

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**

**Директор по цифровизации  
образования**

**Д.И. Гриц**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Программирование на Python
<b>по направлению:</b>	Биотехнология
<b>профиль подготовки:</b>	Системная и синтетическая биология Физтех-школа Биологической и Медицинской Физики центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
<b>курс:</b>	1
<b>квалификация:</b>	магистр

Семестры, формы промежуточной аттестации:

1 (осенний) - Зачет  
2 (весенний) - Зачет

Аудиторных часов: 90 всего, в том числе:

лекции: 30 час.  
семинары: 60 час.  
лабораторные занятия: 0 час.

Самостоятельная работа: 180 час.

Всего часов: 270, всего зач. ед.: 6

Количество контрольных работ, заданий: 2

Программу составили:

Р.С. Кулиев, старший преподаватель  
А.А. Чистов, ассистент  
Д.М. Максимова, преподаватель  
Г.К. Тарасенко, преподаватель  
А.А. Бузулуцкая, методист  
Ж.И. Зубцова, канд. физ.-мат. наук, эксперт

Программа обсуждена на заседании центра дополнительного, дополнительного профессионального и онлайн-образования "Пуск" 28.01.2025

## Аннотация

Дисциплина состоит из 4 модулей:

Модуль 1. Основы программирования на Python

Модуль 2. Объектно-ориентированное программирование, графический интерфейс и основы работы с базами данных в Python

Модуль 3. Создание Web-сервисов на Python

Модуль 4. Анализ данных в Python.

По итогам обучения обучающийся будет способен формализовать и алгоритмизировать поставленную задачу, написать программный код с использованием языков программирования, оформить код в соответствии с установленными требованиями.

### 1. Цели и задачи

#### Цель дисциплины

- формирование/совершенствование компетенций в области решения профессиональных задач по программированию с использованием языка Python, применения шаблонов проектирования на Python, работы с Python библиотеками, применения объектно-ориентированного и функционального программирования.

#### Задачи дисциплины

- научиться использовать базовые типы и конструкции языка программирования Python;
- сформировать умение работать со стандартными структурами данных в Python, писать функции на Python, применять функциональные особенности языка, работать с файлами с помощью языка Python;
- научиться применять механизмы наследования, создавать классы и работать с ними, обрабатывать исключения;
- сформировать умение искать и исправлять ошибки в программе на Python, тестировать программы на Python;
- научиться писать многопоточный код на Python, писать асинхронный код на Python, работать с сетью, создать своё серверное сетевое приложение;
- сформировать умение пользоваться структурным программированием, использовать библиотеку unittest;
- научиться создавать корректную иерархию классов, интерпретировать UML-диаграммы, выполнять рефакторинг существующего кода;
- сформировать умение создавать Декоратор класса, создавать адаптер для интерфейса, несовместимого с системой, реализовывать паттерн Наблюдатель;
- научиться создавать цепочку обязанностей. создавать абстрактную фабрику, создавать обработчик YAML файла;
- сформировать умение работать с библиотекой requests;
- научиться работать с регулярными выражениями из Python, выполнять сложный поиск и замену при помощи регулярных выражений;
- сформировать умение извлекать и изменять данные при помощи модуля BeautifulSoup, использовать API для получения данных со сторонних сайтов;
- научиться создавать и изменять базы данных и таблицы в MySQL, получать данные из баз и таблиц в MySQL;
- сформировать умение создавать приложение на Django, работать с Django-шаблонизатором, работать с базой данных при помощи Django ORM;
- научиться отправлять данные из браузера, валидировать данные на клиентской стороне, валидировать данные на серверной стороне, проводить аутентификацию и авторизацию при помощи Django;
- сформировать умение создавать чат-бота на базе Telegram, работать с системой Git, раскладывать проект на облачный хостинг Heroku;
- научиться применять инструменты библиотеки NumPy, применять инструменты библиотеки SciPy, применять инструменты библиотеки Pandas для работы с данными;
- сформировать умение визуализировать данные при помощи инструментов Python, применить на практике инструменты Python для работы со статистическим анализом;

- научиться применять на практике линейную регрессию, применять на практике кросс-валидацию, оценивать качества моделей, обучать на практике ансамблевые модели;
- сформировать умение применять на практике методы кластеризации, применять на практике методы понижения размерности. создавать рекомендательную сеть;
- научиться умение реализовывать перцептрон, реализовывать свою нейронную сеть.

## 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной задачи	УК-3.1 Организует и координирует работу участников проекта, способствует конструктивному преодолению возникающих разногласий и конфликтов
	УК-3.4 Способен планировать командную работу, распределять поручения членам команды, организовать обсуждение разных идей и мнений
ОПК-2 Имеет представление об актуальных проблемах науки и техники в области своей профессиональной деятельности, способен на научном языке формулировать профессиональные задачи	ОПК-2.2 Способен оценивать актуальность исследований в области своей профессиональной деятельности и их практическую значимость

## 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- принципы организации кода;
- понятия коллекций и функций в Python;
- назначение функций в языках программирования;
- сетевые термины (сокеты, клиент, сервер);
- первичных ключей, типов данных атрибутов умеет строить дерево решений для выбора СУБД;
- принципы проектирования SOLID;
- смысл принципов проектирования SOLID;
- правильную структуру документа;
- номинальные, порядковые, интервальные шкалы и шкалу отношений;
- отличия разных типов шкал;
- категориальные и количественные переменные;
- меры распределения переменных;
- что такое частота распределения;
- что такое нормальное распределение и как его интерпретировать;
- свойства нормального распределения;
- правило трех сигм;
- дискретное, непрерывное и равномерное распределение;
- базовые типы и конструкции Python;
- роль коллекций и функций в программировании на Python;
- принцип работы клиент-серверной архитектуры;
- принципы проектирования БД, применяемые в работе программ и приложений;
- методы модуля BS;
- инструменты для обработки и анализа данных;
- меры разброса;
- что такое асимметрия и эксцесс;
- как посчитать точечную оценку и интерпретировать ее;
- как рассчитать доверительный интервал и интерпретировать его;
- центральную предельную теорему;
- понятия классов и объектов в Python, понимает их взаимосвязь;
- основные паттерны ООП;
- в каких ситуациях применять основные паттерны ООП;
- что такое плейсхолдеры;
- правила использования плейсхолдеров;
- понятие наследования;
- механизмы наследования и его роль в программировании на Python;
- библиотеки Python для обработки данных;
- термины нормализации;
- основные нормальные формы: 1НФ, 2НФ, 3НФ, НФБК прочие нормальные формы: 4NF, 5NF, DKNF, 6NF;
- основные протоколы транспортного уровня;
- как собирать данные из интернета;
- какие существуют способы для извлечения информации из интернета;
- что такое URL, view, шаблоны;
- отличия разных расширений данных;
- выборку и генеральную совокупность;
- где находится каталог библиотек;
- понятия тестирования и отладки;
- способы синхронизации потоков;
- особенности работы с глобальным шлюзом GIL;
- как функционирует взаимодействие в интернете;
- что такое WEB-API;
- как устроена генерация ответа на HTTP-запрос;
- существующие источники данных;
- несколько встроенных функций языка Python;
- понятие особых методов классов;
- роль особых классов в программировании на Python;
- принцип работы асинхронного взаимодействия;
- как должно выглядеть представление данных в табличной форме;
- свойства таблицы стилей QSS и умеет их применять;
- что такое блоки;
- особенности объектно-ориентированной модели в Python;
- механизм формирования исключений;

