Функции<br>6. Структуры.<br>7. Объекты и классы.<br>8. Перегрузка операций.<br>9. Наследование.<br>10. Указатели. Управление памятью.<br>11. Виртуальные функции.<br>12. Потоки и файлы. | Текущий контроль – устный опрос, тестирование. |

## 2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ[1]

Таблица 3

[1] Указываются оценочные средства, применяемые в ходе реализации рабочей программы данной дисциплины.

| № | Наименование оценочного средства | Характеристика оценочного средства | Представление оценочного средства в ФОС |
|---|---|---|---|
| 1 | Устный опрос | Средство контроля усвоения учебного материала темы, раздела или разделов дисциплины, организованное как учебное занятие в виде собеседования преподавателя с обучающимися. | Вопросы по темам/разделам дисциплины |
| 2 | Тест | Средство, позволяющее оценить уровень знаний обучающегося путем выбора им одного из нескольких вариантов ответов на поставленный вопрос. Возможно использование тестовых вопросов, предусматривающих ввод обучающимся короткого и однозначного ответа на поставленный вопрос. | Тестовые задания |

## 3. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ

Оценивание результатов обучения по дисциплине «Алгоритмизация и программирование» осуществляется в соответствии с Положением о текущем контроле успеваемости и промежуточной аттестации обучающихся.

Предусмотрены следующие виды контроля: текущий контроль (осуществление контроля всех видов аудиторной и внеаудиторной деятельности обучающегося с целью получения первичной информации о ходе усвоения отдельных элементов содержания дисциплины) и промежуточная аттестация (оценивается уровень и качество подготовки по дисциплине в целом). Показатели и критерии оценивания компетенций, формируемых в процессе освоения данной дисциплины, описаны в табл. 4.

Таблица 4.

| Код компетенции | Уровень освоения компетенции | Индикаторы достижения компетенции | Критерии оценивания результатов обучения |
|---|---|---|---|
| *ОПК-7* | | Знает | |
| | Недостаточный уровень Оценка «незачтено», «неудовлетворительно» | *ОПК-7.1.* | *Не знает значительной части материала курса, не способен самостоятельно выделять главные положения в изученном материале дисциплины* |
| | Базовый уровень Оценка, «зачтено», «удовлетворительно» | *ОПК-7.1.* | *Знает не менее 50 % основного материала курса, однако испытывает затруднения в его применении* |
| | Средний уровень Оценка «зачтено», «хорошо» | *ОПК-7.1.* | *Знает основную часть материала курса, способен применить изученный материал на практике, испытывает незначительные затруднения в решении задач* |
| | Высокий уровень Оценка «зачтено», «отлично» | *ОПК-7.1.* | *Показывает глубокое знание и понимание материала, способен применить изученный материал на практике* |
| | | Умеет | |
| | Базовый уровень | *ОПК-7.2.* | *Умеет воспроизвести не менее 50 % основного материала курса, однако испытывает затруднения при решении практических задач* |
| | Средний уровень | *ОПК-7.2.* | *Умеет решать стандартные профессиональные задачи с применением полученных знаний, испытывает незначительные затруднения в решении задач* |
| | Высокий уровень | *ОПК-7.2.* | *Умеет решать стандартные профессиональные задачи с применением полученных знаний, показывает глубокое знание и понимание материала, способен решить задачу при изменении формулировки* |
| | | Владеет | |
| | Базовый уровень | *ОПК-7.3.* | *Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности, усвоил основное содержание материала дисциплины, но имеет пробелы в усвоении материала. Имеет несистематизированные знания* |

| | | | |
|---|---|---|---|
| | | | *основных разделов дисциплины.* |
| | Средний уровень | *ОПК-7.3.* | *Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности, способен самостоятельно выделять главные положения в изученном материале. Испытывает незначительные затруднения в решении задач.* |
| | Высокий уровень | *ОПК-7.3.* | *Свободно владеет навыками теоретического и экспериментального исследования, показывает глубокое знание и понимание изученного материала* |

## 4. Методические материалы, определяющие процедуры оценивания результатов обучения Задания

**в форме устного опроса:**

Устный опрос используется для текущего контроля успеваемости обучающихся по дисциплине в качестве проверки результатов освоения терминологии. Каждому студенту выдается свой собственный, узко сформулированный вопрос. Ответ должен быть четким и кратким, содержащим все основные характеристики описываемого понятия, института, категории.

