

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Воронежский государственный технический университет»

УТВЕРЖДАЮ

Декан факультета Бурковский А.В.  
«31» августа 2021 г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

«Прикладное программирование»

Направление подготовки 27.03.04 Управление в технических системах

Профиль Управление и информатика в технических системах

Квалификация выпускника бакалавр

Нормативный период обучения 4 года

Форма обучения очная

Год начала подготовки 2021

Автор программы

А.А. Голиков

Заведующий кафедрой  
Электропривода,  
автоматики и управления в  
технических системах

В.Л. Бурковский

Руководитель ОПОП

Ю.В. Мурзинов

Воронеж 2021

## 1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

### 1.1. Цели дисциплины

обучение студентов практическим навыкам разработки алгоритмов решения инженерных и научных задач и их программной реализации с учетом современных тенденций развития вычислительной техники, информационных технологий в своей профессиональной деятельности, а также в изучении настройки системного, прикладного и инструментального программного обеспечения систем автоматизации и управления.

### 1.2. Задачи освоения дисциплины

к теоретическим задачам относятся: изучение базовых алгоритмов решения вычислительных задач; освоение синтаксиса и базовых функций языка Python; изучения алгоритмов обработки сложных типов данных;  
- прикладные задачи состоят в приобретении навыков программирования с использованием современных языков и программных средств.

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина относится к дисциплинам обязательной части блока Б1.

## 3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Прикладное программирование» направлен на формирование следующих компетенций:

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

ОПК-6 - Способен разрабатывать и использовать алгоритмы и программы, современные информационные технологии, методы и средства контроля, диагностики и управления, пригодные для практического применения в сфере своей профессиональной деятельности

Компетенция	Результаты обучения, характеризующие сформированность компетенции
УК-1	Знает современные проблемы и тенденции развития информационных систем, основы системного подхода к описанию технических и организационных систем, методологии графического моделирования информационных процессов
	Умеет строить функциональные модели информационных процессов
	Владеет программным инструментом для создания графических моделей информационных процессов
ОПК-6	Знать об основах алгоритмизации инженерных задач, основах программирования на алгоритмическом языке высокого уровня.

	Уметь использовать стандартные пакеты (библиотеки) языка для решения практических задач.
	Владеть методами и средствами разработки и оформления технической документации.

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Прикладное программирование» составляет 3 з.е.

Распределение трудоемкости дисциплины по видам занятий  
**очная форма обучения**

Виды учебной работы	Всего часов	Семестры
		7
<b>Аудиторные занятия (всего)</b>	54	54
В том числе:		
Лекции	18	18
Лабораторные работы (ЛР)	36	36
<b>Самостоятельная работа</b>	54	54
Виды промежуточной аттестации - зачет	+	+
Общая трудоемкость: академические часы	108	108
зач.ед.	3	3

#### 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий  
**очная форма обучения**

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего, час
1	История языков программирования. Язык Python.	История языков программирования. Компиляция и интерпретация. Знакомство с Python и средами программирования.	2	6	8	18
2	Простейшие программы. Реализация вычислений и ветвлений.	Типы данных в программировании. Определение переменной. Локальные и глобальные переменные. Ввод данных с клавиатуры. Логические выражения. Условный оператор. Инструкция if. Множественное ветвление	2	6	8	18
3	Реализация циклических, вспомогательных алгоритмов	Цикл While и For. Вложенные циклы. Сумма и произведение цифр числа Числа Фибоначчи (вычисление с помощью цикла while и рекурсии) Алгоритм Евклида (нахождение наибольшего общего делителя). Вычисление факториала на языке программирования Python Перевод чисел из десятичной системы счисления в двоичную Решето Эратосфена - алгоритм определения простых чисел Тестирование простоты числа методом перебора делителей.	4	6	8	18
4	Словари. Массивы. Обработка массивов	Введение в словари. Массивы. Основные задачи обработки массивов: поиск, сортировка, реверс. Отбор элементов массива по условию. Сортировка выбором (поиск минимума и перестановка). Сортировка методом пузырька. Двоичный (бинарный) поиск элемента в массиве Списки — изменяемые последовательности. Отбор элементов массива по условию. Замена элементов в списке	4	6	10	18

5	Символьные строки. Обработка символьных строк.	Символьные строки. Обработка символьных строк. Строки как последовательности символов. Функции для работы с символьными строками. Преобразования «строка-число». Строки в процедурах и функциях. Сравнение и сортировка строк.	4	6	10	18
6	Программирование микроконтроллеров в MicroPython	Установка и настройка IDE. Переменные и математические операции в MicroPython. Кнопка в MicroPython. Условия, циклы и функции.	2	6	10	18
<b>Итого</b>			<b>18</b>	<b>36</b>	<b>54</b>	<b>108</b>

## 5.2 Перечень лабораторных работ

1. Установка Python. Доступ к документации. Ввод и вывод данных. Первая программа на языке Python.
2. Типы данных. Базовые функции для работы с различными типами данных.
3. Условные операторы и циклы. Обработка исключений.
4. Встроенные и пользовательские функции. Итераторы и функции-генераторы. Декораторы функций.
5. Модули. Основы программирования модулей. Модули стандартной библиотеки.
6. Работа с файлами. Модули для работы с файлами.
7. Переменные и математические операции в MicroPython. Кнопка в MicroPython.

