

МИНОБРНАУКИ РОССИИ



Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
ФГАОУ ВО «РГГУ»

**ИНСТИТУТ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ НАУК
СОЦИОЛОГИЧЕСКИЙ ФАКУЛЬТЕТ**

Кафедра теории и истории социологии

ПРОГРАММИРОВАНИЕ НА РУТНОН

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Социология (академический бакалавр) 39.03.01

Цифровая социология и социальная диагностика

Уровень квалификации выпускника (бакалавр)

Форма обучения (очная)

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2026

«Программирование на Python»

Рабочая программа дисциплины

Составитель:

Кандидат экономических наук, доцент *А.Ю. Журавлев*

Ответственный редактор

Доктор философских наук, проф. Ж.Т. Тощенко

УТВЕРЖДЕНО

Протокол заседания кафедры
протокол № 6 от 16.12. 2025

ОГЛАВЛЕНИЕ

1. Пояснительная записка	4
1.1 Цель и задачи дисциплины.....	4
1.2 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций.....	4
2. Структура дисциплины	9
3. Содержание дисциплины	10
4. Образовательные технологии	16
5. Оценка планируемых результатов обучения	20
5.1 Система оценивания.....	20
5.2 Критерии выставления оценки по дисциплине.....	21
5.3 Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине.....	22
6. Учебно-методическое и информационное обеспечение дисциплины	34
6.1 Список источников и литературы.....	37
Основная литература.....	37
Дополнительная литература.....	37
6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет».....	39
7. Материально-техническое обеспечение дисциплины	40
8. Обеспечение образовательного процесса для лиц	40
с ограниченными возможностями здоровья и инвалидов.....	40
9. Методические материалы	42
9.1 Планы практических занятий.....	42
9.2. Методические указания.....	52
Приложение 1. Аннотация рабочей программы дисциплины	54

1. Пояснительная записка

1.1 Цель и задачи дисциплины

Цель дисциплины: формирование у студентов системного представления о фундаментальных основах алгоритмизации и программирования на языке Python как универсальном инструменте решения широкого круга профессиональных задач цифрового социолога, а также развитие устойчивых практических навыков разработки программных средств для сбора, обработки, анализа и визуализации социальной информации, включая первичное знакомство с библиотеками для научных вычислений и машинного обучения.

Задачи дисциплины:

1. Сформировать понимание роли и места языков программирования в структуре современного инструментария цифрового социолога, раскрыть значение Python как лидирующего языка в области анализа данных и искусственного интеллекта, обосновать преимущества автоматизации обработки социальной информации перед ручными методами.
2. Ознакомить студентов с фундаментальными концепциями программирования: раскрыть содержание понятий «алгоритм», «переменная», «тип данных», «оператор», «выражение», объяснить принципы структурного и модульного программирования, сформировать алгоритмическое мышление как основу для решения исследовательских задач.
3. Обучить студентов синтаксису и семантике базовых конструкций языка Python: освоить работу с числовыми и строковыми типами данных, списками, кортежами, словарями и множествами; привить навыки ввода и вывода данных, форматирования строк, использования арифметических и логических операций.
4. Сформировать умение управлять ходом выполнения программы с помощью условных операторов (if, elif, else) и циклов (while, for), а также применять эти конструкции для обработки массивов социологических данных (например, сегментация респондентов, итеративный подсчет частот).
5. Развить навыки структурирования кода с помощью функций: обучить студентов созданию собственных функций, передаче аргументов, возврату значений, объяснить принципы локализации переменных (области видимости) и повторного использования кода для типовых задач анализа.
6. Обучить студентов основам работы с файловой системой: сформировать навыки чтения и записи текстовых файлов, работы с файлами формата CSV как основным

источником социологических данных, освоить методы импорта и экспорта данных для последующей обработки.

7. Привить навыки использования специализированных библиотек Python для научных вычислений и анализа данных: дать общее представление о возможностях библиотек NumPy (работа с многомерными массивами) и Pandas (обработка табличных данных, фильтрация, группировка, агрегация).
8. Сформировать начальные представления о визуализации данных: обучить студентов построению базовых графиков (гистограммы, ящики с усами, точечные диаграммы) с помощью библиотеки Matplotlib и Seaborn для наглядного представления результатов социологических исследований.
9. Дать первичное представление об основах парсинга веб-страниц: познакомить с библиотеками Requests и BeautifulSoup, сформировать понимание принципов сбора данных из открытых интернет-источников для пополнения эмпирической базы исследования.
10. Воспитать культуру разработки программного обеспечения, включая понимание принципов читаемости кода, документирования (комментирования), использования систем контроля версий (Git) и соблюдения стандартов оформления (PEP 8), что является неотъемлемой частью профессиональной компетенции современного аналитика.

