

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего
образования
"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ"

Кафедра № 13

УТВЕРЖДАЮ
Ответственный за образовательную
программу

ст.преподаватель

(должность, уч. степень, звание)

Н.И. Ускова

(инициалы, фамилия)

(подпись)

«21» апреля 2025 г

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Введение в информационные технологии»
(Наименование дисциплины)

Код направления подготовки/ специальности	25.03.01
Наименование направления подготовки/ специальности	Техническая эксплуатация летательных аппаратов и двигателей
Наименование направленности	Эксплуатация и испытания авиационной и космической техники
Форма обучения	заочная
Год приема	2025

Лист согласования рабочей программы дисциплины

Программу составил (а)

ст.преподаватель

(должность, уч. степень, звание)

(подпись, дата)

Н.И. Ускова

(инициалы, фамилия)

Программа одобрена на заседании кафедры № 13

«21» апреля 2025 г, протокол № 9

Заведующий кафедрой № 13

к.т.н.

(уч. степень, звание)

(подпись, дата)

Н.А. Овчинникова

(инициалы, фамилия)

Заместитель директора института №1 по методической работе

доп. к.т.н.

(должность, уч. степень, звание)

(подпись, дата)

В.Е. Таратун

(инициалы, фамилия)

Аннотация

Дисциплина «Введение в информационные технологии» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/специальности 25.03.01 «Техническая эксплуатация летательных аппаратов и двигателей» направленности «Эксплуатация и испытания авиационной и космической техники». Дисциплина реализуется кафедрой «№13».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

УК-1 «Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач»

ОПК-1 «Способен использовать основные законы математики, единицы измерения, фундаментальные принципы и теоретические основы физики, теоретической механики, гидравлики, имеющие отношение к техническому обслуживанию воздушных судов»

ОПК-4 «Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности»

ПК-14 «Способен вести производственно-техническую документацию и документацию установленной отчетности по утвержденным формам»

ПК-15 «Готовность использовать знание основных методов искусственного интеллекта в последующей профессиональной деятельности»

Содержание дисциплины охватывает круг вопросов, связанных с изучением основ программирования и основных концепций компьютерных наук, технологического процесса создания компонент программного обеспечения, удовлетворяющих современным требованиям к программному продукту.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, практические занятия, самостоятельная работа обучающегося, курсовое проектирование.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме дифференцированного зачета.

Общая трудоемкость освоения дисциплины составляет 3 зачетных единицы, 108 часов.

Язык обучения по дисциплине «русский»

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Получение обучающимися знаний основ программирования и основных концепций компьютерных наук, умений и навыков практического программирования, реализации базовых алгоритмов на языках высокого уровня; освоение технологического процесса создания компонент программного обеспечения, удовлетворяющих современным требованиям к программному продукту.

1.2. Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

Категория (группа) компетенции	Код и наименование компетенции	Код и наименование индикатора достижения компетенции
Универсальные компетенции	УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	УК-1.3.1 знать методики поиска, сбора и обработки информации, в том числе с использованием информационных технологий, включая интеллектуальные УК-1.У.1 уметь применять методики поиска, сбора и обработки информации, в том числе с использованием искусственного интеллекта УК-1.У.3 уметь оценивать информацию на достоверность; сохранять и передавать данные с использованием цифровых средств УК-1.В.1 владеть навыками критического анализа и синтеза информации, в том числе с помощью цифровых инструментов
Общепрофессиональные компетенции	ОПК-1 Способен использовать основные законы математики, единицы измерения, фундаментальные принципы и теоретические основы физики, теоретической механики, гидравлики, имеющие отношение к техническому обслуживанию воздушных судов	ОПК-1.У.1 уметь решать прикладные задачи, возникающие в ходе профессиональной деятельности

Общепрофессиональные компетенции	ОПК-4 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК-4.3.1 знать алгоритмы решения прикладных и инженерных задач ОПК-4.3.2 знать основные системные и прикладные программные средства для представления информации в требуемом формате ОПК-4.3.3 знать средства и методы защиты данных в локальных компьютерных сетях ОПК-4.3.4 знать перспективные методы информационных технологий и искусственного интеллекта, направленные на разработку новых научно-технических решений ОПК-4.У.1 уметь представлять информацию в требуемом формате с использованием информационных, компьютерных и сетевых технологий с учетом основных требований информационной безопасности ОПК-4.У.2 уметь решать прикладные и инженерные задачи с применением прикладных программных средств ОПК-4.У.3 уметь применять современные информационные технологии и перспективные методы искусственного интеллекта для решения задач профессиональной деятельности ОПК-4.В.1 владеть информационными, компьютерными и сетевыми технологиями, методами информационной безопасности ОПК-4.В.2 владеть прикладными программными средствами для решения инженерных задач
Профессиональные компетенции	ПК-14 Способен вести производственно-техническую документацию и документацию установленной отчетности по утвержденным формам	ПК-14.3.1 знать производственно-техническую документацию по авиационной технике (наличие, состояние, движение, ресурсы, списание, отчетность) и правила и стандарты ее ведения
Профессиональные компетенции	ПК-15 Готовность использовать знание основных методов искусственного интеллекта в последующей	ПК-15.3.1 знать современные технологии построения систем искусственного интеллекта в условиях неопределенности, основные модели, алгоритмы и методы нечеткой логики, а также базовые модели нейронной сети, которые могут быть использованы при формализации

	профессиональной деятельности	решений прикладных задач
--	-------------------------------	--------------------------

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- «Информатика»,
- «Алгоритмизация и программирование»,
- «Математика. Математический анализ»,
- «Математика. Аналитическая геометрия и линейная алгебра»,
- «Физика».

