

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Ректор

Дата подписания: 08.11.2023 16:21:03

Уникальный программный ключ:

c098bc0c1041cb2a4cf926cf171d6715d99a6ae00adc8e27b55cbe1e2dbd7c78

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

УТВЕРЖДАЮ

Директор Института магистратуры

Иванова Е.А.

« 29 » 08 2022 г.

**Рабочая программа дисциплины  
Программирование на языке Python**

Направление 09.04.03 Прикладная информатика  
магистерская программа 09.04.03.01 "Информационные системы и технологии в бизнесе"

Для набора 2022 года

Квалификация  
магистр

**КАФЕДРА Информационных систем и прикладной информатики****Распределение часов дисциплины по курсам**

Курс	1		Итого	
	уп	рп		
Лабораторные	6	6	6	6
Практические	6	6	6	6
Итого ауд.	12	12	12	12
Контактная работа	12	12	12	12
Сам. работа	92	92	92	92
Часы на контроль	4	4	4	4
Итого	108	108	108	108

**ОСНОВАНИЕ**

Учебный план утвержден учёным советом вуза от 22.02.2022 протокол № 7.

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

Зав. кафедрой: д.э.н., доцент Щербаков С.М. *сш*

Методическим советом направления: д.э.н., доцент, Щербаков С.М. *сш*

### 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-2:**Способен руководить проектами по созданию систем искусственного интеллекта с применением новых методов и алгоритмов машинного обучения со стороны заказчика

**ПК-3:**Способен руководить проектами со стороны заказчика по созданию, поддержке и использованию системы искусственного интеллекта на основе нейросетевых моделей и методов

#### В результате освоения дисциплины обучающийся должен:

<b>Знать:</b>
новые методы и алгоритмы машинного обучения (соотнесено с индикатором ПК-2.1) системы искусственного интеллекта (соотнесено с индикатором ПК-3.1)
<b>Уметь:</b>
руководить разработкой архитектуры комплексных систем искусственного интеллекта со стороны заказчика (соотнесено с индикатором ПК-2.2) 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-3.2)
<b>Владеть:</b>
навыками руководства созданием комплексных систем искусственного интеллекта с применением новых методов и алгоритмов машинного обучения (соотнесено с индикатором ПК-2.3) 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-3.3)

### 3. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература
	<b>Раздел 1. Основы Python</b>				
1.1	Тема 1. Введение в Python. Основные конструкции и базовые типы. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda. /Пр/	1	2	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.2	Тема 2. Структуры данных. Коллекции. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda. /Пр/	1	2	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.3	Тема 3. Функциональное программирование. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda. /Лаб/	1	2	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.4	Тема 1. Введение в Python. Основные конструкции и базовые типы. /Ср/	1	15	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.5	Тема 2. Структуры данных. Коллекции. /Ср/	1	15	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
1.6	Тема 3. Функциональное программирование. /Ср/	1	20	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
	<b>Раздел 2. Объектно-ориентированное программирование</b>				
2.1	Тема 4. Классы и объекты. Наследование в Python. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda. /Пр/	1	2	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
2.2	Тема 5. Отладка и тестирование. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda. /Лаб/	1	2	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
2.3	Тема 6. Построение нейросети на Python. Выполнение заданий с использованием Python, Pandas, Flask, Django, Anaconda. /Лаб/	1	2	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3

2.4	Тема 4. Классы и объекты. Наследование в Python. /Ср/	1	10	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
2.5	Тема 5. Отладка и тестирование. /Ср/	1	10	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
2.6	Тема 6. Построение нейросети на Python. /Ср/	1	22	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3
2.7	/Зачёт/	1	4	ПК-2 ПК-3	Л1.1 Л1.2Л2.1 Л2.2 Л2.3

#### 4. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

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

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

##### 5.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л1.1	Сузи Р. А.	Язык программирования Python: учебное пособие	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ)  Бином. Лаборатория знаний, 2007	<a href="https://biblioclub.ru/index.php?page=book&amp;id=233288">https://biblioclub.ru/index.php?page=book&amp;id=233288</a> неограниченный доступ для зарегистрированных пользователей
Л1.2	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	<a href="https://www.iprbookshop.ru/117194.html">https://www.iprbookshop.ru/117194.html</a> неограниченный доступ для зарегистрированных пользователей

##### 5.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год	Колич-во
Л2.1	Емельянов А. А.	Прикладная информатика: журнал	Москва: Синергия ПРЕСС, 2010	<a href="https://biblioclub.ru/index.php?page=book&amp;id=120321">https://biblioclub.ru/index.php?page=book&amp;id=120321</a> неограниченный доступ для зарегистрированных пользователей
Л2.2	Буйначев С. К., Боклаг Н. Ю.	Основы программирования на языке Python: учебное пособие	Екатеринбург: Издательство Уральского университета, 2014	<a href="https://biblioclub.ru/index.php?page=book&amp;id=275962">https://biblioclub.ru/index.php?page=book&amp;id=275962</a> неограниченный доступ для зарегистрированных пользователей
Л2.3		Язык программирования Python	Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016	<a href="http://www.iprbookshop.ru/52211.html">http://www.iprbookshop.ru/52211.html</a> неограниченный доступ для зарегистрированных пользователей

##### 5.3 Профессиональные базы данных и информационные справочные системы

Национальная электронная библиотека (НЭБ) - <https://rusneb.ru/>

Консультант+

Гарант

##### 5.4. Перечень программного обеспечения

Python

Pandas

Flask

Django

Anaconda

**5.5. Учебно-методические материалы для студентов 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 (МОДУЛЯ)**

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.