1.2 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Компетенция (код и наименование)	Индикаторы компетенций (код и наименование)	Результаты обучения
ОПК-1 Способен понимать принципы работы современных информационных технологий и использовать их при решении задач профессиональной деятельности социолога	ОПК-1.1. Понимает содержание, структуру и принципы работы современных информационных технологий, применяемых для решения задач профессиональной деятельности.	Знать: архитектуру и принципы функционирования интерпретируемых языков программирования на примере Python; классификацию и назначение основных структур данных (числа, строки, списки, словари) как инструментов представления социальной информации; логику работы управляющих конструкций (ветвление, циклы), лежащую в основе алгоритмов

		<p>обработки социологических массивов; понимать разницу между компилируемыми и интерпретируемыми языками, а также возможности экосистемы Python (менеджер пакетов pip, виртуальные окружения) для решения профессиональных задач.</p> <p>Уметь: идентифицировать профессиональную задачу, для решения которой требуется написание программного кода; выбирать оптимальные структуры данных и управляющие конструкции в зависимости от характера исходных данных и поставленного исследовательского вопроса; объяснять логику разработанного алгоритма и интерпретировать результаты его выполнения; использовать официальную документацию и интернет-ресурсы (Stack Overflow) для самостоятельного изучения новых библиотек и методов.</p> <p>Владеть: навыками навигации в интегрированной среде разработки (IDE) или текстовом редакторе; базовой терминологией предметной области (алгоритм, синтаксис, семантика, отладка); навыками структурного (алгоритмического) мышления при постановке задачи на программирование; основами работы с командной строкой для установки пакетов и запуска скриптов.</p>
<p>ОПК-1 Способен понимать принципы работы современных информационных технологий и использовать их при решении задач профессиональной</p>	<p>ОПК-1.2. Использует современные информационные технологии при решении задач профессиональной деятельности.</p>	<p>Знать: методы автоматизации обработки эмпирических данных средствами Python; возможности библиотек Pandas и NumPy для выполнения описательного статистического анализа;</p>

<p>деятельности социолога</p>		<p>правила визуализации данных с помощью Matplotlib, соответствующие академическим стандартам и принципам объективности;</p> <p>понимать, как программные средства позволяют масштабировать анализ на большие массивы данных (Big Data) и воспроизводить результаты исследования.</p> <p>Уметь:</p> <p>разрабатывать программные скрипты на Python для загрузки, очистки и трансформации социологических данных из различных форматов (CSV, TXT, Excel);</p> <p>применять библиотеку Pandas для фильтрации, группировки и агрегации данных (расчет средних, частот, долей по группам);</p> <p>строить визуализации (гистограммы, ящики с усами, точечные диаграммы) для наглядного представления выявленных закономерностей;</p> <p>писать простые парсеры для сбора данных из открытых веб-источников.</p> <p>Владеть:</p> <p>навыками написания читаемого и документированного кода в соответствии со стандартом PEP 8;</p> <p>техниками отладки программ и обработки исключительных ситуаций (try/except);</p> <p>навыками экспорта результатов анализа (таблиц, графиков) в форматы, пригодные для включения в отчеты и презентации;</p> <p>основами работы с системами контроля версий (Git) для управления версиями аналитических скриптов.</p>
<p>ПК-4. Способен к организации работы по сбору, обработке и анализу данных фундаментальных и</p>	<p>ПК-4.3 Обработывает, анализирует и представляет собранные данные с применением</p>	<p>Знать:</p> <p>полный цикл обработки данных средствами языка Python: от первичного импорта и очистки (обработка пропусков, дубликатов,</p>

<p>прикладных социологических исследований</p>	<p>информационных технологий</p>	<p>выбросов) до трансформации и подготовки к анализу; методы статистического анализа данных, реализованные в библиотеках NumPy и Pandas (описательные статистики, группировки, корреляционный анализ); принципы построения информативных визуализаций для различных типов данных (количественных, категориальных, временных рядов) и задачи, которые решает каждая визуализация; способы представления результатов анализа в форме, пригодной для включения в научные отчеты, статьи и презентации (таблицы, графики, дашборды).</p> <p>Уметь: разрабатывать на Python скрипты, реализующие полный цикл обработки эмпирических данных: загрузка данных из различных источников (файлы, API, веб-страницы), их очистка и подготовка (удаление/заполнение пропусков, приведение типов, фильтрация выбросов), проведение аналитических расчетов (расчет средних, медиан, частот, кросс-табуляция, корреляции), визуализация ключевых результатов (гистограммы, ящики с усами, тепловые карты, точечные диаграммы) экспорт итоговых таблиц и графиков во внешние форматы (CSV, PNG, PDF).</p> <p>Владеть: навыками написания воспроизводимых аналитических отчетов в среде Jupyter Notebook, объединяющих код, результаты вычислений и интерпретацию; техниками создания многопанельных графиков и</p>
--	----------------------------------	--