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и используются при изучении других дисциплин:

- «Развитие критического инженерного мышления»,
- «Аэродинамика и динамика полета»,
- «Термодинамика и теплотехника»,
- «Математика. Теория вероятностей и математическая статистика»,
- «Моделирование систем и процессов»,
- «Основы информационной безопасности»,
- «Цифровые информационные управляющие системы»,
- «Интеллектуальные системы»,
- «Конкретная авиационная техника»,
- «Основы испытания авиационной и космической техники»,
- «Системы сбора и обработки полетной информации».

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

Вид учебной работы	Всего	Трудоемкость по семестрам
		№4
1	2	3
Общая трудоемкость дисциплины, ЗЕ/ (час)	3/ 108	3/ 108
Из них часов практической подготовки	3	3
Аудиторные занятия, всего час.	12	12
в том числе:		
лекции (Л), (час)	4	4
практические/семинарские занятия (ПЗ), (час)	4	4
лабораторные работы (ЛР), (час)	4	4
курсовой проект (работа) (КП, КР), (час)	*	*
экзамен, (час)		
Самостоятельная работа, всего (час)	96	96
Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**)	Дифф. Зач.	Дифф. Зач.

Примечание: ** кандидатский экзамен

* - часы , не входящие в аудиторную нагрузку

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

Разделы, темы дисциплины	Лекции (час)	ПЗ (СЗ) (час)	ЛР (час)	КП (час)	СРС (час)
Семестр 4					
Раздел 1. Информатизация. Компьютерное мышление и способы кодирования информации. О-символика.	1				6
Раздел 2. Классификация языков программирования. Процедурные языки программирования. Математические пакеты.		4			17
Раздел 3. Основные команды Linux. Языки C, C++, Java, Python, синтаксис.	1		4		17
Раздел 4. Что такое криптография? Простые криптографические шифры. Баги. C/C++: строки и массивы.	1				10
Раздел 5. Алгоритмы сортировки. Компилятор. Побитовые операции.	1				6
Раздел 6. Рекурсия. Указатели, структура памяти, стек, очереди и связные списки.					10
Раздел 7. Система контроля версий.					3
Раздел 8. Глобальное информационное поле. Искусственный интеллект.					10
Выполнение курсовой работы				0	17
Итого в семестре:	4	4	4		96
Итого	4	4	4	0	96

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

Номер раздела	Название и содержание разделов и тем лекционных занятий
1.	Введение. Общая схема передачи информации. Информационное общество. Цифровая зрелость государства. Этапы развития вычислительной техники. Компьютерное мышление. Бинарная система счисления. Как она связана с десятичной, почему именно она применяется в компьютерах и причём здесь транзисторы. Системы кодирования информации.

	Таблицы ASCII. КОИ-8, UTF-8 и UTF-16. Кодирование изображения. JPEG. Шестнадцатеричная система счисления. Форматы кодирования звуковых и видео файлов. Что такое алгоритм? Эффективность алгоритмов. Псевдокод. Применение алгоритма «Разделяй и властвуй». Баги. Scratch: Демонстрация возможностей. Оценка времени работы алгоритма. О-символика.
2.	Популярные языки программирования. Компилируемые и Интерпретируемые ЯП. Универсальные и специализированные ЯП. Алгоритмические и Языки описания данных. Низкоуровневые и Высокоуровневые. Объектно-Ориентированные и Структурные языки программирования. Процедурные языки программирования. Математические пакеты. Основы MatLab.
3.	Основные команды Linux, компиляция и запуск новосозданных программ в командной строке. Программа типа «Hello, world» с пониманием её синтаксиса. Библиотеки C/C++, для чего и как их подключать к собственным программам. Что такое компилятор, как он устроен в C/C++. Функции C/C++, аргументы и значения, void, функция main и вызов из неё других функций. Численные и символьные типы данных. Ввод и вывод данных, спецификаторы вывода, плейсхолдеры. Точность при использовании разных типов данных. Арифметические и логические операторы. Деление по модулю. Оператор присваивания. Условные выражения if-else. Переключатели switch. Циклы do-while, while-do, for. Бесконечные циклы. Баги и к чему они приводят.
4.	Поиски багов. Бесконечный цикл. Функциональная декомпозиция: прием для повышения читаемости кода и удобства кодирования; Операторные скобки: область действия переменных; Объявление функций до реализации: специально для C/C++; Строки и как с ними работать; Что такое ошибка сегментации; Что такое массивы, одномерные и многомерные; Что такое аргументы командной строки, какова их связь с элементами массивов и как их использовать непосредственно в программах; Криптография — что это такое? Простейшие шифры: Шифр Цезаря, Шифр Виженера.
5.	Сортировка и поиск. Алгоритмы сортировки:

	Пузырьковая сортировка (Bubble Sort), Сортировка вставками (Insertion Sort), Сортировка выбором (Selection Sort), Пирамидальная сортировка (Heap Sort), Быстрая сортировка (Quick Sort), Сортировка слиянием (Merge Sort). Эффективность алгоритмов на языке О-символики. Применение и оптимизация. Компиляторы и интерпретаторы. Этапы разработки программы. Побитовые операции.
6.	Рекурсия. Сумма последовательности чисел: циклический и рекурсивный алгоритмы. Указатели, структура памяти, стек, очереди и связанные списки. Синтаксис указателей и структур. Куча и переполнение буфера. Сжатие. Hash-таблицы. Префиксное дерево.
7.	Что такое VCS? Локальные системы контроля версий. Централизованные системы контроля версий. Распределённые системы контроля версий. Основы работы с Git, репозиторий github.
8.	Основы построения интеллектуальных систем. Обработка знаний. Методы обучения и настройки интеллектуальных систем. Искусственный интеллект и поиск. Деревья. Технологии нечёткого управления. Экспертные системы. Нейросети.

