

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Ростовский государственный экономический университет (РИНХ)»

УТВЕРЖДАЮ  
Директор Таганрогского института  
имени А. П. Чехова (филиала)  
РГЭУ (РИНХ)  
\_\_\_\_\_ С. А. Петрушенко  
«20» мая 2025 г.

**Рабочая программа дисциплины  
Языки и методы программирования**

Направление подготовки  
09.03.03 Прикладная информатика

Направленность (профиль) программы бакалавриата  
09.03.03.02 Разработка программного обеспечения

Для набора 2025 года

Квалификация  
Бакалавр

**КАФЕДРА информатики****Распределение часов дисциплины по семестрам / курсам**

Курс Вид занятий	3		4		Итого	
	уп	рп	уп	рп		
Лекции	6	6			6	6
Лабораторные	12	12	6	6	18	18
Итого ауд.	18	18	6	6	24	24
Контактная работа	18	18	6	6	24	24
Сам. работа	158	158	93	93	251	251
Часы на контроль	4	4	9	9	13	13
Итого	180	180	108	108	288	288

**ОСНОВАНИЕ**

Учебный план утвержден учёным советом вуза от 28.02.2025 протокол № 9.

Программу составил(и): д-р техн. наук, Проф., Джанунц Гарик Апетович

Зав. кафедрой: Тюшнякова И.А.

### 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	формирование у обучающихся системы теоретических знаний и практических навыков программирования на языке Python, включая освоение базовых конструкций языка, объектно-ориентированного подхода, структур данных и алгоритмов для решения прикладных задач в профессиональной деятельности.
-----	--

### 2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

ОПК-1:	Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности
ОПК-1.1:	Знает основы математики, физики, вычислительной техники и программирования
ОПК-1.2:	Умеет решать стандартные профессиональные задачи с применением естественнонаучных и обще-инженерных знаний, методов математического анализа и моделирования
ОПК-1.3:	Владеет навыками теоретического и экспериментального исследования объектов профессиональной деятельности
ОПК-2:	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности
ОПК-2.1:	Знает современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности
ОПК-2.2:	Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства при решении задач профессиональной деятельности
ОПК-2.3:	Владеет навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК-7:	Способен разрабатывать алгоритмы и программы, пригодные для практического применения
ОПК-7.1:	Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий
ОПК-7.2:	Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ
ОПК-7.3:	Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач

#### В результате освоения дисциплины обучающийся должен:

##### **Знать:**

базовые конструкции языка Python (соотнесено с индикатором ОПК-7.1); принципы объектно-ориентированного программирования (соотнесено с индикатором ОПК-7.1); основные структуры данных и алгоритмы: линейные структуры, хеш-таблицы, алгоритмы поиска, алгоритмы сортировки, алгоритмы обхода графов, принципы динамического программирования (соотнесено с индикатором ОПК-1.1); современные программные среды разработки и инструменты для написания, отладки и тестирования кода (соотнесено с индикатором ОПК-2.1).

##### **Уметь:**

разрабатывать программы на Python с использованием условных операторов, циклов, функций и структур данных для решения прикладных задач (соотнесено с индикатором ОПК-7.2); применять объектно-ориентированный подход при разработке программ (соотнесено с индикатором ОПК-7.2); реализовывать базовые алгоритмы поиска, сортировки и обхода графов, анализировать их эффективность (соотнесено с индикатором ОПК-1.2); выбирать и использовать современные инструменты разработки и отладки, включая отечественное программное обеспечение, при решении профессиональных задач (соотнесено с индикатором ОПК-2.2).

##### **Владеть:**

навыками написания программного кода на Python с соблюдением стандартов оформления и требований к структуре программы (соотнесено с индикатором ОПК-7.3); навыками отладки и тестирования программ, включая обработку исключений и проверку граничных случаев (соотнесено с индикатором ОПК-7.3); навыками применения алгоритмических методов для решения практических задач (соотнесено с индикатором ОПК-1.3); навыками работы в современных средах разработки и использования инструментов для автоматизации процессов разработки (соотнесено с индикатором ОПК-2.3).

### 3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

#### Раздел 1. Язык программирования Python: основы и базовые конструкции

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
1.1	Лекция 1. Введение в Python. Переменные и типы данных.	Лекционные занятия	3	2	ОПК-1