		композиций для комплексного представления данных; навыками подготовки дашбордов (информационных панелей) на основе Matplotlib и Seaborn для мониторинга социальных показателей; методами документирования процесса анализа и представления результатов заказчику или научному руководителю в наглядной и убедительной форме.
--	--	--

1.3 Место дисциплины в структуре образовательной программы

Дисциплина «Программирование на Python» относится к вариативной части, формируемой участниками образовательных отношений по направлению «Социология», базового блока дисциплин учебного плана.

Для освоения дисциплины необходимы знания, умения и владения, сформированные в ходе изучения следующих дисциплин и прохождения практик: «Высшая математика», «Инструменты аналитической обработки данных в табличных редакторах», «Базы данных», ознакомительной практики на 1 курсе.

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин и прохождения практик: «Анализ и структуры данных в социологии»; прохождения производственной и педагогической практик.

2. Структура дисциплины

Структура дисциплины для очной формы обучения

Общая трудоёмкость дисциплины составляет 5 з.е., 180 академических часов. Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
5	Практические занятия	20
5	Семинарские занятия	32
6	Практические занятия	20
6	Семинарские занятия	32
Всего:		104

Объем дисциплины в форме самостоятельной работы обучающихся составляет 76 академических часов.

3. Содержание дисциплины

Часть 1. Основы алгоритмизации и программирования на Python

Тема 1. Введение в программирование. Язык Python в работе социолога.

Понятие алгоритма и программы. Роль программирования в автоматизации обработки социальных данных. Обзор языков программирования: критерии выбора для задач социолога. Python как лидер в области анализа данных и машинного обучения: преимущества (простота, читаемость, богатство библиотек, сообщество). Установка интерпретатора Python и интегрированной среды разработки (IDE) — PyCharm / VS Code. Понятие скрипта, запуск первой программы (print). Синтаксис и семантика, понятие об отступах как способе организации блоков кода.

Тема 2. Переменные и базовые типы данных.

Понятие переменной как именованной области памяти для хранения данных. Правила именования переменных (PEP 8). Числовые типы: int (целые числа) и float (числа с плавающей точкой). Арифметические операции и их приоритет. Логический тип bool (True/False). Операции сравнения. Функции type() и id() для исследования объектов.

Тема 3. Строковый тип данных и работа с текстом.

Строковый тип str как основной способ представления текстовой информации (ответы на открытые вопросы, комментарии, наименования). Операции со строками: конкатенация, умножение на число, индексация (доступ по индексу). Срезы строк (извлечение подстрок). Основные методы строк: lower, upper, replace, split, join, strip, find, count. Форматирование строк с помощью f-строк (наиболее современный и удобный способ). Экранирование символов.

Тема 4. Ввод и вывод данных. Взаимодействие с пользователем.

Функция input() для получения данных от пользователя. Преобразование строк, полученных от input(), в нужные типы данных (int, float). Вывод данных с помощью print(): форматирование вывода, параметры sep и end. Простейшие программы-калькуляторы и программы-интервьюеры для сбора данных.

Тема 5. Базовые структуры данных: списки (list).

Понятие коллекции данных. Список как упорядоченная изменяемая коллекция. Создание списков (литералы, преобразование других типов). Индексация и срезы списков. Добавление элементов: `append`, `extend`, `insert`. Удаление элементов: `remove`, `pop`, `del`. Поиск элементов: `index`, `count`. Проверка вхождения (`in`). Использование списков для хранения массива ответов респондентов.

Тема 6. Базовые структуры данных: кортежи (tuple) и множества (set).

Кортеж как неизменяемый аналог списка: создание, особенности, область применения (например, для хранения неизменяемых характеристик). Множество как неупорядоченная коллекция уникальных элементов: создание, добавление (`add`), удаление (`remove`, `discard`). Операции над множествами: пересечение (`&`), объединение (`|`), разность (`-`) для анализа общности признаков (например, общие интересы групп).

Тема 7. Базовые структуры данных: словари (dict).

