МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Донской казачий государственный институт пищевых технологий и бизнеса (филиал) федерального государственного бюджетного образовательного учреждения «Московский государственный университет технологий и управления имени К.Г. Разумовского (Первый казачий университет)»

УТВЕРЖДАЮ
И.о. директора ДКГИПТиБ (филиала)
_____ Г.В. Павлятенко
« 28 » марта 2025 г.

Рабочая программа дисциплины (модуля)

Б1.О.03.01 МОДУЛЬ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ
Основы алгоритмизации и программирования

Кафедра:	Физика, математика и информационные технологии
Направление подготовки:	09.03.03 Прикладная информатика
Направленность (профиль):	Управление проектами в области информационных технологий
Квалификация выпускника:	Бакалавр
Форма обучения:	очно-заочная
Год набора:	2025
Общая трудоемкость:	180 часов/5 з.е.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 00AA7853CEFA35F0837BA866654F3503A6
Владелец: Павлятенко Геннадий Владимирович
Действителен: с 08.12.2025 до 03.03.2027

Ростов-на-Дону, 2025 г.

Программу составил(и):
Преподаватель Горбунов В.С.

Рабочая программа дисциплины (модуля)
"Основы алгоритмизации и программирования"
разработана составлена на основании учебного плана, утвержденного ученым советом 27 марта 2025 г. протокол № 9 в соответствии
с ФГОС ВО Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

Руководитель ОПОП
канд. экон. наук, доцент Левчук В.В._____

Рабочая программа обсуждена на заседании обеспечивающей кафедры
Физика, математика и информационные технологии
Протокол от 27 марта 2025 г. № 3
Зав. кафедрой Левчук Валерия Владимировна_____

СОДЕРЖАНИЕ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ И ОБЪЕМ С РАСПРЕДЕЛЕНИЕМ ПО СЕМЕСТРАМ

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

6. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

9. ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

1.1. Цели:

Целью освоения дисциплины Основы алгоритмизации и программирования является формирование у студентов базовых компетенций в области программирования, в том числе ознакомление студентов с понятием алгоритма, основными видами алгоритмов и способами их составления, алгоритмами некоторых стандартных процессов. В качестве языка обучения используется С++.

1.2. Задачи:

Задачи освоения дисциплины Основы алгоритмизации и программирования:
- формирование у студента навыка перевода конкретной задачи на алгоритмический язык;
- развитие способностей составления кода программы и ее отладки.

2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ И ОБЪЕМ С РАСПРЕДЕЛЕНИЕМ ПО СЕМЕСТРАМ

Цикл (раздел) ОП: Б1.О

Дисциплина относится к обязательной части ОПОП и обязательна для освоения.

Связь с последующими дисциплинами (модулями), практиками

№ п/п	Наименование	Семестр	Шифр компетенции
1	Ознакомительная практика	4	УК-7.1, УК-7.2, УК-7.3, ОПК-7.1, ОПК-7.2, ОПК-7.3, УК-4.1, УК-4.2, УК-4.3
2	Разработка программных приложений	2	ОПК-7.1, ОПК-7.2, ОПК-7.3
3	Программирование	3	ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-7.1, ОПК-7.2, ОПК-7.3
4	Технологическая (проектно-технологическая) практика	6	УК-1.1, УК-1.2, УК-1.3, УК-2.1, УК-2.2, УК-2.3, УК-3.1, УК-3.2, УК-3.3, УК-3.4, УК-3.5, УК-3.6, УК-5.1, УК-5.2, УК-5.3, УК-8.1, УК-8.2, УК-8.3, УК-8.4, УК-8.5, УК-8.6, УК-9.1, УК-9.2, УК-9.3, УК-10.1, УК-10.2, УК-10.3, ОПК-1.1, ОПК-1.2, ОПК-1.3, ОПК-2.1, ОПК-2.2, ОПК-2.3, ОПК-3.1, ОПК-3.2, ОПК-3.3, ОПК-4.1, ОПК-4.2, ОПК-4.3, ОПК-5.1, ОПК-5.2, ОПК-5.3, ОПК-6.1, ОПК-6.2, ОПК-6.3, ОПК-7.1, ОПК-7.2, ОПК-7.3, ОПК-8.1, ОПК-8.2, ОПК-8.3, ОПК-9.1, ОПК-9.2, ОПК-9.3, УК-4.4, УК-4.5, УК-4.6, УК-6.1, УК-6.2, УК-6.3

Распределение часов дисциплины

Семестр (<Курс>.<Семестр на курсе>)	1 (1.1)		Итого	
Недель	17 3/6			
Вид занятий	УП	РП	УП	РП
Лекции	4	4	4	4
Лабораторные	16	16	16	16
В том числе электрон.	56	56	56	56
В том числе в форме практ.подготовки	2	2	2	2
Итого ауд.	20	20	20	20
Контактная работа	20	20	20	20
Сам. работа	124	124	124	124
Часы на контроль	36	36	36	36
Итого	180	180	180	180

Вид промежуточной аттестации:

Экзамен 1 семестр

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), СООТНЕСЕННЫЕ С РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Процесс изучения дисциплины (модуля) направлен на формирование следующих компетенций и индикаторов их

ОПК-7:Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

ОПК-7.1: Знает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и принципы функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ; современные языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий

ОПК-7.2: Умеет выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий

ОПК-7.3: Владеет навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименования разделов, тем, их краткое содержание и результаты освоения /вид занятия/	Семестр	Часов	Интеракт.	Прак. подг.	Индикаторы достижения компетенции	Оценочные средства
	Раздел 1.Основы алгоритмизации и программирования						
1.1	**Тема 1. Теоретические основы алгоритмизации и программирования. Основные виды алгоритмов. Краткое содержание: алгоритм, свойства алгоритма, способы описания алгоритма, назначение функциональных блоков, основные этапы решения задач. Линейный алгоритм, разветвляющийся алгоритм, циклический алгоритм. Знать: свойства алгоритма, способы описания алгоритмов, основные этапы решения задач, основные виды алгоритмов, правила их описания Уметь: разрабатывать алгоритмы решения задач; описывать алгоритмы на языке псевдокода или на естественном языке; описывать каждый вид алгоритма Владеть: основными понятиями и принципами алгоритмизации и программирования; навыками реализации основных видов алгоритмов /Ср/**	1	17	0	0	ОПК-7.1,ОПК-7.2,ОПК-7.3	вопросы для самоподготовки
1.2	**Тема 2. Язык программирования С++. Краткое содержание: язык программирования С++, среды разработки, переменные, функции, операторы цикла, условные операторы, массивы Знать: Основные понятия и концепции языка С++; Синтаксис объявления и инициализации переменных различных типов; Основные операторы С++; Синтаксис и принципы работы условных операторов; Типы циклов Уметь: Устанавливать и настраивать среду разработки С++; Создавать, компилировать и запускать**	1	17	0	0	ОПК-7.1,ОПК-7.2,ОПК-7.3	вопросы для самоподготовки

	программы на C++; Применять различные типы данных в зависимости от задачи; Использовать условные операторы для ветвления алгоритма; Организовывать циклические процессы с помощью операторов цикла; Разбивать программу на функциональные блоки Владеть: Навыками написания программ на C++ с использованием базового синтаксиса; Методами решения типовых алгоритмических задач; Приемами эффективной работы с массивами; Практическими навыками работы в среде разработки C++ /Ср/						
1.3	**Тема 3. Управляющие инструкции. Краткое содержание: оператор цикла: for, do-while, оператор выбора switch, вложенные условные операторы, вложенные операторы цикла, цикл по коллекции, генерация и перехват исключений, инструкция безусловного перехода Знать: Синтаксис и особенности работы каждого оператора; Правила работы switch, допустимые типы выражений; Принципы работы вложенных if-else; Взаимодействие циклов друг с другом; Принципы работы с коллекциями и их итерация; Виды исключений, их обработка через try-catch; Назначение break, continue, return Уметь: Использовать циклы для организации повторяющихся вычислений; Применять switch для обработки множественных условий; Использовать вложенные конструкции для сложных логических проверок; Организовывать вложенные циклы для обработки многомерных данных; Использовать цикл foreach для перебора элементов; Применять исключения для обработки ошибок Владеть: навыками оптимизировать выполнение циклов, выбирать подходящий вид цикла; навыками заменять громоздкие if-else конструкцией switch при необходимости; навыками обеспечивать читаемость и структурированность кода /Ср/**	1	18	0	0	ОПК-7.1,ОПК-7.2,ОПК-7.3	вопросы для самоподготовки
1.4	**Тема 4. Указатели, массивы и ссылки. Краткое содержание: указатели, массивы и указатели, ссылки, динамическое выделение памяти, особенности символьных массивов, двумерные массивы, массивы указателей Знать: Определение указателя, разыменование, арифметика указателей; Связь массивов и указателей, передача массивов в**	1	18	0	0	ОПК-7.1,ОПК-7.2,ОПК-7.3	вопросы для самоподготовки

	функции; Отличия ссылок от указателей, применение ссылок; Функции new и delete, malloc и free, утечки памяти; Особенности хранения строк в С/С++; Организация двумерных массивов, способы их представления в памяти Уметь: Объявлять и использовать указатели, разыменовывать их, применять арифметику указателей; Использовать указатели для работы с массивами, передавать массивы в функции по указателю; Объявлять ссылки, передавать параметры по ссылке в функции; Выделять и освобождать динамическую память, избегать утечек; Создавать и обрабатывать двумерные массивы, передавать их в функции Владеть: навыками работы с указателями в коде, анализ адресов памяти; навыками работы с массивами с использованием указателей; навыками работы с объектами и функциями; навыками управление динамической памятью, отладка утечек /Ср/						
1.5	**Тема 5. Функции. Краткое содержание: объявление и описание функции, перегрузка функций, значение аргументов по умолчанию, рекурсия, механизмы передачи аргументов функциям, передача указателя аргументом функциям, передача массива аргументов функциям, передача текста в функцию, указатель как результат функции, ссылка как результат функции, динамический массив как результат функции, указатель на функцию Знать: Синтаксис объявления и определения функции; Принципы перегрузки, правила выбора перегруженной функции; Принципы работы рекурсии, стек вызовов, базовый и рекурсивный случай; Особенности передачи массива (фактически передача указателя); Синтаксис указателей на функции, применение Уметь: Описывать и вызывать функции; Создавать перегруженные функции; Использовать параметры по умолчанию в функциях; Создавать рекурсивные функции, анализировать их работу; Передавать аргументы разными способами; Передавать указатели в функции, изменять данные через них; Передавать массивы в функции, работать с ними; Передавать строки разными способами; Владеть: техникой перегрузки функций для гибкости кода; навыками работы с**	1	18	0	0	ОПК-7.1,ОПК-7.2,ОПК-7.3	вопросы для самоподготовки

