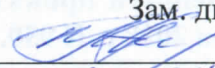


МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного
бюджетного образовательного учреждения высшего образования
«Московский технический университет связи и информатики»

УТВЕРЖДАЮ
Зам. директора по УР
Н.А. Андреева

«28» 04 2026 г.

Б1.О.08 Информационные технологии и программирование рабочая программа дисциплины

Кафедра **Информатики и вычислительной техники**
Направление подготовки **09.03.01. Информатика и вычислительная техника**
Профиль **Искусственный интеллект и машинное обучение**
Формы обучения **очная, заочная**

Распределение часов дисциплины по семестрам (для очной формы обучения (ОФО)), курсам (для заочной формы обучения (ЗФО))

Вид учебной работы	ОФО		ЗФО	
	ЗЕ	часов/сем.	ЗЕ	часов/курс
Общая трудоемкость дисциплины, в том числе (по семестрам, курсам):	7	108/3 144/4	7	252/2
Контактная работа, в том числе (по семестрам, курсам):		54/3 54/4		42/2
Лекции		18/3 18/4		142
Лабораторных работ		18/3 18/4		14/2
Практических занятий		18/3 18/4		14/2
Семинаров				
Самостоятельная работа		54/3 54/4		201/2
Контроль		36/4		9/2
Число контрольных работ (по курсам)				2/2
Число КР (по семестрам, курсам)				
Число КП (по семестрам, курсам)				
Число зачетов с разбивкой по семестрам		1/3		1/2
Число экзаменов с разбивкой по семестрам (курсам)		1/4		1/2

Программу составил:
Доцент кафедры ИВТ, к.т.н. Швидченко С.А.

Рабочая программа дисциплины
«Информационные технологии и программирование»

Разработана в соответствии с ФГОС ВО
направления подготовки 09.03.01 Информатика и вычислительная техника,
утвержденным приказом Министерства образования и науки Российской Федерации от 19
сентября 2017 г. N 930.

Составлена на основании учебного плана
направления 09.03.01 Информатика и вычислительная техника
профиль «Искусственный интеллект и машинное обучение», одобренного Учёным советом
СКФ МТУСИ, протокол № 8 от 31.03 2026 г., и утвержденного директором СКФ
МТУСИ 31.03 2026 г.

Рассмотрена и одобрена на заседании кафедры
«Информатика и вычислительная техника»

Протокол от 28.04 2026 г. № 9

Зав. Кафедрой  С.В. Соколов

1. Цели изучения дисциплины

Целями изучения дисциплины «Информационные технологии и программирование» являются:

- изучение основ алгоритмизации и программирования моделей компонентов информационных систем;
- изучение приемов программирования в различных языках высокого уровня, их инсталляции и использования для разработки компонентов аппаратно- программных комплексов с пользовательскими интерфейсами;
- приобретение студентами знаний и навыков практического использования различных приемов программирования при разработке компонентов информационных систем и средств связи.

2. Планируемые результаты обучения

Изучение дисциплины направлено на формирование у выпускника способности решать профессиональные задачи в соответствии с проектной деятельностью.

Результатом освоения дисциплины являются сформированные у выпускника следующие компетенции:

Компетенции выпускника, формируемые в результате освоения дисциплины (в части, обеспечиваемой дисциплиной)	
ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения	
Знать:	основы алгоритмизации и базовые алгоритмические конструкции, синтаксис и семантику языка C++; принципы структурного и объектно-ориентированного программирования; современные среды разработки программного обеспечения.
Уметь:	формализовать постановку задачи в виде алгоритма и структуры данных, разрабатывать и реализовывать алгоритмы на C++, писать и отлаживать коды на языке программирования C++, тестировать работоспособность программы, интегрировать программные модули.
Владеть:	языком программирования C++, методами отладки и тестирования работоспособности программы.
ОПК-9: Способен осваивать методики использования программных средств для решения практических задач	
Знать:	основные категории программных средств (IDE, компиляторы, отладчики, системы контроля версий) и их назначение при разработке на C++; методики выбора инструментария под тип практической задачи.
Уметь:	подбирать и настраивать программные средства (компилятор, флаги оптимизации, профилировщик) для решения конкретной задачи; использовать отладчики и средства статического анализа для поиска и устранения ошибок в коде;

адаптировать готовые методики под условия практической задачи.
Владеть:
навыками самостоятельной настройки среды разработки под специфику задачи; методами оценки и сравнения альтернативных программных решений по критериям применимости и эффективности; практическими навыками документирования выбранной методики и обоснования выбора инструментария для решения задач.

3. Место дисциплины в структуре образовательной программы

Требования к предварительной подготовке обучающегося (предшествующие дисциплины, модули, темы):	
1	Б1.О.26 «Введение в информационные технологии»
Последующие дисциплины и практики, для которых освоение данной дисциплины необходимо:	
1	Б1.В.07 «Проектный практикум»
2	Б1.О.30 «Структуры и алгоритмы обработки данных»
3	Б1.В.05 «Основы DevOps»
4	Б1.В.ДВ.01.01 «Методы отладки и тестирования программного обеспечения»
5	Б1.В.ДВ.02.01 «Программирование мобильных устройств»
6	Б3.01 Выполнение и защита выпускной квалификационной работы

4. Структура и содержание дисциплины

4.1. Очная форма обучения, 4 года (всего 252 часа, из них 108 часов аудиторных)

Код зан.	Тема и краткое содержание занятия	Вид зан.	Кол. часов	Компетенции	УМИО
1	2	3	4	5	6
Курс 2, Семестр 3					
Модуль 1 - Базовые технологии алгоритмизации и программирования – 57 30 (12 Лек+10 ЛР + 8 ПЗ) часов контактной работы – 27 часов СРС					
1.1	Алгоритмы. Понятие и правила составления. Понятие алгоритма. Правила составления и записи алгоритмов. Типы алгоритмов. Линейные, ветвящиеся и циклические алгоритмы.	Лек.	4	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4
1.2	Технологии алгоритмизации типовых вычислительных задач. Составление блок-схем алгоритмов вычислительных задач. Базовые технологии составления алгоритмов. Требования к технологиям составления блок-схем алгоритмов.	ПЗ	4	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.3	Различные способы описания алгоритмов. Способы составления программ по блок-схемам алгоритмов. Составление алгоритмов по имеющимся кодам программ. Виды тестирования алгоритмов	СРС	12	ОПК-8 ОПК-9	Л1.1 Л1.2 Л2.1 Л2.2
1.4	Управляющие конструкции языка C++. Операторы языка программирования C++. Безусловные конструкции. Условные конструкции. Циклические конструкции.	Лек.	4	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4

1.5	Управляющие конструкции языка C++. Решение прикладных задач в C++. Использование безусловных, условных и циклических конструкций языка.	ПЗ	4	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.6	Массивы. Понятие массива. Свойства массивов. Символьные массивы и строки. Массивы в C++. Статические и динамические массивы. Основные алгоритмы обработки массивов.	Лек.	4	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4
1.7	Исследование технологий создания и использования массивов. Выявление закономерностей в технологии создания и использования массивов различного типа.	Лаб.	4	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.8	Выявление закономерностей использования процедур и функций в решении прикладных задач. Исследование процедур и функций.	Лаб.	6	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.9	Файлы и работа с ними. Структуры, составление и использование структур. Многозадачное программирование. Шаблоны классов. Элементы стандартной библиотеки C++. Практическое использование функций в C++. Функция main(). Шаблоны функций.	СРС	15	ОПК-8 ОПК-9	Л1.1 Л1.2 Л2.1 Л2.2 Л2.3 Л2.4
Модуль 2 - Основные технологии языка C++ – 51 24 (6 Лек + 8 ЛР + 10 ПЗ) часов контактной работы – 27 часов СРС					
2.1	Использование функций при программировании на C++. Объявление функции. Передача параметров в функцию и возврат результата с помощью оператора return. Решение задач с использованием функций.	Лек.	4	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4
2.2	Исследование особенностей использования функций в C++. Исследование перегрузки функций. Область видимости переменных в функциях.	Лаб.	4	ОПК-8 ОПК-9	Л1.1 Л2.2
2.3	Строки и структуры в языке C++. Операции над строками. Тип данных string. Общие сведения о структурах.	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3
2.4	Исследование алгоритмов обработки структур в C++. Обработка массивов и матриц. Обработка строк в C++.	Лаб.	4	ОПК-8 ОПК-9	Л1.1 Л2.2
2.5	Использование многомерных массивов. Практическое изучение использования функций. Структуры, составление и использование структур.	СРС	12	ОПК-8 ОПК-9	Л1.1 Л1.2 Л2.1
2.6	Обработка массивов и матриц. Создания различных структур в C++	СРС	15	ОПК-8 ОПК-9	Л1.2 Л2.1
2.7	Решение прикладных задач. Использование стандартных функций и функций пользователя.	ПЗ	4	ОПК-8 ОПК-9	Л1.1 Л2.2
2.8	Разработка консольных приложений в C++. Обработка массивов и матриц.	ПЗ	4	ОПК-8 ОПК-9	Л1.1 Л2.2
2.9	Создания различных структур в C++. Обработка строк в C++.	ПЗ	2	ОПК-8 ОПК-9	Л1.1 Л2.2
Курс 2, Семестр 4					

Модуль 1 - Реализация объектно-ориентированного программирования на языке C++					
57					
30 (12 Лек+10 ЛР+8 ПЗ) часов контактной работы – 27 часов СРС					
1	Понятия и определения ООП. Особенности программ с ООП по сравнению с другими видами программирования. Методологические основы ООП. Основные понятия и определения. Суть концепции ООП.	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4 Л3.1
1.2	Методология понятия класс. Виды программирования. Основная идея ООП. Понятие метода пользователя. Методы, возвращающие значения и не возвращающие значения. Составление программ с методами пользователя. Main метод программы как основной метод.	Лек.	4	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4
1.3	Составление программ с инкапсуляцией полей и с перегрузкой (полиморфизмом) методов. Практическая реализация различных видов полей и методов	ПЗ	4	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.4	Перегрузка функций и конструкторов. Организация рекурсий в функциях. Файлы и работа с ними.	СРС	12	ОПК-8 ОПК-9	Л1.1 Л2.1
1.5	Классы и объекты. Классы и объекты классов. Прямой доступ к объектам классов. Инкапсуляция. Полиморфизм.	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3
1.6	Виды наследования. Комбинации доступа. Выбор спецификатора доступа. Уровни наследования. Множественное наследование	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3
1.7	Создание приложений с различными видами наследования. Практическая реализация различных видов наследования классов.	ПЗ	4	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.8	Конструкторы при различных видах наследования. Конструкторы с переменными и без. Конструкторы по умолчанию. Действие конструкторов при различных видах наследования.	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3
1.9	Исследование простого наследования классов. Варианты наследования классов в C++.	Лаб.	4	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.10	Исследование взаимодействия методов в C++. Объектно-ориентированный подход в программировании методов.	Лаб.	6	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.11	Структуры, составление и использование структур. Многозадачное программирование. Изучение способов составления программ с полиморфизмом	СРС	15	ОПК-8 ОПК-9	Л1.1 Л1.2 Л2.1
Модуль 2 - Динамические структуры данных в языке C++ – 51					
24 (6 Лек+8 ЛР+10 ПЗ) часов контактной работы – 27 часов СРС					
2.1	Динамическое выделение памяти. Обработка нехватки памяти. Перегрузка операторов new и delete. Выделение и освобождение памяти. Релокация. Деревья. Массивы. Однонаправленные (односвязные) списки. Двухнаправленные (двусвязные) списки.	Лек.	4	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4