Словарь как неупорядоченная коллекция типа «ключ-значение». Создание словарей. Доступ к значениям по ключу (квадратные скобки и метод `get`). Добавление и изменение пар ключ-значение. Перебор ключей, значений и пар ключ-значение (методы `keys`, `values`, `items`). Использование словарей для хранения структурированных данных о респонденте (профиль: имя, возраст, город, ответы). Словарь как база данных в оперативной памяти.

Тема 8. Условный оператор if.

Необходимость ветвления в алгоритмах обработки данных. Условный оператор `if`. Полная и неполная форма ветвления. Отступы как способ выделения блока кода. Простые условия с операциями сравнения. Составные условия с логическими операторами `and`, `or`, `not`. Таблицы истинности.

Тема 9. Множественное ветвление: if-elif-else.

Конструкция `if-elif-else` для множественного выбора. Порядок проверки условий. Вложенные условные операторы. Применение условного оператора для сегментации респондентов (выделение групп по возрасту, доходу, месту жительства), для проверки корректности введенных данных, для присвоения меток на основе значений переменных. Тернарный условный оператор (как краткая форма записи).

Тема 10. Цикл while.

Понятие цикла как способа многократного выполнения действий. Цикл `while` с условием. Бесконечный цикл и способы его предотвращения. Управление циклом: оператор `break` (досрочный выход из цикла) и `continue` (переход к следующей итерации). Применение `while` для обработки данных, когда количество итераций заранее неизвестно (например, чтение данных до появления пустой строки).

Тема 11. Цикл `for`. Итерация по последовательностям.

Цикл `for` для перебора элементов последовательности (строки, списка, кортежа, словаря, множества). Функция `range()` для генерации числовых последовательностей. Особенности перебора словарей. Вложенные циклы для обработки многомерных структур (например, списка списков). Применение циклов для обработки массивов данных: итеративный подсчет частот, суммирование значений, применение одностипных преобразований ко всем элементам выборки.

Тема 12. Генераторы списков (`list comprehension`).

Понятие генератора списка как элегантного и эффективного способа создания списков. Синтаксис генератора списка: `[выражение for элемент in последовательность]`. Генераторы с условием: `[выражение for элемент in последовательность if условие]`. Сравнение с традиционным циклом `for`. Применение для быстрой обработки и фильтрации данных. Генераторы множеств и словарей.

Тема 13. Функции: определение и вызов.

Понятие функции как способа группировки кода для решения повторяющейся задачи. Определение функции с помощью ключевого слова `def`. Вызов функции. Параметры и аргументы функции. Позиционные и именованные аргументы. Значения параметров по умолчанию. Возврат значения с помощью `return`. Функции, возвращающие несколько значений (упаковка в кортеж).

Тема 14. Функции: области видимости переменных.

Локальные и глобальные переменные. Области видимости (`scope`). Правило LEGB (Local, Enclosing, Global, Built-in). Ключевое слово `global` для изменения глобальных переменных внутри функции. Замыкания (краткое знакомство). Документирование функций с помощью `docstring`.

Тема 15. Создание модулей и импорт библиотек.

Понятие модуля как файла с Python-кодом. Создание собственных модулей (библиотек функций). Импорт модуля (`import`). Импорт отдельных функций (`from module import function`). Импорт модуля с псевдонимом (`import module as alias`). Стандартная библиотека Python: обзор полезных модулей (`math`, `random`, `datetime`, `os`, `sys`). Установка сторонних библиотек с помощью `pip` (пакетный менеджер).

Тема 16. Работа с файлами. Ввод-вывод данных.

Необходимость сохранения и загрузки данных. Открытие файла с помощью функции `open()`. Режимы работы с файлами: чтение ('r'), запись ('w'), добавление ('a'), чтение и запись ('r+'). Чтение текстового файла целиком (`read`), построчно (`readline`, `readlines`). Запись данных в текстовый файл (`write`, `writelines`). Менеджер контекста `with...as` для автоматического закрытия файла. Формат CSV (Comma-Separated Values) как основной формат хранения социологических данных. Чтение CSV-файла с помощью встроенного модуля `csv` (`csv.reader`). Запись результатов анализа в CSV-файл (`csv.writer`).

Часть 2. Библиотеки для анализа данных и автоматизации

Тема 17. Обработка исключений.