**Задания в форме тестирования**

Тест представляет собой контрольное мероприятие по учебному материалу каждой темы (раздела) дисциплины, состоящее в выполнении обучающимся системы стандартизированных заданий, которая позволяет автоматизировать процедуру измерения уровня знаний и умений обучающегося.

Тестирование является средством текущего контроля успеваемости обучающихся по дисциплине и может включать в себя следующие типы заданий: задание с единственным выбором ответа из предложенных вариантов, задание на определение верных и неверных суждений; задание с множественным выбором ответов.

В каждом задании необходимо выбрать все правильные ответы.

## 5. Материалы для проведения текущего контроля и промежуточной аттестации

**Задания в форме устного опроса**

**Семестр 1**

1. Понятие алгоритма.
2. Структуры алгоритмов.
3. Алгоритмы линейной структуры.
4. Алгоритмы разветвляющейся структуры.
5. Алгоритмы с циклической структурой.
6. Вычисление суммы членов бесконечного ряда.
7. Вычисление полинома.
8. Нахождение наименьшего и наибольшего значений.
9. Алгоритм со структурой вложенных циклов.
10. Типы данных.
11. Литералы.
12. Переменные.
13. Выражения и операции.
14. Арифметические операции.
15. Логические операции и операции сравнения.
16. Линейная программа.
17. Подключение библиотек.
18. Потоковый ввод и вывод.
19. Математические операции и функции.
20. Разветвляющаяся программа.
21. Оператор IF.
22. Оператор SWITCH.
23. Программа с циклической структурой.
24. Оператор FOR.
25. Операторы DO и WHILE.
26. Указатель (pointer) в C++.
27. Оператор получения адреса.
28. Адрес переменной.  29. Переменная-указатель.

## Семестр 2

1. Определение массивов.
2. Инициализация массивов.
3. Одномерные массивы.
4. Многомерные массивы.
5. Массив и указатель.
6. Динамический массив.
7. Массивы объектов.
8. Массивы строк
9. Массив типа CHAR.
10. Тип данных STRING.
11. Аргументы и тип функции.
12. Прототип функции.
13. Простые функции.
14. Передача аргументов в функцию.
15. Перегруженные функции.

16. Рекурсия.
17. Встраиваемые функции.
18. Ссылки на аргументы.
19. Передача массивов как аргументов.
20. Область видимости и время жизни переменных.
21. Класс памяти.
22. Локальные и глобальные переменные.
23. Определения структур.
24. Доступ к полям структуры.
25. Простая структура.
26. Вложенные структуры. 27. Перечисления.

## Семестр 3

1. Объекты и классы
2. Простой класс.
3. Объекты программы и объекты реального мира.
4. Конструкторы.
5. Структуры.
6. Классы, объекты и память.
7. Перегрузка операций.
8. Перегрузка унарных операций.
9. Перегрузка бинарных операций.
10. Преобразование типов.
11. Наследование.
12. Базовый класс.
13. Производный класс.
14. Конструкторы производного класса. 15. Иерархия классов.

## Семестр 4

1. Адреса и указатели.
2. Управление памятью
3. Связный список.
4. Указатели на объекты и указатели.
5. Виртуальные функции.
6. Дружественные функции.
7. Статические функции.
8. Динамическая информация о типах.
9. Потоки и файлы.
10. Потоковые классы.
11. Потоковый ввод/вывод.
12. Ошибки потока.
13. Указатели файлов.
14. Файловый ввод/вывод. 15. Обработка ошибок ввода/вывода.

Контролируемые компетенции: ОПК-7.
*Оценка компетенций осуществляется в соответствии с таблицей 4.*

## Тестовые задания

## Семестр 1

1. Укажите оператор выбора в языке С ++. CASE choice switch … case default

2. Для чего предназначен оператор continue в языке С ++?
   Пропускает остаток тела цикла и переходит к следующей итерации.
   Пропускает цикл и переходит к следующему оператору в теле программы.
   Определяет условие продолжения цикла.
   Продолжает выполнение текущей итерации цикла.

3. Значение переменной number не лежит между 3 и 6. Укажите правильный вариант записи данного утверждения на языке С ++.
   number> 3 && number <6 !(Number
   <3 && number <6) !(Number <6 ||
   number> 3)
   number <3 || number> 6

4. Что выполняет операция ++ в языке С ++? Уменьшает значение операнда на единицу.

Уменьшает значение операнда на два.
Увеличивает значение операнда на два. Увеличивает
значение операнда на единицу.