	Что такое Python, области применения. Установка и настройка среды. Переменные, правила именования. Функция print(). Основные типы данных: int, float, str, bool.				ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.2	Лекция 2. Работа со строками и числами. Строки: неизменяемость, конкатенация, интерполяция, срезы, основные методы. Числа: целые и вещественные, арифметические операции.	Лекционные занятия	3	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.3	Лекция 3. Условные операторы и логические выражения. Условный оператор if-elif-else. Логические операторы and, or, not. Truthy и falsy значения. Вычисление по короткой схеме.	Лекционные занятия	3	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.4	Функции и области видимости Определение функций, параметры, возвращаемые значения. Области видимости переменных.	Самостоятельная работа	3	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.5	Списки, кортежи и циклы Списки: создание, методы, срезы. Кортежи: особенности, методы. Циклы for и while. Функции range(), enumerate(), zip().	Самостоятельная работа	3	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.6	Списковые включения и lambda-функции Генераторы списков. Лямбда-функции: синтаксис, применение с map(), filter(), sorted().	Самостоятельная работа	3	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.7	Словари и множества Словари: создание, доступ к элементам, методы. Обход словарей. Множества: создание, операции объединения, пересечения, разности.	Самостоятельная работа	3	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3

					ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.8	Модули и обработка ошибок Стандартная библиотека Python. Импорт модулей. Обработка исключений: try-except-else-finally. Оператор raise. Типы исключений.	Самостоятельная работа	3	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.9	Знакомство со средой разработки. Переменные и типы данных	Лабораторные занятия	3	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.10	Работа со строками и числами	Лабораторные занятия	3	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.11	Создание программы «Планировщик погоды для путешествий» с использованием условных операторов	Лабораторные занятия	3	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.12	Разработка функции «Калькулятор скидок»	Лабораторные занятия	3	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.13	Создание RPG-персонажа с проверкой входных данных	Лабораторные занятия	3	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2

					ОПК-1.3
1.14	Генератор числовых паттернов с использованием циклов	Лабораторные занятия	3	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.15	Работа со списками и кортежами	Самостоятельная работа	3	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.16	Списковые включения и лямбда-функции	Самостоятельная работа	3	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.17	Работа со словарями и множествами	Самостоятельная работа	3	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.18	Разработка менеджера конфигурации пользователя	Самостоятельная работа	3	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.19	Обработка исключений. Отладка ISBN-валидатора	Самостоятельная работа	3	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.20	Создание класса «Планета»	Самостоятельная работа	3	6	ОПК-1 ОПК-2 ОПК-7

					ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.21	Разработка приложения для учёта бюджета	Самостоятельная работа	3	10	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.22	Изучение лекционного материала, основной и дополнительной литературы. Конспектирование основных понятий. Разбор примеров из рекомендуемых источников	Самостоятельная работа	3	34	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.23	Подготовка к лабораторным работам и текущему контролю	Самостоятельная работа	3	20	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.24	Подготовка доклада по избранной теме	Самостоятельная работа	3	28	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
1.25	Подготовка к промежуточной аттестации	Зачет	3	4	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3

**Раздел 2. Методы программирования: ООП, структуры данных и алгоритмы**

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
2.1	Объектно-ориентированное программирование: классы и объекты	Самостоятельная	4	6	ОПК-1

	Парадигма ООП. Классы, объекты, атрибуты, методы. Конструктор <code>__init__</code> . Специальные методы ( <code>__str__</code> , <code>__repr__</code> ).	работа			ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.2	Инкапсуляция и управление доступом Инкапсуляция. Приватные атрибуты. Свойства, геттеры и сеттеры. Динамическое управление атрибутами.	Самостоятельная работа	4	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.3	Наследование и полиморфизм. Абстрактные классы и интерфейсы Наследование, переопределение методов. Функция <code>super()</code> . Полиморфизм. Искажение имён. Абстрактные классы (ABC). Абстрактные методы. Принципы абстракции.	Самостоятельная работа	4	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.4	Линейные структуры данных и алгоритмы Понятие алгоритма. Нотация «O большое». Динамические и статические массивы. Стеки и очереди. Связные списки (односвязные, двусвязные).	Самостоятельная работа	4	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.5	Хеш-таблицы и множества Принцип работы хеш-таблиц. Хеш-функции. Коллизии и способы их разрешения. Реализация словарей и множеств в Python.	Самостоятельная работа	4	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.6	Алгоритмы поиска Линейный поиск. Бинарный поиск. Разделяй и властвуй. Метод бисекции.	Самостоятельная работа	4	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.7	Алгоритмы сортировки Квадратичные сортировки: Сортировка выбором. Эффективные сортировки: Быстрая сортировка. Сравнительный анализ.	Самостоятельная работа	4	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3

					ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.8	Графы и алгоритмы обхода Понятие графа. Способы представления: матрица смежности, список смежности. Обход в глубину (DFS) и в ширину (BFS). Примеры применения.	Самостоятельная работа	4	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.9	Деревья и динамическое программирование Понятие дерева. Двоичные деревья поиска. Префиксные деревья. Принципы динамического программирования. Мемоизация и табуляция.	Самостоятельная работа	4	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.10	Создание трекера характеристик игрового персонажа	Лабораторные занятия	4	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.11	Разработка иерархии классов «Фигуры» (прямоугольник, квадрат) с наследованием	Лабораторные занятия	4	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.12	Реализация абстрактного класса «Игрок» и конкретного класса «Пешка»	Лабораторные занятия	4	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.13	Калькулятор площади многоугольников	Самостоятельная работа	4	4	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2