Понятие исключительной ситуации (ошибки времени выполнения). Различные типы исключений: `ValueError`, `TypeError`, `FileNotFoundError`, `ZeroDivisionError`, `IndexError`. Конструкция `try-except` для перехвата и обработки ошибок. Несколько блоков `except` для разных типов исключений. Блок `else` (выполняется, если ошибки не было). Блок `finally` (выполняется всегда). Повышение устойчивости программ к некорректным входным данным (например, пропущенные значения в опросных листах).

Тема 18. Отладка кода и написание читаемого кода.

Основные приемы отладки кода: использование `print` для трассировки, использование отладчика в IDE (точки останова, пошаговое выполнение, просмотр значений переменных). Логирование ошибок с помощью модуля `logging`. Понятие стиля кодирования. Стандарт PEP 8: отступы, длина строк, пробелы, именование переменных и функций. Написание самодокументируемого кода. Важность комментариев (где они действительно нужны).

Тема 19. Библиотека NumPy: массивы и базовые операции.

Понятие библиотеки NumPy для научных вычислений. Преимущества массивов NumPy (`ndarray`) перед стандартными списками: скорость, компактность, векторизация. Создание

массивов из списков и с помощью функций: `arange`, `arange`, `zeros`, `ones`, `linspace`. Атрибуты массива: `shape`, `ndim`, `size`, `dtype`. Индексация и срезы многомерных массивов.

Тема 20. Библиотека NumPy: универсальные и агрегирующие функции.

Универсальные функции (`ufunc`) для поэлементных операций (`+`, `-`, `*`, `/`, `sqrt`, `exp`, `log`). Агрегирующие функции: `sum`, `mean`, `std`, `var`, `min`, `max`, `argmin`, `argmax`. Применение NumPy для быстрой обработки числовых социологических данных (возраст, доход, баллы). Основы линейной алгебры (`dot`, `matmul`). Трансляции (`broadcasting`) — краткое знакомство.

Тема 21. Библиотека Pandas: структуры Series и DataFrame.

Библиотека Pandas как основной инструмент анализа данных в Python. Структура `Series` (одномерный массив с метками-индексами): создание, доступ по индексу, векторизованные операции. Структура `DataFrame` (двумерная таблица): создание из словаря, списка словарей, CSV-файла. Атрибуты `DataFrame`: `index`, `columns`, `values`, `shape`.

Тема 22. Pandas: загрузка данных и первичный анализ.

Чтение данных из CSV-файла с помощью `read_csv()`: основные параметры (`sep`, `encoding`, `header`). Первичный анализ данных: просмотр первых и последних строк (`head`, `tail`), информация о данных (`info`), описательные статистики для числовых столбцов (`describe`). Выборка одного столбца (два способа). Добавление и удаление столбцов.

Тема 23. Pandas: фильтрация и выборка данных.

Выборка данных по меткам с помощью `loc`. Выборка данных по позициям с помощью `iloc`. Булева индексация (фильтрация): `df[df.column > value]`. Множественные условия с использованием `&` (и), `|` (или), `~` (не). Метод `isin()` для фильтрации по списку значений. Метод `between()` для фильтрации по диапазону.

Тема 24. Pandas: обработка пропущенных значений.

Понятие пропущенных значений (`NaN`) в социологических данных. Обнаружение пропусков: `isna()`, `notna()`. Подсчет пропусков по столбцам. Удаление строк или столбцов

с пропусками: `dropna()`. Заполнение пропусков: `fillna()` (константой, средним, методом вперед/назад). Интерполяция пропусков (`interpolate`) для временных рядов.

Тема 25. Pandas: группировка и агрегация данных.

Сортировка данных: `sort_values` (по одному или нескольким столбцам). Группировка данных с помощью `groupby`. Агрегация данных: применение встроенных функций (`mean`, `sum`, `count`, `std`) к группам. Метод `agg()` для применения нескольких функций одновременно. Использование пользовательских функций в `agg`. Трансформация данных с помощью `transform`.

Тема 26. Pandas: анализ частот и таблицы сопряженности.

Анализ частот категориальных переменных: `value_counts()` (обычные и нормированные). Таблицы сопряженности: `pd.crosstab()`. Добавление маргинальных итогов (`normalize`, `margins`). Применение группировки и таблиц сопряженности для выявления различий между социально-демографическими группами (например, уровень образования по полу и возрасту).

Тема 27. Pandas: объединение данных.

Объединение нескольких `DataFrame` по строкам (вертикально) с помощью `pd.concat()`. Объединение по столбцам (горизонтально). Соединение таблиц по ключу (как `VLOOKUP` в Excel): `pd.merge()`. Типы соединений: `inner`, `outer`, `left`, `right`. Обработка дублирующихся ключей. Создание сводных таблиц с помощью `pivot_table`.