5. В программе на языке С ++ есть два объявления переменных int qwerty; int QWERTY; Какое из утверждений верно?
   Такие имена переменных недопустимы.
   Объявления правильные.
   Такие объявления недопустимы, потому что мы пытаемся создать две переменные с одинаковыми идентификаторами.
   Переменные описываются не по такому принципу.

6. Укажите запись экранированного символа языке С ++.
   ‘F’ «\022»
   ‘Ю’
   ‘\t’

7. Зачем в С ++ используют оператор return?
   Чтобы задержать работу программы.
   Функция, в которой он содержится, завершает свое выполнение и управление возвращается в то место программы, из которого вызывалась данная функция.

Чтобы организовать цикл. Чтобы ввести в
программу новые значения.

8. В программе на языке С ++ объявлены такие переменные int x, y; Выражение позволяет вычислить остаток от деления этих переменных?
x % y x
div y x
mod y
x / y

9. Что выведет следующая программа? #include <iostream.h>

```
int main() {
int 1_i ;
    for( 1_i = 0; 1_i < 9;
1_i++)      cout << 1_i +1;
return 1;
}
```

цифры от 0 до 8
программа не будет построена из-за ошибок цифры
от 1 до 9

10. Для того чтобы вывести символ новой строки, надо:
воспользоваться специальным манипулятором endl
при выводе строки символов перевод строки добавляется автоматически закончить оператор точкой с запятой

11. Функция вычисляет произведение двух чисел. Исходные данные вводятся с клавиатуры. Какие проверки целесообразно ввести в программе?
проверка, что исходные данные являются числами и эти числа больше нуля проверка, что исходные данные являются числами проверка исходных данных на равенство нулю
проверки не нужны, все возможные ошибки отловит компилятор

12. Какие компоненты не входят в интегрированную среду программирования
компилятор текстовый редактор отладчик
переводчик

13. Укажите все ключевые слова в приведенном примере int calc(int a, int b, bool f)

```
  {         if
(f==1)
return a+b;
else
      return a*b;
```

}
int, bool, if, else, return int, calc,
bool, return, if, else int, if,
else, return

14. В чем различие использования следующих выражений #include <…> и #include «…» в различии использования заголовочных и исходных файлов нет различий
различие заключается в методе поиска препроцессором включаемого файла

15. Чему будет равен результат вычисления выражения: int d=5; bool b = true, c; c = (!b||(d>3));
true
Ошибка компилятора
false

16. Если после выражения стоит точка с запятой, то выражение вычисляется только если первой стоит операция присваивания это оператор-выражение, действие которого заключается в вычислении выражения выражение вычисляется, а его значение запоминается в специальной переменной, которую можно использовать в следующем операторе

17. Отметьте истинные высказывания:
переменная инициализируется, потом объявляется
переменная объявляется, потом изменяется
переменная объявляется, потом инициализируется и изменяется

18. В каком случае программа выведет строку на консоль

cout < "Hello, world!" < endl; cout
<< "Hello, world!" << endl; cout
>> "Hello, world!" >> endl;

19. Какой результат будет у следующего выражения? int m = 1, n=2; double A = (double)m/n; cout << A;

0
1
ошибка компиляции
0.5

20. Процесс компиляции программы приводит программы к единообразному внешнему виду переводит исходный текст в исполняемый файл для языка Си++ необязателен

21. Чему равно значение выражения
!((1 || 0) && 0) ?

ошибка компиляции
0
1

22. Если int n=3, какой будет результат выполнения кода? switch(n) { case 2: cout << "aaa"; break; case 3: cout << "ббб"; break;
default: cout << "ввв"; break; }

ошибка компилятора
ввв ааа
ббб
неопределенное поведение

23. Что будет выведено на экран в результате выполнения кода? int a=3; if (a>1) cout << "1"; else if(a>2) cout << "2"; else if(a>3)
cout << "3";

1
12
123

24. В каком случае компилятор не выдаст ошибку:

int int iCeloe; bool
LD1LW;
const float fL = 32; float e23 = 1; fL = e23;

25. Укажите правильный идентификатор для имени переменной: FA_Ф12 int
2a
_ri18

26. Если есть два объявления int qwerty; int QWERTY; какое из утверждений верно

такие имена переменных недопустимы объявления
правильные
такие объявления недопустимы, так как мы пытаемся создать две одинаковые переменные

27. Битовой операцией (операцией с одним операндом) является
&
||
+
!=
=

28. Чему равен результат вычисления выражения b – x * 3 + b при x = 12 и b = 8 ?

-4
-20
-124

28. Что является результатом компоновки программы?