уметь:

- использовать базовые типы и конструкции Python для написания простых программ;
- сопоставить и выбрать необходимую структуру данных для конкретной практической задачи;
- составить иерархию классов и описать их методы для конкретной практической задачи;
- выделять в задаче на естественном языке необходимость применения многопоточности;
- типы структур данных, операций над данными;
- составлять описание требований к будущему программному обеспечению;
- пользоваться библиотекой requests;
- загружать данные с расширениями csv., raw., xls.,xlsx., mat., xml., json;
- определить какие переменные принадлежат к какому типу шкалы;
- отличить моду, среднее, медиану;
- найти / вычислить моду, среднее, медиану;
- строить функции и интерпретировать полученный график;
- решать задачи по построению разных видов диаграмм (точечная диаграмма, столбиковая диаграмма, диаграмма соотношения), различать нормальное распределение, определять и удалять выбросы;
- выделять в задаче на естественном языке необходимость применения базовых конструкций языка;
- определить и спроектировать необходимые классы и методы классов для конкретной предметной области;
- создать программу, использующую несколько потоков, для конкретной практической задачи;
- запускать анимации;
- работать с кривой плавности;
- делать анимацию для нескольких объектов;
- с помощью библиотеки requests извлекать информацию из интернета;
- вычислить дисперсию, стандартное отклонение;
- интерпретировать результаты вычислений;
- интерпретировать графики;
- вычислять проценти и квантили;
- решать задачи на комбинаторику, сложение и произведение вероятностей, с зависимыми событиями, случайными событиями, объяснять формулу Байеса, формулу Бернулли, решать задачи на условную вероятность, задачи на математическое ожидание, дисперсию и стандартное отклонение случайной величины, воспроизводить статистическое определение вероятности;
- выполнять проектную деятельность;
- построить график для демонстрации;
- написать простой код на Python;
- написать собственную функцию на языке Python;
- использовать магические методы в написании собственных программ на Python;
- писать многопоточных код на Python;
- таблицу в БД, создавать БД через консоль, удалять таблицу из БД, выгружать БД в SQL формате, отображать все таблицы в БД, отображать структуру БД;
- рисовать средствами QPainter;
- рисовать средствами QCanvas;
- составлять список книг в каталоге при помощи URL;
- определять и заменять пропущенные значения, значения типа NULL;
- создавать вектор;
- выполнять операции сложения и вычитания векторов;
- выполнять операции умножения и деления векторов;
- выполнять операцию векторное точечное произведение;
- вычислять норму векторов;
- вычислять расстояние между векторами;
- вычислять скалярное произведение и угол между векторами;
- создать программу по описанию задачи на естественном языке с использованием коллекций языка Python;
- использовать конструкции для генерации исключений на Python;
- добавлять записи;
- использовать паттерны, стандартизировать код;
- пользоваться модулем для подгрузки данных из интернета для решения собственных задач;
- выявлять данные, которым можно доверять;
- выгружать данные;
- определять правильную (достаточную) выборку;
- интерпретировать результаты выборки и соотносить их с результатами генеральной совокупности;
- установить интерпретатор Python себе на компьютер;

владеть:

- структурным программированием, библиотекой unittest;
- библиотекой requests;
- Django-шаблонизатором;
- системой Git;
- инструментами библиотеки NumPy, инструментами библиотеки SciPy, инструментами библиотеки Pandas для работы с данными.

#### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

##### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Основы программирования на Python	15	30		90
2	Объектно-ориентированное программирование, графический интерфейс и основы работы с базами данных в Python	5	10		30
3	Создание Web-сервисов на Python	5	10		30
4	Анализ данных в Python	5	10		30
Итого часов		30	60		180
Подготовка к экзамену		0 час.			
Общая трудоёмкость		270 час., 6 зач.ед.			

##### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 1 (Осенний)

###### 1. Основы программирования на Python

###### 1.1. Введение в программирование на Python

###### 1.1.1. Введение в программирование на Python

Лекция

Введение в программирование на Python

О языке Python

Установка интерпретатора Python на Windows

Установка интерпретатора Python на Linux

Работа в IDE PyCharm. Первая программа

Введение в Python

Ввод и вывод данных

Примеры простейших программ

Итоги занятия

Практическая работа

Выполнение заданий по теме лекции.

Самостоятельная работа

Дополнительные материалы

Тест на самопроверку по уроку

Задания на программирование

1.1.2. Типы данных. Конструкции языка

Лекция

Числовые типы и операции над ними

Строковый тип данных

Логический тип данных

Условный оператор

Цикл while

Цикл for

Операторы continue и break

Практическая работа

Выполнение заданий по теме лекции

Самостоятельная работа

Дополнительные материалы

Тест на самопроверку по уроку

Задания на программирование

Тест на оценку

1.2. Структуры данных и функции

1.2.1. Коллекции

Лекция

Встроенные структуры данных

Множества. Операции с множествами

Строки. Индексация строк

Списки

Методы split() и join(). Списочные выражения

Кортежи

Словари

Примеры работы с множествами

Примеры работы со строками

Примеры работы со списками

Примеры работы с методами split() и join()