	указателями для эффективного изменения данных; навыками работы с функциями высшего порядка и динамическими вызовами /Ср/						
1.6	**Тема 6. Классы и объекты Краткое содержание: классы и объекты, открытые и закрытые члены класса, перегрузка методов, конструкторы и декструкторы, принципы перегрузки операторов, наследование Знать: Понятие классов и объектов, их назначение; Различия между private, protected и public; Принципы перегрузки методов; Роль конструкторов и деструкторов; Виды наследования (прямое, множественное, виртуальное); Динамическое и статическое связывание методов Уметь: Создавать классы и объекты; Определять уровень доступа для членов класса; Реализовывать методы с разными параметрами; Создавать классы с конструкторами и деструкторами; Перегружать операторы; Создавать классы-наследники; Использовать виртуальные функции Владеть: ООП-подход в программировании; навыками реализации иерархии классов; навыками работы с абстрактными классами /Ср/**	1	18	0	0	ОПК-7.1,ОПК-7.2,ОПК-7.3	вопросы для самоподготовки
1.7	**Тема 7. Использование классов и объектов. Краткое содержание: указатель на объект, создание массива объектов, массив как поле класса, функторы и индексация объектов, конструктор создания копии, наследование и закрытые поля базового класса, виртуальные методы и наследование, множественное наследование, доступ к объектам через переменную базового класса Знать: Основы создания массивов объектов в С++; как массивы могут быть полями классов; концепции функторов и их использования; конструктора копирования и его особенностей; Основы наследования в ООП; виртуальных методов и полиморфизма; принципы множественного наследования и связанных с ним проблем Уметь: создавать указатели на объекты; создавать и инициализировать массивы объектов; определять массивы как члены класса и работать с ними; создавать и использовать функторы для индексации объектов; правильно реализовывать конструктор копирования для классов; правильно использовать наследование и работать с закрытыми полями базового**	1	18	0	0	ОПК-7.1,ОПК-7.2,ОПК-7.3	вопросы для самоподготовки

	класса; создавать виртуальные методы в базовом классе и переопределять их в производных классах; применять множественное наследование и разрешать потенциальные проблемы; работать с объектами через переменные базового класса, включая работу с производными классами Владеть: навыками управления памятью с помощью указателей; навыками работы с динамическим и статическим выделением памяти для объектов; навыками применения функторов для выполнения операций с объектами и индексация объектов; навыками обеспечения инкапсуляции и правильное использование модификаторов доступа для наследования /Ср/						
1.8	**Тема 1. Теоретические основы алгоритмизации и программирования. Основные виды алгоритмов. Краткое содержание: алгоритм, свойства алгоритма, способы описания алгоритма, назначение функциональных блоков, основные этапы решения задач. Линейный алгоритм, разветвляющийся алгоритм, циклический алгоритм. Знать: свойства алгоритма, способы описания алгоритмов, основные этапы решения задач, основные виды алгоритмов, правила их описания /Лек/**	1	1	0	0	ОПК-7.1	устный опрос, тестирование
1.9	**Тема 2. Язык программирования С++. Краткое содержание: язык программирования С++, среды разработки, переменные, функции, операторы цикла, условные операторы, массивы Знать: Основные понятия и концепции языка С++; Синтаксис объявления и инициализации переменных различных типов; Основные операторы С++; Синтаксис и принципы работы условных операторов; Типы циклов /Лек/**	1	1	0	0	ОПК-7.1	устный опрос, тестирование
1.10	**Тема 3. Управляющие инструкции. Краткое содержание: оператор цикла: for, do-while, оператор выбора switch, вложенные условные операторы, вложенные операторы цикла, цикл по коллекции, генерация и перехват исключений, инструкция безусловного перехода Знать: Синтаксис и особенности работы каждого оператора; Правила работы switch, допустимые типы выражений; Принципы работы вложенных if-else; Взаимодействие циклов друг с другом; Принципы работы с коллекциями и их итерация; Виды исключений, их обработка через try-catch;**	1	1	0	0	ОПК-7.1	устный опрос, тестирование

	Назначение break, continue, return /Лек/						
1.11	**Тема 4. Указатели, массивы и ссылки. Краткое содержание: указатели, массивы и указатели, ссылки, динамическое выделение памяти, особенности символьных массивов, двумерные массивы, массивы указателей Знать: Определение указателя, разыменование, арифметика указателей; Связь массивов и указателей, передача массивов в функции; Отличия ссылок от указателей, применение ссылок; Функции new и delete, malloc и free, утечки памяти; Особенности хранения строк в С/С++; Организация двумерных массивов, способы их представления в памяти /Лек/**	1	1	0	0	ОПК-7.1	устный опрос, тестирование
1.12	**Тема 1. Теоретические основы алгоритмизации и программирования. Основные виды алгоритмов. Краткое содержание: алгоритм, свойства алгоритма, способы описания алгоритма, назначение функциональных блоков, основные этапы решения задач. Линейный алгоритм, разветвляющийся алгоритм, циклический алгоритм. Уметь: разрабатывать алгоритмы решения задач; описывать алгоритмы на языке псевдокода или на естественном языке; описывать каждый вид алгоритма Владеть: основными понятиями и принципами алгоритмизации и программирования; навыками реализации основных видов алгоритмов /Лаб/**	1	4	0	1	ОПК-7.2,ОПК-7.3	отчет по практической работе
1.13	**Тема 2. Язык программирования С++. Краткое содержание: язык программирования С++, среды разработки, переменные, функции, операторы цикла, условные операторы, массивы Уметь: Устанавливать и настраивать среду разработки С++; Создавать, компилировать и запускать программы на С++; Применять различные типы данных в зависимости от задачи; Использовать условные операторы для ветвления алгоритма; Организовывать циклические процессы с помощью операторов цикла; Разбивать программу на функциональные блоки Владеть: Навыками написания программ на С++ с использованием базового синтаксиса; Методами решения типовых алгоритмических задач; Приемами эффективной работы с массивами; Практическими навыками работы в среде разработки С++ /Лаб/**	1	4	0	1	ОПК-7.2,ОПК-7.3	отчет по практической работе

1.14	**Тема 3. Управляющие инструкции. Краткое содержание: оператор цикла: for, do-while, оператор выбора switch, вложенные условные операторы, вложенные операторы цикла, цикл по коллекции, генерация и перехват исключений, инструкция безусловного перехода Уметь: Использовать циклы для организации повторяющихся вычислений; Применять switch для обработки множественных условий; Использовать вложенные конструкции для сложных логических проверок; Организовывать вложенные циклы для обработки многомерных данных; Использовать цикл foreach для перебора элементов; Применять исключения для обработки ошибок Владеть: навыками оптимизировать выполнение циклов, выбирать подходящий вид цикла; навыками заменять громоздкие if-else конструкцией switch при необходимости; навыками обеспечивать читаемость и структурированность кода /Лаб/**	1	4	0	0	ОПК-7.2,ОПК-7.3	отчет по практической работе
1.15	**Тема 4. Указатели, массивы и ссылки. Краткое содержание: указатели, массивы и указатели, ссылки, динамическое выделение памяти, особенности символьных массивов, двумерные массивы, массивы указателей Уметь: Объявлять и использовать указатели, разыменовывать их, применять арифметику указателей; Использовать указатели для работы с массивами, передавать массивы в функции по указателю; Объявлять ссылки, передавать параметры по ссылке в функции; Выделять и освобождать динамическую память, избегать утечек; Создавать и обрабатывать двумерные массивы, передавать их в функции Владеть: навыками работы с указателями в коде, анализ адресов памяти; навыками работы с массивами с использованием указателей; навыками работы с объектами и функциями; навыками управление динамической памятью, отладка утечек /Лаб/**	1	4	0	0	ОПК-7.2,ОПК-7.3	отчет по практической работе
	Раздел 2.Контроль						
2.1	**Экзамен** **Знать: процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логику построения и**	1	36	0	0	ОПК-7.1,ОПК-7.2,ОПК-7.3	вопросы к экзамену, тестирование

	принципы функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципы разработки алгоритмов и компьютерных программ; современные языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий **Уметь: выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий** **Владеть: навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач /Экзамен/**						

Перечень применяемых активных и интерактивных образовательных технологий:
Информационные технологии

Личностно ориентированная технология, способ организации самостоятельной деятельности учащихся, направленный на решение задачи учебного проекта

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Самостоятельная (внеаудиторная) работа обучающегося (далее - СРС) – это планируемая учебная, практическая, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное время (свободное от аудиторных учебных занятий) по заданию и при методическом руководстве педагогического работника, отнесенного к профессорско-преподавательскому составу (далее – преподаватель) и (или) лиц, привлекаемых к реализации образовательных программ на иных условиях, но без их непосредственного участия.
СРС по заданию и при методическом руководстве преподавателя и (или) лиц, привлекаемых к реализации образовательных программ на иных условиях, реализуется во время групповых консультаций и (или) индивидуальной работы обучающихся с преподавателями Университета и (или) лиц, привлекаемых к реализации образовательных программ на иных условиях (в

том числе индивидуальных консультаций), а также во время текущего контроля выполнения заданий, отнесенных к самостоятельной работе обучающихся.
Целью СРС является овладение формирование компетенций через овладение знаниями, умениями и навыками профессиональной деятельности по направлению подготовки (специальности). Самостоятельная работа является обязательной для каждого обучающегося.
Формы самостоятельной работы обучающихся определяются преподавателями кафедр Университета при разработке рабочих программ дисциплин (модулей), рабочих программ практик, НИР, программ государственной итоговой (итоговой) аттестации, методических указаний по выполнению практических, лабораторных работ, написанию курсовых работ/проектов и ВКР в соответствии с их содержанием.
В университете оборудованы специальные помещения для самостоятельной работы обучающихся. Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к информационно -телекоммуникационной сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду Университета.

