

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Макаренко Елена Николаевна

Должность: Рector

Дата подписания: 28.06.2023 14:25:08

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.

«01» июня 2023 г.

Рабочая программа дисциплины
Программирование на языке Python

Направление 09.04.03 Прикладная информатика

магистерская программа

09.04.03.03 Машинное обучение и технологии больших данных

Для набора 2023 года

Квалификация

магистр

Кафедра Информационных систем и прикладной информатики

Составители рабочей программы:

д.э.н., зав.каф. Щербаков Сергей Михайлович

СОДЕРЖАНИЕ

I. Цели и задачи освоения дисциплины	4
II. Место дисциплины в структуре образовательной программы.....	4
III. Требования к результатам освоения дисциплины	6
IV. Содержание и структура дисциплины	8
4.1. Содержание дисциплины, структурированное по темам.....	8
4.2. План внеаудиторной самостоятельной работы.....	9
4.3. Содержание учебного материала.....	11
V. Образовательные технологии	12
VI. Учебно-методическое обеспечение дисциплины	12
6.1. Основная литература	12
6.2. Дополнительная литература.....	12
6.3. Периодические издания.....	13
6.4. Перечень ресурсов сети Интернет.....	13
VII. Материально-техническое обеспечение дисциплины	13
VIII. Методические указания для обучающихся по освоению дисциплины	13
IX. Учебная карта дисциплины	15
X. Фонд оценочных средств.....	16
10.1. Паспорт фонда оценочных средств	16
10.2. Контрольная работа № 1 (тестирование).....	16
10.3. Контрольная работа № 2 (тестирование).....	19
10.4. Лабораторные работы №№ 1–6 (выполнение, подготовка отчёта, защита отчёта).....	21
10.5. Индивидуальное задание «Построение нейронной сети».....	22

I. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Цели освоения дисциплины:

- получение студентами теоретических знаний основ построения программного обеспечения для решения профессиональных задач на языке программирования Python;
- повысить профессиональную компетенцию специалиста в области разработки программных приложений на языке Python, в следующих областях: работа с таблицами и структурами данных, их чтение, анализ и последующая обработка в задачах статистики и машинного обучения; работа с библиотекой для математических операций numpy.

Задачи освоения дисциплины:

- ознакомление студентов с основами языка Python;
- изучение базовых алгоритмов и простейших структур данных;
- получение практики применения базовых возможностей и библиотек языка для решения прикладных задач;
- демонстрация современных тенденций в области применения языка Python в промышленных системах.

II. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина относится к факультативным дисциплинам образовательной программы.

Данная дисциплина опирается на базовые знания, умения и навыки, формируемые предшествующими элементами образовательной программы:

Наименование дисциплины (модуля), практики	Требуемые знания, умения, навыки
Программное и аппаратное обеспечение информационных систем	<p><i>Знания:</i></p> <ul style="list-style-type: none">– Основных тенденций развития интеллектуальных технологий в области обработки больших данных.– Методологических основ интеллектуального анализа больших данных.– Классических алгоритмов.– Критериев оценки качества программных решений.– Критериев оценки качества программного кода.– Методов разработки алгоритмических решений.– Способов применения интеллектуальных технологий для разработки программных продуктов.– Способов модернизации программного обеспечения автоматизированных систем.– Способов модернизации аппаратного обеспечения автоматизированных систем. <p><i>Умения:</i></p> <ul style="list-style-type: none">– Применять методы интеллектуального анализа больших данных для решения профессиональных задач.– Осуществлять разработку оригинальных алгоритмов и программных средств Big Data в условиях информационной неопределенности.– Применять критерии оценки качества программного кода и практических решений.– Применять интеллектуальные технологии при разработке программных решений.– Выполнять модернизацию программных и аппаратных решений. <p><i>Навыки:</i></p>