Практическая работа

Выполнение заданий по теме лекции

Самостоятельная работа

Дополнительные материалы

Тест на самопроверку по уроку

Задания на программирование

1.2.2. Функции. Работа с файлами

Лекция

Именные функции, инструкция def

Возврат значений из функции

Множественное присваивание, упаковка и распаковка аргументов

Аргументы по умолчанию и именованные аргументы

Инструкция pass(). Согласованность аргументов

Функция как объект. Функции высших порядков

Лямбда-функция

Принципы работы с файлами на Python

Разбор задач на работу с файлами

Правила записи кода PEP 8

Практическая работа

Выполнение заданий по теме лекции

Самостоятельная работа

Дополнительные материалы

Тест на самопроверку по уроку



Задания на программирование

Тест на оценку

### 1.3. Объектно-ориентированное программирование

#### 1.3.1. Классы и объекты

Лекция

Введение в объектно-ориентированное программирование

Классы и экземпляры классов

Методы

Пример рефакторинга программы на ООП

Инкапсуляция

Полиморфизм

Практическая работа

Выполнение заданий по теме лекции

Самостоятельная работа

Дополнительные материалы

Тест на самопроверку по уроку

Задания на программирование

#### 1.3.2. Наследование

Лекция

Наследование классов

Особенности объектной модели в Python

Элементы статической типизации. Абстрактные классы и протоколы

Множественное наследование

Проблемы, связанные с наследованием

Композиция классов

Практические рекомендации

Практическая работа

Выполнение заданий по теме лекции

Самостоятельная работа

Дополнительные материалы

Тест на самопроверку по уроку

Задания на программирование

Тест на оценку

### 1.4. Углубленный Python

#### 1.4.1. Специальные методы классов. Механизм работы классов

Лекция

Специальные методы классов

Хеширование

Специальные атрибуты

Перегрузка операторов

Коллекции и итераторы

Контекстные менеджеры

Callable-объекты и декораторы

Построитель классов данных dataclass

Практическая работа

Выполнение заданий по теме лекции

Самостоятельная работа

Дополнительные материалы

Тест на самопроверку по уроку

Задания на программирование

#### 1.4.2. Работа с ошибками

Лекция

Обработка ошибок в программировании

Две основные стратегии обработки ошибок

Синтаксис обработки ошибок  
Обработка исключений и производительность  
Генерация исключений  
Инструкция `assert`  
Классы исключений  
Создание пользовательских исключений  
Практика работы с исключениями  
Практическая работа  
Выполнение заданий по теме лекции  
Самостоятельная работа  
Дополнительные материалы  
Тест на самопроверку по уроку  
Задания на программирование  
1.4.3. Установка внешних библиотек. Работа с Git  
Лекция  
Инструкция `import`  
Модули стандартной библиотеки  
Создание своего модуля на Python  
Создание виртуального окружения  
Установка внешних библиотек Python  
Параллельная установка версий интерпретатора  
Git. Работа с распределёнными системами управления версиями  
Практическая работа  
Выполнение заданий по теме лекции  
Самостоятельная работа  
Тест на самопроверку по уроку  
Задания на программирование  
Тест на оценку  
1.5. Работа с сетью  
1.5.1. Работа с сетью. Сокеты  
Лекция  
Сетевые протоколы  
Сокеты, программа клиент-сервер  
Таймауты и обработка сетевых ошибок  
Одновременная обработка нескольких соединений  
Простой HTTP-сервер  
Основные библиотеки для работы с HTTP  
Практическая работа  
Выполнение заданий по теме лекции  
Самостоятельная работа  
Дополнительные материалы  
Тест на самопроверку по уроку  
Задания на программирование  
1.6. Тестирование кода  
1.6.1. Контроль качества программного кода  
Лекция  
Обеспечение качества и тестирование ПО  
Инструменты статического анализа кода  
Инструменты тестирования  
Использование фикстур и мок-объектов в `pytest`  
Покрывание кода тестами. Плагин `pytest-cov`  
Практическая работа  
Выполнение заданий по теме лекции  
Самостоятельная работа

Дополнительные материалы  
Тест на самопроверку по уроку  
Задания на программирование  
1.7. Бонусные материалы для самостоятельного изучения  
1.7.1. Процессы и потоки  
Подходы к параллельному исполнению кода  
Создание потоков  
Создание процессов  
Синхронизация потоков  
Очереди и поддержание пула потоков  
Блокировка GIL в Python  
Использование универсального интерфейса `concurrent.futures`  
Итоги занятия  
Тест на самопроверку по уроку  
Тест на оценку  
1.7.2. Асинхронное программирование  
Асинхронное программирование как альтернатива потокам  
Введение в генераторы  
Генераторы как сопрограммы  
Синтаксис `async`, `await` и `awaitable`-объекты  
Модуль `asyncio`. основные функции  
Модуль `asyncio`. работа с сетью  
Модуль `asyncio`. работа с потоками и процессами  
Асинхронная загрузка данных с помощью библиотеки `httpx`  
Итоги занятия  
Тест на самопроверку по уроку  
Задания на программирование

## Семестр: 2 (Весенний)

2. Объектно-ориентированное программирование, графический интерфейс и основы работы с базами данных в Python

2.1. Работа с базами данных  
2.1.1. Введение в базы данных. Знакомство с SQL  
Лекция  
Введение в базы данных  
Реляционные базы данных СУБД  
SQL. Создание базы данных  
Основные операции с таблицами  
Практическая работа  
Выполнение заданий по теме лекции  
Самостоятельная работа  
Дополнительные материалы  
Данные для практической работы `supermarket_sales`  
Тест на самопроверку по уроку  
2.1.2. Проектирование базы данных  
Лекция  
Сложные запросы `SELECT`, объединение таблиц, подзапросы, группировка  
Функции SQL  
Нормализация и тестирование БД  
Принципы проектирования БД  
Модуль `sqlite3` языка Python  
Практическая работа  
Задание на работу с БД

Самостоятельная работа

Дополнительные материалы

Тест на самопроверку по уроку

Задания на программирование

Тест на оценку

## 2.2. Паттерны ООП на Python для разработки приложения

Лекция

Качество кода

SOLID-принципы

Порождающие паттерны проектирования

Структурные паттерны

Поведенческие паттерны

Практическая работа

Выполнение заданий по теме лекции

Самостоятельная работа

Дополнительные материалы

Тест на самопроверку по уроку

Задания на программирование

Тест на оценку

## 2.3. Общее представление о WEB

Основы организации компьютерных сетей и модель TCP/IP

Популярные протоколы, используемые в WEB-разработке

Библиотека requests

Практика по requests

Тест на оценку

## 2.4. Сбор данных со сторонних сайтов

Введение в обработку данных

Поиск с помощью регулярных выражений

Символьные классы и квантификаторы

Сложный поиск и замена

Тест на оценку

Задание на программирование по теме "Сбор данных со сторонних сайтов"

## 2.5. Beautiful Soup и работа с API

Обзор методов модуля Beautiful Soup

Сложный поиск и изменение с Beautiful Soup

Взаимодействие с API

Практический пример работы с API

Конспект

Задание на программирование по теме "Beautiful Soup и работа с API"