6. ОЦЕНОЧНЫЕ И МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

6.1. Перечень компетенций с указанием этапов формирования индикаторов их достижения в процессе освоения ОПОП

ОПК-7:Способен разрабатывать алгоритмы и программы, пригодные для практического применения;

Недостаточный уровень:

Знания процессов, методов поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логики построения и принципов функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципов разработки алгоритмов и компьютерных программ; современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий не сформированы

Умение выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий не сформировано

Не владеет навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач

Пороговый уровень:

Сформированы базовые структуры знаний процессов, методов поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логики построения и принципов функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципов разработки алгоритмов и компьютерных программ; современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий

Умение выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий фрагментарно и носит репродуктивный характер

Владеет базовыми навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач

Продвинутый уровень:

Знания процессов, методов поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логики построения и принципов функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципов разработки алгоритмов и компьютерных программ; современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий обширные и системные

Умение выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды,

разработки информационных систем и технологий применяется в решении типовых заданий

Владеет обширными и системными навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач

Высокий уровень:

Знания процессов, методов поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логики построения и принципов функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципов разработки алгоритмов и компьютерных программ; современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий твердые, аргументированные и всесторонние

Умение выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий успешно применяется в решении как типовых, так и нестандартных творческих заданий

Способен самостоятельно разрабатывать оригинальные алгоритмы и компьютерные программы, пригодные для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач

6.2. Шкала оценивания в зависимости от уровня сформированности компетенций

Уровень сформированности компетенций

Характеристики индикаторов достижения компетенций	**1. Недостаточный: компетенции не сформированы.**	**2. Пороговый: компетенции сформированы.**	**3. Продвинутый: компетенции сформированы.**	**4. Высокий: компетенции сформированы.**
Знания:	Знания отсутствуют.	Сформированы базовые структуры знаний.	Знания обширные, системные.	Знания твердые, аргументированные, всесторонние.
Умения:	Умения не сформированы.	Умения фрагментарны и носят репродуктивный характер.	Умения носят репродуктивный характер применяются к решению типовых заданий.	Умения успешно применяются к решению как типовых, так и нестандартных творческих заданий.
Навыки:	Навыки не сформированы.	Демонстрируется низкий уровень самостоятельности практического навыка.	Демонстрируется достаточный уровень самостоятельности устойчивого практического навыка.	Демонстрируется высокий уровень самостоятельности, высокая адаптивность практического навыка.

Описание критериев оценивания

Обучающийся демонстрирует: - существенные пробелы в знаниях учебного материала; - допускаются принципиальные ошибки при ответе на основные вопросы билета, отсутствует знание и понимание основных понятий и категорий; - непонимание сущности дополнительных вопросов в рамках заданий билета; - отсутствие умения выполнять практические задания, предусмотренные	Обучающийся демонстрирует: - знания теоретического материала; - неполные ответы на основные вопросы, ошибки в ответе, недостаточное понимание сущности излагаемых вопросов; - неуверенные и неточные ответы на дополнительные вопросы; - недостаточное владение литературой, рекомендованной программой дисциплины;	Обучающийся демонстрирует: - знание и понимание основных вопросов контролируемого объема программного материала; - твердые знания теоретического материала; -способность устанавливать и объяснять связь практики и теории, выявлять противоречия, проблемы и тенденции развития; - правильные и конкретные, без грубых ошибок ответы на поставленные вопросы;	Обучающийся демонстрирует: - глубокие, всесторонние и аргументированные знания программного материала; - полное понимание сущности и взаимосвязи рассматриваемых процессов и явлений, точное знание основных понятий в рамках обсуждаемых заданий; - способность устанавливать и объяснять связь практики и теории; - логически последовательные,

программой дисциплины; - отсутствие готовности (способности) к дискуссии и низкая степень контактности.	- умение без грубых ошибок решать практические задания, которые следует выполнить.	- умение решать практические задания, которые следует выполнить; - владение основной литературой, рекомендованной программой дисциплины; - наличие собственной обоснованной позиции по обсуждаемым вопросам. Возможны незначительные оговорки и неточности в раскрытии отдельных положений вопросов билета, присутствует неуверенность в ответах на дополнительные вопросы.	содержательные, конкретные и исчерпывающие ответы на все задания билета, а также дополнительные вопросы экзаменатора; - умение решать практические задания; - свободное использование в ответах на вопросы материалов рекомендованной основной и дополнительной литературы.
Оценка «незачет», «неудовлетворительно»	**Оценка «зачтено/удовлетворительно», «удовлетворительно»**	**Оценка «зачтено/хорошо», «хорошо»**	**Оценка «зачтено/отлично», «отлично»**

Оценочные средства, обеспечивающие диагностику сформированности компетенций, заявленных в рабочей программе по дисциплине (молулю) для проведения промежуточной аттестации

ОЦЕНИВАНИЕ УРОВНЯ ЗНАНИЙ: Теоретический блок вопросов. Уровень освоения программного материала, логика и грамотность изложения, умение самостоятельно обобщать и излагать материал.
1. Недостаточный уровень
Знания процессов, методов поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логики построения и принципов функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципов разработки алгоритмов и компьютерных программ; современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий не сформированы
Умение выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий не сформировано
Не владеет навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач
2. Пороговый уровень
Сформированы базовые структуры знаний процессов, методов поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логики построения и принципов функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципов разработки алгоритмов и компьютерных программ; современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий
Умение выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий фрагментарно и носит репродуктивный характер
Владеет базовыми навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач
3. Продвинутый уровень
Знания процессов, методов поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логики построения и принципов функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципов разработки алгоритмов и компьютерных программ; современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий обширные и системные

Умение выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий применяется в решении типовых заданий
Владеет обширными и системными навыками разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач
4. Высокий уровень
Знания процессов, методов поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов (информационные технологии); логики построения и принципов функционирования современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий, принципов разработки алгоритмов и компьютерных программ; современных языков программирования и языков работы с базами данных, сред разработки информационных систем и технологий твердые, аргументированные и всесторонние
Умение выбирать языки программирования и языки работы с базами данных, среды разработки информационных систем и технологий, исходя из имеющихся задач; применять современные языки программирования для разработки оригинальных алгоритмов и компьютерных программ, пригодных для практического применения, вести базы данных и информационные хранилища, применять современные программные среды разработки информационных систем и технологий; читать коды программных продуктов, написанных на освоенных языках программирования, и вносить требуемые изменения; анализировать профессиональные задачи, разрабатывать подходящие информационные решения; самостоятельно осваивать новые для себя современные языки программирования и языки работы с базами данных, среды, разработки информационных систем и технологий успешно применяется в решении как типовых, так и нестандартных творческих заданий
Способен самостоятельно разрабатывать оригинальные алгоритмы и компьютерные программы, пригодные для практического применения; навыками отладки и тестирования прототипов программно-технических комплексов задач

6.3. Оценочные средства текущего контроля (примерные темы докладов, рефератов, эссе)

ВОПРОСЫ К УСТНОМУ ОПРОСУ

Тема 1. Теоретические основы алгоритмизации и программирования

1. Что такое алгоритм и какие основные свойства он должен иметь?
2. Какие существуют способы описания алгоритмов? Назови и опиши их.
3. Перечисли основные этапы решения задач с использованием алгоритмов.
4. Что такое линейный алгоритм? Приведи пример задачи, решаемой с его помощью.
5. Чем отличается разветвляющийся алгоритм от линейного? Опиши его структуру.
6. Что такое циклический алгоритм и в каких случаях он применяется?
7. Как описать алгоритм на языке псевдокода? Приведи пример.
8. Какие функциональные блоки используются при описании алгоритма? Для чего они нужны?
9. Каковы правила построения циклического алгоритма?
10. Разработай и опиши на естественном языке алгоритм вычисления суммы чисел от 1 до 10.

Тема 2. Язык программирования C++

1. Что такое язык программирования C++ и какие его основные особенности?
2. Какие среды разработки для C++ ты знаешь? Как установить одну из них?
3. Как объявить и инициализировать переменную в C++? Приведи пример для типа int и double.
4. Какие основные операторы цикла существуют в C++? Опиши их назначение.
5. Как работает условный оператор if-else в C++? Приведи пример кода.
6. Что такое массив в C++ и как его объявить? Приведи пример.
7. Как создать функцию в C++? Напиши пример функции, вычисляющей сумму двух чисел.
8. Как скомпилировать и запустить программу на C++ в среде разработки?
9. В чем разница между типами данных int и float? Когда их использовать?
10. Напиши программу на C++, которая выводит числа от 1 до 5 с помощью цикла.

Тема 3. Управляющие инструкции

1. В чем разница между циклами for и do-while? Приведи пример использования каждого.
2. Как работает оператор switch? Какие типы данных он поддерживает?
3. Что такое вложенные условные операторы? Приведи пример их использования.
4. Как организовать вложенные циклы? Напиши пример вывода таблицы умножения 3x3.
5. Как работает цикл foreach в C++? Для чего он нужен?
6. Что такое исключения в C++? Как их обрабатывать с помощью try-catch?
7. Какую роль выполняют операторы break и continue в циклах? Приведи примеры.
8. Чем отличается оператор return от break? В каких случаях их используют?
9. Напиши программу с использованием switch для определения дня недели по номеру (1–7).
10. Как оптимизировать выполнение цикла? Приведи пример оптимизации.