набор заголовочных файлов с определением в них всех используемых функций исполняемый файл или библиотека заголовочный файл

30. Если имеется объявление «char ch1=’a’,ch2=’b’,ch3=’c’;», допустима ли запись «ch1=ch2+ch3»?

нет
да

31. Чему равен результат вычисления выражения x + 3 * b + x при x = 12 и b = 8 ?

132
300
48

32. Если i = 5, какой будет результат вывода

```
do
{
  cout << (++i)++ << " ";
}
while ( i>=5 && i < 8 ) ;
```

6 7 8
6 8
6 7
6

33. Отметьте правильное объявление переменных:

float; float = y; char
float = 53.5; int x;
int y; int X;

34. Какой из перечисленных типов является встроенным типом языка C++?

float
boolean
real
integer

35. Исходя из данного кода какое высказывание верно?

```
int main()
  {     int
a,b,c,d;
a=1;     b=2;
c=a+b+p;
cout << p;
    ...
  }
```

код не верен, потому что переменным c и d не присвоены значения код не верен, потому что переменная p не объявлена
код верен, потому что по умолчанию все переменные имеют целочисленный тип

## Семестр 2

1. Какая из следующих функций считывает 100 символов из входного потока в строку x?

readline(x, 100, 'n');

cin.getline(x, 100, 'n');
read(x);

cin.getline(100, x, 'n');

2. Строковый типы данных в C++ строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением (‘/0’). строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением (‘’).
строки в C++ представляются как массивы элементов типа char, заканчивающиеся терминатором строки - символом с нулевым значением '0'.

3. Код, указанный ниже объявляет массив ссылок. Правда это или нет?

```
int main()
{
 int& x[50];

 return 0;
}
```

да нет

4. Будет ли напечатано сообщение «не равны»?

```
struct Foo
{
};

struct Bar
{
};

int main(int argc, char** argv)
{
  Foo* f = new Foo;
 Bar* b = new Bar;

  if ( f == b )
    std::cout << "равны" << std::endl;
else
    std::cout << "не равны" << std::endl;

  return 0;
}
```

да нет

5. Программа напечатает строку Я программист или нет?

```
struct
Foo {
int x;
int y; };

int main(int argc, char** argv)
{
  Foo f;


  if ( &f.x < &f.y )
  {
    std::cout << "Я программист" << std::endl;
  }

  return 0;
}
```

да нет
некорректное определение

6. Укажите правильное объявление указателя в C++

int &x;

int x;

int *x;

ptr x;

7. Укажите правильное объявление массива array an array[10]; int anarray[10]; int anarray;
anarray{10};

8. Строка Привет Мир будет показана на экране или нет?

```
int main(int argc, char** argv)
{
  int array[33];

  if ( &array[4] < &array[23] )
  {
    std::cout << "Привет мир" << std::endl;
  }

  return 0;
}
```

да
синтаксическая ошибка нет

9. Что такое ссылка?
используется для переименования объектов нет
правильного ответа
ссылка является псевдонимом для объекта оператор

10. Укажите зарезервированное ключевое слово для высвобождения выделенной памяти clear
remove delete
free

11. Какой заголовочный файл необходимо подключить, чтобы вызвать функцию isalpha()?

cstring
conio.h cctype
ifstream.h

12. Корректное выделение памяти

int a = new sizeof(int * 20);

int *a = new int[20];

int *a = new sizeof(int * 20);

int a = new int[20];

int a = new int(20);

int *a = new 20;

13. Укажите статическую строку

'Статическая строка'
"Статическая строка" char
string[100];

14. Какая из следующих функций добавляет одну строку в конец другой?

stradd();

stringadd ();

strcat ();

Append ();

15. В какой из следующих записей используется операция взятия адреса?

&a;

address(a);

*a;
 a
;

16. В какой из следующих строк выполняется обращение к седьмому элементу массива, размер массива равен 10?

mas; mas(7);
mas[6];
mas[7];

17. В какой из следующих записей используется операция разыменования?

&a;
 a
;

address(a);

*a;

18. Какая из следующих записей возвращает значение переменной a?

*a;

&a;

val(a);
a ;

19. Массив - это ...

Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие общий адрес. Доступ к отдельным элементам массива осуществляется по адресу и индексу
Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие имя. Доступ к отдельным элементам массива осуществляется по имени массива и индексу
Массив - это упорядоченные в памяти элементы одного и того же типа, имеющие имя. Доступ к отдельным элементам массива осуществляется по имени массива и адресу

20. Какой стандартный код используется для Символьных данных типа char в C++?