					ОПК-1.3
2.14	Вычисление квадратного корня методом бисекции	Самостоятельная работа	4	4	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.15	Реализация алгоритма быстрой сортировки	Самостоятельная работа	4	4	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.16	Реализация алгоритма сортировки выбором	Самостоятельная работа	4	4	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.17	Реализация алгоритма Луна для валидации кредитных карт	Самостоятельная работа	4	4	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.18	Преобразование списка смежности в матрицу смежности	Самостоятельная работа	4	7	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.19	Реализация обхода графа в глубину (DFS)	Самостоятельная работа	4	6	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.20	Решение задачи N-Ферзей с использованием DFS	Самостоятельная работа	4	4	ОПК-1 ОПК-2 ОПК-7

					ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
2.21	Вычисление чисел Фибоначчи методом динамического программирования	Самостоятельная работа	4	2	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3
<b>Раздел 3. Экзамен</b>					
№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
3.1	Подготовка к промежуточной аттестации	Экзамен	4	9	ОПК-1 ОПК-2 ОПК-7 ОПК-7.1 ОПК-7.2 ОПК-7.3 ОПК-2.1 ОПК-2.2 ОПК-2.3 ОПК-1.1 ОПК-1.2 ОПК-1.3

#### 4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Структура и содержание фонда оценочных средств для проведения текущего контроля и промежуточной аттестации представлены в Приложении 1 к рабочей программе дисциплины.

#### 5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

##### 5.1. Учебные, научные и методические издания

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
1	Златопольский Д. М.	Программирование: типовые задачи, алгоритмы, методы: учебное пособие	Москва: Лаборатория знаний, 2020	<a href="http://biblioclub.ru/index.php?page=book&amp;id=222873">http://biblioclub.ru/index.php?page=book&amp;id=222873</a>
2	Северенс Ч.	Введение в программирование на Python	Москва: Национальный Открытый Университет «ИНТУИТ», 2016	<a href="http://biblioclub.ru/index.php?page=book&amp;id=429184">http://biblioclub.ru/index.php?page=book&amp;id=429184</a>
3	Дэвид Копец	Классические задачи Computer Science на языке Python	Санкт-Петербург: Питер, 2021	<a href="https://ibooks.ru/reading.php?short=1&amp;productid=365295">https://ibooks.ru/reading.php?short=1&amp;productid=365295</a>

##### 5.1. Учебные, научные и методические издания

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
1	Кнут Д.Э.	Основные алгоритмы	М.; СПб.; Киев: Издат. дом "Вильямс", 2000	
2	Нагаева И. А., Кузнецов И. А.	Алгоритмизация и программирование. Практикум: учебное пособие	Москва Берлин: Директ-Медиа, 2019	<a href="http://biblioclub.ru/index.php?page=book&amp;id=570287">http://biblioclub.ru/index.php?page=book&amp;id=570287</a>

	Авторы, составители	Заглавие	Издательство, год	Библиотека / Количество
3	Билл Любанович	Простой Python. Современный стиль программирования. 2-е изд.	Санкт-Петербург: Питер, 2021	<a href="https://ibooks.ru/reading.php?short=1&amp;productid=373510">https://ibooks.ru/reading.php?short=1&amp;productid=373510</a>

### 5.2. Профессиональные базы данных и информационные справочные системы

Национальная электронная библиотека (НЭБ) <https://rusneb.ru/>  
 ИСС «Гарант» <http://www.internet.garant.ru/>  
 Российская государственная библиотека <https://www.rsl.ru/>  
 Научная электронная библиотека <https://elibrary.ru/defaultx.asp?>

### 5.3. Перечень программного обеспечения

Python  
 OpenOffice

### 5.4. Учебно-методические материалы для обучающихся с ограниченными возможностями здоровья

При необходимости по заявлению обучающегося с ограниченными возможностями здоровья учебно-методические материалы предоставляются в формах, адаптированных к ограничениям здоровья и восприятия информации. Для лиц с нарушениями зрения: в форме аудиофайла; в печатной форме увеличенным шрифтом. Для лиц с нарушениями слуха: в форме электронного документа; в печатной форме. Для лиц с нарушениями опорно-двигательного аппарата: в форме электронного документа; в печатной форме.

## 6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Помещения для всех видов работ, предусмотренных учебным планом, укомплектованы необходимой специализированной учебной мебелью и техническими средствами обучения:

- столы, стулья;
- персональный компьютер / ноутбук (переносной);
- проектор;
- экран / интерактивная доска.

Лабораторные занятия проводятся в компьютерных классах, рабочие места в которых оборудованы необходимыми лицензионными и/или свободно распространяемыми программными средствами и выходом в Интернет, и/или в специализированных лабораториях, предусмотренных образовательной программой.