Тема 4. Указатели, массивы и ссылки
1. Что такое указатель в C++? Как его объявить и разыменовать?
2. Как связаны массивы и указатели? Покажи передачу массива в функцию по указателю.
3. Чем отличается ссылка от указателя? Приведи пример использования ссылки.
4. Как выделить динамическую память в C++ с помощью new? Как её освободить?
5. Что такое утечка памяти и как её избежать при работе с динамической памятью?
6. Как объявить и инициализировать двумерный массив в C++? Приведи пример.
7. Как работает арифметика указателей? Покажи пример увеличения указателя на 2 элемента.
8. Как хранятся строки в C++? В чем особенность символьных массивов?
9. Напиши функцию, которая принимает двумерный массив и выводит его элементы.
10. Как передать параметр в функцию по ссылке? Напиши пример функции, меняющей значение переменной.

ВОПРОСЫ К САМОПОДГОТОВКЕ
Тема 1. Теоретические основы алгоритмизации и программирования
1. Что такое алгоритм? Перечисли и объясни его основные свойства.
2. Какие способы описания алгоритмов ты знаешь? В чем их преимущества и недостатки?
3. Опиши основные этапы решения задачи с использованием алгоритма.
4. Что такое линейный алгоритм? Приведи пример задачи и опиши его на естественном языке.
5. Как работает разветвляющийся алгоритм? Напиши псевдокод для проверки четности числа.
6. В чем суть циклического алгоритма? Опиши алгоритм нахождения суммы чисел от 1 до N.
7. Какие функциональные блоки используются в блок-схемах? Для чего нужен каждый из них?
8. Напиши псевдокод для алгоритма, определяющего, является ли год високосным.
9. Чем отличаются алгоритмы с предусловием и постусловием? Приведи примеры.
10. Разработай алгоритм сортировки трех чисел по возрастанию и опиши его на естественном языке.

Тема 2. Язык программирования C++
1. Что такое C++ и чем он отличается от других языков программирования?
2. Какие среды разработки для C++ ты знаешь? Опиши процесс установки одной из них.
3. Как объявить переменные разных типов в C++? Приведи пример для int, char и double.
4. Напиши программу на C++, которая использует цикл for для вывода чисел от 1 до 10.
5. Как работает условный оператор if в C++? Напиши код для проверки, положительное ли число.
6. Что такое массив в C++? Напиши код для создания массива из 5 элементов и его вывода.
7. Как создать функцию в C++? Напиши функцию, вычисляющую произведение двух чисел.
8. Какие типы данных в C++ ты знаешь? Когда лучше использовать float вместо int?
9. Напиши программу на C++, которая запрашивает у пользователя число и выводит его квадрат.
10. Как разбить программу на функциональные блоки? Приведи пример с двумя функциями.

Тема 3. Управляющие инструкции
1. Чем отличается цикл for от do-while? Напиши пример для каждого.
2. Как работает оператор switch? Напиши код для определения оценки (A, B, C) по баллу.
3. Что такое вложенные условные операторы? Напиши код для проверки возраста и пола.
4. Напиши программу с вложенными циклами для вывода таблицы умножения 5x5.
5. Как использовать цикл foreach в C++? Напиши пример перебора элементов массива.
6. Что такое исключения? Напиши код с try-catch для обработки деления на ноль.
7. Как работают операторы break и continue? Приведи пример их использования в цикле.
8. В чем разница между return и break? Напиши пример функции с return.
9. Как оптимизировать цикл? Перепиши цикл вывода четных чисел от 1 до 10 с шагом 2.
10. Напиши код с switch для определения времени суток по введенному часу (утро, день, вечер).

Тема 4. Указатели, массивы и ссылки
1. Что такое указатель? Напиши код для объявления указателя и его разыменования.
2. Как связаны массивы и указатели? Напиши функцию, выводящую массив через указатель.
3. Чем отличается ссылка от указателя? Напиши пример передачи параметра по ссылке.
4. Как выделить динамическую память в C++? Напиши код с new и delete для массива.
5. Что такое утечка памяти? Как её избежать при работе с динамической памятью?
6. Напиши код для создания и вывода двумерного массива 3x3.
7. Как работает арифметика указателей? Напиши код для сдвига указателя на 3 элемента.
8. Как хранятся строки в C++? Напиши код для работы с символьным массивом.
9. Напиши функцию, которая принимает двумерный массив и возвращает сумму его элементов.
10. Как передать параметр по ссылке? Напиши функцию, меняющую значение переменной.

Тема 5. Функции
1. Как объявить и определить функцию в C++? Напиши пример функции сложения двух чисел.
2. Что такое перегрузка функций? Напиши две перегруженные функции для разных типов данных.
3. Как задать аргументы по умолчанию? Напиши функцию с параметром по умолчанию.
4. Что такое рекурсия? Напиши рекурсивную функцию для вычисления факториала.
5. Как передать массив в функцию? Напиши код для вывода элементов массива.
6. Напиши функцию, которая принимает указатель и изменяет значение переменной.
7. Как вернуть указатель из функции? Напиши пример с динамическим массивом.

8. Как передать строку в функцию? Напиши код для вывода строки через указатель.
9. Что такое указатель на функцию? Напиши пример его использования.
10. Напиши функцию, возвращающую ссылку на максимальный элемент массива.

Тема 6. Классы и объекты
1. Что такое класс и объект? Напиши пример простого класса с двумя полями.
2. Чем отличаются private и public члены класса? Приведи пример их использования.
3. Как перегрузить метод в классе? Напиши класс с двумя версиями метода.
4. Что такое конструктор и деструктор? Напиши класс с обоими.
5. Как перегрузить оператор? Напиши пример перегрузки оператора + для класса.
6. Что такое наследование? Напиши базовый и производный класс.
7. Как работают виртуальные функции? Напиши пример с переопределением метода.
8. Напиши класс "Студент" с полями имя и возраст, и методом вывода данных.
9. Как реализовать множественное наследование? Приведи пример с двумя базовыми классами.
10. Напиши код для создания объекта класса и вызова его метода.

Тема 7. Использование классов и объектов
1. Как создать указатель на объект? Напиши код с указателем на объект класса.
2. Как создать массив объектов? Напиши пример с классом и массивом из 3 объектов.
3. Как определить массив как поле класса? Напиши класс с массивом чисел.
4. Что такое функтор? Напиши пример класса с перегрузкой оператора ().
5. Как работает конструктор копирования? Напиши класс с конструктором копирования.
6. Как получить доступ к закрытым полям базового класса? Напиши пример с наследованием.
7. Как использовать виртуальные методы? Напиши базовый и производный класс с виртуальной функцией.
8. Что такое множественное наследование? Напиши пример с двумя базовыми классами.
9. Как работать с объектом через переменную базового класса? Напиши пример с полиморфизмом.
10. Напиши класс с динамическим массивом как полем и методом для его обработки.

ТЕКУЩЕЕ ТЕСТИРОВАНИЕ
Тема 1. Теоретические основы алгоритмизации и программирования
1. Что из перечисленного является свойством алгоритма?
a) Сложность
b) Детерминированность
c) Абстрактность
d) Непонятность
2. Какой способ описания алгоритма использует графическое представление?
a) Псевдокод
b) Естественный язык
c) Блок-схема
d) Программный код
3. Какой этап решения задачи следует за постановкой задачи?
a) Анализ результатов
b) Разработка алгоритма
c) Тестирование программы
d) Реализация программы
4. Какой алгоритм выполняется последовательно без условий и циклов?
a) Линейный
b) Разветвляющийся
c) Циклический
d) Рекурсивный
5. Что характерно для разветвляющегося алгоритма?
a) Повторение действий
b) Выбор одного из путей выполнения
c) Последовательное выполнение
d) Вызов самого себя
6. Какой алгоритм используется для повторения действий?
a) Линейный
b) Разветвляющийся
c) Циклический
d) Случайный
7. Какой блок в блок-схеме обозначает начало или конец алгоритма?
a) Прямоугольник
b) Ромб
c) Овал
d) Круг
8. Какой элемент псевдокода обозначает начало цикла?
a) if
b) while
c) end

d) input
9. Что из перечисленного не относится к свойствам алгоритма?
a) Массовость
b) Дискретность
c) Бесконечность
d) Результативность
10. Какой алгоритм решает задачу вычисления суммы чисел от 1 до 10?
a) Линейный
b) Разветвляющийся
c) Циклический
d) Статический

Тема 2. Язык программирования C++
1. Какой тип данных используется для хранения целых чисел в C++?
a) float
b) int
c) char
d) double
2. Какая среда разработки не относится к популярным для C++?
a) Visual Studio
b) Code::Blocks
c) Notepad
d) CLion
3. Как правильно объявить переменную x типа int со значением 5?
a) int x = 5;
b) x int = 5;
c) int 5 = x;
d) x = int 5;
4. Какой оператор цикла используется для известного числа итераций?
a) while
b) do-while
c) for
d) switch
5. Какой оператор используется для ветвления в C++?
a) for
b) if
c) return
d) break
6. Как объявить массив из 5 элементов типа int?
a) int arr[5];
b) int arr(5);
c) arr int[5];
d) int[5] arr;
7. Какой заголовок функции вычисляет сумму двух чисел?
a) int sum(int a, int b)
b) sum(int a, int b)
c) int sum(a, b)
d) void sum(int a, int b)
8. Какой оператор завершает выполнение программы?
a) break
b) continue
c) return
d) exit
9. Какой тип данных подходит для хранения символов?
a) int
b) float
c) char
d) double
10. Какая команда выводит текст на экран в C++?
a) print
b) cout
c) scanf
d) write

Тема 3. Управляющие инструкции
1. Какой цикл выполняется хотя бы один раз?
a) for
b) while
c) do-while

d) foreach
2. Какой оператор используется для множественного выбора?
a) if
b) switch
c) for
d) while
3. Какой символ завершает блок case в switch?
a) ;
b) :
c) }
d) break
4. Какой оператор прерывает выполнение цикла?
a) return
b) break
c) continue
d) exit
5. Какой цикл лучше использовать для перебора элементов массива?
a) while
b) do-while
c) for
d) switch
6. Какой блок используется для перехвата исключений?
a) try-catch
b) if-else
c) for-while
d) switch-case
7. Что делает оператор continue в цикле?
a) Завершает программу
b) Прерывает цикл
c) Пропускает текущую итерацию
d) Возвращает значение
8. Какой тип данных не поддерживается в switch?
a) int
b) char
c) float
d) enum
9. Какой оператор возвращает управление из функции?
a) break
b) continue
c) return
d) throw
10. Какой цикл используется для работы с коллекциями в C++11?
a) for
b) while
c) do-while
d) foreach (range-based for)