Код ASCI
Код ASCII
Код UTF-8
Код cp-1251

21. Укажите строку, которая возвращает адрес первого элемента в массиве arr?

&arr; arr;
arr[0];
arr[1];

22. Как правильно высвободить память, после выполнения этого кода?

```
char *a; a = new char[20];
```

delete a[]; delete
a;
delete [] a;

23. После выполнения ряда операций с указателем, что будет выведено на экран?

```
int main(int argc, char** argv)
{
  // предположим, int занимает 4 байта
std::cout << sizeof(int) << std::endl;

  int *x = new int;

 // предположим адрес равен 0x60450000
  std::cout << x << std::endl;

  std::cout << x + 3 << std::endl;

  return 0;
}
```

0x6045000C 0x60450003
некорректное определение
нельзя заранее сказать, каково будет значение адреса 0x60450000

24. Словосочетание "Hello world!" может быть сохранено в символьном массиве размером n элементов. Укажите чему равно n?

11
13

12
10

25. Укажите корректное определение строковой переменной

string mystr[20];

string mystr;

string[20] mystr;

char mystr[20];

26. Каким символом завершается Си-строка?

'\0'
''
'.'
'\n'

27. Какое значение будет напечатано, в результате выполнения следующего кода? #include <iostream>

```
int main()
{
  int sum = 0;

  int array[3][] = {{0, 1, 2}, {3, 4, 5}, {6, 7, 8}};

  for (int i = 0; i < 3 ; ++i)
  {     for (int j = 2; j < 3 ;
j++)

    {
      sum += array[i][j];
    }
  }

  std::cout << sum << std::endl;

  return 0;
}</iostream>
```

21

9 15 синтаксическая
ошибка

28. Объявлена переменная char a; Какое из следующих выражений не верно?

a = "3";

a = 3;

a = '3';

29. Укажите зарезервированное ключевое слово для динамического выделения памяти

new
create malloc
value

30. Какой порядковый номер последнего элемента массива, размер массива 19?

18 порядковый номер определяется
программистом 19

31. Какая из следующих функций сравнивает две строки?

stringcompare();

strcmp();

cmp();

compare();

32. В каком из вариантов ответов объявлен двумерный массив?

int array[20, 20];
char array[20]; int
anarray[20][20];
array anarray[20][20];

33. Допустим, у нас есть код char arr[8];
cin >> arr;

И в массив arr мы попытались записать следующий набор символов Hello World. Что в действительности будет содержать массив arr?

Hello W
Другой ответ
Hello Wo
Hello World
Hello

## Семестр 3

1. Можно ли перегрузить функцию main()?

да нет

2. Какая из переменных хранит количество аргументов, передаваемых в программу?

count
arglen
argc argv

3. Что будет напечатано на экране, после выполнения этого кода?

```
#include <iostream>

int foo(int x, int y)
{
  return x+y;
}

double foo(double x, double y)
{
  return x+y;
}

int main(int argc, char** argv)
{
  double (*ptr)(int, int);

  ptr = foo;

  std::cout << ptr(3,8) << std::endl;

  return 0;
}
```

ошибка компиляции 11
8
3

4. Что будет напечатано на экране, после выполнения этого кода? #include <ostream>

```
int foo(int y);
int foo(int x)
{
  return x+1;
}

int main(int argc, char** argv)
{   int x =
3;   int y
= 6;

  std::cout << foo(x) << std::endl;

  return 0;
}
```

3
9
4
ошибка компиляции

5. Укажите тип возвращаемого значения следующей функции int func(char x, float v, double t);

double
float
int char

6. Какой тип данных имеет переменная ARGV?

это не переменная
char **
int char
*

7. Какие из следующих функций являются встроенными?

inline void foo() {}

void foo() inline {}

inline:void foo() {}

нет правильного ответа

8. Какая строка содержит зарезервированные слова языка программирования C++?

   if, else, for, while do, switch, continue, break char, int,
   float, doubled, short, long, unsigned, signed defaulted,
   goto, return, extern, private, public, protected sizeof,
   const, typedef, static, voided, enum, struct, union

9. Для чего используются встроенные функции?

   Для удаления ненужных функций
   Чтобы уменьшить размер программы
   Для упрощения файла с исходным кодом
   Для увеличения скорости работы программы

10. Что значит ключевое слово inline?