2.2	Исследование динамических структур данных	Лаб.	4	ОПК-8 ОПК-9	Л1.1 Л2.2
2.3	Циклические списки. Стек. Дек. Очередь. Бинарные деревья. Сложные структуры. Связи отдельных элементов и допустимые операции.	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3
2.4	Исследование возможностей работы со списками, организованными в виде очереди	Лаб.	4	ОПК-8 ОПК-9	Л1.1 Л2.2
2.5	Списки. Операции над односвязными списками. Двухнаправленные и кольцевые списки. Операции над кольцевыми списками.	СРС	12	ОПК-8 ОПК-9	Л1.1 Л1.2 Л2.1
2.6	Деревья. Определение и построение. Таблицы. Двоичное (бинарное) дерево: создание и обход. Функция сравнения или получение ключа. Поиск в глубину (Depth-first search, DFS). Поиск в ширину (breadth-first search, BFS)	СРС	15	ОПК-8 ОПК-9	Л1.2 Л2.1
2.7	Составление программ, которые содержат динамическую информацию в виде динамического односвязного списка.	ПЗ	4	ОПК-8 ОПК-9	Л1.1 Л2.2 Л2.3
2.8	Составление программ, которые содержат динамическую информацию в виде динамического двухсвязного списка.	ПЗ	4	ОПК-8 ОПК-9	Л1.1 Л2.2 Л2.3
2.9	Составление программ с использованием динамических структур данных в виде стеков и очередей.	ПЗ	2	ОПК-8 ОПК-9	Л1.1 Л2.2 Л2.3
Экзамен			36	ОПК-8 ОПК-9	Л1.1 - Л3.1
Итого 144 часа					

4.2. Заочная форма обучения, 5 лет

Код зан.	Тема и краткое содержание занятия	Вид зан.	Кол. часов	Компетенции	УМИО
1	2	3	4	5	6
Курс 2					
Модуль 1 - Базовые технологии алгоритмизации и программирования – 36 2 (2 Лек) часа контактной работы – 34 часов СРС					
1.1	Алгоритмы. Понятие и правила составления. Понятие алгоритма. Правила составления и записи алгоритмов. Типы алгоритмов. Линейные, ветвящиеся и циклические алгоритмы. Управляющие конструкции языка C++. Операторы языка программирования C++. Безусловные конструкции. Условные конструкции. Циклические конструкции.	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4
1.2	Управляющие конструкции языка C++. Операторы языка программирования C++. Безусловные конструкции. Условные конструкции. Циклические конструкции.	СРС	34	ОПК-8 ОПК-9	Л1.1 Л1.2 Л1.3
Модуль 1: Базовые технологии алгоритмизации и программирования – 51 6 (2 Лек+2 ЛР+2 ПЗ) часов контактной работы – 45 часов СРС					

1.3	Составление блок-схем алгоритмов вычислительных задач. Управляющие конструкции языка C++. Решение прикладных задач в C++. Использование безусловных, условных и циклических конструкций языка.	ПЗ	2	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.4	Различные способы описания алгоритмов. Способы составления программ по блок-схемам алгоритмов. Составление алгоритмов по имеющимся кодам программ. Виды тестирования алгоритмов	СРС	22	ОПК-8 ОПК-9	Л1.1 Л1.2 Л2.1 Л2.2
1.5	Массивы. Понятие массива. Свойства массивов. Символьные массивы и строки. Массивы в C++. Статические и динамические массивы. Основные алгоритмы обработки массивов. Указатели на функции. Статические и динамические матрицы и их обработка в C++	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4
1.6	Исследование технологий создания и использования массивов. Выявление закономерностей в технологии создания и использования массивов различного типа.	Лаб.	2	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.7	Использование многомерных массивов. Практическое изучение использования функций. Файлы и работа с ними. Структуры, составление и использование структур. Многозадачное программирование. Шаблоны классов. Элементы стандартной библиотеки C++. Практическое использование функций в C++. Функция main(). Шаблоны функций.	СРС	23	ОПК-8 ОПК-9	Л1.1 Л1.2 Л2.1 Л2.2 Л2.3 Л2.4
Модуль 2 - Основные технологии языка C++ – 57 12 (4 Лек+4 ЛР+4 ПЗ) часов контактной работы – 45 часов СРС					
2.1	Использование функций при программировании на C++. Объявление функции. Передача параметров в функцию и возврат результата с помощью оператора return. Решение задач с использованием функций.	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4
2.2	Исследование особенностей использования функций в C++. Исследование перегрузки функций. Область видимости переменных в функциях.	Лаб.	4	ОПК-8 ОПК-9	Л1.1 Л2.2
2.3	Строки и структуры в языке C++. Операции над строками. Тип данных string. Общие сведения о структурах. Исследование алгоритмов обработки структур в C++. Обработка массивов и матриц. Обработка строк в C++.	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3
2.4	Разработка консольных приложений в C++. Обработка массивов и матриц. Обработка строк	ПЗ	4	ОПК-8 ОПК-9	Л1.1 Л2.2
2.5	Использование многомерных массивов. Практическое изучение использования функций. Структуры, составление и использование структур. Обработка массивов и матриц. Создания различных структур в C++	СРС	45	ОПК-8 ОПК-9	Л1.1 Л1.2 Л2.1
	Зачет			ОПК-8 ОПК-9	Л1.1- Л3.1
Модуль 1 - Реализация объектно-ориентированного программирования на языке C++ 52 12 (4 Лек+4 ЛР+4 ПЗ) часов контактной работы – 40 часов СРС					

1.1	Понятия и определения ООП. Методология понятия класс. Виды программирования. Классы и объекты. Классы и объекты классов. Прямой доступ к объектам классов. Инкапсуляция. Полиморфизм.	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4 Л3.1
1.2	Составление программ с инкапсуляцией полей и с перегрузкой (полиморфизмом) методов. Практическая реализация различных видов полей и методов	ПЗ	2	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.3	Перегрузка функций и конструкторов. Организация рекурсий в функциях. Файлы и работа с ними.	СРС	20	ОПК-8 ОПК-9	Л1.1 Л2.1
1.4	Виды наследования. Комбинации доступа. Выбор спецификатора доступа. Уровни наследования. Множественное наследование. Конструкторы при различных видах наследования. Конструкторы с переменными и без. Конструкторы по умолчанию. Действие конструкторов при различных видах наследования.	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3
1.5	Создание приложений с различными видами наследования. Практическая реализация различных видов наследования классов.	ПЗ	2	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.6	Исследование простого наследования классов. Варианты наследования классов в C++.	Лаб.	2	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.7	Исследование взаимодействия методов в C++. Объектно-ориентированный подход в программировании методов.	Лаб.	2	ОПК-8 ОПК-9	Л1.1 Л2.2 Л3.1
1.8	Структуры, составление и использование структур. Многозадачное программирование. Изучение способов составления программ с полиморфизмом	СРС	20	ОПК-8 ОПК-9	Л1.1 Л1.2 Л2.1
Модуль 2 -Динамические структуры данных в языке C++ – 47 10 (4 Лек+4 ЛР+4 ПЗ) часов контактной работы – 40 часов СРС					
2.1	Динамическое выделение памяти. Обработка нехватки памяти. Перегрузка операторов new и delete. Выделение и освобождение памяти. Релокация. Деревья. Массивы. Однонаправленные (односвязные) списки. Двухнаправленные (двусвязные) списки. Циклические списки. Стек. Дек. Очередь. Бинарные деревья. Сложные структуры. Связи отдельных элементов и допустимые операции.	Лек.	2	ОПК-8 ОПК-9	Л1.1 Л2.1 Л2.3 Л2.4
2.2	Исследование динамических структур данных	Лаб.	2	ОПК-8 ОПК-9	Л1.1 Л2.2
2.4	Исследование возможностей работы со списками, организованными в виде очереди	Лаб.	2	ОПК-8 ОПК-9	Л1.1 Л2.2
2.5	Списки. Операции над односвязными списками. Двухнаправленные и кольцевые списки. Операции над кольцевыми списками.	СРС	20	ОПК-8 ОПК-9	Л1.1 Л1.2 Л2.1

2.6	Деревья. Определение и построение. Таблицы. Двоичное (бинарное) дерево: создание и обход. Функция сравнения или получение ключа. Поиск в глубину (Depth-first search, DFS). Поиск в ширину (breadth-first search, BFS)	СРС	17	ОПК-8 ОПК-9	Л1.2 Л2.1
2.7	Составление программ, которые содержат динамическую информацию в виде динамического односвязного списка.	ПЗ	2	ОПК-8 ОПК-9	Л1.1 Л2.2 Л2.3
2.8	Составление программ, которые содержат динамическую информацию в виде динамического двухсвязного списка.	ПЗ	2	ОПК-8 ОПК-9	Л1.1 Л2.2 Л2.3
Экзамен			9	ОПК-8 ОПК-9	Л1.1- Л3.1
Итого 108 часов					

5. Учебно-методическое и информационное обеспечение дисциплины

5.1 Рекомендуемая литература				
5.1.1. Основная литература				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л1.1	Павловская Т.А.	Программирование на языке C++ : Курс лекций	Москва : Интуит НОУ, 2016. — 154 с.	Э1
Л1.2	Фарафонов А.С.	Программирование на языке высокого уровня: методические указания к проведению лабораторных работ	Липецк : ЛГТУ ЭБС АСВ, 2020. — 32 с.	Э2
5.1.2 Дополнительная литература				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л2.1	Зоткин С.П.	Программирование на языке высокого уровня C/C++ : конспект лекций	Москва : МГСУ, Ай Пи Эр Медиа, ЭБС АСВ, 2016. — 140 с.	Э3
Л2.2	Доткулова А.С.	Практикум по дисциплине «Логическое и функциональное программирование»	Москва, МТУСИ, 2016. — 53 с	Э4
Л2.3	Яшина М.В., Барков В.В., Украинский С.В.	Учебно-методическое пособие по дисциплине: «Логическое и функциональное программирование»	Москва, МТУСИ, 2016. — 53 с	Э5
Л2.4	Козырева Г.Ф.	Функциональное и логическое программирование: учебно-методическое пособие	Саратов : Ай Пи Эр Медиа, 2018. — 120 с.	Э6
5.1.3 Методическое обеспечение для самостоятельной работы обучающихся				
Код	Авторы, составители	Заглавие	Издательство, год	Кол.
Л3.1	Кудрявцева Л.Г., Самолетов Р.В.	Информационные технологии. Практикум	Саратов: Вузовское образование, 2020	Э7
5.2 Электронные образовательные ресурсы				