Тема 4. Указатели, массивы и ссылки
1. Что обозначает символ * при объявлении указателя?
a) Умножение
b) Разыменование
c) Адрес переменной
d) Тип данных
2. Какой оператор возвращает адрес переменной?
a) *
b) &
c) ->
d) %
3. Как объявить указатель на int?
a) int* ptr;
b) int ptr*;
c) int ptr;
d) ptr int;
4. Какой оператор выделяет динамическую память в C++?
a) malloc
b) free
c) new
d) delete
5. Как освободить динамическую память?

a) free
b) delete
c) clear
d) remove
6. Как объявить двумерный массив 2x3?
a) int arr[2][3];
b) int arr[2,3];
c) int arr(2)(3);
d) int arr[2,3];
7. Чем отличается ссылка от указателя?
a) Ссылка не может быть nullptr
b) Ссылка изменяет адрес
c) Указатель не требует разыменования
d) Ссылка занимает больше памяти
8. Какой тип данных используется для строк в стиле C?
a) string
b) char*
c) int
d) double
9. Что произойдет, если не освободить динамическую память?
a) Программа завершится
b) Утечка памяти
c) Ошибка компиляции
d) Ничего не произойдет
10. Как передать массив в функцию?
a) По значению
b) По указателю
c) По ссылке
d) Как строку

ЗАДАНИЯ К ПРАКТИЧЕСКИМ РАБОТАМ
Тема 1. Теоретические основы алгоритмизации и программирования
1. Линейный алгоритм: Разработать алгоритм вычисления площади прямоугольника по заданным длине и ширине. Описать его на естественном языке и в виде блок-схемы.
o Входные данные: длина (a), ширина (b).
o Выходные данные: площадь (S).
2. Разветвляющийся алгоритм: Создать алгоритм определения, является ли введенное число положительным, отрицательным или нулем. Описать его в виде псевдокода.
o Входные данные: число (n).
o Выходные данные: сообщение о типе числа.
3. Циклический алгоритм: Разработать алгоритм вычисления суммы чисел от 1 до N. Описать его в виде блок-схемы и псевдокода.
o Входные данные: число N.
o Выходные данные: сумма (sum).

Тема 2. Язык программирования C++
1. Переменные и ввод-вывод: Написать программу, которая запрашивает у пользователя два числа и выводит их сумму и произведение.
o Использовать переменные типа int и double.
o Вывести результат с помощью cout.
2. Условные операторы: Создать программу, которая запрашивает возраст пользователя и определяет, является ли он совершеннолетним (≥18).
o Использовать оператор if-else.
o Вывести соответствующее сообщение.
3. Циклы и массивы: Написать программу, которая заполняет массив из 10 элементов числами от 1 до 10 и выводит их сумму.
o Использовать цикл for для заполнения и подсчета суммы.
o Вывести массив и результат.

Тема 3. Управляющие инструкции
1. Циклы: Написать программу, которая выводит таблицу умножения для числа N (вводится пользователем) с использованием цикла for.
o Входные данные: число N (от 1 до 9).
o Выходные данные: таблица вида "N * i = результат" для i от 1 до 10.
2. Оператор switch: Создать программу, которая запрашивает номер месяца (1–12) и выводит количество дней в этом месяце.
o Учитывать, что в феврале 28 дней (високосный год не учитывать).
o Использовать switch-case.
3. Исключения и вложенные конструкции: Написать программу, которая делит два введенных пользователем числа и обрабатывает деление на ноль с помощью try-catch.
o Использовать вложенный if для проверки, что числа не отрицательные.

o Освободить память с помощью delete.
o Вывести массив и сумму.
3. Ссылки и двумерные массивы: Написать программу, которая создает двумерный массив 3x3, передает его в функцию по ссылке и выводит сумму всех элементов.
o Инициализировать массив числами от 1 до 9.
o Использовать ссылку в качестве параметра функции.

6.4. Оценочные средства промежуточной аттестации.

ВОПРОСЫ К ЭКЗАМЕНУ:
Знать:
Вопросы на установление последовательности
1. Установите последовательность этапов решения задачи с использованием алгоритма:
A. Тестирование и отладка программы
B. Анализ условия задачи
C. Реализация алгоритма на языке программирования
D. Разработка алгоритма
E. Постановка задачи
2. Установите последовательность действий для создания и запуска программы на языке C++:
A. Компиляция программы
B. Написание кода программы
C. Установка среды разработки
D. Запуск программы
E. Настройка проекта в среде разработки
3. Установите последовательность операторов в программе на C++ для обработки исключений при работе с динамической памятью:
A. Освобождение памяти с помощью delete
B. Блок try для выполнения кода с возможной ошибкой
C. Выделение памяти с помощью new
D. Блок catch для обработки исключения
E. Проверка результата выполнения
4. Установите последовательность разработки функции на C++ с передачей массива по указателю:
A. Вызов функции в основной программе
B. Объявление функции с параметром-указателем
C. Определение функции с обработкой массива
D. Передача массива в функцию
E. Объявление массива в основной программе
5. Установите последовательность шагов для создания и использования класса с наследованием в C++:
A. Создание объекта производного класса
B. Определение базового класса
C. Определение производного класса с наследованием
D. Использование методов через объект
E. Объявление виртуального метода в базовом классе

Вопросы на установление соответствия
1. Установите соответствие между видом алгоритма и его описанием:
1. Линейный алгоритм
2. Разветвляющийся алгоритм
3. Циклический алгоритм
4. Комбинированный алгоритм
A. Выполняет действия в зависимости от условия
B. Повторяет набор действий до выполнения условия
C. Выполняет действия строго последовательно
D. Сочетает элементы ветвления и циклов
E. Выполняет случайный набор действий
2. Установите соответствие между оператором C++ и его назначением:
1. for
2. switch
3. try-catch
4. do-while
A. Обработка исключений
B. Организация цикла с заданным числом итераций
C. Выбор одного из множества вариантов
D. Выполнение цикла с проверкой условия после тела
E. Автоматическое выделение памяти
3. Установите соответствие между способом передачи аргумента в функцию на C++ и его особенностью:
1. По значению
2. По указателю
3. По ссылке

4. По умолчанию
A. Позволяет изменять исходный объект через адрес памяти
B. Передает копию данных
C. Позволяет изменять исходный объект без явного указания адреса
D. Задает значение аргумента при его отсутствии в вызове
E. Автоматически создает новый объект
4. Установите соответствие между понятием и его определением:
1. Указатель
2. Ссылка
3. Динамическая память
4. Двумерный массив
A. Переменная, содержащая адрес другой переменной
B. Память, выделяемая во время выполнения программы
C. Альтернативное имя для существующей переменной
D. Набор элементов, организованных в виде таблицы
E. Константа, неизменяемая в процессе работы программы
5. Установите соответствие между элементом класса в C++ и его ролью:
1. Конструктор
2. Деструктор
3. Виртуальный метод
4. Перегрузка метода
A. Освобождает ресурсы объекта при его уничтожении
B. Инициализирует объект при его создании
C. Позволяет переопределять поведение в производных классах
D. Создает несколько версий метода с разными параметрами
E. Автоматически вызывает функцию при запуске программы

Уметь:
1. Выберите правильный ответ и обоснуйте свой выбор. Какое свойство алгоритма связано с завершением его работы?
A. Детерминированность
B. Конечность
C. Результативность
D. Массовость
2. Выберите правильный ответ и обоснуйте свой выбор. Какой оператор в C++ удобен для выполнения цикла с известным числом повторений?
A. while
B. do-while
C. for
D. switch
3. Выберите правильный ответ и обоснуйте свой выбор. Что происходит при отсутствии освобождения динамической памяти в C++?
A. Программа завершится с ошибкой
B. Произойдет утечка памяти
C. Память освободится автоматически
D. Указатель станет недействительным
4. Выберите правильный ответ и обоснуйте свой выбор. Какой оператор в C++ лучше заменяет множество вложенных условий?
A. for
B. switch
C. while
D. try-catch
5. Выберите правильный ответ и обоснуйте свой выбор. Какой способ передачи аргумента в функцию изменяет исходные данные?
A. По значению
B. По указателю
C. По константной ссылке
D. По умолчанию
6. Выберите правильный ответ и обоснуйте свой выбор. Что обязательно указывается при определении класса в C++?
A. Конструктор
B. Деструктор
C. Имя класса
D. Виртуальный метод
7. Выберите правильный ответ и обоснуйте свой выбор. Какой оператор пропускает остаток текущей итерации цикла?
A. break
B. continue
C. return
D. exit
8. Выберите правильный ответ и обоснуйте свой выбор. Что требуется для перегрузки функции в C++?
A. Изменение типа возвращаемого значения
B. Изменение числа или типов параметров
C. Изменение имени функции
D. Изменение области видимости

A. public
B. private
C. protected
D. static
10. Выберите правильный ответ и обоснуйте свой выбор. Что необходимо для реализации полиморфизма в C++?
A. Перегрузка функций
B. Виртуальные методы
C. Статические методы
D. Конструктор копирования

Владеть:
1. Вы разрабатываете алгоритм для подсчета суммы чисел от 1 до 10. Определите, какой вид алгоритма лучше использовать для решения этой задачи.
2. В программе на C++ нужно вывести сообщение "Ошибка", если введенное пользователем число меньше 0. Определите, какой оператор C++ следует использовать для проверки условия.
3. В программе на C++ нужно выделить память для массива из 10 целых чисел. Определите, как правильно выделить динамическую память для этого массива.
4. Вы пишете функцию на C++, которая удваивает значение переменной, переданной в нее. Определите, как правильно передать аргумент в функцию, чтобы исходное значение изменилось.
5. Вы создаете класс Point с координатами x и y в C++. Определите, как правильно объявить конструктор для инициализации координат.
6. В программе нужно обработать деление на ноль с помощью исключений в C++. Определите, как правильно организовать обработку этой ошибки.
7. В классе Rectangle нужно перегрузить оператор + для сложения длин двух прямоугольников. Определите, как правильно реализовать перегрузку оператора +.
8. Вы разрабатываете алгоритм для проверки, является ли введенное пользователем число четным или нечетным. Определите, какой вид алгоритма лучше использовать для этой задачи.
9. В программе на C++ нужно вычислить сумму элементов двумерного массива размером 2x3: {{1, 2, 3}, {4, 5, 6}}. Вычислите сумму всех элементов этого массива с использованием вложенных циклов.
10. Вы разрабатываете алгоритм для вывода таблицы умножения числа 3 (от 1 до 5). Определите, какой вид цикла лучше использовать для этой задачи.