## 7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

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

## ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

### 1 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

#### 1.1 Показатели и критерии оценивания компетенций:

ЗУН, составляющие компетенцию	Показатели оценивания	Критерии оценивания	Средства оценивания
<b>ОПК-1: Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности</b>			
<i>Знать:</i> основные структуры данных и алгоритмы: линейные структуры, хеш-таблицы, алгоритмы поиска, алгоритмы сортировки, алгоритмы обхода графов, принципы динамического программирования	Выполняет содержательный анализ избранной темы. Подготовка доклада по избранной теме. Выполняет лабораторные задания. Изучает лекционный материал, основную и дополнительную литературу. Выполняет поиск и сбор необходимой информации.	Полнота раскрытия темы. Полнота и правильность выполнения лабораторных заданий, наличие выводов. Полнота и содержательность ответа.	Д – доклад (темы 1-20) ЛЗ – лабораторные задания (3 курс – задания 1-13, 4 курс – задания 14-23) З – вопросы к зачету (1-23) Э – вопросы к экзамену (1-25)
<i>Уметь:</i> реализовывать базовые алгоритмы поиска, сортировки и обхода графов, анализировать их эффективность	Выполняет содержательный анализ избранной темы. Подготовка доклада по избранной теме. Выполняет лабораторные задания. Изучает лекционный материал, основную и дополнительную литературу. Выполняет поиск и сбор необходимой информации.	Полнота раскрытия темы. Полнота и правильность выполнения лабораторных заданий, наличие выводов. Полнота и содержательность ответа.	Д – доклад (темы 1-20) ЛЗ – лабораторные задания (3 курс – задания 1-13, 4 курс – задания 14-23) З – вопросы к зачету (1-23) Э – вопросы к экзамену (1-25)
<i>Иметь навыки:</i> навыками применения алгоритмических методов для решения практических задач;	Выполняет содержательный анализ избранной темы. Подготовка доклада по избранной теме. Выполняет лабораторные задания.	Полнота раскрытия темы. Полнота и правильность выполнения лабораторных заданий, наличие выводов.	Д – доклад (темы 1-20) ЛЗ – лабораторные задания (3 курс – задания 1-13, 4 курс – задания

	Изучает лекционный материал, основную и дополнительную литературу. Выполняет поиск и сбор необходимой информации.	Полнота и содержательность ответа.	14-23) З – вопросы к зачету (1-23) Э – вопросы к экзамену (1-25)
ОПК-2: Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности			
<i>Знать:</i> современные программные среды разработки и инструменты для написания, отладки и тестирования кода	Выполняет содержательный анализ избранной темы. Подготовка доклада по избранной теме. Выполняет лабораторные задания. Изучает лекционный материал, основную и дополнительную литературу. Выполняет поиск и сбор необходимой информации.	Полнота раскрытия темы. Полнота и правильность выполнения лабораторных заданий, наличие выводов. Полнота и содержательность ответа.	Д – доклад (темы 1-20) ЛЗ – лабораторные задания (3 курс – задания 1-13, 4 курс – задания 14-23) З – вопросы к зачету (1-23) Э – вопросы к экзамену (1-25)
<i>Уметь:</i> выбирать и использовать современные инструменты разработки и отладки, включая отечественное программное обеспечение, при решении профессиональных задач	Выполняет содержательный анализ избранной темы. Подготовка доклада по избранной теме. Выполняет лабораторные задания. Изучает лекционный материал, основную и дополнительную литературу. Выполняет поиск и сбор необходимой информации.	Полнота раскрытия темы. Полнота и правильность выполнения лабораторных заданий, наличие выводов. Полнота и содержательность ответа.	Д – доклад (темы 1-20) ЛЗ – лабораторные задания (3 курс – задания 1-13, 4 курс – задания 14-23) З – вопросы к зачету (1-23) Э – вопросы к экзамену (1-25)
<i>Иметь навыки:</i> навыками работы в современных средах разработки и использования инструментов для автоматизации процессов разработки	Выполняет содержательный анализ избранной темы. Подготовка доклада по избранной теме. Выполняет лабораторные задания. Изучает лекционный материал, основную и дополнительную литературу. Выполняет поиск и сбор необходимой информации.	Полнота раскрытия темы. Полнота и правильность выполнения лабораторных заданий, наличие выводов. Полнота и содержательность ответа.	Д – доклад (темы 1-20) ЛЗ – лабораторные задания (3 курс – задания 1-13, 4 курс – задания 14-23) З – вопросы к зачету (1-23) Э – вопросы к экзамену (1-25)