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

№ п/п	Темы практических занятий	Формы практических занятий	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 4					
	Основы языка MatLab. Массивы.	Решение задач	1		2
	Решение типовых вычислительных задач в системе MatLab:	Решение задач	3		2
	Вычисление корней полинома.				
	Решение системы линейных уравнений.				
	Вычисление локального минимума и максимума функции.				
	Вычисление определенного интеграла.				
	Решение трансцендентного уравнения.				
Всего			4		

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

№ п/п	Наименование лабораторных работ	Трудоемкость, (час)	Из них практической подготовки, (час)	№ раздела дисциплины
Семестр 4				
1	Графика в среде MatLab	1		2
2	Линейный вычислительный процесс	1		3
3	Разветвляющийся вычислительный процесс:	1		3
	Условный оператор			
	Оператор выбора			
	Циклический вычислительный процесс:	1		3
	Цикл с параметром			
	Цикл с условием			
Всего		4		

4.5. Курсовое проектирование/ выполнение курсовой работы

Цель курсовой работы:

Часов практической подготовки:

Примерные темы заданий на курсовую работу приведены в разделе 10 РПД.

Обязательно указать темы на курсовую работу и выделить для неё время в СРС

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

Вид самостоятельной работы	Всего, час	Семестр 4, час
1	2	3
Изучение теоретического материала дисциплины (ТО)	60	60
Курсовое проектирование (КП, КР)	20	20
Расчетно-графические задания (РГЗ)		
Выполнение реферата (Р)		
Подготовка к текущему контролю успеваемости (ТКУ)		
Домашнее задание (ДЗ)	10	10
Контрольные работы заочников (КРЗ)		
Подготовка к промежуточной аттестации (ПА)	6	6
Всего:	96	96

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8– Перечень печатных и электронных учебных изданий

Шифр/ URL адрес	Библиографическая ссылка	Количество экземпляров в библиотеке (кроме электронных экземпляров)
https://elar.rsvpu.ru/bitstream/123456789/21893/1/Shireva_CCPP.pdf	Ширева, С. Н. Основы программирования на языке C/C++ [Электронный ресурс] : практикум / С. Н. Ширёва ; Рос. гос. проф.-пед. ун-т. - Екатеринбург : РГППУ, 2017. - 147 с.	
004 Ш 96	Шумова, Елена Олеговна. Объектно-ориентированное программирование : учебное пособие / Е. О. Шумова ; С.-Петерб. гос. ун-т аэрокосм. приборостроения. - Санкт-Петербург : Изд-во ГУАП, 2021. - 115 с. - Библиогр.: с. 112	5
316 Ш 20	Шанахан, М. Технологическая сингулярность / М. Шанахан. - М. : Точка : Альпина Паблишер, 2017. - 256 с.	3
004.4 Б24	Бар, Р. Дж. Язык Ада в проектировании систем = System design with Ada / Р. Дж. Бар ; пер. с англ. : О. С. Багатурова, В. М. Храпкин, В. С.	2

	Явнилович; ред. Е. К. Масловский. - М. : Мир, 1988. - 320 с.	
007 Н66	Нильсон, Н. Искусственный интеллект. Методы поиска решений / Н.Нильсон. - М. : Мир, 1973. - 270 с.	1

7. Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

URL адрес	Наименование
https://javarush.ru/	CS-50
https://stepik.org/	<u>Введение в Data Science и машинное обучение</u> <u>Быстрый старт в искусственный интеллект</u> <u>Введение в программирование (C++)</u> <u>Алгоритмы: теория и практика.</u> <u>Методы</u> <u>Алгоритмы: теория и практика. Структуры данных</u> <u>Программирование на языке C++. МНМИЦ СПбГУ</u> <u>Программирование на языке C++ (продолжение).</u> <u>МНМИЦ СПбГУ</u> <u>Веб-разработка для начинающих: HTML и CSS</u> <u>Программирование на Python</u>
https://docs.github.com/ru/get-started	Начало работы с документацией по GitHub
https://docs.exponenta.ru/	Документация MATLAB на русском языке

8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

№ п/п	Наименование
	MatLab не старше 2022г
	Компиляторы C/ C++
	IDE: средства для программирования на Python

8.2. Перечень информационно-справочных систем, используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

№ п/п	Наименование
	Не предусмотрено

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

№ п/п	Наименование составной части материально-технической базы	Номер аудитории (при необходимости)
1	Мультимедийная лекционная аудитория	
2	Компьютерный класс	

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

Вид промежуточной аттестации	Перечень оценочных средств
Дифференцированный зачёт	Список вопросов; Тесты; Задачи.
Выполнение курсовой работы	Экспертная оценка на основе требований к содержанию курсовой работы по дисциплине.

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
«отлично» «зачтено»	– обучающийся глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий.
«хорошо» «зачтено»	– обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения;

Оценка компетенции 5-балльная шкала	Характеристика сформированных компетенций
	– делает выводы и обобщения; – владеет системой специализированных понятий.
«удовлетворительно» «зачтено»	– обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий.
«неудовлетворительно» «не зачтено»	– обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений.