## 6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины не предусматривает выполнение курсового проекта (работы) или контрольной работы.

## 7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

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

#### 7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
УК-1	Знать современные проблемы и тенденции развития информационных систем, основы системного	Тест	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

	подхода к описанию технических и организационных систем, методологии графического моделирования информационных процессов			
	Уметь строить функциональные модели информационных процессов	Решение стандартных практических задач	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть программным инструментом для создания графических моделей информационных процессов	Решение прикладных задач в конкретной предметной области	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-6	знать об основах алгоритмизации инженерных задач, основах программирования на алгоритмическом языке высокого уровня.	Тест	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь использовать стандартные пакеты (библиотеки) языка для решения практических задач.	Решение стандартных практических задач	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть методами и средствами разработки и оформления технической документации.	Решение прикладных задач в конкретной предметной области	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

### 7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 7 семестре для очной формы обучения по двухбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
УК-1	знать современные проблемы и тенденции развития информационных систем, основы	Тест	Выполнение теста на 70-100%	Выполнение менее 70%

	системного подхода к описанию технических и организационных систем, методологии графического моделирования информационных процессов			
	уметь строить функциональные модели информационных процессов	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть программным инструментом для создания графических моделей информационных процессов	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-6	знать об основах алгоритмизации инженерных задач, основах программирования на алгоритмическом языке высокого уровня	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь использовать стандартные пакеты (библиотеки) языка для решения практических задач	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть методами и средствами разработки и оформления технической документации	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

## **7.2 Примерный перечень оценочных средств ( типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)**

### **7.2.1 Примерный перечень заданий для подготовки к тестированию**

1. Какое зарезервированное слово применяется для определения функции в Python?

- a) procedure
- b) func
- c) def
- d) define

2. Что такое рекурсия?

- a) описание какого-либо объекта внутри этого самого объекта
- b) вызов функции в обратном порядке ее алгоритма
- c) возвращение результатов функции
- d) подбор параметров функции

3. Каково основное назначение рекурсивной функции?

- a) возвращение необходимого результата
- b) замена цикла
- c) работа со списками
- d) итерационный перебор через циклы

4. Чем являются функции в Python?

- a) объектами
- b) классами
- c) методами
- d) экземплярами

5. За что отвечает следующий код:

```
(lambda x, y: x + y)
```

- a) за создание анонимной функции
- b) за создание рекурсивной функции
- c) за создание именной функции
- d) за создание lambda-уравнения

6. Какая конструкция служит для создания анонимной функции?

- a) omega
- b) anonymous
- c) lambda
- d) lambda func

7. Какие вычисления происходят при вызове рекурсивной функции?

- a) циклические
- b) односторонние
- c) ленивые
- d) обратные

8. Какая функция создается с помощью следующего кода:

```
def add(x, y):
```

```
    return x + y
```

- a) анонимная функция
- b) это не является функцией

- c) рекурсивная функция
- d) именная функция

9. Какие из видов функций являются уникальными?

- a) итераторы
- b) анонимные
- c) именные
- d) генераторы

10. За создание какой функции отвечает конструкция def?

- a) рекурсивной функции
- b) именной функции
- c) функции-итератора
- d) анонимной функции

### 7.2.2 Примерный перечень заданий для решения стандартных задач

1. Какого типа значение получится в результате вычисления следующего выражения:

(" ")

Ответ:

- 1) str (строка)
- 2) tuple (кортеж)
- 3) это синтаксическая ошибка
- 4) unicode (Unicode-строка)

2. Какого типа значение получится в результате вычисления следующего выражения:

(' , )

Ответ:

- 1) str (строка)
- 2) tuple (кортеж)
- 3) это синтаксическая ошибка
- 4) unicode (Unicode-строка)

3. Какого типа значение получится в результате вычисления следующего выражения:

( , " ")

Ответ:

- 1) str (строка)
- 2) tuple (кортеж)
- 3) это синтаксическая ошибка
- 4) unicode (Unicode-строка)

4. Какого типа значение получится в результате вычисления следующего выражения:

```
[ " " ]
```

Ответ:

- 1) str (строка)
- 2) list (список)
- 3) это синтаксическая ошибка
- 4) array (массив)