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

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
ПК-2: Способен руководить проектами по созданию систем искусственного интеллекта с применением новых методов и алгоритмов машинного обучения со стороны заказчика			
З. новые методы и алгоритмы машинного обучения	знает синтаксис языка Python, основные структуры данных, базовые алгоритмические конструкции	полнота и содержательность ответа умение приводить примеры	ТЗ – тестовые задания (1-30), З – вопросы к зачету (1-18)
У. 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 приводить примеры умение самостоятельно находить решение поставленных задач	ПЗ – практические задания (1-2), ЛЗ – лабораторные задания (1-2)
В. 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	ПЗ – практические задания (1-2), ЛЗ – лабораторные задания (1-2)
ПК-3: Способен руководить проектами со стороны заказчика по созданию, поддержке и использованию системы искусственного интеллекта на основе нейросетевых моделей и методов			
З. 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	ТЗ – тестовые задания (1-30), З – вопросы к зачету (19-36)
У. 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 приводить примеры умение самостоятельно находить решение поставленных задач	ПЗ – практические задания (3-4), ЛЗ – лабораторные задания (3-4)
В. навыками управления проектами по созданию, поддержке и использованию систем искусственного интеллекта на основе аналитики больших данных с применением новых методов и алгоритмов машинного обучения	применяет современные инструментальные методы и средства обучения моделей искусственных нейронных сетей	полнота и содержательность ответа умение приводить примеры умение самостоятельно находить решение поставленных задач	ПЗ – практические задания (5-6), ЛЗ – лабораторные задания (5-6)

1.2 Шкалы оценивания:

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

50-100 баллов (зачет);

0-49 баллов (незачет).

**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 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.

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

### *Критерии оценивания:*

- 50-100 баллов («зачет») – 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; 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; 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 («незачет») – 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.

### Тестовые задания

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
  - .рус
  - .py
4. Что происходит при итерации по генератору?
  - Итерация происходит по списку значений, который вернул генератор при вызове
  - Каждую итерацию вызывает функция next, и генератор исполняется с начала
  - Каждую итерацию вызывает 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?
- pip
  - pip
  - per
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)
– isinstance(Pet, Cat)
– isinstance(Cat, object)
– isinstance(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 тестовых вопросов для одного обучающегося. Каждый тестовый вопрос содержит 2-5 вариантов ответов, один или несколько из которых – верные.

Правильный ответ на один тестовый вопрос – 1 балл, неправильный – 0 баллов.

**Максимальное количество баллов за тестовые задания – 10.**

### Практические задания

Практическое задание 1. Знакомство с IDE Pycharm Community. Основы синтаксиса языка.

Практическое задание 2. Списки и кортежи. Словари. Множества. Примеры программ.

Практическое задание 3. Файловый ввод-вывод. Функции. Рекурсия

Практическое задание 4. Объявление классов. Создание и инициализация экземпляров (объектов) классов. Работа с атрибутами экземпляра класса. Объявление и вызов методов экземпляров. Методы класса. Наследование.

Практическое задание 5. Отладка и тестирование. Запустить отладчик в программе. Написание и запуск тестов.

Практическое задание 6. Представление типов данных библиотеки numpy. Некоторые математические алгоритмы библиотеки numpy.

*Критерии оценивания (для каждого задания):*

4-5 б. – задание выполнено верно;

2-3 б. – при выполнении задания были допущены неточности, не влияющие на результат;

0-1 б. – при выполнении задания были допущены ошибки.

**Максимальное количество баллов за все практические задания – 30 (6 заданий по 5 баллов).**

### 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):*

9-10 б. – задание выполнено верно;

6-8 б. – при выполнении задания были допущены неточности, не влияющие на результат;

3-5 б. – при выполнении задания были допущены ошибки;

1-2 б. – при выполнении задания были допущены существенные ошибки.

***Максимальное количество баллов за все лабораторные задания – 60 (6 заданий по 10 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 зачета.

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

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Учебным планом предусмотрены следующие виды занятий:

- практические занятия,
- лабораторные занятия.

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

При подготовке к практическим и лабораторным занятиям каждый студент должен:

- изучить рекомендованную учебную литературу;
- подготовить ответы на все вопросы по изучаемой теме и представить результаты выполненных заданий.

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

Вопросы, не рассмотренные на практических и лабораторных занятиях, должны быть изучены студентами в ходе самостоятельной работы. Контроль самостоятельной работы студентов над учебной программой курса осуществляется в ходе занятий методом выполнения тестовых, практических и лабораторных заданий. В ходе самостоятельной работы каждый студент обязан прочитать основную и по возможности дополнительную литературу по изучаемой теме, выделить непонятные термины, найти их значение в энциклопедических словарях.

Студент должен готовиться к предстоящему практическому и 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.