

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

УТВЕРЖДАЮ

И.о. декана факультета информационных
технологий и компьютерной безопасности



А.В. Бредихин /
19.03.2024

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
«Программирование на Python»

Направление подготовки 09.03.02 Информационные системы и технологии

Профиль Технологии искусственного интеллекта

Квалификация выпускника бакалавр

Нормативный период обучения 4 года

Форма обучения очная

Год начала подготовки 2024

Автор программы

С.Ю. Белецкая

И.о. заведующего кафедрой
систем

автоматизированного
проектирования и
информационных систем

П.Ю. Гусев

Руководитель ОПОП

Д.В. Иванов

Воронеж 2024

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины

Целью дисциплины является формирование знаний об основных конструкциях языка Python, модулях и библиотеках, основных приёмах программирования, получение практических навыков разработки программного обеспечения на языке Python

1.2. Задачи освоения дисциплины

- формирование знаний о типах данных и управляющих конструкциях языка Python;
- формирование знаний об особенностях реализации методов объектно-ориентированного программирования в Python;
- формирование практических навыков разработки приложений с графическим интерфейсом и клиент-серверных приложений.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Программирование на Python» относится к дисциплинам части, формируемой участниками образовательных отношений блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Программирование на Python» направлен на формирование следующих компетенций:

ПК-1 – Способен анализировать предметную область, определять современные подходы и стандарты автоматизации в процессе проектирования и разработки информационных систем

ПК-2 - Способен разрабатывать структуры программного кода, пользовательский интерфейс и проводить анализ качества кода в процессе разработки информационных систем

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ПК-1	знать синтаксис стандартных конструкций языка Python
	уметь разработать проект информационной системы в соответствии с требованиями
	владеть навыками разработки приложений с графическим интерфейсом
ПК-2	знать стандартные модули и встроенные функции Python
	уметь использовать методы объектно-ориентированного программирования на практике

	владеть навыками разработки клиент-серверных приложений в Python
--	--

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Программирование на языке Python» составляет 6 з.е.

Распределение трудоемкости дисциплины по видам занятий
очная форма обучения

Виды учебной работы	Всего часов	Семестры
		5
Аудиторные занятия (всего)	72	72
В том числе:		
Лекции	18	18
Лабораторные работы (ЛР)	54	54
Самостоятельная работа	108	108
Курсовой проект	+	+
Часы на контроль	36	36
Виды промежуточной аттестации - экзамен	+	+
Общая трудоемкость: академические часы	216	216
зач.ед.	6	6

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	Синтаксис языка Python. Типы данных	Общая характеристика языка Python. Особенности Python. Области применения Python. Основы синтаксиса Python. Идентификаторы, ключевые слова. Переменные в Python. Операторы. Типы данных в Python. Преобразование типов. Числовой тип данных. Арифметические операторы и математические функции. Строки. Функции работы со строками. Списки. Функции работы со списками. Множества. Кортежи. Словари. Математические функции.	4	8	18	30
2	Основные конструкции Python	Условные операторы в Python. Конструкции if-else, if-elif-else, match-case. Организация циклов в Python. Цикл с предусловием (while). Цикл с итератором (for). Досрочное завершение цикла. Генераторы списков. Процедуры и функции в Python. Позиционные и именованные аргументы. Функции с произвольным числом аргументов. Вложенные функции. Области видимости	4	8	18	28

		<p>переменных. Локальные и глобальные переменные. Функции высшего порядка. Анонимные (lambda) функции. Понятие функционального программирования.</p> <p>Модули и пакеты. Импорт модулей. Написание собственных модулей. Стандартные модули Python. Установка и подключение внешних модулей и библиотек.</p> <p>Работа с массивами в Python. Работа с массивами с использованием списков. Основы работы с библиотекой NumPy. Основные функции и методы библиотеки NumPy для работы с многомерными массивами данных.</p> <p>Обработка исключений в Python. Конструкция try-except для обработки исключений.</p>				
3	Ввод/вывод данных	Ввод и вывод данных в Python. Форматированный вывод. Чтение и запись файлов. Функции работы с файлами и файловой системой.	2	8	18	28
4	Объектно-ориентированное программирование в Python	Основы объектно-ориентированного программирования. Принципы ООП. Классы. Синтаксис определения класса. Вызов методов экземпляров классов. Области видимости и пространства имен. Наследование.	4	10	18	22
5	Разработка приложений с графическим интерфейсом	Графические библиотеки в Python. Основы работы с библиотекой Tkinter. Создание виджетов. Изменение параметров виджетов. Обработка событий.	2	10	18	30
6	Клиент-серверное программирование в Python	Особенности реализации клиента и сервера на языке Python. Протокол HTTP. Сокеты.	2	10	18	32
Итого			18	54	108	180