Наименование дисциплины (модуля), практики	Требуемые знания, умения, навыки
	<ul style="list-style-type: none"> – Использование методов интеллектуального анализа больших данных при разработке алгоритмов и программных средств Big Data для решения профессиональных задач в условиях информационной неопределенности. – Разрабатывать программные решения с применением методов алгоритмизации и интеллектуальных технологий. – Оценки качества программных продуктов. – Применения современных методов модернизации программных и аппаратных решений в автоматизированных системах.
Методы машинного обучения	<p><i>Знания:</i></p> <ul style="list-style-type: none"> – Знает классы методов и алгоритмов машинного обучения. – Знает возможности современных инструментальных средств и систем программирования для решения задач машинного обучения. – Знает функциональность современных инструментальных средств и систем программирования в области создания моделей и методов машинного обучения. <p><i>Умения:</i></p> <ul style="list-style-type: none"> – Умеет ставить задачи и адаптировать методы и алгоритмы машинного обучения. – Умеет проводить сравнительный анализ и осуществлять выбор инструментальных средств для решения задач машинного обучения. – Умеет применять современные инструментальные средства и системы программирования для разработки новых методов и моделей машинного обучения.

Знания, умения и навыки, формируемые данной дисциплиной, потребуются при освоении следующих элементов образовательной программы:

- производственная практика, проектно-технологическая практика;
- производственная практика, преддипломная практика;
- выполнение и защита выпускной квалификационной работы.

III. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Освоение дисциплины направлено на формирование следующих компетенций в соответствии с образовательной программой:

Перечень планируемых результатов обучения по дисциплине, соотнесённых с индикаторами достижения компетенций

Компетенция	Индикаторы достижения компетенции	Результаты обучения
ПК-2. Способен руководить проектами по созданию систем искусственного интеллекта с применением новых методов и алгоритмов машинного обучения со стороны заказчика	ПК-2.1. Руководит разработкой архитектуры комплексных систем искусственного интеллекта со стороны заказчика	<p><i>Знания:</i></p> <ul style="list-style-type: none">– Знает возможности современных инструментальных средств и систем программирования для решения задач машинного обучения.– Знает синтаксис языка Python.– Знает основные структуры данных.– Знает базовые алгоритмические конструкции.– Знает инструментарий для отладки и тестирования программ. <p><i>Умения:</i></p> <ul style="list-style-type: none">– Умеет проводить сравнительный анализ и осуществлять выбор инструментальных средств для решения задач машинного обучения.– Умеет определять критерии и метрики оценки результатов моделирования при построении систем искусственного интеллекта в исследуемой области. <p><i>Навыки:</i></p> <ul style="list-style-type: none">– Владеет навыками разработки алгоритмических конструкций.– Владеет навыками отладки программ.

	<p>ПК-2.2. Осуществляет руководство созданием комплексных систем искусственного интеллекта с применением новых методов и алгоритмов машинного обучения</p>	<p><i>Знания:</i></p> <ul style="list-style-type: none"> – Знает функциональность современных инструментальных средств и систем программирования в области создания моделей и методов машинного обучения. – Знает принципы построения систем искусственного интеллекта, методы и подходы к планированию и реализации проектов по созданию систем искусственного интеллекта, методы интеллектуального планирования экспериментов. <p><i>Умения:</i></p> <ul style="list-style-type: none"> – Умеет применять современные инструментальные средства и системы программирования для разработки новых методов и моделей машинного обучения. – Умеет руководить выполнением коллективной проектной деятельности для создания, поддержки и использования систем искусственного интеллекта.
<p>ПК-3. Способен руководить проектами со стороны заказчика по созданию, поддержке и использованию системы искусственного интеллекта на основе нейросетевых моделей и методов</p>	<p>ПК-3.1. Руководит проектами по разработке, систем искусственного интеллекта на основе моделей глубоких нейронных сетей и нечетких моделей и методов со стороны заказчика</p>	<p><i>Знания:</i></p> <ul style="list-style-type: none"> – Знает принципы построения моделей глубоких нейронных сетей и глубокого машинного обучения – Знает функциональность современных инструментальных средств и систем программирования в области создания моделей искусственных нейронных сетей. <p><i>Умения:</i></p> <ul style="list-style-type: none"> – Умеет руководить выполнением коллективной проектной деятельности для создания, поддержки и использования систем искусственного интеллекта на основе моделей глубоких нейронных сетей и нечетких моделей и методов – Умеет применять современные инструментальные методы и средства обучения моделей искусственных нейронных сетей.