## 3. Создание Web-сервисов на Python

### 3. Разработка WEB-приложения

#### 3.1. Python и WEB-фреймворки

Обзор существующих web-фреймворков Python

Архитектура web-приложения

Роутинг и устройство view

Установка и запуск простейшего приложения на Django

Шаблонизация Django

Задание проекта

#### 3.2. Взаимодействие с базами данных с помощью Django

Использование различных баз данных для разработки web-приложений

Работа с Django ORM

Реализация Django ORM в проекте

Задание проекта

### 3.3. Основы Frontend-разработки

Основы HTML-вёрстки

Подключаем CSS

JavaScript и его применение

Дизайн-системы

Настройка проекта с использованием дизайн-системы

Задание проекта

### 3.4. Продвинутое возможности Django

Автоматическое тестирование и Django

Django и Telegram-боты

Контейнеризация

Асинхронные запросы (Celery)

Задание проекта

## 4. Разработка серверных приложений с помощью FastAPI

### 4.1. Введение в FastAPI и REST-сервисы

Введение в FastAPI. Концепция REST. Установка окружения

Первое CRUD-приложение на FastAPI

Документирование API

Тест на самопроверку по уроку

Задание на программирование

Задание проекта

### 4.2. Работа с данными в FastAPI

Подключение базы данных к FastAPI

Асинхронные операции

Миграции базы данных

Тест на самопроверку по уроку

Задание на программирование

Задание проекта

### 4.3. Безопасность и тестирование приложений на FastAPI

Обеспечение безопасности в FastAPI

Хэширование паролей

Защита токенов доступа от подделки

Тестирование приложений на FastAPI

Тест на самопроверку по уроку

Задание на программирование

Задание проекта

### 4.4. Контейнеризация FastAPI-приложения

Контейнеризация FastAPI-приложения и СУБД с помощью Docker Compose

Тест на самопроверку по уроку

Задание на программирование

Задание проекта

## 4. Анализ данных в Python

### 4.1. Введение в анализ данных

Виды данных

Предобработка данных

Инструменты работы с данными

Массивы

Векторы

Библиотека Pandas

df Титаник

Описание переменных df Титаник

Тест на самопроверку по теме " Введение в анализ данных"

Задания на программирование на самопроверку по теме " Введение в анализ данных"

Тест на оценку по теме " Введение в анализ данных"

Задание на оценку по теме " Введение в анализ данных"

#### 4.2. Исследование данных с Python

##### 4.2.1. Описательные статистики. Статистика вывода

Описательные статистики

Выборка и генеральная совокупность

Распределения

Оценки

Тестирование гипотез

Тест на самопроверку по теме "Описательные статистики.Статистика вывода"

Задания на самопроверку на программирование по теме "Описательные статистики.Статистика вывода"

##### 4.2.2. Методы визуализации

Методы визуализации

Библиотека Matplotlib

Библиотека Seaborn

Дополнительные материалы

Тест на самопроверку по теме "Методы визуализации

Задания на программирование на самопроверку по теме "Методы визуализации

Тест на оценку по теме "Методы визуализации"

Задания на оценку по теме "Методы визуализации"

#### 4.3. Проект "Анализ данных в Python"

##### 4.4. Бонус-модуль. Введение в машинное обучение

###### 4.4.1. Теория вероятности

Основы теории вероятности

Тест на самопроверку

Задания на программирование

###### 4.4.2. Основы машинного обучения

Машинное обучение с учителем

Введение в машинное обучение

Линейные модели

Измерение качества модели

Ансамблевые модели

Машинное обучение с учителем

Тест на самопроверку

Задания на программирование

Машинное обучение без учителя

Обучение без учителя. Кластеризация

Методы понижения размерности

Рекомендательные системы

Машинное обучение без учителя (unsupervised learning.ipynb)

Тест на самопроверку

Задания на программирование

###### 4.4.3. Основы нейронных сетей

Основы нейронных сетей

Архитектуры нейронных сетей

Основы нейронных сетей (neural networks.ipynb)

Тест для самопроверки

Задания для самопроверки

Итоговый тест по модулю

Итоговое практическое задание по модулю

#### 4.5. Бонус-модуль. Построение графических интерфейсов

#### 4.5.1. Этап I. Учимся проектировать приложение

Разработка ТЗ

Архитектура проекта

Средства разработки

Структура проекта

Пишем репозиторий

Пишем модели

Задание проекта

#### 4.5.2. Этап II. Основы графического интерфейса приложения

Введение в программирование GUI

Основы PyQt.

Виджеты и сигналы.

Раскладки (layouts)

Программирование виджетов в ООП стиле

Задание проекта

#### 4.5.3. Этап III. Продвинутое возможности библиотеки PySide

Окна в PyQt

Сигналы и события

Многопоточное программирование в PyQt

Тестирование PyQt-приложений

Интегрирование GUI в архитектуру MVP

Задание проекта

#### 4.5.4. Этап IV. Кастомизация интерфейса приложения

Кастомизация окна

Создание виджетов

Определение стилей

Анимации

Задание проекта

### 5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Система дистанционного обучения:

Обучающемуся необходимо наличие доступа в сеть интернет, компьютер.

Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.

### 6. Перечень рекомендуемой литературы

Основная литература

1. Программирование на языке высокого уровня Python, учебное пособие для вузов / Д. Ю. Федоров. — Москва, Юрайт, 2020.— URL: <https://urait.ru/bcode/454100> (дата обращения: 15.12.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)
2. Python для сложных задач: наука о данных и машинное обучение [Текст], [учеб. пособие для вузов] / Дж. Вандер Плас ; [пер. с англ. И. Пальти]. -СПб., Питер, 2018
3. Python и машинное обучение, Электронная версия печатной публикации / С. Рашка. — Москва, ДМК Пресс, 2017

Литература из средств кафедры:

1. Федоров, Д. Основы программирования на примере языка Python: учебное пособие / Д. Федоров. – СПб.: Питер, 2018.
2. Свейгарт, Э. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих / Э. Свейгарт. – М.: Диалектика, 2021.
3. Мартин, Р. Чистый код: создание, анализ и рефакторинг. – СПб.: Питер, 2020.
4. Саммерфилд, М. Python на практике. – М.: ДМК Пресс, 2014.
5. Персиваль, Г., Грегори, Б. Паттерны разработки на Python. – СПб.: Питер, 2022.
6. Прохоренок, Н. А., Дронов, В. А. Python 3 и PyQt6. Разработка приложений. – СПб.: БХВ-Петербург, 2023.
7. Лутц, М. Изучаем Python: авторитетный курс объектно-ориентированного программирования / пер. с англ. Ю. Н. Артеменко. – 5-е изд. – М.: Диалектика, 2020.
8. Любанович, Билл. FastAPI: веб-разработка на Python. — Sprint Book, 2024.