    Сообщает компилятору использовать функцию только в пределах одного файла с исходным кодом
    все вызовы встроенных функций заменяются кодом этой функции нет правильного ответа
    все вызовы встроенных функции заменяются кодом этой функции

11. Какие из следующих утверждений верны?

    компилятор может проигнорировать объявление встроенной функции.
    встроенные функции не могут возвращать значения.
    встроенные функции должны возвращать значение. встроенные функции не должны содержать более 10 строк кода.

12. Что из нижеперечисленного не является прототипом функции?

    double funct(char x)

    char x();

    int funct(char x, char y);

    void funct();

13. Что будет напечатано на экране, после выполнения этого кода?

```
#include <iostream>

int foo(int x, int y)
{
 return x+y;
}

int foo(const int x, const int y)
{   return
x+y+1;

}

int main(int argc, char** argv)
{
 const int x = 3;
const int y = 2;

 std::cout << foo(x,y) << std::endl;

 return 0;
}
```

6
3
5
ошибка компиляции

14. Укажите правильный вызов функции, предполагается, что функция была объявлена ранее.

```
funct; int
funct();;
funct x, y;
funct();
```

15. Что такое ARGV[0]?

ARGV[0] нигде не используется
первый аргумент, который передается в программу из командной строки массив

16. В каком порядке эти два параметра, объявлены в функции main?
Параметры: argc и argv

они не объявлены в main Существует только
один аргумент количество аргументов, затем
массив аргументов массив аргументов, затем
количество элементов

17. Какое значение будет содержать переменная y?

```
const int x = 5;
int main(int argc, char** argv)
{
   int x[x];

   int y = sizeof(x) / sizeof(int);


   return 0;
}
```

5
20

18. Будет ли работать следующий код?

```
int x = 5;

template <typename T>
class x
{
   T member;
};

int main(int argc, char** argv)
{
   class x<int> y;

   return 0;
}
```

да нет

19. Каков будет результат выполнения следующего кода?

```
int f(int a)
{ return
++a; }
int f(unsigned int a)
{

return --a; }
cout << f(5);
```

ошибка компиляции
6
5
4

20. Какое значение будет содержать локальная переменная x, в конце main?

```
int x = 5;
int main(int argc, char** argv)
{     int x =
x;
   return 0;
}
```

неопределенное
0
5

21. Выберите правильное (полное) определение функции.

-
```
void funct(int)
{
cout << "Hello"
}
```

-
```
int funct(int x)
{
return x = x + 1;
}
```

-
```
int funct();

void funct(x)
{
```

- cout << "Hello"
  }

22. Можно ли гарантировать, что объявленная встроенная функция действительно является встроенной?

можно с уверенностью гарантировать, что объявленная вами функция как встроенная, действительно будет встроенной
гарантировать невозможно, в каждом индивидуальном случае бывает по-разному

23. Возможна ли такая ситуация?

```
int x = 5;

class x
{
};

int main(int argc, char** argv)
{
   class x y;

   return 0;
}
```

нет
да

## Семестр 4

**1. Правильное определение структуры в C++**

```
struct a_struct {int a;};

struct {int a;}

struct a_struct {int a;}

struct a_struct int a;
```

**2. Какой из следующих классов обрабатывает процесс записи в файл?**

ofstream другое
input_file

ifstream

**3. Какой заголовочный файл C++ содержит инструкции файлового ввода/вывода?**

fstream infstream
infstream
iostream

**4. Как организовать запись в файл?**

a_file.printf("запись");

a_file="запись";

a_file.out("запись");

a_file << "запись";

**5. Какие из перечисленных констант определяют режим открытия файлов в C++?**

ios_base::create ios_base::application

ios_base::noreplace
ios_base::trunc

**6. Укажите правильный доступ к переменной структуры**

b-var; b.var;
b->var;
b>var;

**7. Что означает константа ios_base::ate, передаваемая в конструктор, в качестве аргумента?**

Открыть файл только для чтения.
Открыть файл, предварительно создав его.
Открыть файл, не создавая его. При открытии
переместить указатель в конец файла.