ИТОГОВОЕ ТЕСТИРОВАНИЕ
1. Какой способ описания алгоритма использует условные конструкции?
a) Блок-схема
b) Естественный язык
c) Псевдокод
d) Все перечисленные
2. Какой алгоритм решает задачу проверки четности числа?
a) Линейный
b) Разветвляющийся
c) Циклический
d) Случайный
3. Какой тип данных используется для целых чисел?
a) float
b) int
c) char
d) bool
4. Как объявить переменную x типа double?
a) double x;
b) x double;
c) double = x;
d) x = double;
5. Какой оператор используется для множественного выбора?
a) if
b) switch
c) for
d) while
6. Какой символ завершает case в switch?
a) ;
b) :
c) }
d) break
7. Что обозначает * при работе с указателями?
a) Умножение
b) Разыменование
c) Адрес
d) Тип
8. Как объявить указатель на int?
a) int* ptr;
b) int ptr*;

a) int
b) void
c) double
d) char
11. Что такое перегрузка функций?
a) Повторное использование имени
b) Изменение типа возврата
c) Увеличение параметров
d) Рекурсия
12. Что такое стек вызовов?
a) Массив функций
b) Память для рекурсии
c) Указатель на функцию
d) Переменные функции
13. Какой модификатор скрывает члены класса?
a) public
b) private
c) protected
d) static
14. Что делает конструктор?
a) Уничтожает объект
b) Инициализирует объект
c) Копирует объект
d) Изменяет объект
15. Как создать массив объектов?
a) Class arr[5];
b) Class arr(5);
c) arr[5] Class;
d) Class[5] arr;

6.5. Примерная тематика курсовых работ (проектов)
Учебным планом не предусмотрено

6.6. Методические указания для обучающихся по освоению дисциплины (модуля)
Методические рекомендации по работе с конспектом лекций

Просмотрите конспект сразу после занятий. Пометьте материал конспекта лекций, который вызывает затруднения для понимания. Попытайтесь найти ответы на затруднительные вопросы, используя предлагаемую литературу. Если самостоятельно не удалось разобраться в материале, сформулируйте вопросы и обратитесь на текущей консультации или на ближайшей лекции за помощью к преподавателю. Каждую неделю рекомендуется отводить время для повторения пройденного материала, проверяя свои знания, умения и навыки по контрольным вопросам.
Работа с рекомендованной литературой:
При работе с основной и дополнительной литературой целесообразно придерживаться такой последовательности. Сначала прочитать весь заданный текст в быстром темпе. Цель такого чтения заключается в том, чтобы создать общее представление об изучаемом материале, понять общий смысл прочитанного. Затем прочитать вторично, более медленно, чтобы в ходе чтения понять и запомнить смысл каждой фразы, каждого положения и вопроса в целом. Чтение приносит пользу и становится продуктивным, когда сопровождается записями. Это может быть составление плана прочитанного текста, тезисы или выписки, конспектирование и др. Выбор вида записи зависит от характера изучаемого материала и целей работы с ним. Если содержание материала несложное, легко усваиваемое, можно ограничиться составлением плана. Если материал содержит новую и трудно усваиваемую информацию, целесообразно его законспектировать. План – это схема прочитанного материала, перечень вопросов, отражающих структуру и последовательность материала. Конспект – это систематизированное, логичное изложение материала источника. Различаются четыре типа конспектов: - план-конспект – это развернутый детализированный план, в котором по наиболее сложным вопросам даются подробные пояснения, - текстуальный конспект – это воспроизведение наиболее важных положений и фактов источника, - свободный конспект – это четко и кратко изложенные основные положения в результате глубокого изучения материала, могут присутствовать выписки, цитаты, тезисы; часть материала может быть представлена планом, - тематический конспект – составляется на основе изучения ряда источников и дает ответ по изучаемому вопросу. В процессе изучения материала источника и составления конспекта нужно обязательно применять различные выделения, подзаголовки, создавая блочную структуру конспекта. Это делает конспект легко воспринимаемым и удобным для работы.

Методические рекомендации по подготовке к практическим занятиям

Практические занятия представляют особую форму сочетания теории и практики. Их назначение – углубление проработки теоретического материала предмета путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к практическим занятиям включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу. Непосредственное проведение практического занятия

предполагает, например: индивидуальные выступления студентов с сообщениями по какому-либо вопросу изучаемой темы; фронтальное обсуждение рассматриваемой проблемы, обобщения и выводы; решение задач и упражнений по образцу; решение вариантных задач и упражнений; решение ситуационных производственных (профессиональных) задач; проектирование и моделирование разных видов и компонентов профессиональной деятельности. выполнение контрольных работ; работу с тестами. При подготовке к практическим занятиям студентам рекомендуется: внимательно ознакомиться с тематикой практического занятия; прочесть конспект лекции по теме, изучить рекомендованную литературу; составить краткий план ответа на каждый вопрос практического занятия; проверить свои знания, отвечая на вопросы для самопроверки; если встретятся незнакомые термины, обязательно обратиться к словарю и зафиксировать их в тетради. Все письменные задания выполнять в рабочей тетради. Практические занятия развивают у студентов навыки самостоятельной работы по решению конкретных задач.

Методические рекомендации по подготовке к лабораторным работам

Лабораторные работы представляют одну из форм освоения теоретического материала с одновременным формированием практических навыков в изучаемой дисциплине. Их назначение – углубление проработки теоретического материала, формирование практических навыков путем регулярной и планомерной самостоятельной работы студентов на протяжении всего курса. Процесс подготовки к лабораторным работам включает изучение нормативных документов, обязательной и дополнительной литературы по рассматриваемому вопросу. Непосредственное проведение лабораторной работы предполагает: изучение теоретического материала по теме лабораторной работы (по вопросам изучаемой темы); выполнение необходимых расчетов и экспериментов; оформление отчета с заполнением необходимых таблиц, построением графиков, подготовкой выводов по проделанным экспериментам и теоретическим расчетам; по каждой лабораторной работе проводится контроль: проверяется содержание отчета, проверяется усвоение теоретического материала. Контроль усвоения теоретического материала является индивидуальным.

Методические указания по выполнению отчёта к лабораторным работам

Основным требованием по выполнению лабораторных и практических работ является полное исчерпывающее описание всей проделанной работы, позволяющее судить о полученных результатах, степени выполнения и профессиональной подготовки студентов.
Методические указания обеспечивают комплексный подход в учебной работе студентов, единство и преемственность требований к оформлению результатов работы на разных этапах обучения. С единых позиций приведены основные требования по структуре, оформлению и содержанию отчета по лабораторным и практическим работам.
Структура отчёта:
- цель работы;
- краткие теоретические сведения;
- ход выполнения работы;
- выводы.
Дополнительными элементы:
- приложения;
- библиографический список.
Требования к содержанию отчёта:
1. Титульный лист
В верхнем поле листа указывают полное наименование учебного заведения.
В среднем поле указывается вид работы, в данном случае лабораторная или практическая работа с указанием курса, по которому она выполнена, и ниже ее название. Название работы приводится без слова тема и в кавычки не заключается.
Далее ближе к правому краю титульного листа указывают фамилию, инициалы и группу учащегося, выполнившего работу, а также фамилию, инициалы преподавателя, принявшего работу.
В нижнем поле листа указывается место выполнения работы и год ее написания (без слова год).
2. Цель работы должна отражать тему работы, а также конкретные задачи, поставленные студенту на период выполнения работы. По объему цель работы в зависимости от сложности и многозадачности работы составляет от нескольких строк до 0,5 страницы.
3. Краткие теоретические сведения. В этом разделе излагается краткое теоретическое описание изучаемой в работе темы. Материал раздела не должен копировать содержание методического пособия или учебника по данной теме, а ограничивается изложением основных понятий, требующихся для дальнейшей обработки полученных результатов. Объем литературного обзора не должен превышать 1/3 части всего отчета.
4. Ход выполнения работы. В данном разделе подробно излагается методика выполнения работы, процесс получения данных и способ их обработки. Если используются стандартные пакеты компьютерных программ для обработки экспериментальных результатов, то необходимо обосновать возможность и целесообразность их применения, а также подробности обработки данных с их помощью.
5. Выводы по работе - кратко излагаются результаты работы, полученные в результате выполнения работы, а также краткий анализ полученных результатов.
Отчет по лабораторной работе оформляется на листе формата А4. Допускается оформление отчета по лабораторной работе в электронном виде средствами Microsoft Office. Текст работы должен быть напечатан через полтора интервала шрифтом Times New Roman, кегль – 12. Поля должны оставаться по всем четырем сторонам печатного листа: левое – не менее 30 мм, правое – не менее 10, нижнее – не менее 20 и верхнее – не 15 мм.
Для защиты лабораторной работы студент должен подготовить отчет, провести самостоятельную работу, иметь отметку о проверенном отчете.
Результаты определяются по пятибалльной системе оценок.