IV. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоёмкость дисциплины составляет 5 зачётных единиц, 180 часов,
 Форма промежуточной аттестации: зачёт

4.1. Содержание дисциплины, структурированное по темам

№ п/п	Темы дисциплины	Семестр	Виды учебной работы и их трудоёмкость, часы (в том числе с использованием онлайн-курсов)				Наименования оценочных средств
			Контактная работа			Самостоя- тельная работа	
			Лекции	Практические занятия	Лабораторные занятия		
Модуль 1. Основы Python							
1	Введение в Python. Основные конструкции и базовые типы.	1	–	4	4	18	– Лабораторные работы
2	Структуры данных. Коллекции.	1	–	6	6	18	– Лабораторные работы
3	Функциональное программирование.	1	–	6	6	18	– Контрольная работа №1 (тест по теме «Язык Python»); – Лабораторные работы
Модуль 2. Объектно-ориентированное программирование							
4	Классы и объекты. Наследование в Python.	1	–	6	6	18	– Лабораторные работы
5	Отладка и тестирование.	1	–	6	6	18	– Лабораторные работы
6	Построение нейросети на Python.	1	–	8	8	18	– Лабораторные работы; – Контрольная работа №2 (Тест по теме «Объектно-ориентированное программирование в Python»); Индивидуальное задание «Построение нейронной сети».
Итого часов			–	36	36	108	–

4.2. План внеаудиторной самостоятельной работы

№ п/п	Темы дисциплины	Семестр	Вид самостоятельной работы	Сроки выполнения (нед.)	Затраты времени (часы)	Учебно-методическое обеспечение
Модуль 1. Основы Python						
1	Тема 1. Введение в Python. Основные конструкции и базовые типы.	1	– проработка и повторение материала практических занятий; – подготовка к лабораторной работе.	1–2	18	[1], [4]
2	Тема 2. Структуры данных. Коллекции.	1	– проработка и повторение материала практических занятий; – подготовка к лабораторным работам, подготовка отчётов о выполнении лабораторных работ, подготовка к защите отчётов о выполнении лабораторных работ.	3-4	18	[2], [3]
3	Тема 3. Функциональное программирование.	1	– проработка и повторение материала практических занятий; – подготовка к лабораторным работам, подготовка отчётов о выполнении лабораторных работ, подготовка к защите отчётов о выполнении лабораторных работ; – подготовка к тестированию.	4-8	18	[2]
Модуль 2. Объектно-ориентированное программирование						
4	Тема 4. Классы и объекты. Наследование в Python.	1	– проработка и повторение материала практических занятий; подготовка к лабораторной работе.	8-10	18	[2], [3]
5	Тема 5. Отладка и тестирование.	1	– проработка и повторение материала практических занятий; подготовка к лабораторным работам, подготовка отчётов о выполнении лабораторных работ, подготовка к защите отчётов о выполнении лабораторных работ.	10-12	18	[3]

№ п/п	Темы дисциплины	Семестр	Вид самостоятельной работы	Сроки выполнения (нед.)	Затраты времени (часы)	Учебно- методическое обеспечение
6	Тема 6. Построение нейросети на Python.	1	<ul style="list-style-type: none"> – проработка и повторение материала практических занятий; – подготовка к лабораторным работам, подготовка отчётов о выполнении лабораторных работ, подготовка к защите отчётов о выполнении лабораторных работ; – выполнение и подготовка к защите индивидуального задания «Построение нейронной сети»; – подготовка к тестированию. 	12-17	18	[1]–[4]
Общая трудоёмкость самостоятельной работы по дисциплине					108	–

4.3. Содержание учебного материала

Модуль 1. Основы Python.

Тема 1. Введение в Python. Основные конструкции и базовые типы.

Выбор среды разработки (IDE). Базовые типы. Условные операторы. Циклы. Файлы.

Модули и пакеты. Виртуальное окружение (Virtualenv).

Установка и запуск Jupyter Notebook.

Тема 2. Структуры данных. Коллекции.

Списки и кортежи. Словари. Множества.

Тема 3. Функциональное программирование. Функции.

map, filter, reduce, partial, lambda — анонимные функции. Списочные выражения

Декораторы. Генераторы.

Модуль 2. Объектно-ориентированное программирование.

Тема 4. Классы и объекты. Наследование в Python.