10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

№ п/п	Перечень вопросов (задач) для экзамена	Код индикатора
	Учебным планом не предусмотрено	

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

№ п/п	Перечень вопросов (задач) для зачета / дифф. зачета	Код индикатора
1.	Общая схема передачи информации. Информационное общество. Этапы развития вычислительной техники.	УК-1.3.1
2.	Компьютерное мышление. Бинарная система счисления. Как она связана с десятичной, почему именно она применяется в компьютерах и причём здесь транзисторы.	ОПК-4.3.1
3.	Что такое алгоритм? Эффективность алгоритмов.	ОПК-4.3.2
4.	Псевдокод. Типы языков программирования.	ОПК-4.3.3
5.	Таблицы ASCII. Системы кодирования информации.	ОПК-4.3.4
6.	Библиотеки C/C++. Пространство имён.	ПК-14.3.1
7.	Что такое компиляторы и интерпретаторы? Зачем нужны и как устроены.	ПК-15.3.1
8.	Препроцессор, компиляция, компоновка.	УК-1.3.1
9.	Этапы разработки программы.	ОПК-4.3.1
10.	Функции C/C++, аргументы и значения, void, функция main и вызов из неё других функций.	ОПК-4.3.2
11.	Численные и символьные типы данных.	ОПК-4.3.3
12.	Арифметические и логические операторы. Деление по	ОПК-4.3.4

	модулю. Остаток от деления.	
13.	Условные выражения if-else. Переключатели switch.	ПК-14.3.1
14.	Циклы do-while, while-do, for. Бесконечные циклы.	ПК-15.3.1
15.	Что такое массивы, одномерные и многомерные?	ОПК-4.3.1
16.	Криптография — что это вообще такое? Простейшие шифры.	ОПК-4.3.2
17.	Алгоритмы сортировки.	ОПК-4.3.3
18.	Рекурсия. Сумма последовательности чисел: циклический и рекурсивный алгоритмы.	ОПК-4.3.4
19.	Указатели, структура памяти, стек, очереди и связные списки	ПК-14.3.1
20.	Оценка времени работы алгоритма на примере поиска чисел Фибоначчи. Факторы, влияющие на точное время работы программы.	ПК-15.3.1
21.	Скорость работы алгоритма. O- символика, преимущества и недостатки. Общие правила оценки.	ОПК-4.3.1
22.	Кодирование изображения. Способы кодирования цвета.	ОПК-4.3.2
23.	Кодирование звуковой информации. Способы кодирования звуковой информации. Критерии качества. Форматы звуковых файлов.	ОПК-4.3.3
24.	Системы контроля версий. Git.	ОПК-4.3.4
25.	Основы построения интеллектуальных систем.	ПК-14.3.1
26.	Обработка знаний.	ПК-15.3.1
27.	Методы обучения и настройки интеллектуальных систем.	ОПК-4.3.1
28.	Искусственный интеллект и поиск. Деревья.	ОПК-4.3.2
29.	Технологии нечёткого управления.	ОПК-4.3.3
30.	Экспертные системы.	ОПК-4.3.4
31.	Нейросети.	ПК-14.3.1 ПК-15.3.1
32.	1. Напишите функцию power, реализующую возведение целого числа в неотрицательную целую степень. Функция power должна принимать на вход два целых числа и возвращать целое число. При выполнении задания учтите, что функция обязательно должна называться power, функция ничего не должна читать со входа или выводить. Требования к реализации: в этом задании вам нужно реализовать <i>только</i> функцию power. Вы можете определять вспомогательные функции, если они вам нужны. Реализовывать функции main не нужно. Ограничения: библиотеку cmath (и math.h) использовать запрещено.	УК-1.У.1
33.	2. Напишите программу, которая суммирует целые числа. На вход программе подаются целые числа в следующем формате: на первой строке идет целое число T — количество тестов, далее следует T строк, в каждой из которых через пробел идут два	УК-1.У.3

	целых числа ai и bi . На выводе для каждой из T строк нужно вывести сумму $ai+bi$ в том порядке, в котором пары поступают на вход. Ничего, кроме этого, выводить не нужно.	
34.	3. Напишите программу для решения квадратных уравнений вида $ax^2+bx+c=0$ (относительно x). На вход программа получает три целых числа: a , b и c , соответственно. При этом гарантируется, что $a \neq 0$. На вывод программа должна вывести два вещественных корня уравнения, разделённые пробелом. Если вещественных корней нет, то программа должна вывести строку "No real roots". Если у уравнения имеется только один корень (кратный корень), то программа должна вывести его дважды. Порядок вывода корней не важен. Ничего, кроме этого, выводить не нужно. Для вычислений с плавающей точкой используйте тип <code>double</code> . При выполнении задания вам может оказаться полезной функция <code>sqrt</code> из заголовочного файла <code>cmath</code> .	УК-1.B.1
35.	4. Дано натуральное число, выведите сумму его первой и последней цифры. Формат входных данных На вход дается натуральное число N , не превосходящее 10000. Формат выходных данных Выведите одно целое число - ответ на задачу.	ОПК-1.Y.1
36.	5. Дано целое трехзначное число. Найдите сумму его цифр. Формат входных данных На вход дается число от 100 до 999. Формат выходных данных Выведите одно целое число - ответ на задачу.	ОПК-4.Y.1
37.	6. Напишите программу, которая запрашивает пятиразрядное число и выводит цифры данного числа задом наперёд.	ОПК-4.Y.2
38.	7. Напишите программу, которая запрашивает пять натуральных чисел и выводит 1, если это число нечётное и 0 -- если число чётное.	ОПК-4.Y.3
39.	8. В этой задаче необходимо было ввести два числа a и b , и вывести их в обратном порядке через пробел. Программист ошибся, и в итоге программа не работает. Найдите и исправьте ошибки в тексте программы.	ОПК-4.B.1