Э1	https://book.ru/book/918123
Э2	https://www.iprbookshop.ru/22912.html
Э3	https://www.iprbookshop.ru/48037.html
Э4	https://www.iprbookshop.ru/61489.html
Э5	https://www.iprbookshop.ru/61490.html
Э6	https://www.iprbookshop.ru/71596.html
Э7	https://www.iprbookshop.ru/97631.html

5.3 Информационно-справочные системы и профессиональные базы данных

5.3.1	Консультант Плюс – https://www.consultant.ru/
5.3.2	Гарант – https://www.garant.ru/
5.3.3	Консорциум Кодекс – https://docs.cntd.ru/
5.3.4	Научная электронная библиотека – https://www.elibrary.ru/
5.3.5	Профессиональная база данных: Федеральный портал «Российское образование»: [Электронный ресурс] – Режим доступа: http://www.edu.ru/ (открытый доступ)
5.3.6	Профессиональная база данных: Федеральный центр информационно-образовательных ресурсов: [Электронный ресурс] – Режим доступа: http://fcior.edu.ru/ (открытый доступ)

5.4 Программное обеспечение

П.1	Libra Office Свободное программное обеспечение
П.2	Яндекс браузер для организаций
П.3	Kaspersky Endpoint Security

6. Материально-техническое обеспечение дисциплины

6.1 МТО лекционных занятий	
1	Лекционная аудитория, оснащенная проектором, ПК (ноутбуком), экраном
6.2 МТО лабораторных работ и практических занятий	
1	Учебные аудитории, оборудованные компьютерной техникой, с возможностью подключения к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ
6.3 МТО рубежных контролей, зачетов, экзаменов	
1	Учебные аудитории, оборудованные компьютерной техникой, с возможностью подключения к сети «Интернет», и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ
6.4 МТО самостоятельной работы обучающихся	
1	Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду СКФ МТУСИ

7. Оценочные материалы

Оценочные материалы и перечень видов оценочных средств текущего контроля и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.

8. Методические рекомендации для обучающихся по освоению дисциплины

Методические рекомендации по освоению дисциплины представлены в Приложении 2 к рабочей программе дисциплины.
--

9. Особенности реализации дисциплины (модуля) при обучении инвалидов и лиц с ограниченными возможностями здоровья

Рабочая программа дисциплины (модуля) при необходимости может быть адаптирована для обучения (в том числе с применением дистанционных образовательных технологий) лиц с ограниченными возможностями здоровья, инвалидов. Для этого требуется заявление обучающихся, являющихся лицами с ограниченными возможностями здоровья, инвалидами, или их законных представителей и рекомендации психолого-медико-педагогической комиссии. При обучении лиц с ограниченными возможностями здоровья учитываются их индивидуальные психофизические особенности. Обучение инвалидов осуществляется также в соответствии с индивидуальной программой реабилитации инвалида (при наличии).

Для лиц с нарушением слуха возможно предоставление учебной информации в визуальной форме (краткий конспект лекций; тексты заданий, напечатанные увеличенным шрифтом), на аудиторных занятиях допускается присутствие ассистента, а также сурдопереводчиков и тифлосурдопереводчиков.

Текущий контроль успеваемости осуществляется в письменной форме: обучающийся письменно отвечает на вопросы, письменно выполняет практические задания.

Доклад (реферат) также может быть представлен в письменной форме, при этом требования к содержанию остаются теми же, а требования к качеству изложения материала (понятность, качество речи, взаимодействие с аудиторией и т. д.) заменяются на соответствующие требования, предъявляемые к письменным работам (качество оформления текста и списка литературы, грамотность, наличие иллюстрационных материалов и т.д.).

Промежуточная аттестация для лиц с нарушениями слуха проводится в письменной форме, при этом используются общие критерии оценивания. При необходимости время подготовки к ответу может быть увеличено.

Для лиц с нарушением зрения допускается аудиальное предоставление информации, а также использование на аудиторных занятиях звукозаписывающих устройств (диктофонов и т.д.). Допускается присутствие на занятиях ассистента (помощника), оказывающего обучающимся необходимую техническую помощь.

Текущий контроль успеваемости осуществляется в устной форме.

При проведении промежуточной аттестации для лиц с нарушением зрения тестирование может быть заменено на устное собеседование по вопросам.

Для лиц с ограниченными возможностями здоровья, имеющих нарушения опорно-двигательного аппарата, на аудиторных занятиях, а также при проведении процедур текущего контроля успеваемости и промежуточной аттестации могут быть предоставлены необходимые технические средства (персональный компьютер, ноутбук или другой гаджет); допускается присутствие ассистента (ассистентов), оказывающего обучающимся необходимую техническую помощь (занять рабочее место, передвигаться по аудитории, прочитать задание, оформить ответ, общаться с преподавателем).

10. Дополнения и изменения в Рабочей программе

Лист актуализации рабочей программы дисциплины «Б1.О.08 Информационные технологии и программирование» для использования в 20__/20__ учебном году

Утверждаю

Зам. директора по УР _____
«__» _____ 20__ г.

Направление: 09.03.01. Информатика и вычислительная техника

Профиль: Искусственный интеллект и машинное обучение

Форма обучения: очная, заочная

(Возможны следующие варианты):

а) Рабочая программа действует без изменений.

б) В рабочую программу вносятся следующие изменения:

1)

2)

3)

Разработчик (и): _____
(ФИО, ученая степень, ученое звание)

«__» _____ 20__ г.

Рабочая программа пересмотрена и одобрена на заседании кафедры ИВТ

Протокол № _____ от «__» _____ 20__ г.

Заведующий кафедрой _____

Оценочные материалы для проведения текущего контроля и промежуточной аттестации по дисциплине

1. Оценочные материалы для проведения промежуточной аттестации по дисциплине

1.1 Шкала оценивания компетенций

Шкала оценивания компетенций		
Оценка	Уровень освоения компетенции	Критерии оценивания
«Отлично»	Высокий уровень	Обучающийся показывает всестороннее, систематическое и глубокое знание основного и дополнительного учебного материала, умеет свободно выполнять задания, предусмотренные программой; усвоил основную и знаком с дополнительной рекомендованной литературой; может объяснить взаимосвязь основных понятий дисциплины в их значении для последующей профессиональной деятельности; проявляет творческие способности в понимании, изложении и использовании учебного материала.
«Хорошо»	Повышенный уровень	Обучающийся показывает достаточный уровень знаний в пределах основного учебного материала, без существенных ошибок выполняет предусмотренные в программе задания; усвоил основную литературу, рекомендованную в программе; способен объяснить взаимосвязь основных понятий дисциплины при дополнительных вопросах преподавателя. Допускает не существенные погрешности в ответах, устраняет их без помощи преподавателя.
«Удовлетворительно»	Пороговый уровень	Обучающийся показывает знания основного учебного материала в минимальном объеме, необходимом для дальнейшей учебы; справляется с выполнением заданий, предусмотренных программой, допуская при этом большое количество не принципиальных ошибок; знаком с основной литературой, рекомендованной программой. Допускает существенные погрешности в ответах, но обладает необходимыми знаниями для их устранения под руководством преподавателя.
«Неудовлетворительно»	Минимальный уровень не достигнут	Обучающийся обнаруживает пробелы в знаниях основного учебного материала, допускает принципиальные ошибки в выполнении предусмотренных программой заданий, не знаком с рекомендованной

		литературой, не может исправить допущенные ошибки. Как правило, оценка «неудовлетворительно» ставится обучающимся, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании вуза без дополнительных занятий по соответствующей дисциплине.
Зачет		0-100 «Не зачтено» - 0-40 «Зачтено» - 41-100
Экзамен		0-100 «Неудовлетворительно» - 0-40 «Удовлетворительно» - 41-60 «Хорошо» - 61-80 «Отлично» - 81-100

1.2 Показатели, критерии и шкалы оценивания компетенций

ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения		
Знать:		
основы алгоритмизации и базовые алгоритмические конструкции, синтаксис и семантику языка C++; принципы структурного и объектно-ориентированного программирования; современные среды разработки программного обеспечения	Контрольная работа	Модуль 1. 0 ÷ 7 «Неудовлетворительно» - 0,1 «Удовлетворительно» - 2,3 «Хорошо» - 4,5 «Отлично» - 6,7
	Контрольная работа	Модуль 2. 0 ÷ 7 «Неудовлетворительно» - 0,1 «Удовлетворительно» - 2,3 «Хорошо» - 4,5 «Отлично» - 6,7
Уметь:		
формализовать постановку задачи в виде алгоритма и структуры данных; разрабатывать и реализовывать алгоритмы на C++; писать и отлаживать коды на языке программирования C++; тестировать работоспособность программы; интегрировать программные модули	Практические занятия 1, 2, 4, 6, 8 Лабораторная работа 1, 3, 4, 8	Модуль 1 0 ÷ 10 «Неудовлетворительно» - 0-4 «Удовлетворительно» - 5-6 «Хорошо» - 7-8 «Отлично» - 9-10
Владеть:		

языком программирования C++; методами отладки и тестирования работоспособности программы	Практические занятия 2, 3, 5, 7, 9, 10 Лабораторная работа 2, 5, 6, 7	Модуль 2 0 ÷ 20 «Неудовлетворительно» - 0-9 «Удовлетворительно»- 11- 13 «Хорошо» - 14-17 «Отлично» - 18-20
Зачет		0-100: «Не зачтено» - 0-40 «Зачтено» - 41-100
Экзамен		0-100: «Неудовлетворительно» - 0- 40 «Удовлетворительно» - 41- 60 «Хорошо» - 61-80 «Отлично» - 81-100
ОПК-9: Способен осваивать методики использования программных средств для решения практических задач		
Знать:		
основные категории программных средств (IDE, компиляторы, отладчики, системы контроля версий) и их назначение при разработке на C++; методики выбора инструментария под тип практической задачи	Контрольная работа Контрольная работа	Модуль 1. 0 ÷ 7 «Неудовлетворительно» - 0,1 «Удовлетворительно» - 2,3 «Хорошо» - 4,5 «Отлично» - 6,7 Модуль 2. 0 ÷ 7 «Неудовлетворительно» - 0,1 «Удовлетворительно» - 2,3 «Хорошо» - 4,5 «Отлично» - 6,7
Уметь:		
подбирать и настраивать программные средства (компилятор, флаги оптимизации, профилировщик) для решения конкретной задачи; использовать отладчики и средства статического анализа для поиска и устранения ошибок в коде; адаптировать готовые методики под условия практической задачи.	Практические занятия 1, 3, 5, 7, 8, 9 Лабораторная работа 1, 3, 5, 6	Модуль 1 0÷10 «Неудовлетворительно» - 0-4 «Удовлетворительно» - 5-6 «Хорошо» - 7-8 «Отлично» - 9-10
Владеть:		

навыками самостоятельной настройки среды разработки под специфику задачи; методами оценки и сравнения альтернативных программных решений по критериям применимости и эффективности; практическими навыками документирования выбранной методики и обоснования выбора инструментария для решения задач	Практические занятия 2, 4, 6, 10 Лабораторная работа 2, 4, 6, 7, 8	Модуль 2 0 ÷ 20 «Неудовлетворительно» - 0-9 «Удовлетворительно»- 11-13 «Хорошо» - 14-17 «Отлично» - 18-20
--	---	--

1.3 Оценочные материалы для текущего контроля: типовые контрольные задания, иные материалы

1.3.1 Оценочные средства для очной и заочной формы обучения

Модуль 1 семестр 1 (50 баллов)

Модуль содержит 3 лекционных занятия, 2 практических занятия и 2 лабораторных работы, а также самостоятельную работу студентов. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 10 баллов, за выполненное и защищённое практическое занятие студент получает максимально 10+10=20 баллов, за выполненную и защищённую лабораторную работу студент получает максимально 10+10=20 баллов. Общее максимальное количество баллов за модуль 1 составляет 50 баллов.