Классы и экземпляры. Методы. Наследование. Классы исключений.

Тема 5. Отладка и тестирование.

Тема 6. Построение нейросети на Python.

Основные архитектуры нейронных сетей. Теорема Байеса. Алгоритм EM.

Работа с математической библиотекой numpy. Введение в Tensor Flow и Keras.

Перечень тем практических занятий

№ п/п	Тема практического занятия	Количество часов
Модуль 1. Основы Python.		
1	Знакомство с IDE Pycharm Community. Основы синтаксиса языка.	4
2	Списки и кортежи. Словари. Множества. Примеры программ.	6
3	Файловый ввод-вывод. Функции. Рекурсия	6
Модуль 2. Объектно-ориентированное программирование.		
4	Объявление классов. Создание и инициализация экземпляров (объектов) классов Работа с атрибутами экземпляра класса. Объявление и вызов методов экземпляров. Методы класса. Наследование.	6
5	Отладка и тестирование. Запустить отладчик в программе. Написание и запуск тестов.	6
6	Представление типов данных библиотеки numpy. Некоторые математические алгоритмы библиотеки numpy.	8
Всего часов		36

Перечень лабораторных работ

№ п/п	Название лабораторной работы	Количество часов
Модуль 1. Основы Python.		
1	Лабораторная работа №1. Виртуальное окружение (Virtualenv). Установка и запуск Jupyter Notebook	4
2	Лабораторная работа №2. Множества.	6
3	Лабораторная работа №3. Декораторы.	6
Модуль 2. Объектно-ориентированное программирование.		
4	Лабораторная работа №4. Классы и объекты.	6

№ п/п	Название лабораторной работы	Количество часов
5	Лабораторная работа №5. Отладка и тестирование.	6
6	Лабораторная работа №6. Работа с функциями библиотеки numpy для построения нейронных сетей.	8
Всего часов		36

V. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

По дисциплине предусмотрены следующие методы обучения и интерактивные формы проведения занятий:

- дискуссионные (обсуждение новых информационным технологий);
- групповой работы (работа в малых группах на практических занятиях).

Наряду с традиционными образовательными технологиями, для реализации дисциплины могут использоваться технологии электронного обучения и дистанционные образовательные технологий в электронной информационно-образовательной среде университета. Контактная работа обучающихся с преподавателем может проводиться с использованием платформ Microsoft Teams, Cisco, Moodle (BigBlueButton) и др., что позволяет обеспечить онлайн и офлайн взаимодействие преподавателя с обучающимися в рамках дисциплины.

Основными методами текущего контроля являются электронный учёт и контроль учебных достижений студентов (использование средств сервиса балльно-рейтинговой системы; ведение электронного журнала успеваемости, проведение электронного тестирования и применение других средств контроля с использованием системы электронного обучения).

VI. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Основная литература

1. Забелин А. А. Реализация алгоритмов вычислительной математики на языке Python [Электронный ресурс]: учебное пособие / Забелин А. А. - Чита: ЗабГУ, 2020. - 130 с. – URL: <https://e.lanbook.com/book/173632>.

2. Тарланов А. Т. Основы языка программирования Python [Электронный ресурс]: учебно-методическое пособие / Тарланов А. Т., Магомедов Ш. Г. - Москва: РТУ МИРЭА, 2019. - 107 с. – Режим доступа: по подписке. – URL: <https://e.lanbook.com/book/171465>.

3. Шелудько В. М. Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули: учебное пособие / В.М. Шелудько; Министерство науки и высшего образования РФ; Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет»; Институт компьютерных технологий и информационной безопасности - Ростов-на-Дону|Таганрог: Издательство Южного федерального университета, 2017. - 108 с. – Режим доступа: по подписке. – URL: [biblioclub: http://biblioclub.ru/index.php?page=book&id=500060](http://biblioclub.ru/index.php?page=book&id=500060).

6.2. Дополнительная литература

4. Сузи, Р.А. Язык программирования Python: учебное пособие: [16+] / Р.А. Сузи. – 2-е изд., испр. – Москва: Интернет-Университет Информационных Технологий (ИНТУИТ): Бином. Лаборатория знаний, 2007. – 327 с. – (Основы информационных технологий). – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=233288>. – ISBN 978-5-9556-0109-0. – Текст: электронный.