	информации.		
ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения			
<i>Знать:</i> базовые конструкции языка Python; принципы объектно-ориентированного программирования	Выполняет содержательный анализ избранной темы. Подготовка доклада по избранной теме. Выполняет лабораторные задания. Изучает лекционный материал, основную и дополнительную литературу. Выполняет поиск и сбор необходимой информации.	Полнота раскрытия темы. Полнота и правильность выполнения лабораторных заданий, наличие выводов. Полнота и содержательность ответа.	Д – доклад (темы 1-20) ЛЗ – лабораторные задания (3 курс – задания 1-13, 4 курс – задания 14-23) З – вопросы к зачету (1-23) Э – вопросы к экзамену (1-25)
<i>Уметь:</i> разрабатывать программы на Python с использованием условных операторов, циклов, функций и структур данных для решения прикладных задач; применять объектно-ориентированный подход при разработке программ	Выполняет содержательный анализ избранной темы. Подготовка доклада по избранной теме. Выполняет лабораторные задания. Изучает лекционный материал, основную и дополнительную литературу. Выполняет поиск и сбор необходимой информации.	Полнота раскрытия темы. Полнота и правильность выполнения лабораторных заданий, наличие выводов. Полнота и содержательность ответа.	Д – доклад (темы 1-20) ЛЗ – лабораторные задания (3 курс – задания 1-13, 4 курс – задания 14-23) З – вопросы к зачету (1-23) Э – вопросы к экзамену (1-25)
<i>Иметь навыки:</i> навыками написания программного кода на Python с соблюдением стандартов оформления и требований к структуре программы; навыками отладки и тестирования программ, включая обработку исключений и проверку граничных случаев	Выполняет содержательный анализ избранной темы. Подготовка доклада по избранной теме. Выполняет лабораторные задания. Изучает лекционный материал, основную и дополнительную литературу. Выполняет поиск и сбор необходимой информации.	Полнота раскрытия темы. Полнота и правильность выполнения лабораторных заданий, наличие выводов. Полнота и содержательность ответа.	Д – доклад (темы 1-20) ЛЗ – лабораторные задания (3 курс – задания 1-13, 4 курс – задания 14-23) З – вопросы к зачету (1-23) Э – вопросы к экзамену (1-25)

## 1.2 Шкалы оценивания:

Текущий контроль успеваемости и промежуточная аттестация осуществляется в рамках накопительной балльно-рейтинговой системы в 100-балльной шкале:

50-100 баллов (оценка «зачет»);

0-49 баллов (оценка «незачет»).

84-100 баллов (оценка «отлично»);

67-83 баллов (оценка «хорошо»);

50-66 баллов (оценка «удовлетворительно»);

0-49 баллов (оценка «неудовлетворительно»);

## **2 Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

### **Вопросы к зачету**

1. Понятие переменной в Python. Правила именования переменных. Базовые типы данных.
2. Функция print(): назначение, форматирование вывода, параметры sep и end.
3. Строки в Python: свойства, неизменяемость, конкатенация, интерполяция.
4. Срезы строк: синтаксис, примеры использования, отрицательные индексы.
5. Основные методы строк: upper(), lower(), split(), join(), strip().
6. Числовые типы данных: int и float. Арифметические операции и их приоритет.
7. Условный оператор if-elif-else: синтаксис, примеры использования.
8. Логические операторы and, or, not. Таблицы истинности.
9. Понятие truthy и falsy значений в Python. Примеры.
10. Функции: определение, вызов, параметры, возвращаемые значения.
11. Области видимости переменных: локальные, глобальные, вложенные функции.
12. Списки: создание, индексация, срезы, основные методы (append(), remove(), sort()).
13. Кортежи: особенности, отличия от списков, методы кортежей.
14. Цикл for: синтаксис, итерация по последовательностям, функция range().
15. Цикл while: синтаксис, управление циклом (break, continue).
16. Функции enumerate() и zip(): назначение, примеры использования.
17. Генераторы списков: синтаксис, примеры.
18. Лямбда-функции: синтаксис, применение с map(), filter(), sorted().
19. Словари: создание, доступ к элементам, методы keys(), values(), items().
20. Обход словарей в цикле. Добавление и удаление элементов.
21. Множества: создание, основные операции (объединение, пересечение, разность).
22. Модули в Python: импорт, стандартная библиотека, установка сторонних модулей.
23. Обработка исключений: конструкция try-except-else-finally. Оператор raise.