Контрольные вопросы к практическим занятиям.

Вопросы для контрольной работы 1 (ОПК-8, ОПК-9):

1. Какие базовые инструменты (пакетные менеджеры, компиляторы) обычно используют при установке среды для разработки на C++?
2. Зачем системному администратору важно понимать разницу между статической и динамической линковкой при развёртывании C++-приложений?
3. Как проверить, корректно ли установлены переменные окружения (например, PATH, LD_LIBRARY_PATH) для сборки C++-кода на сервере?
4. Какие типичные ошибки возникают при установке зависимостей для C++-проекта и как их диагностировать через логи сборки?
5. Как администратор может контролировать версии компиляторов C++ на разных узлах инфраструктуры, чтобы избежать проблем совместимости?.
6. Какие три базовые алгоритмические конструкции вы знаете?
7. В чём разница между синтаксисом и семантикой в языке программирования?
8. Приведите пример синтаксической ошибки в коде на C++.
9. Что означает ключевое слово const в C++ с точки зрения семантики?

Практическое занятие №1. Технологии алгоритмизации типовых вычислительных задач. Составление блок-схем алгоритмов вычислительных задач. Базовые технологии составления алгоритмов. Требования к технологиям составления блок-схем алгоритмов. (ОПК-8, ОПК-9)
Контрольные вопросы к ПЗ

1. Что такое алгоритмизация типовой вычислительной задачи?
2. Назови три базовые структуры алгоритмов и кратко поясни каждую.

3. Какие геометрические фигуры используют в блок-схемах и что они обозначают?
4. В чём главное требование к чёткости описания шагов в алгоритме?
5. Приведи пример простой вычислительной задачи и опиши, с чего начнёшь составлять для неё блок-схему.

Практическое занятие №2. Управляющие конструкции языка C++. Решение прикладных задач в C++. Использование безусловных, условных и циклических конструкций языка (ОПК-8, ОПК-9)

Контрольные вопросы к ПЗ

1. Для чего в C++ используют условные конструкции (например, if)?
2. Как записать простейшее условие с if в C++?
3. Чем отличается конструкция if-else от одиночного if?
4. В каких случаях удобно применять оператор switch вместо цепочки if-else?
5. Какой цикл в C++ подойдёт, если нужно повторять действия, пока выполняется условие (например, пока число больше нуля)?
6. Чем цикл for обычно удобнее while при переборе элементов?
7. Как досрочно выйти из цикла в C++ и как пропустить одну итерацию?
8. Приведи короткий пример прикладной задачи (например, подсчёт суммы или поиск максимума), где логично использовать цикл и условие вместе..

Лабораторная работа №1. Исследование технологий создания и использования массивов. Выявление закономерностей в технологии создания и использования массивов различного типа (ОПК-8, ОПК-9)

Контрольные вопросы к ЛР

1. Что такое массив в программировании?
2. Чем статический массив отличается от динамического?
3. Как обратиться к отдельному элементу массива?
4. Почему индексация в массивах часто начинается с нуля?
5. Какие типы данных можно хранить в однородном массиве?
6. В чём особенность многомерных массивов — как к ним обращаться?
7. Какая закономерность есть в выделении памяти под элементы массива одного типа?
8. Приведи пример задачи, где использование массива заметно упрощает решение..

Лабораторная работа №2. Выявление закономерностей использования процедур и функций в решении прикладных задач. Исследование процедур и функций (ОПК-8, ОПК-9)

Контрольные вопросы к ЛР

1. Какие основные различия между процедурами и функциями в программировании?
2. Как часто используются процедуры и функции в реальных прикладных задачах?
3. Какие преимущества даёт использование процедур и функций в программировании?
4. Какие сложности могут возникнуть при применении процедур и функций в решении задач?
5. Как влияет правильное использование процедур и функций на эффективность программы?
6. Какие методы можно использовать для оптимизации процедур и функций в задачах?
7. В каких случаях предпочтительнее использовать процедуры вместо функций и наоборот?
8. Как влияет структурирование кода с помощью процедур и функций на его читаемость и поддержку?

Модуль 2 семестр 1 (50 баллов):

Модуль 2 содержит 2 лекционных занятия, 1 практическое занятие и 1 лабораторную работу, а также самостоятельную работу студентов. Знание лекционного материала оценивается по контрольной работе, состоящей из четырех вопросов, максимальное количество баллов за

контрольную работу составляет 10 баллов, за выполненное и защищённое практическое занятие студент получает максимально 20 баллов, за выполненную и защищённую лабораторную работу студент получает максимально 20 баллов. Общее максимальное количество баллов за модуль 2 составляет 50 баллов.

Практическое занятие №3. Решение прикладных задач. Использование стандартных функций и функций пользователя (ОПК-8, ОПК-9)

Контрольные вопросы к ПЗ

1. Что понимают под прикладной задачей в программировании?
2. Приведи пример простой прикладной задачи, которую можно решить на C++.
3. Чем стандартная функция отличается от функции пользователя?
4. Назови 2–3 стандартные функции C++, которые часто используют при решении прикладных задач.
5. Зачем в программе создают собственные (пользовательские) функции?
6. Как объявить и вызвать пользовательскую функцию в C++?
7. При каком условии удобно вынести часть решения прикладной задачи в отдельную функцию?
8. Как передать данные из основной программы в пользовательскую функцию и вернуть результат обратно?

Практическое занятие №4. Разработка консольных приложений в C++. Обработка массивов и матриц (ОПК-8, ОПК-9)

Контрольные вопросы к ПЗ

1. Как объявить одномерный массив в консольном приложении на C++?
2. Какой цикл чаще всего используют для заполнения массива данными в C++?
3. Как вывести элементы массива на консоль в C++?
4. Чем отличается обработка одномерного массива от обработки матрицы в C++?
5. Как объявить двумерный массив (матрицу) в C++?
6. Какой тип цикла нужен для обхода всех элементов матрицы?
7. Как найти сумму всех элементов массива в консольной программе на C++?
8. Как поменять местами два элемента массива в C++?

Практическое занятие №5. Создания различных структур в C++. Обработка строк в C++ (ОПК-8, ОПК-9)

Контрольные вопросы к ПЗ

1. Что такое структура (struct) в C++ и зачем она нужна?
2. Как объявить и инициализировать структуру в коде?
3. Как обратиться к полю структуры через переменную и через указатель?
4. Чем структура отличается от класса в C++ (в базовом понимании)?
5. Что хранит тип std::string и какой заголовочный файл для него нужен?
6. Как считать строку с пробелами с клавиатуры в C++?
7. Как узнать длину строки и проверить, пустая ли она?
8. Как объединить две строки или добавить символ в конец строки в C++?

Лабораторная работа №3. Исследование особенностей использования функций в C++. Исследование перегрузки функций. Область видимости переменных в функциях (ОПК-8, ОПК-9)

Контрольные вопросы к ЛР

1. Что значит «использовать функцию» в C++ — приведи короткий пример вызова.
2. Зачем в C++ делают отдельные объявления и определения функций?

3. Что такое перегрузка функций и какой параметр отличает одну перегруженную функцию от другой?
4. Приведи два примера сигнатур перегруженных функций с одним именем.
5. Что видно внутри функции: все переменные программы или только некоторые? Почему?
6. Чем локальная переменная в функции отличается от глобальной по области видимости?
7. Что произойдёт, если в функции объявить переменную с тем же именем, что и у глобальной?
8. Как компилятор выбирает, какую из перегруженных функций вызвать при вызове?

Лабораторная работа №4. Исследование алгоритмов обработки структур в C++.
Обработка массивов и матриц. Обработка строк в C++. (ОПК-8, ОПК-9)
Контрольные вопросы к ЛР

1. Как в C++ обратиться к отдельному элементу одномерного массива?
2. Чем отличается обработка одномерного массива от обработки матрицы в C++?
3. Какой цикл чаще всего используют для перебора всех элементов массива и почему?
4. Как найти сумму элементов матрицы в C++ — опиши базовый алгоритм в двух трёх шагах.
5. Что нужно учитывать при обходе матрицы по столбцам, а не по строкам?
6. Как в C++ проверить, содержит ли строка определённый символ?
7. Приведи пример простой операции над строкой в C++ (например, получение длины или конкатенация).
8. Как можно объединить обработку массива и работу со строкой в одной задаче — придумай краткий пример сценария.

Модуль 1 семестр 2 (50 баллов)

Модуль 1 содержит 3 лекционных занятия, 3 практических занятия и 3 лабораторных работы, а также самостоятельную работу студентов. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 14 баллов, за выполненное и защищённое практическое занятие студент получает максимально 6+6+6=18 баллов, за выполненную и защищённую лабораторную работу студент получает максимально 6+6+6=18 баллов. Общее максимальное количество баллов за модуль 1 составляет 50 баллов.

Вопросы для контрольной работы 2 (ОПК-8, ОПК-9):

1. Какие способы обмена данными между информационными системами чаще всего реализуют на C++?
2. Как в C++ работают с JSON или XML при взаимодействии систем?
3. В чём плюсы и минусы использования сокетов для информационного взаимодействия в C++?
4. Как в современных C++-приложениях организуют взаимодействие с REST-API?
5. В чём главное отличие структурного подхода к программированию от объектно-ориентированного?
6. Назови три ключевых принципа ООП в C++ и кратко поясни каждый.
7. Какие задачи в C++ удобнее решать со структурным подходом, а какие — с ООП? Приведи по одному простому примеру.
8. Назови 2–3 популярные среды разработки для C++ и укажи по одному преимуществу каждой.

Практическое занятие №6. Составление программ с инкапсуляцией полей и с перегрузкой (полиморфизмом) методов. Практическая реализация различных видов полей и методов (ОПК-8, ОПК-9)

Контрольные вопросы к ПЗ

1. Что значит инкапсулировать поля в классе C++ и зачем это делают?
2. Какие модификаторы доступа используют для инкапсуляции полей — назови основные и кратко поясни разницу.
3. Приведи простой пример класса, где поля объявлены private, а доступ к ним идёт через публичные методы.
4. В чём разница между перегрузкой методов и полиморфизмом в C++?
5. Как компилятор различает перегруженные методы — какие параметры сигнатуры для этого важны?
6. Покажи два перегруженных метода в одном классе — с разной сигнатурой.
7. Как реализовать полиморфизм в C++ на практике — какой механизм для этого чаще всего используют?
8. Что произойдёт, если вызвать виртуальный метод через указатель на базовый класс, указывающий на объект производного класса?