	<pre>#include <iostream>using namespace std;int main(){ int a, b; cin << a << b; cout >> a >> " " >> b >> endl;return 0; }</pre>	
40.	9. В этой задаче вводятся три числа a, b, c и должна выводиться в первой строке сумма d этих чисел, а во второй -- их произведение ee. Программист, создававший код, допустил ошибку. Исправьте её. <pre>#include <iostream>using namespace std;int main(){ int a, b, c, d, e; cin >> a >> b >> c; d = a + b + c; e = a * b * c; cout << d; cout << e; return 0; }</pre>	ОПК-4.В.2
41.	10. Напишите программу, которая переводит показания секундомера в минуты и секунды. На вход программе подаётся одно число -- показание секундомера в секундах. Программа должна вывести два числа через пробел -- количество минут и количество секунд в этом показании.	УК-1.У.1
42.	11. Напишите программу, которая запрашивает число (от 500 до 5000), и выводит, к какому веку относится год с этим номером.	УК-1.У.3
43.	12. Напишите программу, которая запрашивает два промежутка времени в формате «часы минуты секунды», и выводит количество секунд между ними. Гарантируется, что второй промежуток времени наступил после первого, и показания взяты в одни и те же сутки.	УК-1.В.1
44.	13. Напишите программу, которая выдаёт из банкомата запрошенную сумму денег (кратную 10) в имеющихся купюрах (100 рублей, 50 рублей, 10 рублей). Сумму необходимо выдавать по возможности наиболее крупными купюрами. Сумма не будет превышать 20000 руб.	ОПК-1.У.1
45.	14. Вводится информационный объём сообщения – X бит (кратно 8). Найдите количество Килобайт (КВ) и Байт (В) в данном сообщении. В 1 Килобайте – 1024 Байта; В 1 Байте – 8 бит.	ОПК-4.У.1
46.	15. В доме N подъездов, в каждом из них M этажей, на каждом этаже K квартир. В первой строке через пробел вводятся три числа N, M и K. Во второй строке вводится X – номер квартиры. Определить, в каком подъезде находится эта квартира и на каком этаже. Гарантируется, что квартира с таким номером в данном доме существует.	ОПК-4.У.2
47.	16. По данному целому числу N распечатайте все квадраты натуральных чисел, не превосходящие N, в порядке возрастания.	ОПК-4.У.3
48.	17. По данному числу N распечатайте все целые	ОПК-4.В.1

	степени двойки, не превосходящие N, в порядке возрастания.	
49.	18. Напишите программу, которая запрашивает натуральное трёхзначное число, и если сумма его крайних цифр равна средней цифре – пишет «YES», в противном случае – «NO»	ОПК-4.В.2
50.	19. Напишите программу, которая запрашивает натуральное число N (не более 50), и выводит два ряда из N звёздочек.	УК-1.У.1
51.	20. Напишите программу, которая запрашивает два натуральных числа a и b , ($a \leq b$) и выводит все натуральные числа от a до b включительно через пробел.	УК-1.У.3
52.	21. В этой задаче необходимо было ввести два числа a и b, и вывести сумму, разность и произведение для этих чисел и выводит текст в следующем формате: <число a> + <число b> = <сумма чисел a и b> <число a> - <число b> = <разность чисел a и b> <число a> * <число b> = <произведение чисел a и b> Программист ошибся, и в итоге программа не работает. Найдите и исправьте ошибки в тексте программы. <pre> a = input() b = input() print(a, '+', b, '=', a+b, '\n', a, '-', b, '=', a-b, '\n', a, '*', b, '=', a*b, sep='') </pre>	УК-1.В.1
53.	21. В MatLab выведите несколько графиков в одном окне: график линейной функции, полином 2ой степени, экспоненциальную зависимость, тригонометрическую функцию. Укажите тип функций в заголовках и обозначьте оси.	ОПК-1.У.1
54.	22. Вычислите корни полинома $x^4 + 2x^3 - x - 1.$	ОПК-4.У.1
55.	23. Найдите решение системы линейных уравнений $\begin{cases} 3x_1 + 2x_2 + x_3 = 5 \\ 2x_1 + 3x_2 + x_3 = 1 \\ 2x_1 + x_2 + 3x_3 = 11 \end{cases}$	ОПК-4.У.2
56.	24 Спортсмен в первый день пробежал 10 км. Каждый следующий день он увеличивал дневную норму на 10% от результата предыдущего дня. Найти через сколько дней спортсмен пробежит суммарный путь более 100 км? Выведите таблицу с указанием дня и пройденного пути.	ОПК-4.У.3