Методические рекомендации по выполнению реферата

Реферат – письменная работа объемом 8–10 страниц. Это краткое и точное изложение сущности какого-либо вопроса, темы. Тему реферата студент выбирает из предложенных преподавателем или может предложить свой вариант. В реферате нужны развернутые аргументы, рассуждения, сравнения. Содержание темы излагается объективно от имени автора. Функции реферата. Информативная, поисковая, справочная, сигнальная, коммуникативная. Степень выполнения этих функций зависит от содержательных и формальных качеств реферата и для каких целей их использует. Требования к языку реферата. Должен отличаться точностью, краткостью, ясностью и простотой.
Структура реферата:
1. Титульный лист
2. Оглавление (на отдельной странице). Указываются названия всех разделов (пунктов плана) реферата и номера страниц, указывающие начало этих разделов в тексте реферата.
3. Введение. Аргументируется актуальность исследования, т.е. выявляется практическое и теоретическое значение данного исследования. Далее констатируется, что сделано в данной области предшественниками, перечисляются положения, которые должны быть обоснованы. Обязательно формулируются цель и задачи реферата.
4. Основная часть. Подчиняется собственному плану, что отражается в разделении текста на главы, параграфы, пункты. План основной части может быть составлен с использованием различных методов группировки материала. В случае если используется чья-либо неординарная мысль, идея, то обязательно нужно сделать ссылку на того автора, у кого взят данный материал.
5. Заключение. Последняя часть научного текста. В краткой и сжатой форме излагаются полученные результаты, представляющие собой ответ на главный вопрос исследования.
6. Приложение. Может включать графики, таблицы, расчеты.
7. Библиография (список литературы). Указывается реально использованная для написания реферата литература. Названия книг располагаются по алфавиту с указанием их выходных данных. Общие требования к построению, содержанию и оформлению».
При проверке реферата оцениваются:
 знание фактического материала, усвоение общих представлений, понятий, идей;
 характеристика реализации цели и задач исследования;
 степень обоснованности аргументов и обобщений;
 качество и ценность полученных результатов;
 использование литературных источников;
 культура письменного изложения материала;
 культура оформления материалов работы.
Правила написания научных текстов (реферат, дипломная работа):
Здесь приводятся рекомендации по консультированию студентов относительно данного вида самостоятельной работы. Во время консультаций руководителю следует предложить к обсуждению следующие вопросы.
• Какова истинная цель Вашего научного текста – это поможет Вам разумно распределить свои силы и время.
• Важно разобраться, кто будет «читателем» Вашей работы.
• Начинать писать серьезную работу следует не раньше, чем возникнет ощущение, что по работе с источниками появились идеи, которыми можно поделиться.
• Должна быть идея, а для этого нужно научиться либо относиться к разным явлениям и фактам несколько критически (своя идея – как иная точка зрения), либо научиться увлекаться какими-то известными идеями, которые нуждаются в доработке (идея – как оптимистическая позиция и направленность на дальнейшее совершенствование уже известного).
• Писать следует ясно и понятно, стараясь основные положения формулировать четко и недвусмысленно, а также стремясь структурировать свой текст.
• Объем текста и различные оформительские требования во многом зависят от принятых в конкретном учебном заведении порядков.

Методические рекомендации по выполнению контрольных работ

Контрольная работа выполняется по вариантам. На бланке указывается факультет, курс, группа, ФИО студента. Вопросы строятся на основе тестовых и ситуативных заданий. В тестовых заданиях, выбирается правильный(ые) ответ(ы). При решении ситуативных заданий выбирается правильная последовательность действий в рассматриваемой ситуации. Проверка контрольной работы позволяет выявить и исправить допущенные студентами ошибки, указать, какие вопросы дисциплины ими недостаточно усвоены и требуют доработки. Студент должен внимательно ознакомиться с письменными замечаниями преподавателя и приступить к их исправлению, для чего еще раз повторить соответствующий материал.

Методические рекомендации по подготовке к коллоквиуму

Коллоквиумом называется собеседование преподавателя и студента по заранее определенным контрольным вопросам. Целью коллоквиума является формирование у студента навыков анализа теоретических проблем на основе самостоятельного изучения учебной и научной литературы. На коллоквиум выносятся крупные, проблемные, нередко спорные теоретические вопросы. Упор делается на монографические работы профессора-автора данного спецкурса. От студента требуется:
- владение изученным в ходе учебного процесса материалом, относящимся к рассматриваемой проблеме;

- знание разных точек зрения, высказанных в научной литературе по соответствующей проблеме, умение сопоставлять их между собой;
- наличие собственного мнения по обсуждаемым вопросам и умение его аргументировать.
Коллоквиум - это не только форма контроля, но и метод углубления, закрепления знаний студентов, так как в ходе собеседования преподаватель разъясняет сложные вопросы, возникающие у студента в процессе изучения данного источника. Однако коллоквиум не консультация и не экзамен. Его задача добиться глубокого изучения отобранного материала, пробудить у студента стремление к чтению дополнительной социологической литературы. Подготовка к коллоквиуму начинается с установочной консультации преподавателя, на которой он разъясняет развернутую тематику проблемы, рекомендует литературу для изучения и объясняет процедуру проведения коллоквиума. Как правило, на самостоятельную подготовку к коллоквиуму студенту отводится 3-4 недели. Методические указания состоят из рекомендаций по изучению источников и литературы, вопросов для самопроверки и кратких конспектов ответа с перечислением основных фактов и событий, относящихся к пунктам плана каждой темы. Это должно помочь студентам целенаправленно организовать работу по овладению материалом и его запоминанию. При подготовке к коллоквиуму следует, прежде всего, просмотреть конспекты лекций и практических занятий и отметить в них имеющиеся вопросы коллоквиума. Если какие-то вопросы вынесены преподавателем на самостоятельное изучение, следует обратиться к учебной литературе, рекомендованной преподавателем в качестве источника сведений.
Коллоквиум проводится в форме индивидуальной беседы преподавателя с каждым студентом или беседы в небольших группах (2-3 человека). Обычно преподаватель задает несколько кратких конкретных вопросов, позволяющих выяснить степень добросовестности работы с литературой, проверяет конспект. Далее более подробно обсуждается какая-либо сторона проблемы, что позволяет оценить уровень понимания. По итогам коллоквиума выставляется дифференцированная оценка по пятибалльной системе.

Методические рекомендации по устному опросу/самоподготовке

После изучения определенной темы по записям в конспекте и учебнику, а также
решения достаточного количества соответствующих задач на практических занятиях и
самостоятельно студенту рекомендуется, используя лист опорных сигналов,
воспроизвести по памяти определения, выводы формул, формулировки основных
положений и доказательств. В случае необходимости следует рекомендовать еще раз внимательно разобраться в материале. Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала – умение решать задачи или пройти тестирование по пройденному материалу. Однако преподавателю следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

Методические рекомендации по подготовке к семинарским занятиям

Одним из видов внеаудиторной самостоятельной работы является подготовка к семинарским занятиям. Семинар – форма учебно-практических занятий, при которой студенты обсуждают сообщения, доклады и рефераты, выполненные ими по результатам учебных или научных исследований под руководством преподавателя. Преподаватель в этом случае является координатором обсуждений темы семинара, подготовка к которому является обязательной. Поэтому тема семинара и основные источники обсуждения предъявляются до обсуждения для детального ознакомления, изучения. Цели обсуждений направлены на формирование навыков профессиональной полемики и закрепление обсуждаемого материала. Семинар – это такая форма организации обучения, при которой на этапе подготовки доминирует самостоятельная работа учащихся с учебной литературой и другими дидактическими средствами над серией вопросов, проблем и задач, а в процессе семинара идут активное обсуждение, дискуссии и выступления учащихся, где они под руководством преподавателя делают обобщающие выводы и заключения. Семинар предназначен для углубленного изучения дисциплины, овладения методологией научного познания, то главная цель семинарских занятий – обеспечить студентам возможность овладеть навыками и умениями использования теоретического знания применительно к особенностям изучаемой отрасли.

Методические рекомендации по подготовке к эссе

Одним из видов самостоятельной работы студентов является написание творческой работы по заданной либо согласованной с преподавателем теме. Творческая работа (эссе) представляет собой оригинальное произведение объемом 500-700 слов, посвященное какой-либо значимой классической либо современной проблеме в определенной теоретической и практической области. Творческая работа не является рефератом и не должна носить описательный характер, большое место в ней должно быть уделено аргументированному представлению своей точки зрения студентами, критической оценке рассматриваемого материала и проблематики, что должно способствовать раскрытию творческих и аналитических способностей. Цели написания эссе – научиться логически верно и аргументировано строить устную и письменную речь; работать над углублением и систематизацией своих философских знаний; овладеть способностью использовать основы знаний для формирования мировоззренческой позиции. Приступая к написанию эссе, изложите в одном предложении, что именно вы будете утверждать и доказывать (свой тезис).

Методические рекомендации по подготовке к докладу

Для подготовки доклада необходимо выбрать актуальную тему. Желательно, чтобы тема была интересна докладчику и вызывала желание качественно подготовить материалы. Подготовка доклада предполагает: определение цели доклада;

подбор необходимого материала, определяющего содержание доклада; составление плана доклада, распределение собранного материала в необходимой логической последовательности.
Композиция доклада имеет вступление, основную часть и заключение.
Вступление должно содержать: название доклада; сообщение основной идеи; современную оценку предмета изложения; краткое перечисление рассматриваемых вопросов; интересную для слушателей форму изложения. Основная часть, в которой необходимо раскрыть суть темы, обычно строится по принципу отчёта. Задача основной части: представить достаточно данных для того, чтобы слушатели заинтересовались темой.
Заключение – чёткое обобщение и краткие выводы по излагаемой теме.

Методические рекомендации по подготовке к собеседованию

Собеседование – средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.
Цель собеседования: проверка усвоения знаний; умений применять знания; сформированности профессионально значимых личностных качеств.
Подготовка к собеседованию предполагает повторение пройденного материала и приобретение навыка свободного владения терминологией и фактическими данными по определенному разделу дисциплины.

