Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Макаренко Елена Николаевна
Должность: Ректор
Дата подписания: 17.09.2026 18:21:01
Уникальный программный ключ:
c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

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

УТВЕРЖДАЮ
Начальник
учебно-методического управления
Т.К. Платонова
«25» мая 2026 г.

Рабочая программа дисциплины
Программирование на языке Python

Направление подготовки
09.03.03 Прикладная информатика

Направленность (профиль) программы бакалавриата
09.03.03.02 Разработка и управление программными проектами в цифровой экономике

Для набора 2026 года

Квалификация
Бакалавр

КАФЕДРА **Информационных систем и прикладной информатики**

Распределение часов дисциплины по семестрам / курсам

Семестр (<Курс>.<Семестр на курсе>)	**5 (3.1)**		Итого	
Недель	16 5/6			
Вид занятий	УП	РП	УП	РП
Лекции	16	16	16	16
Лабораторные	32	32	32	32
Итого ауд.	48	48	48	48
Контактная работа	48	48	48	48
Сам. работа	60	60	60	60
Часы на контроль	36	36	36	36
Итого	144	144	144	144

ОСНОВАНИЕ

Учебный план утвержден учёным советом Университета (протокол № 9 от 03.03.2026 г.).

Программу составил(и): д.э.н., доцент, Щербаков С.М.

Зав. кафедрой: д.э.н., профессор С.М. Щербаков

Методический совет: д.э.н., профессор Е.Н. Тищенко

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1.1	получение обучающимися теоретических знаний и основ построения программного обеспечения для решения профессиональных задач на языке программирования Python.

2. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

ПК-8. Способен проектировать, разрабатывать, модифицировать и адаптировать прикладное программное обеспечение

ПК-3. Способен разрабатывать решения для сбора, обработки, хранения, анализа и визуализации данных

В результате освоения дисциплины обучающийся должен:

Знать:

новые методы и алгоритмы машинного обучения (соотнесено с индикатором ПК-3.1)
системы искусственного интеллекта (соотнесено с индикатором ПК-8.1)

Уметь:

руководить разработкой архитектуры комплексных систем искусственного интеллекта со стороны заказчика (соотнесено с индикатором ПК-3.2)
руководить проектами по разработке, систем искусственного интеллекта на основе моделей глубоких нейронных сетей и нечетких моделей и методов со стороны заказчика (соотнесено с индикатором ПК-8.2)

Владеть:

навыками руководства созданием комплексных систем искусственного интеллекта с применением новых методов и алгоритмов машинного обучения (соотнесено с индикатором ПК-3.3)
навыками управления проектами по созданию, поддержке и использованию систем искусственного интеллекта на основе аналитики больших данных с применением новых методов и алгоритмов машинного обучения (соотнесено с индикатором ПК-8.3)

3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Раздел 1. Основы Python

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
1.1	Тема 1. Введение в Python. Основные конструкции и базовые типы. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лекционные занятия	5	2	ПК-8 ПК-3
1.2	Тема 1. Введение в Python. Основные конструкции и базовые типы. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лабораторные занятия	5	2	ПК-8 ПК-3
1.3	Тема 2. Структуры данных. Коллекции. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лекционные занятия	5	2	ПК-8 ПК-3
1.4	Тема 2. Структуры данных. Коллекции. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лабораторные занятия	5	4	ПК-8 ПК-3
1.5	Тема 3. Функциональное программирование. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лекционные занятия	5	2	ПК-8 ПК-3
1.6	Тема 3. Функциональное программирование. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лабораторные занятия	5	6	ПК-8 ПК-3
1.7	Тема 1. Введение в Python. Основные конструкции и базовые типы.	Самостоятельная работа	5	6	ПК-8 ПК-3
1.8	Тема 2. Структуры данных. Коллекции.	Самостоятельная работа	5	6	ПК-8 ПК-3
1.9	Тема 3. Функциональное программирование.	Самостоятельная работа	5	6	ПК-8 ПК-3