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

№ п/п	Примерный перечень тем для курсового проектирования/выполнения
-------	--

	курсовой работы

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

№ п/п	Примерный перечень вопросов для тестов	Код индикатора
1.	Выберите все верные утверждения из списка. 1. С++ компилируемый язык программирования. 2. С++ язык с динамической типизацией. 3. С++ интерпретируемый язык программирования. 4. С++ ориентирован на быстрое создание прототипов приложений. 5. С++ язык со статической типизацией. 6. С++ не поддерживает процедурное программирование.	УК-1.3.1
2.	Отметьте все верные утверждения. Выберите все подходящие ответы из списка 1. Для запуска программы, код которой был написан на интерпретируемом языке, на компьютере должен быть установлен интерпретатор этого языка. 2. Код программы, написанный на языке, который компилируется в байт код виртуальной машины, достаточно скомпилировать однажды, чтобы программу можно было запускать на любой операционной системе, где есть соответствующая виртуальная машина. 3. Код программы, написанный на языке, который компилируется в машинный код, достаточно скомпилировать однажды, и потом программу можно будет запустить на любой операционной системе, для которой существует компилятор этого языка. 4. Для запуска программы, код которой был написан на компилируемом языке, на компьютере должен быть установлен компилятор этого языка. 5. Код программы, написанный на интерпретируемом языке, можно без предварительной компиляции запустить на любой операционной системе, где установлен интерпретатор этого языка. 6. Скомпилировать программу на С++ для некоторой архитектуры X можно только на компьютере с архитектурой X.	УК-1.У.1
3.	Выберите из списка объявления, которые не стоит помещать в заголовочные файлы	УК-1.У.3

	Выберите все подходящие ответы из списка 1. <code>void foo();</code> 2. <code>extern int a;</code> 3. <code>int a;</code> 4. <code>void foo() { std::cout << "Hello, World!\n"; }</code> 5. <code>void bar() { foo(); }</code>	
4.	Предположим, что в программе определён макрос <code>sqr</code>. <pre>#define sqr(x) x * x</pre> Какое значение будет иметь следующее выражение? <pre>sqr(3 + 0)</pre> Введите численный ответ	УК-1.B.1
5.	В коде программы определена следующая функция: <pre>int foo(int n) { if (n <= 0) return 1; return foo((n * 2) / 3) + foo(n - 2); }</pre> Нужно посчитать, сколько всего раз будет вызвана функция <code>foo</code>, если ее вызвать с аргументом 3 (т.е. <code>foo(3)</code>). Самый первый вызов тоже нужно посчитать. Подсказка: для решения этой задачи будет полезно нарисовать дерево рекурсивных вызовов. Введите численный ответ В коде определена следующая структура: <pre>struct ivector3d { int array[3]; };</pre> И определена следующая функция: <pre>void scale(ivector3d *v, int k) {</pre>	ОПК-1.Y.1

	<pre>for (int i = 0; i != 3; ++i) v->array[i] *= k; }</pre> Пусть у вас есть экземпляр iv3d структуры ivector3d. Изначально массив array экземпляра iv3d заполнен единицами, и вы вызываете функцию scale следующим образом: <pre>scale(&iv3d, 2);</pre> Какова будет сумма элементов массива array внутри iv3d по завершению функции? Введите численный ответ	
6.	структура ivector3d определена следующим образом: <pre>struct ivector3d { int *array; };</pre> У вас есть экземпляр iv3d этой структуры, поле array которого указывает на массив из трех элементов, заполненный единицами. Кроме того у вас есть функция scale, определенная следующим образом: <pre>void scale(ivector3d v, int k) { for (int i = 0; i != 3; ++i) v.array[i] *= k; }</pre> Задача та же, какова будет сумма элементов в массиве, на который указывает поле array, после следующего вызова: <pre>scale(iv3d, 2);</pre> Введите численный ответ	ОПК-4.3.1
7.	Компилятору C++ позволено выполнять оптимизации, связанные с удалением ненужных вызовов конструктора копирования (copy elision) при определенных условиях, даже если конструктор копирования содержит сайд эффекты (например, вывод в std::cout), поэтому на большинстве современных компиляторов конструктор копирования не будет вызван ни в одной из указанных ситуаций. Однако, компилятор не обязан выполнять эту оптимизацию. В каких из приведенных ниже ситуаций может быть вызван конструктор копирования класса String: 1. <pre>String spaces(size_t n) { const String s(n, ' '); return s; }</pre>	ОПК-4.3.2

	<pre> int main() { std::cout << spaces(10).str << "\n"; return 0; } 2. int main() { String ten_spaces; ten_spaces = String(10, ' '); std::cout << ten_spaces.str << "\n"; return 0; } 3. void foo(String str) { std::cout << str.str << "\n"; } int main() { foo(String(10, ' ')); return 0; } 4. void bar(const String &str) { std::cout << str.str << "\n"; } int main() { bar(" "); // ten spaces; return 0; } Выберите все подходящие ответы из списка 1. 1 2. 2 3. 3 4. 4 </pre>	
8.	Ниже даны несколько вариантов реализации оператора присваивания для класса String, выберите из них некорректные реализации, т.е. такие, которые неправильно работают с памятью (допускают утечки, обращаются к освобожденной памяти,	ОПК-4.3.3

	используют неинициализированные указатели), нарушают семантику оператора присваивания или инвариант класса. 1. <pre>String &operator=(const String &other) { str = new char[other.size + 1]; strcpy(str, other.str); size = other.size; return *this; }</pre> 2. <pre>String &operator=(const String &other) { delete[] str; str = new char[other.size + 1]; strcpy(str, other.str); size = other.size; return *this; }</pre> 3. <pre>String &operator=(const String &other) { if (this != &other) { str = other.str; size = other.size; } return *this; }</pre> Выберите все подходящие ответы из списка 1. 1 2. 2 3. 3	
9.	Есть три версии функции foo: <pre>void foo(char) { std::cout << "char" << std::endl; } void foo(signed char) { std::cout << "signed char" << std::endl; } void foo(unsigned char) { std::cout << "unsigned char" << std::endl; }</pre> Отметьте все верные утверждения относительно вызова функции foo. Выберите все подходящие ответы из списка 1. в результате вызова foo('a') будет выведено signed char	ОПК-4.3.4