Дополнительная литература

1. Программирование на Python 3 : Подробное руководство [Текст] = Programming in Python 3 : [учеб. пособие для вузов] / М. Саммерфилд; пер. с англ. А. Киселева. — СПб : Символ-Плюс, 2015. — 608 с.
2. Python на практике [Текст], создание качественных программ с использованием параллелизма, библиотек и паттернов/М. Саммерфилд, -М, ДМК Пресс, 2014
3. Изучаем Python / Эрик Мэтиз, Санкт-Петербург, Питер, 2020
4. Python и анализ данных, Первичная обработка данных с применением pandas, NumPy и IPython / У. Маккини. — Москва, ДМК Пресс, 2020.— URL: <https://e.lanbook.com/book/131721> (дата обращения: 26.01.2021). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

Дополнительная литература из средств кафедры:

1. Бэрри, П. Изучаем программирование на Python. — М.: Питер, 2017.
2. Савельев, В. Статистика и коттики. — М.: Питер, 2018.
3. Бослав, С. Статистика для всех. — М.: Питер, 2015.
4. Хамидуллин, Р. Я. Теория вероятностей и математическая статистика. — М.: Питер, 2020.

## **7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)**

Think Python [Электронный ресурс]. – Режим доступа:

<https://greenteapress.com/wp/think-python-2e/>

Automate the Boring Stuff with Python [Электронный ресурс]. – Режим доступа:

<https://automatetheboringstuff.com/>

Dive Into Python 3 [Электронный ресурс]. – Режим доступа:

<http://diveintopython3.problemsolving.io/>

Problem Solving with Algorithms and Data Structures using Python [Электронный ресурс]. – Режим доступа: <https://runestone.academy/runestone/static/pythonds/index.html>

Chitlur, S. A Byte of Python [Электронный ресурс]. – Режим доступа:

<https://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf>, 2020

Percival, H. J. W. Test-Driven Web Development with Python [Электронный ресурс]. – Режим доступа: <http://www.obeythetestinggoat.com/pages/book.html#toc>

The Hitchhiker's Guide to Python: Best Practices for Development / K. Reitz, T. Schlusser

[Электронный ресурс]. – O'Reilly, 2016. – Режим доступа: [docs.python-guide.org](https://docs.python-guide.org)

Учебная статья: Errors and Exceptions [Электронный ресурс]. – Режим доступа:

<https://docs.python.org/3/tutorial/errors.html>

Книга издательства O'Reilly [Электронный ресурс]. – Режим доступа:

<https://docs.python-guide.org/>

Официальный сайт Python - <https://www.python.org/>

Введение в анализ данных с помощью Pandas - <https://habr.com/ru/post/196980/>

Среда разработки IDE PyCharm - <https://www.jetbrains.com/pycharm/>

PEP 8 – Style Guide for Python Code - <https://peps.python.org/pep-0008/>



Что такое TCP/IP и зачем они нужны: <https://thecode.media/tcp-ip/>  
TCP и UDP — в чем разница: <https://wiki.merionet.ru/seti/23/tcp-i-udp-v-chem-raznica/>  
User-Agent: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent>  
Тестер регулярных выражений: <https://regex101.com/>  
HTTP Request & Response Service: <http://httpbin.org>  
Введение в анализ данных с помощью Pandas - <https://habr.com/ru/post/196980/>  
Открытые датасеты - <https://www.kaggle.com/datasets>  
Федеральная служба государственной статистики - <https://rosstat.gov.ru/>  
Официальная документация (англ.): <https://docs.python.org/3/>  
Документация по исключениям: <https://docs.python.org/3/library/exceptions.html>  
Модель GitFlow: <https://nvie.com/posts/a-successful-git-branching-model/>  
Современный подход к версионированию для CI/CD:  
<https://www.atlassian.com/continuous-delivery/continuous-integration/trunk-based-development>  
Документация Postgres про сравнение строк -  
<https://postgrespro.ru/docs/postgrespro/9.5/functions-matching>  
Документация Postgres про другие функции работы со строками -  
<https://postgrespro.ru/docs/postgrespro/9.5/functions-string>  
Интерактивный учебник по SQL - <http://www.sql-tutorial.ru/ru/content.html>  
<https://www.sqlitetutorial.net/>  
<https://docs.python.org/3/library/sqlite3.html> - документация по модулю sqlite3  
<https://doc.qt.io/qtforpython/index.html> – документация  
<https://www.pythonguis.com/pyside6/> – учебник  
<https://docs.python-requests.org/en/latest/index.html>  
<https://www.useragentstring.com/>  
<https://docs.python.org/3/library/re.html>  
<https://www.crummy.com/software/BeautifulSoup/bs4/doc.ru/bs4ru.html>  
<https://world.openfoodfacts.org/>  
<https://publicapis.io/>  
<https://www.djangoproject.com/>  
<https://awesomedjango.org/>  
Игровой тренажер по Python: [py.checkio.org](http://py.checkio.org)  
NumPy documentation — NumPy v1.24 Manual <https://numpy.org/doc/stable/>  
MP3 to Numpy and back | Kaggle <https://www.kaggle.com/code/gabrielmilan/mp3-to-numpy-and-back>  
User Guide — pandas 2.0.2 documentation ([pydata.org](http://pydata.org))  
Pandas Pivot Table Explained - Practical Business Python ([pbpython.com](http://pbpython.com))  
Моем датасет: руководство по очистке данных в Python ([proglab.io](http://proglab.io))  
pandas documentation — pandas 1.5.3 documentation  
Наглядная шпаргалка по операциям с DataFrame в pandas для data wrangling и не только ([tproger.ru](http://tproger.ru))  
Statistical functions (scipy.stats) — SciPy v1.10.1 Manual  
Understanding Confidence Intervals | Easy Examples & Formulas ([scribbr.com](http://scribbr.com))  
6 Steps to Evaluate a Statistical Hypothesis Testing ([enago.com](http://enago.com))  
17 Statistical Hypothesis Tests in Python (Cheat Sheet) - [MachineLearningMastery.com](http://MachineLearningMastery.com)  
Matplotlib documentation &#8212; Matplotlib 3.7.1 documentation  
Animation — Matplotlib 3.7.1 documentation  
Example gallery — [seaborn 0.12.2 documentation](http://seaborn.pydata.org)

## **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

Документация Postgres про сравнение строк -  
<https://postgrespro.ru/docs/postgrespro/9.5/functions-matching>

Документация Postgres про другие функции работы со строками - <https://postgrespro.ru/docs/postgrespro/9.5/functions-string>

Тестер регулярных выражений - <https://www.regextester.com>

Интерактивный учебник по SQL - <http://www.sql-tutorial.ru/ru/content.html>

Введение в анализ данных с помощью Pandas - <https://habr.com/ru/post/196980/>

Начало работы с Power BI -

<https://docs.microsoft.com/ru-ru/power-bi/fundamentals/desktop-getting-started>

Профессиональный информационно-аналитический ресурс, посвященный машинному обучению, распознаванию образов и интеллектуальному анализу данных – <http://www.machinelearning.ru/>

Информационная система «Единое окно доступа к образовательным ресурсам» (ИС «Единое окно») – <http://window.edu.ru/>

Платформа открытое образование – <https://openedu.ru/>

## **9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Самостоятельная работа подразделяется на аудиторную и внеаудиторную. Аудиторную самостоятельную работу составляют практические задания, которые выполняются слушателями во время учебных занятий, результаты ее выполнения проверяются и оцениваются преподавателем в учебном процессе.