Раздел 2. Объектно-ориентированное программирование

№	Наименование темы, краткое содержание	Вид занятия / работы / форма ПА	Семестр / Курс	Количество часов	Компетенции
2.1	Тема 4. Классы и объекты. Наследование в Python. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лекционные занятия	5	2	ПК-8 ПК-3

2.2	Тема 4. Классы и объекты. Наследование в Python. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лабораторные занятия	5	6	ПК-8 ПК-3
2.3	Тема 5. Отладка и тестирование. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лекционные занятия	5	2	ПК-8 ПК-3
2.4	Тема 5. Отладка и тестирование. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лабораторные занятия	5	6	ПК-8 ПК-3
2.5	Тема 6. Построение нейросети на Python. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лекционные занятия	5	6	ПК-8 ПК-3
2.6	Тема 6. Построение нейросети на Python. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda.	Лабораторные занятия	5	8	ПК-8 ПК-3
2.7	Тема 4. Классы и объекты. Наследование в Python.	Самостоятельная работа	5	6	ПК-8 ПК-3
2.8	Тема 5. Отладка и тестирование.	Самостоятельная работа	5	12	ПК-8 ПК-3
2.9	Тема 6. Построение нейросети на Python.	Самостоятельная работа	5	24	ПК-8 ПК-3
2.10	Подготовка к промежуточной аттестации	Экзамен	5	36	ПК-8 ПК-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	Емельянов А. А.	Прикладная информатика: журнал	Москва: Синергия ПРЕСС, 2010	ЭБС «Университетская библиотека онлайн»
2	Сузи Р. А.	Язык программирования Python: учебное пособие	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ)| Бином. Лаборатория знаний, 2007	ЭБС «Университетская библиотека онлайн»
3		Вестник Ростовского государственного экономического университета (РИНХ)	, 1996	ЭБС «IPR SMART»
4	Букунов, С. В., Букунова, О. В.	Объектно ориентированное программирование на языке Python: учебное пособие	Санкт-Петербург: Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2020	ЭБС «IPR SMART»

5.2. Профессиональные базы данных и информационные справочные системы

Национальная электронная библиотека (НЭБ) - https://rusneb.ru/
ИСС "КонсультантПлюс"
ИСС "Гарант"

5.3. Перечень программного обеспечения

Операционная система РЕД ОС
Python
Pandas
Flask
Django
Anaconda

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.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде.

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

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

Приложение 1

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
ПК-3: Способен разрабатывать решения для сбора, обработки, хранения, анализа и визуализации данных			
Знать новые методы и алгоритмы машинного обучения	Знает синтаксис языка Python, основные структуры данных, базовые алгоритмические конструкции	Полнота и содержательность ответа Умение приводить примеры	Вопросы к экзамену (1-36), тесты (1-30), лабораторные задания (1-6)
Уметь руководить разработкой архитектуры комплексных систем искусственного интеллекта со стороны заказчика	Проводит сравнительный анализ и осуществляет выбор инструментальных средств для решения задач машинного обучения	Полнота и содержательность ответа Умение самостоятельно находить решение поставленных задач	Вопросы к экзамену (1-36), тесты (1-30), лабораторные задания (1-6)
Владеть навыками руководства созданием комплексных систем искусственного интеллекта с применением новых методов и алгоритмов машинного обучения	Разрабатывает алгоритмические конструкции, отлаживает программы	Полнота и содержательность ответа Умение приводить примеры Умение самостоятельно находить решение поставленных задач	Вопросы к экзамену (1-36), тесты (1-30), лабораторные задания (1-6)
ПК-8: Способен проектировать, разрабатывать, модифицировать и адаптировать прикладное программное обеспечение			
Знать системы искусственного интеллекта. Системы искусственного интеллекта	Знает функциональность современных инструментальных средств и систем программирования в области создания моделей искусственных нейронных сетей	Полнота и содержательность ответа Умение приводить примеры	Вопросы к экзамену (1-36), тесты (1-30), лабораторные задания (1-6)
Уметь руководить проектами по разработке, систем искусственного интеллекта на основе моделей глубоких нейронных сетей и нечетких моделей и методов со стороны заказчика	Руководит выполнением коллективной проектной деятельности для создания, поддержки и использования систем искусственного интеллекта на основе моделей глубоких нейронных сетей и нечетких моделей и методов	Полнота и содержательность ответа Умение самостоятельно находить решение поставленных задач	Вопросы к экзамену (1-36), тесты (1-30), лабораторные задания (1-6)
Владеть навыками управления проектами по созданию, поддержке и	Применяет современные инструментальные методы и средства обучения	Полнота и содержательность ответа Умение приводить	Вопросы к экзамену (1-36), тесты (1-