5.2 Перечень лабораторных работ

1. Установка Python и сред разработки. Знакомство с интегрированными средами разработки приложений.
2. Основные конструкции Python.
3. Работа со сложными структурами данных.
4. Работа с файлами.
5. Модули и библиотеки
6. Объектно-ориентированное программирование в Python.
7. Разработка приложений с графическим интерфейсом.
8. Клиент-серверное программирование в Python.

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

В соответствии с учебным планом освоение дисциплины предусматривает выполнение курсового проекта в 5 семестре для очной формы обучения.

Примерная тематика курсового проекта: «Разработка приложения в

Python (по вариантам)».

Задачи, решаемые при выполнении курсового проекта:

- изучение синтаксиса языка Python;
- изучение встроенных функций для работы с различными структурами данных;
- изучение встроенных классов для реализации обмена данными между распределенными компонентами приложения.

Курсовой проект включает в себя графическую часть и расчетно-пояснительную записку.

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
ПК-1	знать синтаксис стандартных конструкций языка Python	Тест, защита лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь разработать проект информационной системы в соответствии с требованиями	Тест, защита лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками разработки приложений с графическим интерфейсом	Тест, защита лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ПК-2	знать стандартные модули и встроенные функции Python	Тест, защита лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь использовать методы объектно-ориентированного программирования на практике	Тест, защита лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеть навыками разработки клиент-серверных приложений в Python	Тест, защита лабораторных работ	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

«ОТЛИЧНО»;

«хорошо»;
 «удовлетворительно»;
 «неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
ПК-1	знать синтаксис стандартных конструкций языка Python	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80- 90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	уметь разработать проект информационной системы в соответствии с требованиями	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками разработки приложений с графическим интерфейсом	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ПК-2	знать стандартные модули и встроенные функции Python	Тест	Выполнение теста на 90- 100%	Выполнение теста на 80- 90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	уметь использовать методы объектно-ориентированного программирования на практике	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеть навыками разработки клиент-серверных приложений в Python	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	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. Какие из перечисленных записей возведут число x в степень y ?

- а) `pow(x, y)`;
- б) `sqr(x, y)`;
- в) `x**y`;
- г) `x%y`.

2. Какой результат выведет функция `round(4.56666, 3)`?

- а) 4;
- б) 4.5;
- в) **4.567**;

г) 4.56666.

3. Какая из перечисленных конструкций определяет функцию?

а) `function fun(n);`

б) `def fun (n);`

в) `procedure fun (n);`

г) `begin fun (n).`

4. Какой метод позволяет добавить элемент в конец списка?

а) `append();`

б) `insert();`

в) `add();`

г) `sum().`

5. Каким способом можно объявлять переменные в Python:

а) `a=5;`

б) `a=int (5);`

в) `int a=5;`

г) `a:=5.`

6. Какая функция отвечает за вывод на экран?

а) `output (a);`

б) `out (a);`

в) `print (a);`

г) `input (a).`

7. Какая функция отвечает за открытие файла?

а) `file();`

б) `open();`

в) `open_file();`

г) `append().`

8. Какая библиотека Python предназначена для работы с многомерными массивами данных?

а) Math

б) Numpy

в) Scipy

г) Tkinter

9. Какая из перечисленных библиотек Python может быть использована для разработки графического интерфейса?

а) Pandas

- б) Keras
- в) **Tkinter**
- г) Scikit Learn

10. Отметьте правильно оформленные комментарии в Python

- а) `*comments\`
- б) `*comments*`
- в) `(*comments*)`
- г) **`# comments`**

7.2.2 Примерный перечень заданий для решения стандартных задач

1. Какое из перечисленных объявлений функции является неверным:

- а) `def fun (n);`
- б) `def fun (n, a = 2);`
- в) **`def fun (n = 1, a);`**
- г) `def fun (n = 1, a = 2).`

2. Какой метод используется для замены подстроки?

- а) `string.renew()`
- б) `string.change()`
- в) `string.update()`
- г) **`string.replace()`**

3. Какая из перечисленных записей определяет кортеж?