Внеаудиторная самостоятельная работа включает формы: изучение дополнительной литературы, подготовка итоговых проектов по модулям, подготовка проекта.

Основными критериями качества организации самостоятельной работы служит наличие контроля результатов самостоятельной работы.

Основными современными формами организации самостоятельной работы являются творческие работы и работа с информационными компьютерными технологиями.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Биотехнология
<b>профиль подготовки:</b>	Системная и синтетическая биология Физтех-школа Биологической и Медицинской Физики центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
<b>курс:</b>	<u>1</u>
<b>квалификация:</b>	магистр

Семестры, формы промежуточной аттестации:

- 1 (осенний) - Зачет
- 2 (весенний) - Зачет

**Разработчики:**

Р.С. Кулиев, старший преподаватель  
А.А. Чистов, ассистент  
Д.М. Максимова, преподаватель  
Г.К. Тарасенко, преподаватель  
А.А. Бузулуцкая, методист  
Ж.И. Зубцова, канд. физ.-мат. наук, эксперт

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-3 Способен организовывать и руководить работой команды, вырабатывая командную стратегию для достижения поставленной задачи	УК-3.1 Организует и координирует работу участников проекта, способствует конструктивному преодолению возникающих разногласий и конфликтов
	УК-3.4 Способен планировать командную работу, распределять поручения членам команды, организовать обсуждение разных идей и мнений
ОПК-2 Имеет представление об актуальных проблемах науки и техники в области своей профессиональной деятельности, способен на научном языке формулировать профессиональные задачи	ОПК-2.2 Способен оценивать актуальность исследований в области своей профессиональной деятельности и их практическую значимость

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Программирование на Python» обучающийся должен:

**знать:**

- принципы организации кода;
- понятия коллекций и функций в Python;
- назначение функций в языках программирования;
- сетевые термины (сокеты, клиент, сервер);
- первичных ключей, типов данных атрибутов умеет строить дерево решений для выбора СУБД;
- принципы проектирования SOLID;
- смысл принципов проектирования SOLID;
- правильную структуру документа;
- номинальные, порядковые, интервальные шкалы и шкалу отношений;
- отличия разных типов шкал;
- категориальные и количественные переменные;
- меры распределения переменных;
- что такое частота распределения;
- что такое нормальное распределение и как его интерпретировать;
- свойства нормального распределения;
- правило трех сигм;
- дискретное, непрерывное и равномерное распределение;
- базовые типы и конструкции Python;
- роль коллекций и функций в программировании на Python;
- принцип работы клиент-серверной архитектуры;
- принципы проектирования БД, применяемые в работе программ и приложений;
- методы модуля BS;
- инструменты для обработки и анализа данных;
- меры разброса;
- что такое асимметрия и эксцесс;
- как посчитать точечную оценку и интерпретировать ее;
- как рассчитать доверительный интервал и интерпретировать его;
- центральную предельную теорему;
- понятия классов и объектов в Python, понимает их взаимосвязь;
- основные паттерны ООП;
- в каких ситуациях применять основные паттерны ООП;
- что такое плейсхолдеры;
- правила использования плейсхолдеров;
- понятие наследования;
- механизмы наследования и его роль в программировании на Python;
- библиотеки Python для обработки данных;
- термины нормализации;
- основные нормальные формы: 1НФ, 2НФ, 3НФ, НФБК прочие нормальные формы: 4NF, 5NF, DKNF, 6NF;
- основные протоколы транспортного уровня;
- как собирать данные из интернета;
- какие существуют способы для извлечения информации из интернета;
- что такое URL, view, шаблоны;
- отличия разных расширений данных;
- выборку и генеральную совокупность;
- где находится каталог библиотек;
- понятия тестирования и отладки;
- способы синхронизации потоков;
- особенности работы с глобальным шлюзом GIL;
- как функционирует взаимодействие в интернете;
- что такое WEB-API;
- как устроена генерация ответа на HTTP-запрос;
- существующие источники данных;
- несколько встроенных функций языка Python;
- понятие особых методов классов;
- роль особых классов в программировании на Python;
- принцип работы асинхронного взаимодействия;
- как должно выглядеть представление данных в табличной форме;
- свойства таблицы стилей QSS и умеет их применять;
- что такое блоки;
- особенности объектно-ориентированной модели в Python;
- механизм формирования исключений;