Тема 28. Библиотека Matplotlib: основы визуализации.

Принципы эффективной визуализации. Базовая библиотека `Matplotlib`. Импорт `pyplot`. Создание фигуры и осей. Построение линейных графиков (`plot`) для отображения трендов. Настройка размера рисунка (`figsize`), подписей осей (`xlabel`, `ylabel`), заголовка (`title`), легенды (`legend`). Использование различных стилей линий и маркеров.

Тема 29. Matplotlib: гистограммы, ящики с усами и точечные диаграммы.

Построение гистограмм (`hist`) для анализа распределения признаков (настройка количества бинов). Ящики с усами (`boxplot`) для выявления выбросов и сравнения распределений нескольких групп. Точечные диаграммы (`scatter`) для первичного анализа взаимосвязи двух переменных. Добавление регрессионной линии с помощью `np.polyfit`. Сохранение графиков в файл (`savefig`) с различными параметрами (`dpi`, `format`).

Тема 30. Библиотека Seaborn: статистическая визуализация.

Библиотека Seaborn как надстройка над Matplotlib. Особенности интерфейса: использование DataFrame напрямую. Построение гистограмм с ядерной оценкой плотности (histplot, kdeplot). Ящики с усами (boxplot) и скрипичные диаграммы (violinplot). Точечные диаграммы с регрессионной линией (lmpplot, regplot). Матрицы корреляций и тепловые карты (heatmap). Настройка стилей оформления (set_style, set_palette).

Тема 31. Введение в веб-скрапинг: сбор данных из интернета.

Понятие веб-скрапинга и его этические границы (robots.txt, условия использования сайта). Библиотека Requests для отправки HTTP-запросов: получение HTML-страницы (get), проверка статуса ответа. Библиотека BeautifulSoup для парсинга (разбора) HTML-структуры. Поиск элементов по тегам, классам, id (find, find_all). Извлечение текста и атрибутов (get_text(), [attr]). Сбор структурированных данных (таблиц, списков) с веб-страниц и сохранение их в DataFrame.

Тема 32. Основы работы с API для получения данных.

Понятие API (Application Programming Interface) как современного способа получения данных. Формат обмена данными JSON. Библиотека Requests для работы с API. Отправка GET-запросов к открытым API (например, ВКонтакте, GitHub, данные государственных порталов). Передача параметров в запросе (params). Обработка JSON-ответа: преобразование в словарь Python (response.json()). Извлечение нужной информации из сложных JSON-структур. Сохранение данных в CSV. Примеры использования API в социологических исследованиях (сбор постов, комментариев, статистики сообществ, данных переписи).

4. Образовательные технологии

№	Тема	Образовательные технологии
1	Введение в программирование. Язык Python в работе социолога.	Лекция-визуализация, установка ПО (тренинг), самостоятельная работа по настройке среды, консультация.
2	Переменные и базовые типы данных.	Лекция с демонстрацией примеров, лабораторное занятие (решение задач в интерактивной среде), экспресс-опрос.
3	Строковый тип данных и	Проблемная лекция (как обрабатывать открытые

№	Тема	Образовательные технологии
	работа с текстом.	вопросы?), лабораторное занятие (кейс «Очистка текстовых ответов»), тренинг.
4	Ввод и вывод данных. Взаимодействие с пользователем.	Лекция-визуализация, лабораторное занятие (создание программы-интервьюера), парное программирование.
5	Базовые структуры данных: списки (list).	Лекция с разбором примеров, лабораторное занятие (кейс «Хранение массива ответов»), самостоятельная работа.
6	Базовые структуры данных: кортежи (tuple) и множества (set).	Лекция-визуализация, лабораторное занятие (кейс «Анализ уникальных интересов групп»), мини-дискуссия.
7	Базовые структуры данных: словари (dict).	Проблемная лекция (как хранить профиль респондента?), лабораторное занятие (создание базы данных в памяти), тренинг.
8	Условный оператор if.	Лекция с демонстрацией, лабораторное занятие (кейс «Сегментация респондентов по возрасту»), консультация.
9	Множественное ветвление: if-elif-else.	Лекция-визуализация, лабораторное занятие (кейс «Классификация уровней дохода»), работа в малых группах.
10	Цикл while.	Проблемная лекция (как обрабатывать данные неизвестного объема?), лабораторное занятие (игра «Угадай число»), тренинг.
11	Цикл for. Итерация по последовательностям.	Лекция с разбором примеров, лабораторное занятие (кейс «Подсчет частот вручную»), самостоятельная работа.
12	Генераторы списков (list comprehension).	Лекция-визуализация, лабораторное занятие (сравнение циклов и генераторов), экспресс-соревнование.
13	Функции: определение и вызов.	Лекция с объяснительно-побуждающим обучением, лабораторное занятие (создание библиотеки функций), консультация.
14	Функции: области видимости переменных.	Лекция-визуализация, лабораторное занятие (эксперименты с локальными и глобальными переменными), тренинг.
15	Создание модулей и импорт библиотек.	Лекция с демонстрацией, лабораторное занятие (создание собственного модуля), самостоятельная работа.
16	Работа с файлами. Ввод-вывод данных.	Проблемная лекция (как сохранить результаты?), лабораторное занятие (чтение и запись CSV-файлов), кейс-стади.
17	Обработка исключений.	Лекция с разбором популярных ошибок, лабораторное занятие (повышение устойчивости кода), тренинг «Отлов ошибок».
18	Отладка кода и написание читаемого кода.	Лекция-визуализация (стандарты PEP 8), лабораторное занятие (рефакторинг и отладка чужого кода), дискуссия.
19	Библиотека NumPy: массивы и базовые операции.	Лекция с демонстрацией, лабораторное занятие (создание и индексация массивов), самостоятельная работа.