использованию систем искусственного интеллекта на основе аналитики больших данных с применением новых методов и алгоритмов машинного обучения	моделей искусственных нейронных сетей	примеры Умение самостоятельно находить решение поставленных задач	30), лабораторные задания (1-6)

1.2 Шкалы оценивания:

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

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 экзамену

1. Понятие алгоритма, программы.
2. Переменные.
3. Типы данных.
4. Строки.
5. Массивы.
6. Ввод и вывод.
7. Линейные алгоритмы.
8. Ветвления.
9. Условия.
10. Циклические алгоритмы.
11. Виды циклов.
12. Цикл по коллекции.
13. Цикл по счетчику.
14. Цикл с предусловием и постусловием.
15. Обработка массивов.
16. Выход из цикла.
17. Понятие подпрограммы.
18. Передача и возврат параметров.
19. Локальные и глобальные переменные.
20. Рекурсия.
21. Ввод и вывод данных.
22. Среда разработки.
23. Системы управления кодом, git.
24. Модули.
25. Списки.
26. Кортежи.
27. Словари.
28. Преобразование данных.
29. Срезы.
30. Циклическая обработка списка.
31. Генератор списка.
32. Понятие объектно-ориентированного программирования.

33. Класс. Объект.
34. Инкапсуляция, наследование и полиморфизм в Python.
35. Библиотека Pandas, анализ и обработка данных.
36. Визуализация в matplotlib и seaborn.

Задание включает в себя один теоретический вопрос из представленного перечня и одно практико-ориентированное задание из подраздела «Лабораторные задания».

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 и знакомство с 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 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.

Тесты

1. Совместимы ли Python 2 и Python 3?
− Совместимы полностью
− Несовместимы

2. На каком языке программирования написана основная реализация спецификации Python?
− Python
− Java
− C

3. Какое расширение обычно дают файлам с кодом на Python?
− .python
− .pyc
− .py

4. Что происходит при итерации по генератору?
− Итерация происходит по списку значений, который вернул генератор при вызове
− Каждую итерацию вызывается функция next, и генератор исполняется с начала
− Каждую итерацию вызывается функция next, и исполнение генератора возобновляется с момента после yield

5. Выберите верные утверждения про множества:
− проверка на вхождение элемента в множество происходит за линейное время
− множества изменяемые
− множества неизменяемые
− проверка на вхождение элемента в множество происходит за константное время

6. bool(0.000001) - True или False?
− True
− False

7. Какой записи эквивалентно применение декоратора
@login_required
def send_feedback(request)?
− login_required = send_feedback(login_required)
− send_feedback = login_required(send_feedback)
− def login_required(send_feedback)(request)
− def login_required(send_feedback)

8. Какая функция позволяет считать ввод пользователя из терминала?
− readline()
− input()
− read()
− enter()

9. Какой метод превратит байтовую строку в строку?
−.encode()
−.decode()

10. Можно ли использовать изменяемые объекты в качестве значений по умолчанию в функциях?
−Нет, случится синтаксическая ошибка
−Да, но это может привести к неочевидным ошибкам

11. Предположим, есть пакет foo, в котором находится модуль bar.py, внутри bar.py определена функция с именем run. Какая конструкция импорта является правильной?
− import run from foo.bar
− from foo.bar import run
− import foo.bar.run