*Зачетное задание включает два вопроса – один теоретический вопрос и одно практико-ориентированное задание из числа приведенных ниже лабораторных заданий.*

#### **Критерии оценки:**

- оценка «зачтено» (50-100 баллов) выставляется студенту, если он в ходе ответа показал наличие твердых знаний по вопросу, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

- оценка «не зачтено» (менее 50 баллов) выставляется студенту, если ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

## Вопросы к экзамену

1. Понятие класса и объекта в объектно-ориентированном программировании.
2. Конструктор `__init__`: назначение, параметры, создание атрибутов объекта.
3. Специальные методы `__str__` и `__repr__`: отличия, назначение, примеры использования.
4. Инкапсуляция в Python: приватные атрибуты, соглашения об именовании.
5. Свойства (property): геттеры, сеттеры, deleter. Декоратор `@property`.
6. Динамическое управление атрибутами: функции `getattr()`, `setattr()`, `hasattr()`.
7. Наследование классов: создание подклассов, переопределение методов.
8. Функция `super()`: назначение, использование для вызова методов родительского класса.
9. Полиморфизм: понятие, примеры реализации в Python.
10. Множественное наследование: особенности, порядок разрешения методов (MRO).
11. Искажение имён: принцип работы, примеры использования.
12. Абстрактные классы и методы: модуль `ABC`, декоратор `@abstractmethod`.
13. Понятие алгоритма и его свойства. Нотация «O большое».
14. Линейный и бинарный поиск: принципы работы, сравнение эффективности.
15. Метод бисекции: применение для поиска корней уравнений и приближённых вычислений.
16. Рекурсивные алгоритмы: понятие, примеры (факториал, числа Фибоначчи).
17. Динамическое программирование: принципы, мемоизация и табуляция.
18. Алгоритмы сортировки: сравнительный анализ (сортировка выбором, быстрая сортировка).
19. Стек и очередь: принципы организации, применение в алгоритмах.
20. Связные списки: односвязные и двусвязные, основные операции.
21. Хеш-таблицы: принцип работы, хеш-функции, разрешение коллизий.
22. Графы: способы представления (матрица смежности, список смежности).
23. Обход графов: алгоритмы DFS (в глубину) и BFS (в ширину), сравнение.
24. Задача N-Ферзей: постановка, решение с использованием поиска с возвратом.
25. Деревья: двоичные деревья поиска, префиксные деревья. Динамическое программирование на примере чисел Фибоначчи.

Экзаменационное задание включает три вопроса – два теоретических вопроса и одно практико-ориентированное задание из числа приведенных ниже лабораторных заданий.

### Критерии оценивания:

– 84-100 баллов (оценка «отлично») – изложенный материал фактически верен, наличие глубоких исчерпывающих знаний в объеме пройденной программы дисциплины в соответствии с поставленными программой курса целями и задачами обучения; правильные, уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, усвоение основной и знакомство с дополнительной литературой;

– 67-83 баллов (оценка «хорошо») – наличие твердых и достаточно полных знаний в объеме пройденной программы дисциплины в соответствии с целями обучения, правильные действия по применению знаний на практике, четкое изложение материала, допускаются

отдельные логические и стилистические погрешности, обучающийся усвоил основную литературу, рекомендованную в рабочей программе дисциплины;

– 50-66 баллов (оценка «удовлетворительно») – наличие твердых знаний в объеме пройденного курса в соответствии с целями обучения, изложение ответов с отдельными ошибками, уверенно исправленными после дополнительных вопросов; правильные в целом действия по применению знаний на практике;

– 0-49 баллов (оценка «неудовлетворительно») – ответы не связаны с вопросами, наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

## **Лабораторные задания**

### **3 курс**

Лабораторное задание 1

Знакомство со средой разработки. Переменные и типы данных

Лабораторное задание 2

Работа со строками и числами

Лабораторное задание 3

Создание программы «Планировщик погоды для путешествий» с использованием условных операторов

Лабораторное задание 4

Разработка функции «Калькулятор скидок»

Лабораторное задание 5

Создание RPG-персонажа с проверкой входных данных

Лабораторное задание 6

Генератор числовых паттернов с использованием циклов

Лабораторное задание 7

Работа со списками и кортежами

Лабораторное задание 8

Списковые включения и лямбда-функции

Лабораторное задание 9

Работа со словарями и множествами

Лабораторное задание 10

Разработка менеджера конфигурации пользователя

Лабораторное задание 11  
Обработка исключений. Отладка ISBN-валидатора

Лабораторное задание 12  
Создание класса «Планета»

Лабораторное задание 13  
Разработка приложения для учёта бюджета

*Лабораторные работы 7-13 выполняются самостоятельно в домашних условиях и представляются на проверку в виде отчета к установленному сроку.*

**Критерии оценивания (для каждого задания):**

5-6 баллов – задача решена верно; студент формулирует собственные, самостоятельные, обоснованные, аргументированные суждения, представляет полные и развернутые ответы на дополнительные вопросы;

3-4 баллов – при решении задачи были допущены неточности, не влияющие на результат; студент формулирует собственные, самостоятельные, обоснованные, аргументированные суждения, допуская незначительные ошибки на дополнительные вопросы;

0-2 балла – при решении задачи были допущены существенные ошибки; студент допускает грубые ошибки, испытывает затруднения в формулировке собственных суждений, неспособен ответить на дополнительные вопросы.