Практическое занятие №7. Создание приложений с различными видами наследования.

Практическая реализация различных видов наследования классов (ОПК-8, ОПК-9)

Контрольные вопросы к ПЗ.

1. Что означает ключевое слово public при объявлении наследования (например, class Derived : public Base)?
2. Чем отличается доступ к членам базового класса при private и protected наследовании?
3. Как вызвать конструктор базового класса из конструктора производного класса? Приведи краткий пример.
4. Что произойдёт, если в базовом и производном классе есть метод с одинаковым именем и сигнатурой — какой метод будет вызван?
5. Зачем в базовом классе делают деструктор виртуальным при использовании наследования?
6. В чём суть множественного наследования в C++ и как его объявить?
7. Что такое «ромбовидное наследование» и как в C++ решают возникающую из-за него проблему?
8. Приведи пример простой иерархии из 3 классов (базовый и два производных) с реализацией одного общего метода в каждом — как будет работать переопределение?

Лабораторная работа №5. Исследование простого наследования классов.

Варианты наследования классов в C++ (ОПК-8, ОПК-9)

Контрольные вопросы к ЛР

1. Что значит «простое наследование» в C++?
2. Как в коде записать, что один класс наследует другой? Приведи минимальный пример синтаксиса.
3. Чем базовый класс отличается от производного при простом наследовании?
4. Какие спецификаторы доступа (public/protected/private) используют при наследовании и как они влияют на видимость членов?
5. Наследуются ли конструкторы и деструкторы в C++? Если нет — как тогда инициализировать базовый класс?
6. Могут ли приватные члены базового класса использоваться в производном классе напрямую? Если нет — какими способами к ним можно получить доступ?
7. Приведи пример ситуации, где уместно использовать простое наследование, и кратко опиши, что будет в базовом и производном классах.
8. В чём разница между public наследованием и private наследованием на уровне доступа к членам базового класса?

Лабораторная работа №6. Безопасный ввод/вывод данных. Программная реализация линейных алгоритмов (ОПК-8, ОПК-9)

Контрольные вопросы к ЛР

1. Что значит «взаимодействие методов» в контексте ООП на C++?
2. Как методы одного класса могут вызывать методы другого класса? Приведи простой пример.
3. В чём разница между вызовом метода через объект и через указатель на объект?
4. Какую роль играет указатель this при взаимодействии методов внутри класса?
5. Как виртуальные функции помогают методам взаимодействовать в иерархии классов?
6. Чем отличается перегрузка методов от их переопределения с точки зрения взаимодействия в программе?
7. Как область видимости (public/private/protected) влияет на возможность вызова одних методов из других?
8. Приведи пример, когда вызов одного метода запускает цепочку других методов — и объясни, зачем так делают.

Модуль 2 семестр 2 (50 баллов)

Модуль 2 содержит 3 лекционных занятия, 3 практических занятия и 3 лабораторных работы, а также самостоятельную работу студентов. Знание лекционного материала оценивается по контрольной работе, состоящей из двух вопросов, максимальное количество баллов за контрольную работу составляет 14 баллов, за выполненное и защищённое практическое занятие студент получает максимально $6+6+6=18$ баллов, за выполненную и защищённую лабораторную работу студент получает максимально $6+6+6=18$ баллов. Общее максимальное количество баллов за модуль 1 составляет 50 баллов.

Вопросы для контрольной работы (ОПК-8, ОПК-9)

1. Что понимают под информационным взаимодействием в информационных системах?
2. Какие основные способы обмена данными между информационными системами используют сегодня?
3. В чём разница между синхронным и асинхронным взаимодействием систем? Приведи по одному простому примеру.
4. Что такое API и какую роль он играет в информационном взаимодействии систем?
5. В чём суть структурного программирования — назови 2–3 ключевых принципа.
6. Чем объектно-ориентированный подход принципиально отличается от структурного?
7. Какие три основные характеристики (принципы) объектно-ориентированного программирования обычно выделяют? Кратко поясни каждую одним предложением.
8. Как принципы структурного или объектно-ориентированного программирования помогают организовать надёжное взаимодействие между частями системы?

Практическое занятие №8. Составление программ, которые содержат динамическую информацию в виде динамического односвязного списка (ОПК-8, ОПК-9)

Контрольные вопросы к ПЗ

1. Что такое динамический односвязный список в C++?
2. Из каких частей обычно состоит узел односвязного списка?
3. Как выделить память под новый узел списка в C++?
4. Как добавить элемент в начало динамического односвязного списка?
5. Как пройти по всем элементам списка и вывести их значения?
6. Как удалить элемент из односвязного списка, зная его значение?

7. Почему важно освобождать память при удалении узлов списка?
8. Как проверить, пуст ли динамический односвязный список?

Практическое занятие №9. Составление программ, которые содержат динамическую информацию в виде динамического двухсвязного списка (ОПК-8, ОПК-9)

Контрольные вопросы к ПЗ

1. Что хранится в узле динамического двухсвязного списка в C++?
2. Зачем в узле двухсвязного списка нужны два указателя — на следующий и на предыдущий узел?
3. Как создать пустой двухсвязный список в C++ (какие переменные понадобятся)?
4. Как добавить элемент в начало двухсвязного списка — какие связи нужно обновить?
5. Как добавить элемент в конец двухсвязного списка, если список уже не пустой?
6. Как удалить узел из середины двухсвязного списка и не потерять связи между оставшимися узлами?
7. Как обойти двухсвязный список в прямом и в обратном направлении?
8. Какие ошибки чаще всего допускают при работе с динамической памятью в реализации двухсвязного списка?

Практическое занятие №10. Составление программ с использованием динамических структур данных в виде стеков и очередей (ОПК-8, ОПК-9)

Контрольные вопросы к ПЗ

Чтение

1. Что такое стек и по какому принципу он работает в C++?
2. Как реализовать стек на базе массива или связного списка в C++?
3. Какие основные операции поддерживают стеки (push, pop и др.) и как их запрограммировать?
4. В чём отличие стека от очереди с точки зрения порядка обработки элементов?
5. Как реализовать очередь на базе массива или связного списка в C++?
6. Какие операции типичны для очереди (enqueue, dequeue и др.) и как их реализовать?
7. Как обработать ситуацию переполнения стека/очереди или обращения к пустому контейнеру?
8. Приведи пример простой задачи, где использование стека или очереди упрощает решение — как её оформить в коде на C++?

Лабораторная работа №7. Исследование динамических структур данных (ОПК-8, ОПК-9)

Контрольные вопросы к ЛР

1. Что значит «динамическая структура данных» в C++?
2. Чем динамические структуры отличаются от статических массивов?
3. Какую роль в динамических структурах играют указатели?
4. Как в C++ выделить и освободить память для элементов динамической структуры?
5. Приведи 3 примера распространённых динамических структур данных в C++.
6. В чём главная опасность при работе с динамическими структурами и как её избежать?
7. Как проверить, что в динамической структуре нет утечек памяти?
8. Зачем в некоторых динамических структурах хранят указатель на голову и/или хвост?

Лабораторная работа №8. Исследование возможностей работы со списками, организованными в виде очереди (ОПК-8, ОПК-9)

Контрольные вопросы к ЛР

1. Какой принцип работы очереди в C++ (как расшифровывается FIFO и что он означает)?

2. Какой заголовочный файл нужно подключить, чтобы использовать очередь в C++?
3. Как объявить очередь для хранения целых чисел в C++?
4. Какая функция добавляет элемент в очередь, а какая удаляет — назови обе и кратко поясни, что делает каждая.
5. Как проверить, пуста ли очередь, и как узнать количество элементов в ней?
6. Какие два элемента очереди можно получить напрямую (без удаления), и какими методами это делается?
7. Почему в `std::queue` нельзя обратиться к элементу по индексу (например, `queue[0]`)?
8. Приведи короткий пример фрагмента кода, где в очередь добавляют три числа, а потом по одному извлекают и выводят на экран.

1.3.2. Вопросы, выносимые на зачет по дисциплине «Информационные технологии и программирование» (ОПК-8, ОПК-9)

1. Структура программы на C++. Элементы языка C++. Алфавит языка, Идентификаторы. Ключевые (зарезервированные) слова. Знаки операций. Константы. Разделители.
2. Переменные автоматические и статические. Зона видимости и время существования переменных.
3. Типы данных в C++. Переменные и константы. Операции. Выражения. Ввод и вывод данных. Логический тип.
4. Способы организация ветвлений в программах на C++. Операторы перехода. Оператор `switch` (многовариантное ветвление).
5. Операторы перехода. Операторы `break` и `continue` в циклах, примеры программ, `goto <метка>`.
6. Поточковый ввод-вывод в стиле C++, типы, пример, особенности, библиотеки
7. Одномерный массив данных. Создание, ввод и вывод данных. Одномерные массивы и указатели. Пример программы и блок-схемы.
8. Двумерный массив данных. Создание, ввод и вывод данных. Пример программы и блок-схемы.
9. Возможности типа данных `vector`. Программа демонстрирующая управления размерами векторов и копирование векторов.
10. Понятие указателя. Знак `*`. Операция разыменования
11. Строки. Организация работы со строками. Примеры использования функций для работы со строками.
12. Работа с файлами – подключение, чтение, запись, дозапись в файлы. Примеры программ.
13. Операторы цикла: с параметром `For` – синтаксис, пример, блок-схема.
14. Операторы цикла: с условием `While` – синтаксис, пример, блок-схема.
15. Операторы цикла: с условием `Do While` – синтаксис, пример, блок-схема.
16. Общая структура процедур и функций. Параметры процедур и функций. Отличия и особенности реализации.
17. Функции в C++: передача по значению, передача по адресу. Локальные и глобальные переменные. Формальные параметры и фактические аргументы.
18. Локальные и глобальные переменные. Передача одномерных массивов как параметров функции. Передача строк в качестве параметров функций. Передача многомерных массивов в функцию

Вопросы, выносимые на экзамен по дисциплине «Информационные технологии и программирование» (ОПК-8, ОПК-9)