	2. вызов foo('a') приведет к ошибке компиляции 3. в результате вызова foo(97) будет выведено char 4. вызов foo(97) приведет к ошибке компиляции 5. в результате вызова foo('a') будет выведено unsigned char 6. в результате вызова foo(97) будет выведено signed char 7. в результате вызова foo('a') будет выведено char	
10.	В программе определены две функции: <pre>float square(float value) { return value * value; } double square(float value) { return (double)value * value; }</pre> Далее в программе есть вызов: <pre>double sq = square(2.0);</pre> Отметьте все верные утверждения из списка. Выберите все подходящие ответы из списка 1. программа не скомпилируется, потому что такая перегрузка функции square не допустима 2. программа не скомпилируется, потому что вызов double sq = square(2.0) неоднозначен 3. программа скомпилируется, будет вызвана функция double square(float value) 4. программа скомпилируется, будет вызвана функция float square(float value)	ОПК-4.У.1
11.	В программе есть следующие определения: <pre>void promotion(char &) { std::cout << "char" << std::endl; } void promotion(int &) { std::cout << "int" << std::endl; } void promotion(long &) { std::cout << "long" << std::endl; }</pre> Кроме того в программе есть вызов: <pre>short sh = 10; promotion(sh);</pre> Отметьте все верные утверждения. Выберите все подходящие ответы из списка 1. вызов promotion(sh) не скомпилируется, так как есть несколько подходящих функций для вызова 2. при вызове promotion(sh) произойдет преобразование типа и будет вызвана функция promotion(char &)	ОПК-4.У.2

	3. при вызове <code>promotion(sh)</code> произойдет преобразование типа и будет вызвана функция <code>promotion(int &)</code> 4. при вызове <code>promotion(sh)</code> произойдет преобразование типа и будет вызвана функция <code>promotion(long &)</code> 5. вызов <code>promotion(sh)</code> не скомпилируется, так как нет ни одной подходящей функции для вызова	
12.	Выберите все верные утверждения. Выберите все подходящие ответы из списка 1. Все виртуальные методы базового класса являются виртуальными и для производных классов. 2. Невиртуальные методы не могут вызывать виртуальные методы. 3. Для того, чтобы вызвать виртуальный метод, объект должен быть создан через <code>new</code>. 4. Виртуальные методы могут вызывать неvirtуальные методы. 5. Обращение к объекту по ссылке не позволяет вызывать виртуальные методы. 6. Виртуальный метод можно вызвать только через указатель на объект. 7. Виртуальные методы могут быть константными.	ОПК-4.У.3
13.	Выберите все верные утверждения. Выберите все подходящие ответы из списка 1. Производные классы не могут переопределять <code>private</code>-виртуальные методы базового класса, если они унаследованы от базового класса с модификатором <code>private</code>. 2. У интерфейсов нет таблиц виртуальных методов. 3. Если в базовом классе виртуальная функция определена как <code>public</code>, то в производном классе её можно переопределить как <code>private</code>. 4. Производные классы не видят <code>protected</code>-предков своего (непосредственного) базового класса, если они унаследованы от базового класса с модификатором <code>private</code>. 5. Чистый виртуальный метод с определением — это не то же самое, что обычный виртуальный метод.	ОПК-4.В.1

	6. Производные классы не видят private-предков своего базового класса. 7. Если в базовом классе виртуальная функция определена как private, то в производном классе её можно переопределить как public.	
14.	В заголовочном файле count.hpp определена следующая функция: <pre>static int count() { static int counter = 0; return ++counter; }</pre> Этот файл подключается тремя файлами foo.cpp, bar.cpp и zoo.cpp. Сколько различных экземпляров переменной counter будут существовать в программе? Введите численный ответ	ОПК-4.В.2
15.	В заголовочном файле foo.hpp есть определение функции: <pre>inline void foo(int i) { std::cout << "i = " << i << std::endl; }</pre> В программе есть три корректных файла с кодом first.cpp, second.cpp и third.cpp, которые подключают foo.hpp. Отметьте все верные утверждения из списка. Выберите все подходящие ответы из списка 1) При компоновке лишние определения функции foo будут отброшены. 2) Программа компилируется и компоуется без проблем. 3) Файлы first.cpp, second.cpp и third.cpp компилируются без проблем. 4) Файлы first.cpp, second.cpp и third.cpp компилируются без проблем, но на этапе компоновки возникает ошибка из-за множественного определения функции foo.	ПК-14.3.1
16.	В заголовочном файле count.hpp определена следующая функция: <pre>inline int count() { static int counter = 0; return ++counter; }</pre> Этот файл подключается тремя файлами foo.cpp, bar.cpp и zoo.cpp. Сколько различных экземпляров переменной counter будут существовать в программе?	ПК-15.3.1