12. Выберите верные утверждения про кортежи:
− Проверка на вхождение элемента в кортеж происходит за константное время
− Кортежи изменяемые
− Проверка на вхождение элемента в кортеж происходит за линейное время
− Кортежи неизменяемые
− Кортежи могут содержать элементы различных типов

13.Зачем нужен virtualenv (виртуальное окружение)?
− Возможность запускать несколько интерпретаторов Python одновременно
− Изоляция зависимостей
− Возможность увеличить скорость запуска скомпилированных в байткод Python-программ

14.Какая утилита позволяет ставить внешние Python пакеты в систему?
− pypi
− pip
− pep

15.Что содержат файлы с расширением .pyc?
− Код на Python, скомпилированный в байткод
− Код на Python, скомпилированный в машинный код

16. Для чего используются декораторы?

− Для эффективного использования памяти при итерации
− Для модификации поведения функций
− Чтобы иметь возможность импортировать функцию в другой модуль

17. Какие имена переменных правильные?
−1name
−!name
−Name
−_name

18. Как пишутся комментарии в Python?
− # это комментарий
− // это комментарий
− """ это комментарий """

19. Что получится в результате выполнения среза [2:3] для строки "привет"?
"и"
"ри"
"р"
"ив"

20. pi = 3.1415926
pi_fmt = f"{pi:#0.2f}"
Чему будет равен pi_fmt?
− Строке "3.14"
− Числу 3.14

21. Отметьте все исключения стандартной библиотеки Python
− ValueError
− RequestException
− KeyboardInterrupt
− IndexError

22. Какие из методов генерации исключения разрешены в Python?
− raise None
− raise ValueError
− raise "ValueError"
− raise ValueError("error")

23. Обращение к несуществующему атрибуту экземпляра
сгенерирует исключение KeyError
− сгенерирует исключение AttributeError
− вернет False
− вернет None

24. Отметьте верные утверждения про classmethod
− Метод не принимает дополнительных аргументов кроме указанных программистом
− Метод первым аргументом принимает класс
− К этому методу можно обращаться от экземпляра класса
− Метод первым аргументом принимает ссылку на экземпляр класса
− К этому методу можно обращаться от имени класса
In [1]:
```
class Tmp:
    @classmethod
    def foo(cls, val):
        print(val)
```

```
tmp = Tmp()
Tmp.foo(5)
tmp.foo(6)
```

25. Отметьте верные утверждения про staticmethod
− К этому методу можно обращаться от экземпляра класса
− Метод первым аргументом принимает ссылку на экземпляр класса
− Метод первым аргументом принимает класс
− Метод не принимает дополнительных аргументов кроме указанных программистом
− К этому методу можно обращаться от имени класса

```
In [2]:class Tmp:
    @staticmethod
    def foo(val):
        print(val)
tmp = Tmp()
Tmp.foo(5)
tmp.foo(6)
```

26. Для чего используются @property?
− Чтобы делать методы приватными
− Чтобы делать атрибуты приватными
− Чтобы создать вычисляемый атрибут

```
In [3]: class Celsius:
    def __init__(self, temperature = 0):
        self.temperature = temperature
    def to_fahrenheit(self):
        return (self.temperature * 1.8) + 32
    def get_temperature(self):
        print("Getting value")
        return self._temperature
    def set_temperature(self, value):
        if value < -273:
            raise ValueError("Temperature below -273 is not possible")
        print("Setting value")
        self._temperature = value
    temperature = property(get_temperature,set_temperature)
    celsius = Celsius()
celsius.temperature = -280;
```

27. Если имя метода в классе начинается с символа нижнего подчеркивания, например: _get_name
− если вызвать метод у экземпляра класса, то сгенерируется исключение AttributeError
− обращаться к методу объекта не рекомендуется, так как метод не считается публичным API класса
− символ нижнего подчеркивания в начале метода не добавляет никакого дополнительного значения