1. Одномерный массив данных. Создание, ввод и вывод данных. Одномерные массивы и указатели. Пример программы и блок-схемы.
2. Двумерный массив данных. Создание, ввод и вывод данных. Пример программы и блок-схемы.
3. Возможности типа данных vector. Программа демонстрирующая управления размерами векторов и копирование векторов.
4. Понятие указателя. Знак *. Операция разыменования
5. Строки. Организация работы со строками. Примеры использования функций для работы со строками.
6. Работа с файлами – подключение, чтение, запись, дозапись в файлы. Примеры программ.
7. Операторы цикла: с параметром, с пред- и пост-условием – синтаксис, пример, блок-схема.
8. Общая структура процедур и функций. Параметры процедур и функций. Отличия и особенности реализации. Функции в C++: передача по значению, передача по адресу. Локальные и глобальные переменные. Формальные параметры и фактические аргументы.
9. Локальные и глобальные переменные. Передача одномерных массивов как параметров функции. Передача строк в качестве параметров функций. Передача многомерных массивов в функцию
10. Динамические структуры данных. Динамические массивы – использование, выделение и освобождение памяти. Структура узла, создание узла, добавление узла в начало/конец списка или перед/после заданного, проход по списку, поиск в списке, удаление узла.
11. Тип vector (библиотека STL) – заголовочный файл, объявление, заполнение (добавление в конец), обработка – примеры использования в программе. Динамические матрицы.
12. Списки – определение, отличие и особенности от други типов данных. Использование контейнера map (STL). Итератор. Связные списки (list) - циклический и двусвязный список.
13. Стек, дек, очередь – определение, отличие и особенности от други типов данных. Использование контейнера (STL). Основные операции со стеком.
14. Использование стека, вычисление арифметических выражений - скобочные выражения
15. Деревья - рекурсивные определения. Обход дерева – 3 вида, отличия: обход «в глубину» и «в ширину». Двоичное дерево в массиве.
16. Графы – определение, отличие и особенности от други типов данных. Связный граф. Связь с деревом. Ориентированные графы. Взвешенные графы. Жадные алгоритмы
17. Сортировка – определение. Виды алгоритмов. Метод пузырька, метод выбора, быстрая сортировка - пояснить принцип действия, отличия. Двоичный поиск.
18. Понятие ООП. Класс – понятие на примере моделей. Объявление класса, использование конструктора с/без параметров. Скрытие внутреннего устройства. Иерархия классов. Полиморфизм. Области видимости.

Примерные практические задания, выносимых на экзамен:

1. Действительные числа A,B вводятся с клавиатуры в функции vvod1, а действительные числа C,D- в функции vvod2. Составить функцию Pr, в которой можно было бы выяснить, можно ли уместить прямоугольник со сторонами A,B внутри прямоугольника со сторонами C,D. Функцию Pr включить в класс Q2, результат записать в файл и вывести на экран.
2. Составить функцию определения среднего арифметического положительных чисел, введенных с клавиатуры. Всего ввести N различных чисел в функции, входящей в CL1, результат записать в файл и вывести на экран.

3. Составить процедуру вывода на печать положительных значений выражения $y = \sin(x/5) + 5\cos 3(x-2)$ для x изменяющегося на отрезке $[-5, 12]$ с шагом 1,2. Результат записать в файл и вывести на экран.
4. Записать процедуру для вычисления выражения $F(X,Y) = (2X^3 - 4X^2 + X + 1)/(9Y^3 + Y + 4) + 3Y^2 + 5Y$. Вычислить при $X=5.1$, $Y=6.2$ вводимых с файла, результат записать в файл и вывести на экран.
5. Для арифметической прогрессии 4, 9, 14, 19... найти первые n членов этой прогрессии. Использовать цикл с параметром - FOR. Оформить в виде процедуры, результат записать в файл и вывести на экран.
6. Вывести на печать значения функции $z = \ln(x) + \operatorname{tg}(2x)$, большие 1, для x изменяющегося на отрезке $[3, 8]$ (границы отрезка задать из файла) с шагом 0,9. Оформить в виде функции, результат записать в файл и вывести на экран.
7. Дано натуральное число N . Получить наименьшее число вида $2k$, большее N . Использовать цикл с постусловием. Оформить в виде процедуры, результат записать в файл и вывести на экран.
8. Даны числа $x = 1, 2, \dots, N$. Вычислить: $3 \cdot P(X+3) \cdot P(X)$ для x , введенных с клавиатуры. Использовать функцию: $P(X) = 10 \cdot X^3 - 14 \cdot X^2 + 12 \cdot X - 2$, результат записать в файл и вывести на экран.
9. Даны круг радиуса R и квадрат со стороной A . Определить можно ли вписать их друг в друга. Оформить в виде функции, результат записать в файл и вывести на экран.
10. Найти n членов последовательности $x_1 = x_2 = x_3 = 1$; $x_k = x_{k-1} + x_{k-3}$. Использовать цикл с параметром. Оформить в виде процедуры, результат записать в файл и вывести на экран.
11. Вывести на печать значения функции $z = \sin(x) + \cos(x)$, находящиеся в интервале $(-0,2; 0,8)$ для x изменяющегося на отрезке $[4, -6]$ (границы отрезка задать из файла) с шагом 0,91. Оформить в виде функции, результат записать в файл и вывести на экран.
12. Найти среднее арифметическое отрицательных чисел введенных с клавиатуры. Всего ввести N различных чисел. Оформить в виде процедуры, результат записать в файл и вывести на экран.
13. Задав функцию, вывести на печать среднее арифметическое элементов двух одномерных массивов, введенных с клавиатуры. Массив передается функции в качестве параметра, результат записать в файл и вывести на экран.
14. Для введенных с клавиатуры чисел найти сумму положительных из них. При вводе отрицательного числа суммирование завершить. Оформить в виде процедуры, результат записать в файл и вывести на экран.

2. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

2.1 Порядок и методика проведения текущего контроля и промежуточной аттестации

Предусмотрены следующие виды контроля:

- текущий контроль по каждому модулю – в форме написания контрольной работы для оценки теоретических знаний;
 - текущий контроль по каждому модулю – в форме отчетов по лабораторным работам и практическим занятиям для оценки практических навыков;
 - промежуточная аттестация по дисциплине – в форме зачета, экзамена
- Текущий контроль успеваемости в форме защиты лабораторных и практических проводится в два этапа:

- 1-й этап: допуск к выполнению лабораторной работы – проводится в форме письменной «летучки» (5-10 мин) с целью контроля знаний студентов теоретической части лабораторной работы и готовности к выполнению практических исследований;
- 2-й этап выполняется по окончании каждой лабораторной работы (практического занятия) в форме индивидуального собеседования по выполненным исследованиям или расчетам. Проводится с

целью контроля закрепления теоретической части материала и степени отработки студентом практических навыков исследования на аппаратуре.

Контрольные работы выполняются в виде короткого письменного ответа на один вопрос, изученный на предыдущей лекции в начале каждой последующей лекции. Ответ на вопрос дается в течение 5-10 минут. Таким образом, после лекционного курса каждого модуля формируется общая оценка за теоретические знания.

С целью повышения качества обучения за счет побуждения студентов к активной текущей учебной работе, четкого и оперативного контроля всего хода учебного процесса, снижения роли случайных и субъективных факторов при оценивании учебной деятельности студентов в образовательном процессе реализована модульно-рейтинговая система.

Правила ее использования прописаны в «Положении об МРС».

Набранные обучающимся баллы могут быть переведены в оценку:

- «не зачет» - от 0 до 40 от максимального количества баллов;
- «зачет» - от 41 до 100 максимального количества баллов.

Для получения зачета студенту достаточно набрать от 41 и более баллов.

Соотношения максимального количества баллов, полученных студентом по блокам модулей, показаны в Таблице .

Таблица 1 – Распределение баллов по блокам модулей дисциплины «Информационные технологии и программирование»

Модуль	Всего баллов (Максимальное значение)	Теоретический блок (Контрольная работа)	Практический блок (Распределение баллов по занятиям)
Модуль 1	50	14	36
Модуль 2	50	14	36
Зачет		100	

Как правило, теоретический блок оценивается по результатам контрольной работы. Практический блок оценивается по результатам выполнения лабораторных работ и заданий на практических занятиях.

На зачете и экзамене производится оценка тех компетенций, которые должны быть в той или иной форме освоены в процессе изучения. Рекомендуются формировать вопросы в билетах к зачету таким образом, чтобы преподаватель смог оценить все компетенции данной дисциплины.

2.2 Методика проведения зачета и экзамена в группах заочной форм

Зачет по дисциплине «Информационные технологии и программирование» у студентов заочной форм обучения проводится письменно по классической методике. Вопросы сгруппированы в билетах. В каждом билете содержится по 2-3 вопроса, скомпонованные таким образом, чтобы преподаватель смог оценить все компетенции данной дисциплины. Количество билетов должно быть не менее числа студентов в группе. Студенты готовят письменные ответы на вопросы.

Экзамен по дисциплине «Информационные технологии и программирование» у студентов очной и заочной форм обучения проводится письменно по классической методике. Вопросы сгруппированы в билетах. В каждом билете содержится по 2-3 вопроса, скомпонованные таким образом, чтобы преподаватель смог оценить все компетенции данной дисциплины. Количество билетов должно быть не менее числа студентов в группе. Студенты готовят письменные ответы на вопросы.

3. Промежуточная аттестация

Промежуточная аттестация проводится в форме зачета.

Зачет проводится по расписанию экзаменационной сессии в письменном виде. Количество вопросов в билете к зачету – 2. Проверка ответов и объявление результатов производится в день

зачета. Результаты аттестации заносятся в зачетную ведомость и зачетную книжку студента.

Студенты, не прошедшие промежуточную аттестацию по графику сессии, должны ликвидировать задолженность в установленном порядке.

Образец билета к зачету

	МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ Северо-Кавказский филиал ордена Трудового Красного Знамени федерального государственного бюджетного образовательного учреждения высшего образования «Московский технический университет связи и информатики»»	Утверждаю Зав. кафедрой ИВТ _____ Соколов С.В. " __ " _____ 20__ г.
Направление подготовки: 09.03.01. Информатика и вычислительная техника Курс: 2, семестр 3 Дисциплина: Информационные технологии и программирование		
Билет № 1 1. Работа с файлами – подключение, чтение, запись, дозапись в файлы. Примеры программ. 2. Функции в C++; передача по значению, передача по адресу. Локальные и глобальные переменные. Формальные параметры и фактические аргументы Доцент кафедры ИВТ _____ С.А. Швидченко " __ " _____ 20__ г.		

Один комплект отпечатанных билетов к зачету, подписанных преподавателем кафедры и утвержденных заведующим кафедрой хранится у заведующего кафедрой, другой комплект – у преподавателя, ведущего дисциплину.

Промежуточная аттестация во 2 семестре проводится в форме экзамена. Экзамен проводится по расписанию промежуточной аттестации в устном виде по экзаменационным билетам. Результаты аттестации заносятся в ведомость и зачетную книжку обучающегося.

Образец экзаменационного билета.

	МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ Северо-Кавказский филиал ордена Трудового Красного Знамени федерального государственного бюджетного образовательного учреждения высшего образования «Московский технический университет связи и информатики»»	Утверждаю Зав. кафедрой ИВТ _____ Соколов С.В. " __ " _____ 20__ г.
Направление подготовки: 09.03.01. Информатика и вычислительная техника Курс: 2, семестр 4 Дисциплина: Информационные технологии и программирование		

Билет № 1

1. Сортировка – определение. Виды алгоритмов. Метод пузырька, метод выбора, быстрая сортировка - пояснить принцип действия, отличия. Двоичный поиск
2. Стек, дек, очередь – определение, отличие и особенности от других типов данных. Использование контейнера (STL). Основные операции со стеком.
3. Вывести на печать значения функции $z=\sin(x)+\cos(x)$, находящиеся в интервале $(-0,2; 0,8)$ для x изменяющегося на отрезке $[4,-6]$ (границы отрезка задать из файла) с шагом 0,91. Оформить в виде функции, результат записать в файл и вывести на экран

Доцент кафедры ИВТ

С.А. Швидченко

"__" _____ 20__ г.