5. Северенс, Ч. Введение в программирование на Python / Ч. Северенс. – 2-е изд., испр. – Москва: Национальный Открытый Университет «ИНТУИТ», 2016. – 231 с.: схем., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=429184>.

6.3. Периодические издания

- Научный журнал «Машинное обучение и анализ данных» <http://jmla.org/ru/journal>

6.4. Перечень ресурсов сети Интернет

- Документация языка Python (<https://www.python.org/>);
- Документация библиотеки numpy (<http://www.numpy.org/>).

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

При реализации дисциплины используются следующие помещения, оборудование и программное обеспечение:

Аудитория технологии и методов программирования:

Интерактивная доска с проектором, персональные компьютеры (10 шт.), ноутбук. Windows 7, Microsoft Office 2007, Adobe Acrobat Reader (Бесплатное проприетарное ПО, <https://acrobat.adobe.com/ru/ru/acrobat/pdf-reader/volume-distribution.html>), Foxit (Бесплатное проприетарное ПО, <https://www.foxitsoftware.com/ru/>), i2 Analyst's Notebook (Бесплатная лицензия для образовательных целей, <https://developer.ibm.com/academic/>), Notepad++, Бесплатное ПО (GNU GPL 2), <https://notepad-plus-plus.org/>, Total Commander 7.x, WinRAR, XAMPP, Бесплатное ПО (GNU GPL), <http://www.apachefriends.org/en/xampp.html>, актуальные версии браузеров Google Chrome (Свободное ПО, <https://google.com/chrome/browser/>), Mozilla Firefox, Бесплатное ПО (GNU GPL), <https://firefox.com/>, Edge, Safari с поддержкой протокола WebRTC, PyCharm 2017.1.2 <https://www.jetbrains.com/pycharm/> Свободное ПО, <https://www.python.org/>, Evolus Pencil, Свободное ПО (GNU GPL 2), <https://pencil.evolus.vn/>, Team Foundation Server 2015, Visual Studio 2015

VIII. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Дисциплина включает в себя практические и лабораторные занятия, а также самостоятельную работу обучающихся.

Организация образовательного процесса по дисциплине осуществляется с использованием системы электронного обучения.

Лабораторные работы требуют предварительной теоретической подготовки по соответствующей теме: изучения основной и дополнительной литературы по дисциплине. На лабораторных работах ход выполнения контролируется преподавателем, возникающие проблемы интерактивно обсуждаются и подробно разбираются. Допускается завершать выполнение лабораторных работ дома с последующей сдачей этой работы на следующем занятии.

Занятия могут проводиться с применением дистанционных образовательных технологий с использованием платформ Microsoft Teams, Cisco, Moodle (BigBlueButton) и др.

Проведение практических занятий осуществляется с постановкой проблемных вопросов, допускающих возникновение дискуссий, что предполагает активное включение студентов в образовательный процесс.

Используемые технологии обеспечивают:

- формирование компетенций, осознанное усвоение знаний, качественное освоение умений их применять и формирование заинтересованного отношения к изучаемым объектам в единстве;
- продуктивность познавательной деятельности, научный поиск, создание субъективно и объективно новых знаний или других продуктов;
- ориентацию на студентов, стимулирование их активности, самостоятельности, инициативы и ответственности;
- контекстный характер обучения, то есть привязку к реальным профессиональным задачам;

– вовлеченность студентов в выполняемую деятельность, возможность проявить и развить свой интеллектуальный, творческий, личностный, деловой потенциал.

Самостоятельная работа направлена на повышение качества обучения, углубление и закрепление знаний студента, развитие аналитических навыков по проблематике учебной дисциплины, активизацию учебно-познавательной деятельности студентов и снижение аудиторной нагрузки.

Максимальное количество баллов по каждому виду контрольных мероприятий указано в учебной карте дисциплины.