№	Тема	Образовательные технологии
20	NumPy: универсальные и агрегирующие функции.	Проблемная лекция (скорость vs удобство), лабораторное занятие (расчет статистик для массива), тренинг.
21	Библиотека Pandas: структуры Series и DataFrame.	Лекция-визуализация, лабораторное занятие (создание DataFrame из различных источников), консультация.
22	Pandas: загрузка данных и первичный анализ.	Лекция с демонстрацией, лабораторное занятие (загрузка реального датасета), кейс «Первое знакомство с данными».
23	Pandas: фильтрация и выборка данных.	Проблемная лекция (как отобразить нужное?), лабораторное занятие (кейс «Выделение подвыборки»), тренинг.
24	Pandas: обработка пропущенных значений.	Лекция-визуализация (проблема «грязных» данных), лабораторное занятие (кейс «Очистка опроса»), дискуссия.
25	Pandas: группировка и агрегация данных.	Лекция с разбором примеров, лабораторное занятие (кейс «Сравнение социальных групп»), мини-исследование.
26	Pandas: анализ частот и таблицы сопряженности.	Лекция-визуализация, лабораторное занятие (кейс «Зависимость между признаками»), работа в парах.
27	Pandas: объединение данных (merge, concat).	Проблемная лекция (как соединить данные из разных источников?), лабораторное занятие (кейс «Обогащение данных»).
28	Библиотека Matplotlib: основы визуализации.	Лекция-визуализация с элементами дискуссии «Как не обмануть графиком», лабораторное занятие.
29	Matplotlib: гистограммы, ящики с усами, точечные диаграммы.	Лекция с демонстрацией, лабораторное занятие (визуализация распределений), тренинг.
30	Библиотека Seaborn: статистическая визуализация.	Лекция-визуализация, лабораторное занятие (создание «красивых» графиков), конкурс на лучшую визуализацию.
31	Введение в веб-скрапинг (BeautifulSoup).	Лекция с разбором этических аспектов, лабораторное занятие (парсинг учебного сайта), групповая работа.
32	Работа с API. Итоговый проект.	Лекция-визуализация, лабораторное занятие (получение данных из API), проектная деятельность, защита проектов.

При реализации программы дисциплины «Программирование на Python» используются следующие образовательные технологии:

Лекционные занятия проводятся в интерактивных формах:

- ~ Лекция-визуализация — подача материала сопровождается разбором программного кода, демонстрацией работы скриптов и визуализацией структур данных.
- ~ Проблемная лекция — перед изложением нового материала формулируется проблема из области социологии (например, «как обработать 10 000 открытых вопросов?»), и в ходе лекции ищутся пути ее решения средствами Python.

- ~ Лекция с разбором конкретных ситуаций — анализируются реальные кейсы из практики цифровых социологов.

Лабораторные занятия (основная форма практической работы):

- ~ Работа за персональным компьютером с установленным интерпретатором Python и IDE.
- ~ Решение кейсов — выполнение практических заданий, моделирующих реальные исследовательские задачи (обработка данных опроса, визуализация результатов, сбор данных из открытых источников).
- ~ Парное программирование — работа в парах, где один студент пишет код, а другой анализирует и предлагает улучшения, с последующей сменой ролей.
- ~ Тренинги — отработка конкретных навыков (например, написание циклов, отладка кода) на серии коротких упражнений.