28. Можно ли использовать экземпляры классов в качестве ключей словаря (dict)?
− Да
− Нет

29. Можно ли передавать экземпляр класса как аргумент в функцию?
− Да
− Нет

30. Предположим есть базовый класс питомец - Pet и класс наследник - Cat. Отметьте варианты, которые вернут True
− isinstance(Cat(), Cat)

− isinstance(Cat(), Pet)
− issubclass(Pet, Cat)
− issubclass(Cat, object)
− issubclass(Cat, Pet)
In [5]:

```
class Pet:
    def __init__(self, name):
        self.name = name
        class Cat(Pet):
    def __init__(self, name, breed=None):
        super().__init__(name)
        self.__breed = breed
    def say(self):
        return '{0}: meow!'.format(self.name)
    def get_breed(self):
        return self.__breed
    print(isinstance(Cat('Buddy'), Cat))
print(isinstance(Cat('Buddy'), Pet))
print(issubclass(Pet, Cat))
print(issubclass(Cat, object))
print(issubclass(Cat, Pet))
```

Критерии оценивания:
Из имеющегося банка тестов формируется тестовое задание, содержащее 10 тестов.
Правильный ответ на одно задание– 1 балл, неправильный – 0 баллов.
Максимальное количество баллов – 10.

Лабораторные задания

Лабораторное задание №1. Виртуальное окружение (Virtualenv). Установка и запуск Jupyter Notebook
Лабораторное задание №2. Множества.
Лабораторное задание №3. Декораторы.
Лабораторное задание №4. Классы и объекты.
Лабораторное задание №5. Отладка и тестирование.
Лабораторное задание №6. Работа с функциями библиотеки numpy для построения нейронных сетей.

Критерии оценивания (для каждого задания):

- 11-15 б. – задание выполнено верно;
- 7-10 б. – при выполнении задания были допущены неточности, не влияющие на результат;
- 3-6 б. – при выполнении задания были допущены ошибки;
- 0-2 б. – при выполнении задания были допущены существенные ошибки.

Максимальное количество баллов за все лабораторные задания – 90 (6 заданий по 15 баллов).

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 экзамена.

Экзамен проводится по расписанию промежуточной аттестации. Количество вопросов в задании – 2 (один теоретический вопрос и одно практико-ориентированное задание). Проверка ответов

и 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.

Приложение 2

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Учебным планом предусмотрены следующие виды занятий:

- лекции;
- лабораторные занятия;.

В ходе лекционных занятий рассматриваются основные теоретические вопросы, даются рекомендации для самостоятельной работы и подготовке к лабораторным занятиям.

В ходе лабораторных занятий углубляются и закрепляются знания студентов по ряду рассмотренных на лекциях вопросов, развиваются навыки практической работы.

При подготовке к лабораторным занятиям каждый студент должен:

– изучить рекомендованную учебную литературу;

– изучить конспекты лекций;

– подготовить ответы на все вопросы по изучаемой теме.

В процессе подготовки к лабораторным и практическим занятиям студенты могут воспользоваться консультациями преподавателя.

Вопросы, не рассмотренные на лекциях и лабораторных занятиях, должны быть изучены студентами в ходе самостоятельной работы. Контроль самостоятельной работы студентов над учебной программой курса осуществляется в ходе занятий посредством тестирования и выполнения лабораторных заданий. В ходе самостоятельной работы каждый студент обязан прочитать дополнительную литературу по изучаемой теме, дополнить конспекты лекций недостающим материалом, выписками из рекомендованных первоисточников, выделить непонятные термины, найти их значение в энциклопедических словарях.

Студент должен готовиться к предстоящему лабораторному занятию по всем обозначенным в рабочей программе дисциплины вопросам.

Для подготовки к занятиям, текущему контролю и промежуточной аттестации студенты могут воспользоваться электронно-библиотечными системами. Также обучающиеся могут взять на дом необходимую литературу на абонементе университетской библиотеки или воспользоваться читальными залами.