**8. Какое значение будет напечатано?**

```
#include <iostream>

const int SIZE = 5;
```

```
struct tester
{
   int array[::SIZE];

   enum
   {
      SIZE = 3
   };

   void size()
   {
      std::cout << sizeof(array) / sizeof(int);
   }
};

int main(int argc, char** argv)
{
   tester t;
   t.size();
return 0;
}
```

3 заранее
неизвестно 5

**9. Какое значение будет напечатано?**

```
#include <iostream>

const int x = 12;

int main(int argc, char** argv)
{
   enum dog
   {
x = x,
y
   };

   std::cout << y << std::endl;

   return 0;
}
```

13
12
Неопределенное

**10. В каком из следующих вариантов ответов выполнен корректный доступ к переменной структуры, причём структура объявлена через указатель?**

b-var; b.var;
b->var;
b>var;

**11. При определении структуры необходимо использовать следующее ключевое слово**

record object
structure
struct

**12. Правильное объявление переменной, типа структуры foo**

foo var; int
foo; foo;
struct foo;

**13. Правильная конструкция определения класса в C++**

Class < имя класса >
{
Private:

- <список объявлений данных-элементов и функций-элементов, скрытых от доступ

Public:
<список прототипов функций-элементов общедоступного интерфейса>
};

Class = < имя класса >
{
Private

- <список объявлений данных-элементов и функций-элементов, скрытых от досту

Public
<список прототипов функций-элементов общедоступного интерфейса>};

class < имя класса >
{
private:

- <список объявлений данных-элементов и функций-элементов, скрытых от доступ

public:

<список прототипов функций-элементов общедоступного интерфейс >
};

**14. Какое значение должен возвращать деструктор?**

код состояния о правильном удалении класса
деструкторы не возвращают значение объект
класса.
указатель на класс

**15. Может ли деструктор иметь аргументы?**

да нет

**16. Понятие this в классе**

Указатель this является скрытым аргументом метода, превращает функцию в область памяти только для чтения.
Объект this является аргументом метода другого класса, существует во всех методах и указывает на адрес: this -><объект>
Указатель this является скрытым аргументом метода, существует во всех методах объекта и указывает на его (объект) адрес: this ->< объект >

**17. Назовите принципы объектно-ориентированного представления программных систем**

абстрагирование; инкапсуляция; модульность; абстрагирование; модульность; иерархическая организация абстрагирование; инкапсуляция; модульность; иерархическая организация

**18. Что такое деструктор?**

Деструктор - это специальная функция-элемент, которая должна отслеживать данные в экземпляре класса в процессе работы
Деструктор - это функция, которая должна открывать динамическую область для экземпляра класса
Деструктор - это специальная функция-элемент, которая должна уничтожать экземпляр класса после завершения его работы

**19. Назовите преимущества классов**

все указанные ответы
удобный способ моделирования объектов реального мира
инкапсуляция данных удобство повторного
использования кода

**20. Какого спецификатора доступа в классах нет?**

protected
public hidden
private

**21. Что такое объект?**

Объект - это конкретное представление абстракции с характеристиками – модификатором, селектором, итератором.
Объекты это инструмент борьбы для борьбы со сложностью различных систем реальных сущностей с характеристиками – агрегация, зависимость, конкретизация.
Объект - это конкретное представление абстракции, которое обладает индивидуальностью, состоянием и поведением.

**22. Что такое класс?**

Класс - это копия характеристик объекта , которые разделяют одинаковые свойства, операции, отношения семантику (смысл).
Класс - это абстракция существенных характеристик объекта или это описание множества объектов, которые разделяют одинаковые свойства, операции, отношения семантику (смысл)
Класс - это абстракция существенных характеристик системы или это описание множества систем, которые имеют свойства с одним смыслом

**23. Укажите корректное объявление класса**

public class A { }

object A { int x; };

class B { }

class A { int x; };

**24. Какие функции есть у любого класса?**

конструктор нет
таких
деструктор
конструктор и деструктор

**25. Основные типы наследования в классах**

Открытое наследование классов позволяет выполнить образование производного класса и объекта. Закрытое наследование классов позволяет выполнить

образование только производного класса. Множественное наследование - если у производного класса имеется несколько закрытых классов
Открытое наследование классов позволяет выполнить образование производного класса и объекта. Закрытое наследование классов позволяет выполнить образование только производного объекта. Множественное наследование - если у производного класса имеется несколько базовых классов
Открытое наследование классов позволяет выполнить образование производного класса и объекта. Закрытое наследование классов позволяет выполнить образование только производного класса. Множественное наследование - если у производного класса имеется несколько базовых классов

**26. Какая функция выполняет начальную инициализацию данных в классе?**

деструктор нет
правильного ответа
конструктор

Контролируемые компетенции: ОПК-7.
*Оценка компетенций осуществляется в соответствии с таблицей 4.*

## **Темы курсовых работ** Не

предусмотрено **Вопросы**

## **к зачету с оценкой и к**

## **экзамену**

**Первый семестр**

1. Структуры алгоритмов.
2. Алгоритмы линейной структуры.
3. Алгоритмы разветвляющейся структуры.
4. Алгоритмы циклической структуры.
5. Вычисления в цикле с несколькими одновременно изменяющимися параметрами.
6. Типы данных.
7. Литералы. Переменные.
8. Выражения и операции.
9. Арифметические операции.
10. Логические операции и операции сравнения.
11. Линейная программа.
12. Подключение библиотек.
13. Потоковый ввод и вывод.