	Введите численный ответ	
17.	Давайте вспомним как решается линейное уравнение $ax+b=0$. Будем считать, что $a \neq 0$. На следующем шаге необходимо выбрать правильный вариант формулы. Выберите один вариант из списка 1) $x = a / b$ 2) $x = b / a$ 3) $x = - a / b$ 4) $x = - b / a$	УК-1.3.1
18.	Используя предыдущую программу нахождения корня линейного уравнения как образец, создайте программу нахождения площади прямоугольника. Нарисуйте блок-схему (выполняется в случае очного обучения). Изображение блок-схемы можно загрузить с помощью пункта "Insert Image". При любом ответе система выдает положительную оценку, но качество решения оценивает только преподаватель! Напишите развернутый ответ	УК-1.У.1
19.	Создайте программу решения квадратного уравнения (считать, что дискриминант D всегда больше 0). Нарисуйте блок-схему (выполняется в случае очного обучения). Изображение блок-схемы можно загрузить с помощью пункта "Insert Image". При любом ответе система выдает положительную оценку, но качество решения оценивает только преподаватель! Напишите развернутый ответ	УК-1.У.3
20.	Можно ли угадать задуманное целое число от 1 до 1000, задав только 500 вопросов? (Разрешаются вопросы с ответом "да" или "нет".) Выберите один вариант из списка 1) да 2) нет	УК-1.В.1
21.	Слева указаны формулы, записанные с помощью логических операций, а справа — записанные с помощью арифметических операций по модулю 2 (значения 0 и 1, умножение обычное, а сложение по модулю 2, то есть $1+1=0$). Составьте пары из формул, задающих одну и ту же функцию (в разных обозначениях). Сопоставьте значения из двух списков 1) A or B	ОПК-1.У.1

	2) $A \text{ and } B$ 3) $\text{not } (A \Rightarrow B)$ 4) $A \text{ xor } B$ a) $A + B - A * B$ b) $A * B$ c) $A - A * B$ d) $(A + B) \bmod 2$	
22.	окажите, что разные логические связки можно выразить как многочлены, если считать истину единицей, а ложь нулём. (Отсюда, зная о возможности представления любой булевой функции в конъюнктивной нормальной форме, можно было бы вывести, что любая функция представляется многочленом. Но нам нужна оценка для числа операций в этом многочлене, так что мы дадим другое доказательство.) Сопоставьте значения из двух списков 1) $a \text{ или } b$ 2) $a \text{ и } b$ 3) $\text{не } a$ 4) $a \text{ xor } b$ a) $a \cdot b$ b) $a + 1$ c) $a + b + a \cdot b$ d) $a + b$	ОПК-4.3.1
23.	Мы видели, что любая булева функция $B(x_1, \dots, x_n)$ может быть представлена многочленом по модулю 2, то есть суммой мономов, каждый из которых является произведением некоторых переменных (допустим также пустой моном, не содержащий переменных; он равен 1). Какие свойства функции B соответствуют каким свойствам представляющего её многочлена? Сопоставьте значения из двух списков 1) $B(0,0,0,\dots) + B(1,0,0,\dots) = 1 \pmod{2}$ 2) $B(0,0,0,\dots) = 1$ 3) в таблице значений булевой функции нечётное число единиц a) многочлен включает в себя моном, содержащий все переменные b) многочлен включает в себя пустой моном c) многочлен включает в себя моном x_1 (содержащий единственную переменную)	ОПК-4.3.2
24.	Соотнесите проблемы, связанные с обучением нейронной сети с помощью градиентного спуска, с наиболее очевидными методами их решения.	ОПК-4.3.3

	Сопоставьте значения из двух списков 1) Застревание в локальном минимуме 2) Веса и целевая функция почти/вообще не меняются 3) Переобучение 4) Форма изменения целевой функции хорошая, но обучение протекает долго a) Устроить кросс-валидацию b) Попробовать несколько начальных точек (начальных весов) сети c) Увеличить learning rate d) Нормировать входы	
25.	Расставьте китов алгоритма обратного распространения ошибки в том порядке, в котором они нужны, чтобы вычислить $\partial J / \partial b_3^2$ для сети, в которой $L=6$ слоёв. Неиспользованных китов, если таковые найдутся, поставьте последними. Расположите элементы списка в правильном порядке $\frac{\partial J}{\partial b_j^l} = \delta_j^l$ $\delta^l = \left((W^{l+1})^T \delta^{l+1} \right) \odot (a^l)'(z^l)$ $\delta^L = \nabla_{a^L} J \odot (a^L)'(z^L)$ $\frac{\partial J}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l$	ОПК-4.3.4

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

№ п/п	Перечень контрольных работ
	Не предусмотрено

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины

11.1. Методические указания для обучающихся по освоению лекционного материала .

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления.
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Структура предоставления лекционного материала:

Согласно таблице 4. С представлением демонстрационного материала и интерактивной работой со обучающимися.

11.2. Методические указания для обучающихся по выполнению лабораторных работ

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ

См. таблицу 6.

Структура и форма отчета о лабораторной работе

Согласно требованиям, представленным на сайте ГУАП <https://guap.ru/standart/doc>

Требования к оформлению отчета о лабораторной работе

Согласно требованиям, представленным на сайте ГУАП <https://guap.ru/standart/doc>

11.3. Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине;
- методические указания по выполнению контрольных работ (для обучающихся по заочной форме обучения).

11.4. Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

11.5. Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

- дифференцированный зачет – это форма оценки знаний, полученных обучающимся при изучении дисциплины, при выполнении курсовых проектов, курсовых работ, научно-исследовательских работ и прохождении практик с аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Система оценок при проведении промежуточной аттестации осуществляется в соответствии с требованиями Положений «О текущем контроле успеваемости и промежуточной аттестации студентов ГУАП, обучающихся по программам высшего образования» и «О модульно-рейтинговой системе оценки качества учебной работы студентов ГУАП».

Лист внесения изменений в рабочую программу дисциплины

Дата внесения изменений и дополнений. Подпись внесшего изменения	Содержание изменений и дополнений	Дата и № протокола заседания кафедры	Подпись зав. кафедрой