Один комплект отпечатанных билетов к зачету, подписанных преподавателем кафедры и утвержденных заведующим кафедрой хранится у заведующего кафедрой, другой комплект – у преподавателя, ведущего дисциплину.

4. Тестовые задания для проведения оценки сформированности компетенций (ОПК-8, ОПК-9)

Данная дисциплина формирует у обучающегося ОПК-8, ОПК-9 компетенцию, объективно оценить которую можно при ответе студента на тест или при выполнении практического или лабораторного задания.

Один комплект тестовых вопросов с указанием правильных ответов хранится у заведующего кафедрой, другой комплект – у преподавателя, ведущего дисциплину.

Тестовые задания позволяют оценить уровень сформированности компетенции ОПК-8, ОПК-9.

Задания распределены по блокам в соответствии с уровнем сложности. Каждый блок содержит номер задания, и текст задания.

Базовый уровень содержит примерно 50 % заданий.

Он формируется из заданий с выбором одного или нескольких верных ответов из предложенных или воспроизведения фактического материала (терминология, факты, классификация, параметры и др.);

Повышенный уровень в среднем составляет 35 % заданий.

В нем используются задания на сопоставление, сравнение, установление последовательности;

Высокий уровень в среднем составляют – 15 % заданий.

В задании используются: решения нетиповых задач, алгоритмы, доказательства, задания с развернутым ответом – определения, перечисление, изображение схемы, структуры и др.

Количество заданий по каждой компетенции равно 20. Всего 40 заданий.

Ключи к ответам представлены в отдельной таблице.

ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения

ОПК-9: Способен осваивать методики использования программных средств для решения практических задач

БЛОК А (базовый уровень) – Задание с выбором одного или нескольких верных ответов из предложенных

Инструкция по тестам Блока А: Прочитайте текст и выберите один или несколько правильных ответов

№ задания	Тексты заданий
1	Объем цифровой информации может измеряться в 1. Килокалориях, 2. Мегагерцах, 3. Мегабайтах , 4. Килоботах.
2	Шифрование информации проводится с целью 1. Удобства передачи по каналам связи 2. Сокращения информации 3. Защиты информации от помех
3	Скорость передачи цифровой информации может измеряться в 1. Килобайтах 2. Мегабитах в секунду 3. Мегаголов в час 4. Пикселях
4	Кодирование информации производится с целью 1. Удобства передачи по каналам связи 2. Сокращения информации от третьих лиц 3. Отличия цифровой от аналоговой передачи данных
5	Информация — это 1. Сведения (сообщения, данные) 2. Цифровые системы передачи 3. Цифровые устройства хранения
6	К преимуществам цифровой информации относятся: 1. Возможность математической обработки 2. Возможность сжатия 3. Наличие ошибок квантования 4. Идеально точное преобразование аналоговых сигналов в цифровые
7	Что такое бит? 1. Байт 2. Единица измерения количества информации 3. Единица измерения количества электричества
8	Почему компьютер работает с двоичными числами? 1. Основой компьютера являются переключающие устройства с двумя устойчивыми состояниями. 2. Проще перевести числа из десятичной в двоичную систему 3. Другая система отсутствует.
9	Компьютерный текст кодируется таблицами: 1. ASCII 2. Unicode 3. Bitmap 4. Teletype
10	Принтеры бывают: 1. Струйные, 2. Лазерные, 3. Матричные , 4. Системные, 5. Копировальные.

БЛОК Б (повышенный уровень) – Задание закрытого типа на установление соответствия

Инструкция: Прочитайте текст и установите соответствие

№ задания	Тексты заданий
11	Установите соответствие между операционными системами и их разработчиками: 1. Windows 2. MAC OS 3. Linux А. Линус Торвальдс Б. Бил Гейтс В. Стив Джобс
12	Расставьте в порядке соответствия аппаратные средства компьютера и характеризующие их параметры 1. Процессор 2. Память 3. Сетевой адаптер 4. Блок питания А. Напряжение и ток Б. Мегабиты в секунду В. Гигабайты Г. Количество операций в секунду
13	Установите соответствие компьютерных систем и реализующим их устройствам 1. ПЗУ 2. ОЗУ 3. Устройства ввода 4. Устройства вывода А. Принтеры, мониторы Б. Сканеры, микрофоны В. Планки оперативной памяти Г. BIOS, DVD-disk
14	Установите соответствие между типом графического изображения и его цифровым описанием 1. Черно-белое 2. Полутоновое 3. Полноцветное А. 24-битное Б. 8-битное Г. 1-битное

БЛОК Б (повышенный уровень) – Задание открытого типа на дополнение

Инструкция: Прочитайте и дополните пропущенный текст

№ задания	Тексты заданий
15	Расставьте типы компьютеров по возрастанию вычислительной мощности 1. Суперкомпьютеры 2. Микрокомпьютеры 3. Мэйнфреймы 4. Персональные

№ задания	Тексты заданий
16	Расставьте в порядке хронологии появления виды программирования 1. Объектно-ориентированное программирование, 2. Процедурные языки, 3. Ассемблеры, 4. Программирование в машинных кодах.
17	Расставьте в порядке увеличения следующие числа: 1. 111-двоичная, 2. 111-шестнадцатичная, 3. 111-восьмиричная, 4. 111-десятичная.

БЛОК В (высокий уровень) – Задание открытого типа с развернутым ответом

Инструкция: Прочитайте текст и запишите ответ

№ задания	Тексты заданий
18	Python, Java, C++ относятся к языкам _____
19	Google, Bing (Microsoft), Яндекс, Rambler являются _____
20	Совокупностью программ, алгоритмов и данных, которые позволяют компьютеру выполнять определённые задачи и функции является _____

Система оценивания выполнения тестовых заданий по компетенции ОПК-8, ОПК-9

Номер задания	Указания по оцениванию	Результат оценивания (баллы, полученные за выполнение задания/характеристика правильности ответа)
Задания 1-10	Задания с выбором одного или нескольких верных ответов из предложенных считается верным, если правильно указана цифра или цифры, означающие верные ответы.	Совпадение с верным ответом оценивается 1 баллом; неверный ответ или его отсутствие - 0 баллов.
Задания 11-17	Задания закрытого типа на установление соответствия считается верным, если правильно установлены все соответствия (позиции из одного столбца верно сопоставлены с позициями другого)	Полное совпадение с верным ответом оценивается 1 баллом; неверный ответ или его отсутствие - 0 баллов.

Задания 18-20	Задание открытого типа с развернутым ответом считается верным, если ответ совпадает с эталонным по содержанию и полноте.	Полный правильный ответ на задание оценивается 3 баллами; если допущена одна ошибка/неточность/ответ правильный, но не полный - 1 балл, если допущено более одной ошибки/ответ неправильный/ответ отсутствует- 0 баллов.
------------------	--	--

Методика проведения тестирования

Максимальное время выполнения теста - 60 минут.

Количество тестовых заданий в тесте не менее 20.

Для проведения контроля остаточных знаний компьютер формирует в случайном порядке 20 тестовых заданий.

При прохождении тестового контроля знаний студент должен выбрать один или несколько правильных ответов в каждом тестовом задании.

В случае правильного ответа на 16 и более тестовых заданий (81-100% правильных ответов) студент подтверждает освоение компетенций с уровнем «отлично». При ответе на 12-15 тестовых заданий (61-80% правильных ответов) студент подтверждает освоение компетенций с уровнем «хорошо». При ответе на 8-14 тестовых заданий (41-60% правильных ответов) студент подтверждает освоение компетенций с уровнем «удовлетворительно». При ответе менее чем на 8 тестовых заданий (0-40% правильных ответов) студент не подтверждает необходимый уровень знаний и оценивается «неудовлетворительно».

ОПК-8: Способен разрабатывать алгоритмы и программы, пригодные для практического применения

ОПК-9: Способен осваивать методики использования программных средств для решения практических задач

2 семестр

БЛОК А (базовый уровень) – Задание с выбором одного или нескольких верных ответов из предложенных

Инструкция по тестам Блока А: Прочитайте текст и выберите один или несколько правильных ответов

№ задания	Тексты заданий
1	При написании безопасного кода необходимо использовать объектно-ориентированное программирование. ООП – это: 1. Объектно-ориентированное программирование, методология программирования, основанная на понятии объектов 2. Общеобязательный обработчик протоколов, используемый для передачи данных в компьютерных сетях 3. Оператор цикла, который повторяет блок кода до тех пор, пока условие истинно
2	В безопасном коде зачастую используют стек. Стек – это структура данных, работающая по принципу: 1. "последний вошел - первый вышел" 2. "первый вошел - первый вышел" 3. "последний вошел - последний вышел"
3	Оператор switch в C++ – это: 1. Оператор условия, который выполняет блок кода при определенном условии 2. Оператор цикла, который повторяет блок кода до тех пор, пока условие истинно 3. Оператор множественного выбора, который выполняет блок кода в зависимости

№ задания	Тексты заданий
	от значения переменной
4	Выберите верное утверждение про циклы: 1. Цикл for не работает со словарями 2. Цикл while может быть не выполнен ни разу 3. Нельзя использовать отрицательный шаг в for
5	В ООП на C++ не существует следующего модификатора доступа: 1. protected 2. included 3. private
6	Полиморфизм в C++ – 1. Это возможность создавать классы, которые могут использовать методы родительского класса, но при этом могут переопределять их поведение 2. Это возможность создавать объекты разных классов с одинаковыми именами методов 3. Это возможность создавать методы с одинаковыми именами в разных классах, которые выполняют одинаковые действия
7	Выберите верное утверждение для типов данных языка C++: 1. Переменная типа bool занимает столько же места, как и переменная типа int 2. Переменная типа float занимает столько же места, как и переменная типа int 3. Размер переменной типа str зависит от количества символов в строке
8	Выберите верный синтаксис объявления класса на C++: 1. class MyClass { int x; void func(); }; 2. class MyClass int x; void func() } 3. class MyClass (int x; void func();)
9	Наследование в ООП – это 1. Механизм, который позволяет создавать новый класс на основе уже существующего класса 2. Механизм, который позволяет передавать данные между объектами класса 3. Механизм, который позволяет создавать новые объекты на основе существующих объектов
10	Данный фрагмент кода на C++ выводит на экран <pre>#include <iostream> using namespace std; int main() { int num = 5; for (int i = 1; i <= 10; ++i) { cout << num << "*" << i << "=" << (num * i) << endl; } return 0; }</pre> 1. Таблицу умножения на 5 2. Произведение первых 10 натуральных чисел, увеличенное в 5 раз 3. Программа выдаст ошибку во время выполнения