- а) **`t = 12345, 54321, 'hello!';`**
- б) `t = 5.8;`
- в) `t = {'jack': 4098, 'sape': 4139};`
- г) `t = 5.`

4. Укажите, какому коду эквивалентен код `x += 10`

- а) `x + x = 10`
- б) `x == +10`
- в) **`x = x + 10`**
- г) `x = x - 10`

5. Какая функция определяет длину строки?

- а) **`len(s);`**

- б) length(s);
- в) count(s);
- г) long(s).

6. Какой метод позволяет определить позицию заданного элемента в списке?

- а) append();
- б) insert();
- в) index();**
- г) sum().

7. Какое ключевое слово используется в Python для создания конструктора?

- а) `__class__`
- б) `__main__`
- в) `__name__`
- г) `__init__`

8. Укажите, чем является строка 'Length' в словаре `myDictionary = {'Length':3, 'Width':5, 'Height':12}`

- а) логическим выражением
- б) функцией
- в) ключом**
- г) условием

9. Какой метод позволяет добавить элемент в произвольное место списка?

- а) append();
- б) insert();**
- в) add();
- г) sum().

10. Какой метод позволяет извлечь элемент из вершины стека?

- а) append();
- б) pop();**
- в) index();
- г) sum().

7.2.3 Примерный перечень заданий для решения прикладных задач

1 Какой результат будет получен при выполнении данного кода?

```
for i in range(5):
    if i % 2 == 0:
        continue
    print(i)
```

- а) ошибка, так как *i* не присвоено
- б) числа 1,3,5
- в) числа 0,2,4
- г) **числа 1,3**

2 Какие ошибки допущены в коде ниже?

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)
print(factorial(5))
```

- а) функция не может вызывать сама себя
- б) необходимо указать тип возвращаемого значения
- в) функция всегда будет возвращать 1
- г) **в коде нет никаких ошибок**

3 Какой результат будет получен при выполнении данного кода?

```
for j in 'Hi! I\'m mister Robert':
    if j == '!':
        print("Найдено")
        break
else:
    print ("Готово")
```

- а) ошибку в коде
- б) Найдено и Готово
- в) **Найдено**
- г) Готово

4 Что будет результатом данного кода?

```
x = 23
num = 0 if x > 10 else 11
print(num)
```

- а) 11
- б) **0**
- с) 23
- д) ошибка

5 Какая из строчек кода выведет структуру, отсортированную в порядке

увеличения возрастов животных?

```
animals = [  
    {'type': 'penguin', 'name': 'Stephanie', 'age': 8},  
    {'type': 'elephant', 'name': 'Devon', 'age': 3},  
    {'type': 'puma', 'name': 'Moe', 'age': 5},  
]
```

а) `sorted(animals, key='age')`

б) `sorted(animals, key=lambda animal: animal['age'])`

в) `sorted(animals)`

г) ни один вариант не является верным, два словаря нельзя сравнивать друг с другом.

6 Как можно более кратко представить следующую запись?

if X:

 A = Y

else:

 A = Z

а) `A = Y if Z else Y`

б) `A = Y if X else Z`

в) `A = X if Z else Y`

г) `A = X if Y else Z`

7 Что будет выведено в результате работы данного кода?

```
cost = 1000
```

```
if cost < 1000:
```

```
    print ( "Скидок нет" )
```

```
elif cost < 2000 and cost > 1500 :
```

```
    print ( "Скидка 22%" )
```

```
elif cost > 5000:
```

```
    print ( "Скидка 5%" )
```

```
else:
```

```
    print ( "Скидка 10%" )
```

а) Скидок нет

б) Скидка 22%

в) Скидка 5%

г) Скидка 10%

8 Строка кода `while (x=1):` выдает ошибку `SyntaxError: invalid syntax`. Укажите правильную причину ошибки

- а) в скобках не логическое выражение
- б) недопустимый пробел перед скобками
- в) пропущен пробел перед while
- г) x не определен заранее

9 Укажите правильный результат работы кода