**уметь:**

- использовать базовые типы и конструкции Python для написания простых программ;
- сопоставить и выбрать необходимую структуру данных для конкретной практической задачи;
- составить иерархию классов и описать их методы для конкретной практической задачи;
- выделять в задаче на естественном языке необходимость применения многопоточности;
- типы структур данных, операций над данными;
- составлять описание требований к будущему программному обеспечению;
- пользоваться библиотекой requests;
- загружать данные с расширениями csv., raw., xls.,xlsx., mat., xml., json;
- определить какие переменные принадлежат к какому типу шкалы;
- отличить моду, среднее, медиану;
- найти / вычислить моду, среднее, медиану;
- строить функции и интерпретировать полученный график;
- решать задачи по построению разных видов диаграмм (точечная диаграмма, столбиковая диаграмма, диаграмма соотношения), различать нормальное распределение, определять и удалять выбросы;
- выделять в задаче на естественном языке необходимость применения базовых конструкций языка;
- определить и спроектировать необходимые классы и методы классов для конкретной предметной области;
- создать программу, использующую несколько потоков, для конкретной практической задачи;
- запускать анимации;
- работать с кривой плавности;
- делать анимацию для нескольких объектов;
- с помощью библиотеки requests извлекать информацию из интернета;
- вычислить дисперсию, стандартное отклонение;
- интерпретировать результаты вычислений;
- интерпретировать графики;
- вычислять проценти и квантили;
- решать задачи на комбинаторику, сложение и произведение вероятностей, с зависимыми событиями, случайными событиями, объяснять формулу Байеса, формулу Бернулли, решать задачи на условную вероятность, задачи на математическое ожидание, дисперсию и стандартное отклонение случайной величины, воспроизводить статистическое определение вероятности;
- выполнять проектную деятельность;
- построить график для демонстрации;
- написать простой код на Python;
- написать собственную функцию на языке Python;
- использовать магические методы в написании собственных программ на Python;
- писать многопоточных код на Python;
- таблицу в БД, создавать БД через консоль, удалять таблицу из БД, выгружать БД в SQL формате, отображать все таблицы в БД, отображать структуру БД;
- рисовать средствами QPainter;
- рисовать средствами QCanvas;
- составлять список книг в каталоге при помощи URL;
- определять и заменять пропущенные значения, значения типа NULL;
- создавать вектор;
- выполнять операции сложения и вычитания векторов;
- выполнять операции умножения и деления векторов;
- выполнять операцию векторное точечное произведение;
- вычислять норму векторов;
- вычислять расстояние между векторами;
- вычислять скалярное произведение и угол между векторами;
- создать программу по описанию задачи на естественном языке с использованием коллекций языка Python;
- использовать конструкции для генерации исключений на Python;
- добавлять записи;
- использовать паттерны, стандартизировать код;
- пользоваться модулем для подгрузки данных из интернета для решения собственных задач;
- выявлять данные, которым можно доверять;
- выгружать данные;
- определять правильную (достаточную) выборку;
- интерпретировать результаты выборки и соотносить их с результатами генеральной совокупности;
- установить интерпретатор Python себе на компьютер;

**владеть:**

- структурным программированием, библиотекой unittest;
- библиотекой requests;
- Django-шаблонизатором;
- системой Git;
- инструментами библиотеки NumPy, инструментами библиотеки SciPy, инструментами библиотеки Pandas для работы с данными.

**3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю**

Тестовые задания

1. Какая версия языка Python является актуальной в настоящее время?

- Python 3.X
- Python 2.6.X
- Python 1.X

2. Что такое инструкция в языках программирования?

- команда для выполнения какого-либо действия
- документ с перечислением всех команд языка
- данные, которые пользователь вводит с консоли в программу
- спецсимволы, используемые для написания кода

3. Какие элементы используют для формирования структуры программы в Python? Выберите все верные варианты.

- пробелы
- двоеточие
- табуляция
- фигурные скобки
- команды языка
- знак #

4. Какая строка кода выведет на экран сумму двух переменных a и b?

a = 2

b = 4

5. Есть большой объём данных по клиентам. В ходе работы необходимо обращаться к небольшому числу столбцов. При этом критична скорость работы. Какой тип модели данных лучше выбрать?

- Колоночные (столбчатые) базы данных (Clickhouse, Vertica)
- Реляционные базы данных (SQLite, Postgres)
- Графовые базы данных
- Иерархические базы данных

6. Данные, которые нужно хранить в базе данных представимы в виде набора связанных таблиц. Какой тип базы данных будет подходящим?

- Реляционная БД
- База данных с широкими столбцами
- Графовая база данных
- Иерархическая

7. В таблице с информацией о клиентах есть поля возраст клиента, адрес клиента, номер телефона, дата регистрации. Какое из полей может использоваться в качестве первичного ключа?

- номер телефона
- возраст клиента
- дата регистрации



- адрес клиента

8. В таблице с информацией о недвижимости (estate) есть поле с этажом квартир (floor). Нужно выбрать строки с нечётным этажом. Как это можно сделать?

- `SELECT * FROM estate WHERE floor % 2 != 0;`
- `SELECT * FROM estate;`
- `SELECT * FROM estate WHERE floor is ODD;`
- `SELECT * FROM estate WHERE floor // 2 != 0;`

9. Что из перечисленного не может быть IP-адресом?

- а) 192.167.3.5
- б) 255.255.255.255
- в) 2001:0db8:0001:0000:0000:0ab9:C0A8:0102
- г) 192.256.2.2

10. Для приёма и отправки почты, как и для некоторых других действий (например, отправки HTTP-запросов), зарезервированы порты в \_\_\_\_\_ пуле.

- а) общеизвестном
- б) зарегистрированном
- в) пользовательском
- г) частном

11. Известно, что IPv4 адреса — это 4 блока чисел от 0 до 255 (т.е. всего доступно 256 вариантов). Сколько всего IPv4 адресов существует? Запишите количество цифр в этом числе

12. Даны компьютеры с двумя IP-адресами: 215.154.247.178 и 215.55.238.117. Какой должна быть маска подсети, чтобы эти компьютеры были в одной подсети?

- а) /24
- б) /8
- в) /18
- г) /215

13. Какой из нижеприведенных источников данных хорошо подходит для тренировки и отработки навыка анализ данных:

- a. GitHub
- b. Kaggle
- c. Всемирный Банк
- d. Росстат

14. Какой из приведённых форматов файлов основан на синтаксисе языка программирования?

- a. .sav
- b. .json
- c. .csv
- d. .tsv

15. Какая информация не содержится в методе `info()`:

- a. размер датасета
- b. имена колонок
- c. количество пропущенных значений
- d. все это содержится в выдаче метода

16. Что делает данный код: `df.dropna(how = 'all', inplace = True )`:

- a. удаляет все строки, где есть все пропущенные значение по колонкам и не перезаписывает исходный датафрейм

- b. удаляет все строки, где есть хотя бы одно пропущенное значение по колонкам и перезаписывает исходный датафрейм
- c. удаляет все строки, где есть хотя бы одно пропущенное значение по колонкам и не перезаписывает исходный датафрейм
- d. все выше перечисленное неверно

#### Вопрос 17

Какой декоратор используется в FastAPI для определения маршрута, который обрабатывает GET-запросы?

- A. `@app.post("/")`
- B. `@app.get("/")`
- C. `@app.put("/")`
- D. `@app.delete("/")`

#### Вопрос 18

Какую функцию выполняет файл `settings.py` в Django-проекте?

- A) Определяет маршруты (URL)
- Б) Содержит настройки проекта, включая конфигурацию базы данных и middleware
- В) Хранит логи работы приложения
- Г) Управляет шаблонами и их отображением

### 4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

#### Задание на программирование

##### 1. Приветствие

На вход программа получает строку — имя пользователя. Считайте её и выведите на экран фразу: «Привет, [имя]!».

Обратите внимание, что выводимая строка должна быть написана в точности так, как в задании. Если будет отличаться регистр букв или будут отсутствовать знаки препинания, программа засчитана не будет.

Ввод	Вывод
Валентин	Привет, Валентин!