БЛОК Б (повышенный уровень) – Задание закрытого типа на установление соответствия
Инструкция: Прочитайте текст и установите соответствие

№ задания	Тексты заданий
11	Установите соответствие между фрагментами кода на C++ и задачей: <pre>long long productOdd(int n) { long long prod = 1; for (int i = 1; i <= n; i += 2) { prod *= i; } return prod; }</pre> 1. } <pre>long long productEven(int n) { if (n < 2) return 1; long long prod = 1; for (int i = 2; i <= n; i += 2) { prod *= i; } return prod; }</pre> 2. } <pre>long long factorial(int n) { long long res = 1; for (int i = 2; i <= n; ++i) { res *= i; } return res; }</pre> 3. } <pre>long long productMod(int n) { long long res = 1; for (int i = 1; i <= n; ++i) { res = (res * i) % MOD; } return res; }</pre> 4. } А. Вычисление произведения нечетных натуральных чисел до n Б. Вычисление произведения четных натуральных чисел до n В. Вычисление факториала n! Г. Вычисление произведения чисел по модулю
12	Установите соответствие: 1. метод класса, реализация которого отсутствует 2. виртуальный метод, который переопределяет виртуальный метод базового класса 3. метод класса, который инициализирует атрибуты класса при создании объекта А) абстрактный метод Б) метод с использованием ключевого слова "super" В) метод-конструктор
13	Установите соответствие между названиями функций для типа данных string языка C++ и их назначением: 1) substr(pos, len) 2) length() 3) size() А) Возвращает длину строки

№ задания	Тексты заданий
	Б) Возвращает подстроку, начиная с позиции pos длиной len символов В) Возвращает количество символов в строке
14	Установите соответствие между названиями операторов на C++ и их назначением: 1) for 2) while 3) switch А) цикл с заданным числом повторений Б) цикл с предусловием В) оператор множественного выбора

БЛОК Б (повышенный уровень) – Задание закрытого типа на дополнение
Инструкция: Прочитайте и дополните пропущенный текст

№ задания	Тексты заданий
15	В программе на C++ введите слово, которое заменено знаками вопроса: <pre>#include <iostream> using namespace std; int main() { int i = 0; ??? (i < 5) { cout << "i = " << i << endl; ++i; } return 0; }</pre>
16	В программе на Python введите слово, которое заменено знаками вопроса: <pre>#include <iostream> void printSquare(int n) { std::cout << "Квадрат числа " << n << " равен " << n * n << std::endl; } int main() { printSquare(5); ??? (int i = 1; i <= 3; ++i) { printSquare(i); } return 0; }</pre>
17	Напишите, что будет выведено на экран при выполнении нижеприведенного кода: <pre>#include <iostream> int main() {</pre>

№ задания	Тексты заданий
	<pre>int a = 5, b = 3; std::cout << a << "=Z(" << b << ")\n"; return 0; }</pre>

БЛОК В (высокий уровень) – Задание открытого типа с развернутым ответом
Инструкция: Прочитайте текст и запишите ответ

№ задания	Тексты заданий
18	В языке C++ функции, начинающиеся со слова <code>lambda</code> называются: _____
19	Динамическая структура доступа, реализующая принцип: «последний пришел, первый ушел», получила название: _____
20	Неупорядоченная коллекция пар «ключ-значение» называется _____

Методические рекомендации по освоению дисциплины

1. Методические рекомендации преподавателю

Дисциплина «**Информационные технологии и программирование**» включает в себя:

- лекционные занятия – 36 часов;
- практические занятия – 36 часов;
- лабораторные работы – 36 часов;
- промежуточную аттестацию III семестр– зачет.
- промежуточную аттестацию IV семестр – экзамен.

Перед началом изучения дисциплины преподаватель должен ознакомить студентов с рабочей программой и оценочными материалами по дисциплине, с видами учебной и самостоятельной работы, перечнем литературы и интернет-ресурсов, с формами текущей и промежуточной аттестации, с критериями оценки качества знаний для итоговой оценки по дисциплине.

При проведении лекций, преподаватель:

- 1) формулирует тему и цель занятия, объявляет учебные вопросы;
- 2) излагает основные теоретические положения;
- 3) с помощью технических средств обучения и/или под запись дает определения основных понятий, расчетных формул;
- 4) проводит примеры из отечественного и зарубежного опыта, дает текущие статистические данные для наглядного и образного представления изучаемого материала;
- 5) в конце занятия выдает вопросы для самостоятельного изучения.

На занятиях лабораторного цикла следует обратить внимание на соответствие выбираемых студентом средств выполнения решаемым в работе задачам.

Каждая лабораторная работа должна быть оформлена и защищена в соответствии с требованиями. Защита производится после оформления отчета по работе.

Во время выполнения заданий в учебной аудитории студент может консультироваться с преподавателем, определять наиболее эффективные методы решения поставленных задач. Если какая-то часть задания остается не выполненной, студент может продолжить её выполнение во время внеаудиторной самостоятельной работы.

Для оценки полученных знаний и освоения учебного материала по каждому разделу и в целом по дисциплине преподаватель использует формы текущего, промежуточного и итогового контроля знаний обучающихся.

2. Методические рекомендации студентам

Приступая к изучению учебной дисциплины, студенты должны ознакомиться с рабочей программой, оценочными материалами, учебной, научной и методической литературой, имеющейся в ЭИОС СКФ МТУСИ,

Глубина усвоения дисциплины зависит от активной и систематической работы студента на лекциях и практических занятиях, а также в ходе самостоятельной работы, по изучению рекомендованной литературы.

2.1 Методические указания для обучающихся по лекционным занятиям

Важно сосредоточить внимание на содержании лекции. Это поможет лучше воспринимать учебный материал и уяснить взаимосвязь проблем по всей дисциплине.

Основное содержание лекции целесообразнее записывать в тетради в виде ключевых фраз, понятий, тезисов, обобщений, схем, опорных выводов. Необходимо обращать внимание на термины,

формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставлять в конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющей материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений.

С целью уяснения теоретических положений, разрешения спорных ситуаций необходимо задавать преподавателю уточняющие вопросы. Для закрепления содержания лекции в памяти, необходимо во время самостоятельной работы внимательно прочесть свой конспект и дополнить его записями из учебников и рекомендованной литературы.

Конспектирование читаемых лекций и их последующая доработка способствует более глубокому усвоению знаний, и поэтому являются важной формой учебной деятельности студентов.

2.2 Методические указания для обучающихся по подготовке к лабораторным работам

Лабораторные занятия по любой учебной дисциплине проводятся в соответствии с рабочей программой этой учебной дисциплины в установленные расписанием часы.

Инструктаж по технике безопасности, правилам пожарной безопасности и правилам внутреннего распорядка в лаборатории проводится преподавателем на первом лабораторном занятии. На этом же занятии студентам сообщаются:

- программа всего предстоящего лабораторного цикла;
- условия взаимодействия студентов с преподавателем в процессе выполнения лабораторных работ;
- условия контроля самостоятельной работы студентов, включая правила оформления отчетов по лабораторным работам и их последующей защиты;
- другая необходимая информация.

Организация лабораторных работ включает:

- самостоятельную внеаудиторную подготовку студента к выполнению каждой отдельной лабораторной работы в соответствии с ее программой в рамках часов, выделенных на самостоятельную проработку материала в программе дисциплины;
- входной контроль преподавателем степени подготовленности каждого студента к выполнению лабораторных работ;
- выполнение программы лабораторных работ в полном объеме;
- оформление отчета и его защиту каждым студентом в установленные сроки;
- формирование преподавателем рейтингов каждого из студентов по результатам выполнения и защиты им отдельных лабораторных работ и их циклов (используется Модульно-рейтинговая система).

Оценка качества выполнения лабораторных работ каждым студентом производится преподавателем отдельно за подготовку к работе, ее выполнение и защиту.

2.3 Методические указания для обучающихся по подготовке к практическим занятиям

Целью практических занятий является закрепление теоретических знаний, полученных при изучении дисциплины.

При подготовке к практическому занятию целесообразно выполнить следующие рекомендации:

- изучить основную литературу; ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях: журналах, газетах и т.д.;
- при необходимости доработать конспект лекций. При этом учесть рекомендации преподавателя и требования учебной программы.

При выполнении практических занятий основным методом обучения является самостоятельная работа студента под управлением преподавателя. На них пополняются теоретические знания студентов, их умение творчески мыслить, анализировать, обобщать изученный материал, проверяется отношение студентов к будущей профессиональной деятельности.

Оценка выполненной работы осуществляется преподавателем комплексно: по результатам

выполнения заданий, устному сообщению и оформлению работы.

После подведения итогов занятия студент обязан устранить недостатки, отмеченные преподавателем при оценке его работы.

3. Методические указания для обучающихся по подготовке контрольной работы

В ходе выполнения контрольной работы студент решает следующие задачи:

1. Выполняется контрольное задание №1 по теме «Алгоритмы на основе линейной структуры», пишется программа на языке Python, оформляется отчет с блок-схемой.
2. Выполняется контрольное задание №1 по теме «Алгоритмы на основе разветвляющейся структуры», пишется программа на языке Python, оформляется отчет с блок-схемой.
3. Выполняется контрольное задание №1 по теме «Алгоритмы на основе регулярной циклической структуры», пишется программа на языке Python, оформляется отчет с блок-схемой.
4. Выполняется контрольное задание №1 по теме «Списки / Одномерный массив», пишется программа на языке C++, оформляется отчет с блок-схемой.

Порядок выполнения контрольной работы приведен в методических материалах по ее выполнению. Критерии оценки контрольной работы приведены в оценочных материалах дисциплины.

4. Методические рекомендации для обучающихся по самостоятельной работе

Самостоятельная работа студентов является составной частью учебной работы и имеет целью закрепление и углубление полученных знаний и навыков, поиск и приобретение новых знаний, в том числе с использованием автоматизированных обучающих курсов (систем), а также выполнение учебных заданий, подготовку к предстоящим занятиям, зачетам и экзаменам.

Самостоятельная работа во внеаудиторное время включает в себя:

- повторение лекционного материала;
- подготовки к лабораторным работам (практическим занятиям);
- изучения учебной и научной литературы;
- решения задач, выданных на практических занятиях;
- подготовки к тестированию;
- выделение наиболее сложных и проблемных вопросов по изучаемой теме, получение разъяснений и рекомендаций по данным вопросам с преподавателями кафедры на их еженедельных консультациях.
- проведение самоконтроля путем ответов на вопросы текущего контроля знаний, решения представленных в учебно-методических материалах кафедры задач, тестов, по изучаемой дисциплине.

Перед выполнением внеаудиторной самостоятельной работы преподаватель проводит инструктаж (консультацию) с определением цели задания, его содержания, сроков выполнения, основных требований к результатам работы, критериев оценки, форм контроля и перечня источников и литературы.

Обычно постановку задачи обучаемым на проведение самостоятельной работы преподаватель осуществляет на одном из занятий, предшествующем данному.

Методику самостоятельной работы все обучаемые выбирают индивидуально.