IX. УЧЕБНАЯ КАРТА ДИСЦИПЛИНЫ

Курс 1, семестр 1, очная форма обучения

№ п/п	Виды контрольных мероприятий (наименования оценочных средств)	Количество баллов	
		Текущий контроль	Рубежный контроль
Модуль 1. Наименование модуля 1			
1	Лабораторные работы №№ 1–3 (выполнение, подготовка отчёта, защита отчётов)	30 (3 работы × 10 баллов)	–
2	Контрольная работа № 1 (тестирование)	–	10
Модуль 2. Наименование модуля 2			
3	Лабораторные работы №№ 4–6 (выполнение, подготовка отчёта, защита отчётов)	30 (3 работы × 10 баллов)	–
4	Контрольная работа № 2 (тестирование)	–	10
5	Индивидуальное задание «Построение нейронной сети»	20	
Всего		80	20
Бонусные баллы		до 10 баллов Участие в профильных соревнованиях, конкурсах, конференциях по направлению подготовки (в зависимости от успешности участия). Участие в подготовке и организации профильных мероприятий по направлению подготовки (в зависимости от объема выполненной работы).	
Промежуточная аттестация в форме зачёта		Оценка по дисциплине выставляется по сумме баллов за текущий контроль и рубежный контроль. Для получения оценки «зачтено» необходимо набрать не менее 60 баллов	

Х. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

10.1. Паспорт фонда оценочных средств

№ п/п	Индикатор достижения компетенции	Наименование оценочного средства
1	ПК-2.1. Руководит разработкой архитектуры комплексных систем искусственного интеллекта со стороны заказчика	– контрольная работа № 2 (тестирование); – лабораторные работы № 1-6 (выполнение, подготовка отчёта, защита отчёта)
2	ПК-2.2. Осуществляет руководство созданием комплексных систем искусственного интеллекта с применением новых методов и алгоритмов машинного обучения	– контрольная работа № 2 (тестирование); – лабораторные работы № 1-6 (выполнение, подготовка отчёта, защита отчёта)
3	ПК-3.1. Руководит проектами по разработке систем искусственного интеллекта на основе моделей глубоких нейронных сетей и нечетких моделей и методов со стороны заказчика	– контрольная работа № 1 (тестирование); – лабораторная работа № 6 (выполнение, подготовка отчёта, защита отчёта); – индивидуальное задание «Построение нейронной сети»

10.2. Контрольная работа № 1 (тестирование)

Контрольная работа №1 служит рубежным контролем модуля 1, оценивается в 10 баллов и содержит 20 вопросов по 0,5 балла каждый. Время выполнения — 30 минут. Тест принимается к оцениванию, если верно выполнено 60% вопросов.

1. Совместимы ли Python 2 и Python 3?

- Совместимы полностью
- Несовместимы (Правильно)

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

- Python
- Java
- C (Правильно)

3. Какое расширение обычно дают файлам с кодом на Python?

- .python
- .рус
- .py (Правильно)

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 (Правильно)

5. Выберите верные утверждения про множества:

- проверка на входение элемента в множество происходит за линейное время
- множества изменяемые (Правильно)
- множества неизменяемые
- проверка на входение элемента в множество происходит за константное время (Правильно)

6. `bool(0.000001)` - True или False?

- True (Правильно)
- 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)` (Правильно)
- `def login_required(send_feedback)(request)`
- `def login_required(send_feedback)`

8. Какая функция позволяет считать ввод пользователя из терминала?

- `readline()`
- `input()` (Правильно)
- `read()`
- `enter()`

9. Какой метод превратит байтовую строку в строку?

- `.encode()`
- `.decode()` (Правильно)

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 (Правильно)

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` (Правильно)
- `import foo.bar.run`

12. Выберите верные утверждения про кортежи:

- Проверка на входение элемента в кортеж происходит за константное время
- Кортежи изменяемые
- Проверка на входение элемента в кортеж происходит за линейное время (Правильно)
- Кортежи неизменяемые (Правильно)

– Кортежи могут содержать элементы различных типов (Правильно)

13. Зачем нужен `virtualenv` (виртуальное окружение)?

– Возможность запускать несколько интерпретаторов Python одновременно

– Изоляция зависимостей (Правильно)

– Возможность увеличить скорость запуска скомпилированных в байткод Python-программ

14. Какая утилита позволяет ставить внешние Python пакеты в систему?

– `pip`

– `pip` (Правильно)

– `per`

15. Что содержат файлы с расширением `.pyc`?