```
myList = ['The','Python', 'is', 'Great']  
print('@'.join(myList))
```

- а) **The@Python@is@Great**
- б) ThePythonisGreat@
- в) The Python is Great @
- г) @ The Python is Great

10 Укажите корректный код, перемешивающий список в произвольном порядке

- а) **random.shuffle(myList)**
- б) myList.shuffle(random)
- в) shuffle.myList(random)
- г) shuffle(myList).random

7.2.4 Примерный перечень вопросов для подготовки к зачету

Не предусмотрено учебным планом

7.2.5 Примерный перечень заданий для подготовки к экзамену

1. Основы синтаксиса Python. Общие правила оформления кода.
2. Переменные и операторы Python.
3. Типы данных Python. Списки. Функции работы со списками.
4. Типы данных Python. Строки. Функции работы со строками.
5. Типы данных Python. Кортежи и множества.
6. Типы данных Python. Словари.
7. Условные операторы Python.
8. Организация циклов в Python.
9. Процедуры и функции в Python.
10. Модули и библиотеки. Написание собственных модулей.
11. Обработка исключений в Python/
12. Ввод и вывод в Python. Форматированный вывод.
13. Чтение и запись файлов. Функции работы с файлами и файловой системой.
14. Объектно-ориентированное программирование в Python. Принципы ООП.
15. Классы. Синтаксис определения класса. Вызов методов экземпляров классов.
16. Области видимости и пространства имен. Наследование.
17. Графические библиотеки Python. Основы работы с библиотекой Tkinter.

18. Создание виджетов. Изменение параметров виджетов.
19. Обработка событий в Tkinter.
20. Особенности реализации клиента и сервера на языке Python.
21. Протокол HTTP. Сокеты.

7.2.6. Методика выставления оценки при проведении промежуточной аттестации

Экзамен проводится по тест-билетам, каждый из которых содержит 10 вопросов. Каждый правильный ответ на вопрос в тесте оценивается 1 баллом. Максимальное количество набранных баллов – 10.

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 4 баллов.
2. Оценка «Удовлетворительно» ставится в случае, если студент набрал от 5 до 6 баллов
3. Оценка «Хорошо» ставится в случае, если студент набрал от 7 до 8 баллов.
4. Оценка «Отлично» ставится, если студент набрал от 9 до 10 баллов.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Синтаксис языка Python. Типы данных	ПК-1, ПК-2	защита лабораторных работ, вопросы к экзамену
2	Основные конструкции Python	ПК-1, ПК-2	защита лабораторных работ, вопросы к экзамену
3	Ввод/вывод данных	ПК-1, ПК-2	защита лабораторных работ, вопросы к экзамену
4	Объектно-ориентированное программирование в Python	ПК-1, ПК-2	защита лабораторных работ, вопросы к экзамену
5	Разработка приложений с графическим интерфейсом	ПК-1, ПК-2	защита лабораторных работ, вопросы к экзамену
6	Клиент-серверное программирование в Python	ПК-1, ПК-2	защита лабораторных работ, вопросы к экзамену

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.

Защита курсовой работы, курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

1 Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 349 с. — Текст : электронный. — Режим доступа: <https://urait.ru/bcode/544190>

2 Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2024. — 227 с. — Текст : электронный. — Режим доступа: <https://urait.ru/bcode/539651>

3 Маккини, У. Python и анализ данных / У. Маккини ; перевод А. А. Слинкин. — 2-е изд. — Москва : ДМК Пресс, 2020. — 540 с. — Текст : электронный. — Режим доступа: <https://www.iprbookshop.ru/125361>

4 Сузи, Р. А. Язык программирования Python : учебное пособие / Р. А. Сузи. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 350 с. — Текст : электронный. — Режим доступа: <https://www.iprbookshop.ru/97589.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:

Программное обеспечение

Python
Pycharm
Jupyter Notebook
Microsoft Visual Studio Community

Ресурсы информационно-телекоммуникационной сети «Интернет»
<http://www.edu.ru/>
Образовательный портал ВГТУ

Информационные справочные системы
<http://window.edu.ru>
<https://wiki.cchgeu.ru/>

Современные профессиональные базы данных
<https://habr.com/ru/>
<https://sources.ru/>
<https://proglib.io/>

9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Для проведения обучения по дисциплине используются:

Компьютерный класс

Учебная аудитория для проведения учебных занятий, включающая:

- рабочее место преподавателя (стол, стул);
- рабочие места обучающихся (столы, стулья)
- персональные компьютеры с установленным ПО, подключенные к сети Интернет (12 шт.);
- принтер;
- доска магнитно-маркерная поворотная
- оборудование для лекционных демонстраций и проекционная аппаратура.

Помещение для самостоятельной работы. Читальный зал с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду.

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

По дисциплине «Программирование на Python» читаются лекции, проводятся лабораторные работы, выполняется курсовой проект.

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.

Методика выполнения курсового проекта изложена в учебно-методическом пособии. Выполнять этапы курсового проекта должны своевременно и в установленные сроки.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, защитой курсового проекта.

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 усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: <ul style="list-style-type: none"> - 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 экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.

11 Лист регистрации изменений

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