***Максимальное количество баллов за все лабораторные задания 3-го курса – 78 (13 задания по 6 баллов).***

***4 курс***

Лабораторное задание 14  
Создание трекера характеристик игрового персонажа

Лабораторное задание 15  
Разработка иерархии классов «Фигуры» (прямоугольник, квадрат) с наследованием

Лабораторное задание 16  
Калькулятор площади многоугольников

Лабораторное задание 17  
Вычисление квадратного корня методом бисекции

Лабораторное задание 18  
Реализация алгоритма быстрой сортировки

Лабораторное задание 19  
Реализация алгоритма Луна для валидации кредитных карт

Лабораторное задание 20  
Преобразование списка смежности в матрицу смежности

Лабораторное задание 21  
Реализация обхода графа в глубину (DFS)

Лабораторное задание 22  
Решение задачи N-Ферзей с использованием DFS

Лабораторное задание 23  
Вычисление чисел Фибоначчи методом динамического программирования

*Лабораторные задания 17-23 выполняются самостоятельно в домашних условиях и представляются на проверку в виде отчета к установленному сроку.*

**Критерии оценивания (для каждого задания):**

8-10 баллов – задача решена верно; студент формулирует собственные, самостоятельные, обоснованные, аргументированные суждения, представляет полные и развернутые ответы на дополнительные вопросы;

5-7 баллов – при решении задачи были допущены неточности, не влияющие на результат; студент формулирует собственные, самостоятельные, обоснованные, аргументированные суждения, допуская незначительные ошибки на дополнительные вопросы;

3-4 балла – при решении задачи были допущены ошибки; студент испытывает затруднения в формулировке собственных обоснованных и аргументированных суждений, допуская ошибки на дополнительные вопросы;

1-2 балла – при решении задачи были допущены существенные ошибки; студент допускает грубые ошибки, испытывает затруднения в формулировке собственных суждений, неспособен ответить на дополнительные вопросы.

*Максимальное количество баллов за все лабораторные задания 4-го курса – 100 (10 задания по 10 баллов).*

## **Доклад**

**3 курс**

### **Темы докладов**

1. История развития языка Python.
2. Сравнительный анализ Python с другими языками программирования (Java, C++, JavaScript).
3. Функциональное программирование в Python: возможности и ограничения.
4. Декораторы в Python: принципы работы, создание и применение.
5. Генераторы и корутины: итераторы, yield, асинхронное программирование.
6. Менеджеры контекста: конструкция with, создание собственных менеджеров.
7. Метапрограммирование в Python: рефлексия, дескрипторы, метаклассы.
8. Многопоточность и многопроцессорность в Python.

9. Паттерны проектирования в Python: реализация и примеры использования.
10. Принципы SOLID в объектно-ориентированном программировании на Python.
11. Тестирование программного обеспечения: unittest, pytest, mock-объекты.
12. Документирование кода: docstrings, Sphinx, генерация документации.
13. Стандартная библиотека Python: обзор наиболее полезных модулей.
14. Визуализация данных в Python: библиотеки matplotlib, seaborn, plotly.
15. Веб-разработка на Python: обзор фреймворков (Django, Flask, FastAPI).
16. Работа с базами данных в Python: SQLAlchemy, ORM, драйверы БД.
17. Научные вычисления в Python: библиотеки NumPy, SciPy, Pandas.
18. Машинное обучение на Python.
19. Графовые алгоритмы: реализация и применение в Python.
20. Оптимизация кода на Python: профилирование, Cython, PyPy.

#### **Критерии оценки:**

- 17-22 баллов, если студент перечисляет все существенные характеристики обозначенного в вопросе предмета и возможные варианты дальнейшего развития решения проблемы, если это возможно;
- 12-16 баллов, если студент раскрыл только часть основных положений вопроса, продемонстрировал неточность в представлениях о предмете вопроса;
- 6-11 баллов, если студент обозначил общую траекторию ответа, но не смог конкретизировать основные компоненты;
- 0-5 балла, если студент не продемонстрировал знаний основных понятий, представлений об изучаемом предмете.

***Максимальное количество баллов за доклад во 3-м курсе – 22.***

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

Процедуры оценивания включают в себя текущий контроль и промежуточную аттестацию.

**Текущий контроль** успеваемости проводится с использованием оценочных средств, представленных в п. 2 данного приложения. Результаты текущего контроля доводятся до сведения студентов до промежуточной аттестации.

**Промежуточная аттестация** проводится в форме зачета / экзамена.

Зачет проводится по расписанию промежуточной аттестации. Количество вопросов в задании – 2 (один теоретический вопрос и одно практико-ориентированное задание). Объявление результатов производится в день зачета. Результаты аттестации заносятся в ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику, должны ликвидировать задолженность в установленном порядке.