– Код на Python, скомпилированный в байткод (Правильно)

– Код на Python, скомпилированный в машинный код

16. Для чего используются декораторы?

– Для эффективного использования памяти при итерации

– Для модификации поведения функций (Правильно)

– Чтобы иметь возможность импортировать функцию в другой модуль

17. Какие имена переменных правильные?

– `!name`

– `!name`

– `Name` (Правильно)

– `_name` (Правильно)

18. Как пишутся комментарии в Python?

– `#` это комментарий (Правильно)

– `//` это комментарий

– `"""` это комментарий `"""`

19. Что получится в результате выполнения среза `[2:3]` для строки "привет"?

"и"

"ри"

"р"

"ив" (Правильно)

20. `pi = 3.1415926`

`pi_fmt = f"{pi:#0.2f}"`

Чему будет равен `pi_fmt`?

– Строке "3.14" (Правильно)

– Числу 3.14

10.3. Контрольная работа № 2 (тестирование)

Контрольная работа №2 1 служит рубежным контролем модуля 2, оценивается в 10 баллов и содержит 10 вопросов по 1 баллу каждый. Время выполнения — 15 минут. Тест принимается к оцениванию, если верно выполнено 60% вопросов.

1. Отметьте все исключения стандартной библиотеки Python

- ValueError (Правильно)
- RequestException
- KeyboardInterrupt (Правильно)
- IndexError (Правильно)

2. Какие из методов генерации исключения разрешены в Python?

- raise None
- raise ValueError (Правильно)
- raise "ValueError"
- raise ValueError("error") (Правильно)

3. Обращение к несуществующему атрибуту экземпляра сгенерирует исключение KeyError

- сгенерирует исключение AttributeError (Правильно)
- вернет False
- вернет None

4. Отметьте верные утверждения про classmethod

- Метод не принимает дополнительных аргументов кроме указанных программистом
- Метод первым аргументом принимает класс (Правильно)
- К этому методу можно обращаться от экземпляра класса (Правильно)
- Метод первым аргументом принимает ссылку на экземпляр класса
- К этому методу можно обращаться от имени класса (Правильно)

```
In [1]: class Tmp:
        @classmethod
        def foo(cls, val):
            print(val)
```

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

5. Отметьте верные утверждения про staticmethod

- К этому методу можно обращаться от экземпляра класса (Правильно)
- Метод первым аргументом принимает ссылку на экземпляр класса
- Метод первым аргументом принимает класс
- Метод не принимает дополнительных аргументов кроме указанных программистом (Правильно)
- К этому методу можно обращаться от имени класса (Правильно)

```
In [2]: class Tmp:
        @staticmethod
        def foo(val):
            print(val)
```

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

6. Для чего используются `@property`?

- Чтобы делать методы приватными
- Чтобы делать атрибуты приватными
- Чтобы создать вычисляемый атрибут (Правильно)

```
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;
```

7. 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 (Правильно)

- символ нижнего подчеркивания в начале метода не добавляет никакого дополнительного значения

8. Можно ли использовать экземпляры классов в качестве ключей словаря (dict)?

- Да (Правильно)
- Нет

9. Можно ли передавать экземпляр класса как аргумент в функцию?

- Да (Правильно)
- Нет

10. Предположим есть базовый класс питомец - `Pet` и класс наследник - `Cat`. Отметьте варианты, которые вернут `True`

- `isinstance(Cat(), Cat)` (Правильно)
- `isinstance(Cat(), Pet)` (Правильно)
- `issubclass(Pet, Cat)`
- `issubclass(Cat, object)` (Правильно)
- `issubclass(Cat, Pet)` (Правильно)