Методические рекомендации по подготовке к тестированию/ итоговому тестированию

Тестирование/итоговое тестирование – это система стандартизированных заданий, определенного содержания и упорядоченных в рамках определенной стратегии предъявления, позволяющая качественно оценить структуру и эффективно измерить уровень знаний, умений и навыков по учебной дисциплине (разделу). Тестирование позволяет оценить у студентов уровень знаний фактического материала, умения логически мыслить, способности к рефлексии и творческому подходу к решению поставленной задачи на каждом этапе их обучения.
При самостоятельной подготовке к тестированию студенту необходимо:
- четко выяснить все условия тестирования заранее (сколько тестов будет предложено, сколько времени отводится на тестирование, какова система оценки результатов);
- проконсультироваться с преподавателем по вопросу выбора учебной литературы;
- проработать информационный материал по разделу дисциплине;
- конспективно выписать из соответствующих разделов учебной дисциплины основные понятия, ключевые идеи, теоретические концепции и подходы;
- повторить теоретический материал лекций, самостоятельно изученной учебной и научной литературы, семинарских занятий.
При выполнении тестовых заданий рекомендуется:
- внимательно полностью прочитать вопрос и предлагаемые варианты ответов, выбрать правильные из них (их может быть несколько);
- при затруднении при ответе на сложный для студента вопрос целесообразно переходить к другим вопросам теста. По завершению ответов на все последующие вопросы теста необходимо вернуться к трудному вопросу и дать на него продуманный ответ;
- по окончании тестирования следует проверить правильность данных ответов на вопросы теста, чтобы избежать механических ошибок.

Методические рекомендации по подготовке к экзамену

Изучение многих общепрофессиональных и специальных дисциплин завершается экзаменом. Подготовка к экзамену способствует закреплению, углублению и обобщению знаний, получаемых, в процессе обучения, а также применению их к решению практических задач. Готовясь к экзамену, студент ликвидирует имеющиеся пробелы в знаниях, углубляет, систематизирует и упорядочивает свои знания. На экзамене студент демонстрирует то, что он приобрел в процессе обучения по конкретной учебной дисциплине. Экзаменационная сессия – это серия экзаменов, установленных учебным планом. Между экзаменами интервал 2-4 дня, в течение студент систематизирует уже имеющиеся знания. На консультации перед экзаменом студенты должны быть ознакомлены с основными требованиями и получить ответы на возникающие в процессе подготовки вопросы. Необходимо ориентировать студентов на систематическую подготовку к занятиям в течение семестра, что позволит использовать время экзаменационной сессии для систематизации знаний.

Методические рекомендации по подготовке к зачету/ зачету с оценкой

В ходе подготовки к зачету/зачету с оценкой студент, в первую очередь, должен систематизировать знания, полученные в ходе изучения дисциплины. К зачету/зачету с оценкой необходимо готовиться целенаправленно, регулярно, систематически и с первых дней обучения по данной дисциплине. В самом начале учебного курса познакомьтесь со следующей учебно-методической документацией:
- программой дисциплины;
- перечнем знаний, умений и навыков, которыми студент должен владеть;
- тематическими планами лекций, семинарских занятий;
- учебниками, учебными пособиями по дисциплине, а также электронными ресурсами;

- перечнем вопросов к зачету/зачету с оценкой.

После этого у обучающихся должно сформироваться четкое представление об объеме и характере знаний и умений, которыми надо будет овладеть по дисциплине. Систематическое выполнение учебной работы на лекциях и лабораторных занятиях позволит успешно освоить дисциплину и создать хорошую базу для сдачи зачета/зачета с оценкой.

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Рекомендуемая литература	
7.1.1. Основная литература	
Л.1.1	Батасова В. С., Воробьева И. А., Голубева И. В., Гречкина П. В., Маран М. М., Чибизова Н. В., Маран М. М. Программирование в примерах и задачах [Электронный ресурс]:учебное пособие для вузов. - Санкт-Петербург: Лань, 2024. - 260 с. – Режим доступа: https://e.lanbook.com/book/362825
Л.1.2	В.В. Трофимов и др. Основы алгоритмизации и программирования:учебник. - Москва: Издательство Юрайт, 2022. - 137 с.
Л.1.3	И.Г. Семакин и др. Основы алгоритмизации и программирования:учебник. - Москва: Издательский центр "Академия", 2021. - 304 с.
Л.1.4	Андрианова А. А., Исмагилов Л. Н., Мухтарова Т. М. Алгоритмизация и программирование. Практикум [Электронный ресурс]:учебное пособие. - Санкт-Петербург: Лань, 2022. - 240 с. – Режим доступа: https://e.lanbook.com/book/206258
Л.1.5	Юрина Т. А. Программирование и алгоритмизация [Электронный ресурс]:учебно-методическое пособие. - Омск: СибАДИ, 2021. - 88 с. – Режим доступа: https://e.lanbook.com/book/179228
Л.1.6	Конова Е. А., Поллак Г. А. Алгоритмы и программы. Язык С++ [Электронный ресурс]:учебное пособие для спо. - Санкт-Петербург: Лань, 2024. - 384 с. – Режим доступа: https://e.lanbook.com/book/382052
7.2. Лицензионное и свободно распространяемое программное обеспечение в том числе отечественного производства	
7.2.1	Microsoft Windows 10
7.2.2	Kaspersky Endpoint Security
7.3. Перечень профессиональных баз данных, информационных справочных систем и ресурсов сети Интернет	
7.3.1	Электронно-библиотечная система "Лань". Режим доступа: https://e.lanbook.com/
7.3.2	Электронно-библиотечная система "Университетская библиотека онлайн". Режим доступа: https://biblioclub.ru/
7.3.3	Электронно-библиотечная система "BOOK.ru". Режим доступа: https://book.ru/
7.3.4	Сайт национального открытого университета "ИНТУИТ". Режим доступа: https://intuit.ru/
7.3.5	ПЛАТФОРМА ОНЛАЙН-ОБРАЗОВАНИЯ «РАЗУМ». Режим доступа: https://razoom.mgutm.ru/

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

8.1	Адрес: 344002, Ростовская область, г. Ростов-на-Дону, пер. Семашко, д. 55: аудитория 20-28 - Компьютерный класс : 15 персональных компьютеров, объединённых в локальную вычислительную сеть на базе выделенного сервера приложений и web-сервера, наглядными пособиями, Мультимедийная установка «Toshiba» с экраном 3,5 м. Лицензионные программы:. ПО ОС Windows XP professional, MS Office 2003. Microsoft Visual Studio 2010-2013. Opera, Mozilla Firefox, Adobe Acrobat Reader 3.01,. WinDjView, КОМПАС-3D LT, 1С Предприятие 8.1, Система для моделирования бизнеса и оценки бизнес-проектов «Project Expert 7 ─ Эксперт Системс» (сетевая версия). CorelDRAW Graphics Suite X7. GRAPHISOFT ArchiCAD 15, Autodesk DWG Viewer, AVS DVD Player 2.4, CorelDraw Graphics Suite X4, Google SketchUp6.7, QuickTime, QBacic, Pascal, Delphi, Kaspersky Endpoint Security. IQlib. Все компьютеры подключены к сети Internet
8.2	Адрес: 344002, Ростовская область, г. Ростов-на-Дону, пер. Семашко, д. 55: аудитория 20-29 - Компьютерный класс : Оснащен: 12 персональными компьютерами, объединёнными в локальную вычислительную сеть на базе выделенного сервера приложений и web-сервера, наглядными пособиями, Мультимедийным проектором: Nec VT48 ANSI, с экраном 3,5 м. Лицензионные программы: ПО ОС Windows XP professional, MS Office 2003. Microsoft Visual Studio 2010-2013. Google Chome, Opera, Mozilla Firefox, Adobe Acrobat Reader 3.01, QBacic, Pascal, Delphi. WinDjView, КОМПАС-3D LT, GRAPHISOFT ArchiCAD 15, Audit XP. Система для моделирования бизнеса и оценки бизнес-проектов «Project Expert 7» (сетевая версия). 1С Предприятие 8.1, Autodesk DWG Viewer, AVS DVD Player 2.4, CorelDraw Graphics Suite X4, Google SketchUp6.7, QuickTime, Kaspersky Endpoint Security. IQlib. Инженерный калькулятор. Консультант Плюс. Все компьютеры подключены к сети Internet.

9. ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ

Организация образовательного процесса для лиц с ограниченными возможностями осуществляется в соответствии с «Методическими рекомендациями по организации образовательного процесса для инвалидов и лиц с ограниченными возможностями здоровья в образовательных организациях высшего образования, в том числе оснащенности образовательного процесса» Министерства образования и науки РФ от 08.04.2014г. № АК-44/05вн. В образовательном процессе используются социально-активные и рефлексивные методы обучения, технологии социокультурной реабилитации с целью оказания помощи в установлении полноценных межличностных отношений с другими студентами, создании комфортного психологического климата в студенческой группе. Студенты с ограниченными возможностями здоровья, в отличие от остальных студентов, имеют свои специфические особенности восприятия, переработки материала. Подбор и разработка учебных материалов производится с учетом индивидуальных особенностей. Предусмотрена возможность обучения по индивидуальному графику, при составлении которого

возможны различные варианты проведения занятий: в академической группе и индивидуально, на дому с использованием дистанционных образовательных технологий.

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
канд. экон. наук, доцент Левчук В.В._____ _____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2026 г. № __
Зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2026 г. № __
Зав. кафедрой Левчук Валерия Владимировна _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
канд. экон. наук, доцент Левчук В.В._____ _____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2026 г. № __
Зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2026 г. № __
Зав. кафедрой Левчук Валерия Владимировна _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
канд. экон. наук, доцент Левчук В.В._____ _____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2026 г. № __
Зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2026 г. № __
Зав. кафедрой Левчук Валерия Владимировна _____

==========

Актуализация с учетом развития науки, техники, культуры, экономики, техники, технологий и социальной сферы
Руководитель ОПОП
канд. экон. наук, доцент Левчук В.В._____ _____

Рабочая программа актуализирована, обсуждена и одобрена на заседании обеспечивающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2026 г. № __
Зав. кафедрой Левчук Валерия Владимировна _____

Рабочая программа согласована на заседании выпускающей кафедры

Физика, математика и информационные технологии

Протокол от __ _____ 2026 г. № __
Зав. кафедрой Левчук Валерия Владимировна _____