5. Какого типа значение получится в результате вычисления следующего выражения:

```
(r'\u0432')
```

Ответ:

- 1) str (строка)
- 2) unicode (Unicode-строка)
- 3) tuple (кортеж)
- 4) это синтаксическая ошибка

6. Что выведет следующая программа

```
S = 0
for i in range(1, 10):
    if i % 2 == 0:
        S = S + i
print S
```

Ответ:

- 1) 0
- 2) 10
- 3) 20
- 4) 30

7. Что выведет следующая программа:

```
S = 0
for i in range(1, 10):
    if i % 2 == 0:
        S = S + i
print S
```

Ответ:

- 1) 10
- 2) 12
- 3) 14

4) 20

8. Что выведет следующая программа:

```
S = 0
for i in range(1, 10, 2):
    if i % 2 == 0:
        S = S + i
print S
```

Ответ:

- 1) 0
- 2) 1
- 3) 10
- 4) 20

9. Что выведет следующая программа:

```
S = 0
for i in range(10, 2, -1):
    if i % 2 == 0:
        S = S + i
print S
```

Ответ:

- 1) 0
- 2) 20
- 3) 28
- 4) 30

10. Что выведет следующая программа:

```
S = 0
for i in range(1, 10):
    for j in range(1, 3):
        if i == j:
            S = S + i + j
            break
print S
```

Ответ:

- 1) 0
- 2) 2
- 3) 6
- 4) 12

### 7.2.3 Примерный перечень заданий для решения прикладных задач

1. Написать программу определения минимального из трех чисел.
2. Написать программу определения размера стипендии. Пользователь вводит 3 натуральных числа: общее количество оценок, количество пятерок, количество четверок. Если все пятерки – стипендия 6000 руб., если одна четверка – 4500 руб., если две четверки – 3750 руб., если нет троек – 3000 руб., иначе – нет стипендии.
3. Написать программу определения четности/нечетности числа.
4. Создать список, состоящий из степеней двойки от 0й до 10й. 1 2 4 8 16 ... 1024. Двумя способами - с помощью цикла и с помощью генератора.
5. Создать список, состоящий из квадратов натуральных чисел от 1 до 10. 1 4 9 ... 10000. Двумя способами - с помощью цикла и с помощью генератора.
6. Подсчитать количество уникальных чисел в последовательности, вводимой пользователем.
7. Вывести слова, входящие в состав предложения, без повторений.
8. Написать процедуру, здоровающуюся с пользователем. Имя пользователя вводится как параметр процедуры.
9. Написать функцию определения корней квадратного уравнения. В качестве параметров подаются коэффициенты  $a$ ,  $b$ ,  $c$  ( $ax^2 + bx + c = 0$ ).
10. Написать программу, разбивающую строку на отдельные слова. Вывести слова в алфавитном порядке.
11. Написать программу, осуществляющую замену подстроки в строке. Например, дана строка: “Привет! Меня зовут Ольга”, произвести замену “Ольга -> Анна”.
12. Написать программу, вычисляющую пересечение и объединение двух множеств. Элементы множества вводятся пользователем с клавиатуры.
13. Создать словарь, состоящий из пар «страна – столица» (РФ-Москва, США-Вашингтон, Белоруссия-Минск, Украина-Киев, Китай-Пекин).
14. Написать программу, обрабатывающую исключение – деление на 0.
15. Написать программу, реализующую класс с конструктором. Создать несколько объектов данного класса.

#### **7.2.4 Примерный перечень вопросов для подготовки к зачету**

1. Определение алгоритма. Виды алгоритмов. Способы описания алгоритмов.
2. Блок-схемы.
3. Организация ввода и вывода на экран в Python.
4. Математические операции в Python.
5. Ветвления в Python. Множественное ветвление в Python.
6. Циклы в Python. Операторы управления циклами.
7. Тип данных список в Python. Методы, функции и операции для работы со списками.
8. Тип данных кортеж в Python. Методы, функции и операции для работы с кортежами.
9. Создание подпрограмм в Python. Способы передачи параметров. Возврат значений.
10. Тип данных строка в Python. Методы, функции и операции для работы со строками.

11. Типы данных.
12. Переменные.
13. Числовые типы данных.
14. Операции над числовыми типами данных.
15. Вывод данных.
16. Ввод данных.
17. Форматированный ввод/вывод.

### 7.2.5 Примерный перечень заданий для подготовки к экзамену

Не предусмотрено учебным планом

### 7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Зачет проводится по тест-билетам, каждый из которых содержит 10 вопросов и задачу. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом, задача оценивается в 10 баллов (5 баллов верное решение и 5 баллов за верный ответ). Максимальное количество набранных баллов – 20.

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 6 баллов.

2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 6 до 10 баллов

3. Оценка «Хорошо» ставится в случае, если студент набрал от 11 до 15 баллов.