14. Математические операции и функции.
15. Разветвляющаяся программа.
16. Оператор IF.
17. Оператор SWITCH.
18. Программа с циклической структурой.
19. Оператор FOR.
20. Операторы DO и WHILE.
21. Указатель (pointer) в C++.
22. Оператор получения адреса.
23. Адрес переменной.
24. Переменная-указатель.

**Четвертый семестр**

1. Структуры алгоритмов. Алгоритмы линейной структуры.
2. Алгоритмы разветвляющейся структуры. Алгоритмы циклической структуры.
3. Вычисления в цикле с несколькими одновременно изменяющимися параметрами.
4. Типы данных. Литералы. Переменные.
5. Выражения и операции. Арифметические операции. Логические операции и операции сравнения.
6. Линейная программа.
7. Подключение библиотек.
8. Потоковый ввод и вывод.
9. Математические операции и функции.
10. Разветвляющаяся программа.
11. Оператор IF. Оператор SWITCH.
12. Программа с циклической структурой.
13. Оператор FOR. Операторы DO и WHILE.
14. Указатель (pointer) в C++. Оператор получения адреса.
15. Адрес переменной.
16. Переменная-указатель.
17. Адреса и указатели.
18. Операции получения адреса.
19. Указатели- константы и указатели-переменные.
20. Определение массивов. Многомерные массивы.
21. Массивы объектов. Массивы строк.
22. Простые функции. Передача аргументов в функцию.
23. Ссылки на аргументы. Область видимости и время жизни переменных.
24. Класс памяти.
25. Структуры. Локальные и глобальные структур.
26. Доступ к полям структуры.
27. Вложенные структуры.
28. Перечисления
29. Простой класс.

30. Объекты программы и объекты реального мира.
31. Конструкторы.
32. Структуры. Классы, объекты и память.
33. Перегрузка унарных операций.
34. Перегрузка бинарных операций. Преобразование типов.
35. Базовый и производный классы.
36. Конструкторы производного класса.
37. Иерархия
38. Адреса и указатели.
39. Управление памятью.
40. Связный список.
41. Указатели на объекты и указатели.
42. Виртуальные функции.
43. Дружественные функции.
44. Статические функции.
45. Потоковые классы.
46. Потоковый ввод/вывод.
47. Указатели файлов.
48. Файловый ввод/вывод.

**9.5. Вопросы к зачету с оценкой**

**Третий семестр**

1. Структуры алгоритмов. Алгоритмы линейной структуры.
2. Алгоритмы разветвляющейся структуры.
3. Алгоритмы циклической структуры.
4. Вычисления в цикле с несколькими одновременно изменяющимися параметрами.
5. Типы данных. Литералы. Переменные.
6. Выражения и операции.
7. Арифметические операции.
8. Логические операции и операции сравнения.
9. Линейная программа.
10. Подключение библиотек.
11. Потоковый ввод и вывод.
12. Математические операции и функции.
13. Разветвляющаяся программа.
14. Оператор IF. Оператор SWITCH.
15. Программа с циклической структурой.
16. Оператор FOR. Операторы DO и WHILE.
17. Указатель (pointer) в C++. Оператор получения адреса.
18. Адрес переменной.
19. Переменная-указатель.
20. Адреса и указатели.
21. Операции получения адреса.
22. Указатели- константы и указатели-переменные.
23. Определение массивов. Многомерные массивы.

24. Массивы объектов. Массивы строк.
25. Простые функции. Передача аргументов  в функцию.
26. Ссылки на аргументы.
27. Область видимости и время жизни переменных.
28. Класс памяти.
29. Структуры. Локальные и глобальные структур.
30. Доступ к полям структуры.
31. Вложенные структуры.
32. Перечисления
33. Простой класс.
34. Объекты программы и объекты реального мира.
35. Конструкторы.
36. Структуры. Классы, объекты и память.
37. Перегрузка унарных операций.
38. Перегрузка бинарных операций.
39. Преобразование типов.
40. Базовый и производный классы.
41. Конструкторы производного класса.
42. Иерархия