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))

```

10.4. Лабораторные работы №№ 1–6 (выполнение, подготовка отчёта, защита отчёта)

Лабораторные работы проводятся в аудитории, оборудованной в соответствии с п.7.2. рабочей программы дисциплины (РПД). Основанием для допуска к лабораторной работе являются:

- знания теоретического материала и методических указаний, которые должен продемонстрировать студент в начале занятия.

- отсутствие задолженностей по предыдущим лабораторным работам, если таковые проводились.

При домашней подготовке к выполнению каждой лабораторной работы студент знакомится с теоретическим материалом по тематике работы и готовит макет отчета о выполнении.

По каждой из лабораторных работ отчет оформляется отдельно, но после сдачи добавляется к файлу, в котором хранятся отчеты всех работ по курсу. Страницы в файле имеют сквозную нумерацию; титульный лист один.

Содержание отчета:

Вариант задания.

Листинг программы и расширенный комментарий к ней.

Краткие выводы.

Защита отчета о выполнении лабораторной работы сопровождается демонстрацией полученных результатов, теоретических знаний и ответов на контрольные вопросы теме работы.

В процессе подготовки и выполнения лабораторных работ студент руководствуется учебной и методической литературой, указанной в п. 6.1 РПД.

Пример задания для лабораторной работы:

Написать python-модуль solution.py, внутри которого необходимо поместить код класса FileReader. Конструктор этого класса принимает один параметр: путь до файла на диске. В классе FileReader должен быть реализован метод read, возвращающий строку - содержимое файла, путь к которому был указан при создании экземпляра класса. Python модуль должен быть написан таким образом, чтобы импорт класса FileReader из него не вызвал ошибок.

При написании реализации метода read, необходимо учитывать случай, когда при инициализации был передан путь к несуществующему файлу. Требуется обработать возникающее при этом исключение FileNotFoundError и вернуть из метода read пустую строку.

Критерии оценки:

- 9-10 баллов выставляется студенту, если он своевременно выполнил все задачи, предусмотренные в лабораторной работе, подготовил отчет в соответствии с требованиями преподавателя и в процессе защиты продемонстрировал полноту теоретических знаний в объеме содержания учебной дисциплины, относящейся к лабораторной работе. Сумел ответить на дополнительные вопросы, связанные не только с процессом выполнения лабораторной работы, но и с пониманием совершенных действий и решенных задач.
- 7-8 баллов выставляется студенту, если он выполнил все задачи, предусмотренные в лабораторной работе, подготовил отчет в соответствии с требованиями преподавателя и в процессе защиты продемонстрировал наличие достаточных теоретических знаний в объеме содержания учебной дисциплины, относящейся к лабораторной работе. Сумел ответить на вопросы, связанные с процессом выполнения лабораторной работы.
- 5-6 баллов выставляется студенту, если он более чем на половину выполнил поставленные в лабораторной работе задачи, способен ответить на вопросы, касающиеся теоретической составляющей в объеме содержания учебной дисциплины, относящейся к лабораторной работе.
- 1-4 балла выставляется студенту, если он не выполнил задание к лабораторной работе, не изучил достаточно теоретический материал или не имеет отчета.
- 0 баллов выставляется студенту, при невыполнении требований, предусмотренных в случае удовлетворительной оценки.

10.5. Индивидуальное задание «Построение нейронной сети»

В качестве индивидуального задания студенту предлагается на основе лабораторной работы №6 самостоятельно попрактиковаться в построении нейронной сети на языке Python, ее обучении на самостоятельно выбранных наборах данных с получением последующих результатов. Индивидуальное задание оценивается в 20 баллов, срок выполнения — 10 дней.

Критерии оценки:

17 - 20 баллов выставляется студенту, если он своевременно выполнил все задачи, предусмотренные в задании, подготовил отчет в соответствии с требованиями преподавателя и в процессе защиты продемонстрировал полноту теоретических знаний в объеме содержания учебной дисциплины, относящейся к задаче. Сумел ответить на дополнительные вопросы, связанные не только с процессом выполнения задания, но и с пониманием совершенных действий и решенных задач.

14 -17 баллов выставляется студенту, если он выполнил все задачи, предусмотренные в задании, подготовил отчет в соответствии с требованиями преподавателя и в процессе защиты продемонстрировал наличие достаточных теоретических знаний в объеме содержания учебной дисциплины, относящейся к заданию. Сумел ответить на вопросы, связанные с процессом выполнения лабораторной работы.

12-13 баллов выставляется студенту, если он более чем на половину выполнил поставленные задачи, способен ответить на вопросы, касающиеся теоретической составляющей в объеме содержания учебной дисциплины, относящейся к заданию.

0-12 баллов выставляется студенту, при невыполнении требований, предусмотренных в случае удовлетворительной оценки.