##### 2. Вычисление значения предыдущего числа и следующего

На вход программа получает целое число  $a$ . Получите значения предыдущего числа и следующего. Выведите результат  $a-1$ ,  $a$ ,  $a+1$  по одному числу в строку.

Ввод	Результат
5	4
	5
	6

##### 3. Питомец

Егор разрабатывает приложение «Кошечки-собачки». Ему нужно считать данные о питомце, которые введены в консоль, и подтвердить их ввод путем вывода на экран.

На вход программа получает четыре параметра: имя питомца (`name`), вид животного (`animal`), порода (`breed`) и возраст (`age`).

Необходимо вывести введенную информацию в виде:

Ваш питомец: `animal name`

Порода питомца: breed  
Возраст питомца: age  
Ввод            Результат  
Кузя            Ваш питомец: кот Кузя  
кот  
мейн-кун      Порода питомца: мейн-кун  
5                Возраст питомца: 5

4. А) Создайте базу данных (БД), используя программу DB Browser for SQLite, с названием Python\_<фамилия на латинице>.

Б) Добавьте в созданную БД:

1) Таблицу Users с полями: уникальный идентификатор пользователя, имя, фамилия, возраст, город проживания, страна, дата регистрации в онлайн-магазине.

2) Таблицу Products с полями: уникальный идентификатор товара, название товара, категория товара, стоимость товара за ед. в \$.

3) Таблицу Sales с полями: уникальный идентификатор транзакции, дата совершения транзакции, уникальный идентификатор купленного товара, количество ед. товара, наличие скидки, пункт выдачи/адрес доставки товара

В) Заполните таблицы вымышленными данными (по 5-10 строк).

Г) Перейдите во вкладку SQL и напишите скрипты, чтобы выполнить следующие действия с БД:

1) В таблице Users присвойте всем пользователям город проживания Москва.

2) В таблице Users измените возраст клиента во 2-й строке таблицы на 35 лет.

3) В таблице Products добавьте новый столбец со стоимостью товара в рублях (курс можно взять с сайта ЦБ на дату выполнения ДЗ)

4) В таблице Sales выберите строки, для которых дата совершения транзакции наиболее ранняя или наиболее поздняя, т. е. строки с минимальной и максимальной датами.

5. Загрузите данные Supermarket\_sales и импортируйте их в DB Browser for SQLite с названием SupermarketSales\_<фамилия на латинице>. По ссылке можно посмотреть описание базы данных <https://www.kaggle.com/datasets/aungpyaeap/supermarket-sales>

Перейдите во вкладку SQL и напишите скрипты, чтобы выполнить следующие действия с БД:

1) Выберите записи таблицы, в которых поле “Branch” принимает значение А.

2) Выберите записи таблицы, в которых поле “Branch” принимает значение А или С.

3) Выберите записи таблицы, в которых поле “Branch” принимает значение отличное от А.

4) Выберите записи таблицы, в которых поле “Branch” принимает значение В, а поле Productline равно “Fashion accessories”.

5) Удалите записи таблицы, для которых поле “Rating” принимает значение меньше 5.

6. С помощью SQL запросов к БД из задания 2 ответьте на вопросы:

1) Какое количество транзакций совершили клиенты?

Одна строка соответствует одной транзакции. После удаления строк (п.5 задания 2) у нас остается 847 строк

2) Сколько транзакций с товарами категории «Electronic accessories» было совершено?

Запрос должен выглядеть так: `SELECT * FROM sales WHERE Productline == “Electronic accessories”`

3) Были ли покупки товара категории «Electronic accessories» за период с 3 февраля 2019 года? Если такие покупки были, то найдите их число.

4) Клиентов с каким рейтингом больше

а) 5-7

б) больше 8?

5) Сколько типов оплаты (поле Payment)?

6) Найдите стоимости (Unitprice) и категории (Productline) для товаров, купленных в количестве (Quantity) 10 шт. за одну транзакцию. Напишите число таких транзакций.

## 7. Регулярные выражения

### Данные

Для этого задания потребуется CSV-файл, скачать который можно по ссылке:

[https://gist.githubusercontent.com/Grammer/4880a2e474515b6b1c5fd4b3dc1e11aa/raw/94e99ab2d2d2d8e3fc10daf50cb7af23c517a1af/enron\\_3000.csv](https://gist.githubusercontent.com/Grammer/4880a2e474515b6b1c5fd4b3dc1e11aa/raw/94e99ab2d2d2d8e3fc10daf50cb7af23c517a1af/enron_3000.csv)

Назовите скачанный файл `enron_3000.csv`

Источник данных: Enron Email Dataset

### Задание

Считайте полученный файл построчно. Из каждой строки извлеките все email-адреса, которые в ней употреблялись, и сделайте большой частотный словарь, где ключом будет email-адрес, а значением — количество употреблений этого email-адреса во всём файле.

Выведите 20 самых частотных email-адресов с их частотностью, разделённых знаком табуляции.

Обратите внимание: email-адреса расположены не только в полях адресата и адресанта, иногда они встречаются и в других полях

### Пример вывода

```
top1email@enron.com 100500
```

```
top2email@enron.com 100499
```

```
top3email@enron.com 100498
```

## 8. Вам дан датафрейм:

```
import numpy as np
```

```
import pandas as pd
```

```
df = pd.DataFrame(np.nan, index=[0, 1, 2, 3], columns=['A', 'B'])
```

```
df.loc[0,"A"] = 1
```

```
df.loc[1,"A"] = 2
```

```
df.loc[2,"A"] = 3
```

```
df.loc[3,"A"] = 4
```

```
df.loc[0,"B"] = 5
```

```
df.loc[1,"B"] = 6
```

```
df.loc[2,"B"] = 7
```

```
df
```

```
   A  B
0  1.0  5.0
1  2.0  6.0
2  3.0  7.0
3  4.0 NaN
```

Выполните задания.

### Задание 1

Переименуйте названия строк в последовательность чисел от 1 до 4, используя соответствующий метод pandas.

### Задание 2

Определите какой тип данных у второй колонки датафрейма и выведите его на экран.

### Задание 3

Замените пропущенное значение в датафрейме числом 6 и пересохраните датафрейм.

## Критерии оценивания

Максимальная сумма, которую можно набрать, успешно выполнив все контрольные мероприятия, составляет 100 баллов. Для получения положительной оценки «зачтено» необходимо набрать не менее 30 баллов.

Оценка «зачтено» выставляется студенту, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в зачет представлены ниже:

Модуль	Вклад в зачет, %
Оценка за модуль 1	20
Оценка за модуль 2	20
Оценка за модуль 3	20
Оценка за модуль 4	20
Зачет	20

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

Во время проведения зачета обучающиеся могут пользоваться программой дисциплины.