4. Оценка «Отлично» ставится, если студент набрал от 16 до 20 баллов.)

### 7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	История языков программирования. Компиляция и интерпретация. Знакомство с Python и средами программирования.	УК-1, ОПК-6	Тест, защита лабораторных работ
2	Типы данных в программировании. Определение переменной. Локальные и глобальные переменные. Ввод данных с клавиатуры. Логические выражения. Условный оператор. Инstrukция if. Множественное ветвление.	УК-1, ОПК-6	Тест, защита лабораторных работ
3	Цикл While и For. Вложенные циклы. Сумма и произведение цифр числа Числа Фибоначчи (вычисление с помощью цикла while и рекурсии) Алгоритм Евклида (нахождение	УК-1, ОПК-6	Тест, защита лабораторных работ

	<p>наибольшего общего делителя)  Вычисление факториала на языке программирования Python  Перевод чисел из десятичной системы счисления в двоичную  Решето Эратосфена - алгоритм определения простых чисел  Тестирование простоты числа методом перебора делителей</p>		
4	<p>Введение в словари.  Массивы. Основные задачи обработки массивов: поиск, сортировка, реверс. Отбор элементов массива по условию.  Сортировка выбором (поиск минимума и перестановка).  Сортировка методом пузырька.  Двоичный (бинарный) поиск элемента в массиве  Списки — изменяемые последовательности.  Отбор элементов массива по условию. Замена элементов в списке</p>	УК-1, ОПК-6	Тест, защита лабораторных работ
5	<p>Символьные строки. Обработка символьных строк.  Строки как последовательности символов. Функции для работы с символьными строками.  Преобразования «строка-число».  Строки в процедурах и функциях.  Сравнение и сортировка строк.</p>	УК-1, ОПК-6	Тест, защита лабораторных работ
6	<p>Установка и настройка IDE.  Переменные и математические операции в MicroPython. Кнопка в MicroPython. Условия, циклы и функции.</p>	УК-1, ОПК-6	Тест, защита лабораторных работ

### 7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач

на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

## **8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)**

### **8.1 Перечень учебной литературы, необходимой для освоения дисциплины**

1. Северенс Ч. Введение в программирование на Python: Учебная литература для ВУЗов Национальный Открытый Университет «ИНТУИТ», 2016 [http://biblioclub.ru/index.php?page=book\\_red&id=429184](http://biblioclub.ru/index.php?page=book_red&id=429184)

2. Федоров Д.Ю. ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ PYTHON. Учебное пособие для прикладного бакалавриата: Гриф УМО ВО М.:Издательство Юрайт, 2018 <https://biblio-online.ru/book/1EE056CF-F11A-4C18-8D33-40B703D49AC5>

1. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули : учебное пособие / В. М. Шелудько. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2017. — 107 с. — ISBN 978-5-9275-2648-2. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/87530.html>

### **8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:**

1. LibreOffice;
2. Microsoft Office Word 2013/2007;
3. Microsoft Office Excel 2013/2007;
4. Microsoft Office Power Point 2013/2007;
5. Windows Professional 8.1 (7 и 8) Single Upgrade MVL A Each Academi;c
6. ABBYY FineReader 9.0.
7. FEMM 4.2;
8. SciLab
9. MATLAB Classroom
10. Simulink Classroom

Интернет-ресурсы:

Википедия - <http://ru.wikipedia.org/wiki/Python>

ИНТУИТ - <http://www.INTUIT.ru> (<http://www.intuit.ru/department/pl/python>)  
Сайт разработчиков на Питоне - <http://diveinto.python.ru/toc.html>  
Самоучитель Python - <http://pythonworld.ru/samouchitel-python>

Перечень информационных справочных систем:

1. Основы алгоритмизации и программирования на языке Python [Электронный ресурс]: <https://portal.edu.asu.ru/course/view.php?id=3237>
2. Издательство «Лань» - электронно-библиотечная система [Электронный ресурс]: <http://e.lanbook.com>
3. Издательство МЦНМО. Свободно распространяемые книги издательства Московского центра непрерывного математического образования [Электронный ресурс]: [www.mcsme.ru/free-books](http://www.mcsme.ru/free-books)
4. Электронная библиотечная система Алтайского государственного университета [Электронный ресурс]: <http://elibrary.asu.ru>
5. Электронная база данных ZBMATH: <https://zbmath.org/>

## **9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

1. Специализированная аудитория, оснащенная проекционной аппаратурой.
2. Специализированная учебная аудитория для проведения лабораторных занятий.

## **10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**

По дисциплине «Прикладное программирование» читаются лекции, проводятся лабораторные работы.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.

Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоения учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом три дня эффективнее всего использовать для повторения и систематизации материала.

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
----------	-----------------------------	----------------------------	--