Экзамен проводится по расписанию промежуточной аттестации. Количество вопросов в задании – 3 (два теоретических вопроса и одно практико-ориентированное задание). Объявление результатов производится в день экзамена. Результаты аттестации заносятся в ведомость и зачетную книжку студента. Студенты, не прошедшие промежуточную аттестацию по графику, должны ликвидировать задолженность в установленном порядке.

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Методические указания по освоению дисциплины адресованы студентам всех форм обучения.

Учебным планом предусмотрены следующие виды занятий:

- лекции;
- лабораторные занятия.

Важным условием успешного освоения дисциплины «Языки и методы программирования» является создание системы правильной организации труда, позволяющей распределить учебную нагрузку равномерно в соответствии с графиком образовательного процесса. Большую помощь в этом может оказать составление плана работы на семестр, месяц, неделю, день. Его наличие позволит подчинить свободное время целям учебы, трудиться более успешно и эффективно. С вечера всегда надо распределять работу на завтрашний день. В конце каждого дня целесообразно подвести итог работы: тщательно проверить, все ли выполнено по намеченному плану, не было ли каких-либо отступлений, а если были, по какой причине они произошли. Нужно осуществлять самоконтроль, который является необходимым условием успешной учебы. Если что-то осталось невыполненным, необходимо изыскать время для завершения этой части работы, не уменьшая объема недельного плана. Все задания к лабораторным работам, а также задания, вынесенные на самостоятельную работу, рекомендуется выполнять непосредственно после соответствующей темы лекционного курса, что способствует лучшему усвоению материала, позволяет своевременно выявить и устранить «пробелы» в знаниях, систематизировать ранее пройденный материал, на его основе приступить к овладению новыми знаниями и навыками.

Знакомство с дисциплиной происходит уже на первой лекции, где от студента требуется не просто внимание, но и самостоятельное оформление конспекта. При работе с конспектом лекций необходимо учитывать тот фактор, что одни лекции дают ответы на конкретные вопросы темы, другие – лишь выявляют взаимосвязи между явлениями, помогая студенту понять глубинные процессы развития изучаемого предмета как в истории, так и в настоящее время.

Конспектирование лекций – сложный вид вузовской аудиторной работы, предполагающий интенсивную умственную деятельность студента. Конспект является полезным тогда, когда записано самое существенное и сделано это самим обучающимся. Не надо стремиться записать дословно всю лекцию. Такое «конспектирование» приносит больше вреда, чем пользы. Целесообразно вначале понять основную мысль, излагаемую лектором, а затем записать ее. Желательно запись осуществлять на одной странице листа или оставляя поля, на которых позднее, при самостоятельной работе с конспектом, можно сделать дополнительные записи, отметить непонятные места.

Конспект лекции лучше подразделять на пункты, соблюдая красную строку. Этому в большой степени будут способствовать вопросы плана лекции, предложенные преподавателям. Следует обращать внимание на акценты, выводы, которые делает лектор, отмечая наиболее важные моменты в лекционном материале замечаниями «важно», «хорошо запомнить» и т.п. Можно делать это и с помощью разноцветных маркеров или ручек, подчеркивая термины и определения.

Целесообразно разработать собственную систему сокращений, аббревиатур и символов. Однако при дальнейшей работе с конспектом символы лучше заменить обычными словами для быстрого зрительного восприятия текста.

Работая над конспектом лекций, всегда необходимо использовать не только учебник, но и ту литературу, которую дополнительно рекомендовал лектор. Именно такая серьезная, кропотливая работа с лекционным материалом позволит глубоко овладеть теоретическим материалом.

В процессе подготовки к лабораторным занятиям, студентам необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы. При всей полноте конспектирования лекции в ней невозможно изложить весь материал из-за лимита аудиторных часов. Поэтому самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала, формирует у студентов свое отношение к конкретной проблеме.

Изучение дисциплины проходит с акцентом на лабораторные работы. По темам лабораторных работ разработаны учебно-методические материалы, в которых изложены подробные методические рекомендации по изучению каждой темы и выполнению заданий. Наличие таких учебно-методических и дидактических материалов позволяет каждому студенту работать в своем индивидуальном темпе, а также дополнительно прорабатывать изучаемый материал во время самостоятельных занятий.

Для успешного овладения предлагаемым курсом студент должен обладать определённой информационной культурой: навыками работы с литературой, умением определять и находить информационные ресурсы, соответствующие целям и задачам образовательного процесса, получать к ним доступ и использовать в целях повышения эффективности своей профессиональной деятельности. При изучении данного курса необходимо максимально использовать компьютер, изучать дополнительные информационные ресурсы.

#### **Подготовка к промежуточной аттестации**

При подготовке к промежуточной аттестации целесообразно:

- внимательно изучить перечень вопросов и определить, в каких источниках находятся сведения, необходимые для ответа на них;
- внимательно прочитать рекомендованную литературу;
- составить краткие конспекты ответов (планы ответов).