Самостоятельная работа студентов включает:

- ~ Изучение теоретического материала и документации к библиотекам.
- ~ Выполнение домашних заданий по написанию программ.
- ~ Подготовку к контрольным работам и тестированиям.
- ~ Работу над итоговым проектом (индивидуально или в малых группах).

Проектная деятельность:

- ~ Выполнение итогового проекта, включающего полный цикл анализа данных (от постановки задачи и сбора данных до визуализации и формулировки выводов).
- ~ Защита проекта с презентацией результатов.

Консультации с преподавателем: индивидуальные и групповые консультации по сложным темам и проектам.

Дискуссии и мини-конференции: обсуждение результатов анализа, интерпретация полученных данных, этические аспекты сбора и обработки информации.

В период временного приостановления посещения обучающимися помещений и территории РГГУ для организации учебного процесса с применением электронного обучения и дистанционных образовательных технологий могут быть использованы следующие образовательные технологии:

- ~ видеолекции и онлайн-лекции в режиме реального времени;
- ~ электронные учебники и учебные пособия в электронном виде;
- ~ системы для электронного тестирования (Stepik, Google Forms);
- ~ облачные среды разработки (Replit, Google Colab) для выполнения лабораторных работ;

консультации с использованием телекоммуникационных средств.

5. Оценка планируемых результатов обучения

5.1 Система оценивания

Оценка знаний студентов складывается из *промежуточных* аттестаций (60%), а также *итоговой* аттестации в письменной форме (40%).

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль:		
~ экспресс-тест;	1 балл	15 баллов
~ участие в дискуссии на семинаре;	1 балл	13 баллов
~ подготовка доклада;	5 баллов	5 баллов
~ промежуточная контрольная работа;	10 баллов	10 баллов
~ реферат.	17 баллов	17 баллов
<i>Промежуточная аттестация (итоговая практическая работа)</i>		40 баллов
Итого за семестр экзамен		100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала		Шкала ECTS
95 – 100	отлично	зачтено	A
83 – 94			B
68 – 82	хорошо		C
56 – 67	удовлетворительно		D
50 – 55			E
20 – 49	неудовлетворительно	не зачтено	FX
0 – 19			F

5.2 Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A,B	«отлично»/ «зачтено (отлично)»/ «зачтено»	Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации. Обучающийся исчерпывающе и логически стройно

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
		<p>излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения.</p> <p>Свободно ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>
82-68/ С	«хорошо»/ «зачтено (хорошо)»/ «зачтено»	<p>Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей.</p> <p>Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами.</p> <p>Достаточно хорошо ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>
67-50/ D,E	«удовлетворительно»/ «зачтено (удовлетворительно)»/ «зачтено»	<p>Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами.</p> <p>Демонстрирует достаточный уровень знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>
49-0/ F,FX	«неудовлетворительно»/ не зачтено	<p>Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его</p>

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
		<p>изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами.</p> <p>Демонстрирует фрагментарные знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляются обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p>

5.3 Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

Оценка знаний, умений, опыта деятельности студента осуществляется преподавателем в ходе реализации каждой из предусмотренных рабочей программой форм занятий. Основным способом оценки выступает контроль за посещением занятий, уровнем подготовленности к ним, качеством предъявляемых устных и письменных знаний, умений, навыков, формирующих профессиональный опыт студентов-социологов. Предусмотрены следующие виды контроля:

Экспресс-опрос

Формируемые компетенции и индикаторы их достижения: ОПК-1 (ОПК-1.1 и ОПК-1.2), и ПК-4 (ПК-4.3).

В конце лекции преподаватель задает один вопрос по теме лекции. В течение максимум 5 минут, студент обязан написать краткий (максимум 10 строк) ответ на поставленный вопрос. Экспресс-тест оценивается в 1 балл.

При экспресс-опросе по каждой из прочитанных лекций выявляется формирование компетенции по критическому восприятию, обобщению, анализу профессиональной информации.

Доклад

Формируемые компетенции и индикаторы их достижения: ОПК-1 (ОПК-1.1 и ОПК-1.2), и ПК-4 (ПК-4.3).

При оценивании устного доклада, учитываются следующие характеристики:

- ~ полнота подготовки задания, информативность;
- ~ глубина проработки изучаемых вопросов (проработка материала должна отражать владение основными понятиями и концепциями);
- ~ системность, наличие выводов.
- ~ максимально за один доклад можно получить 5 баллов.

Участие в дискуссии на семинарах

Формируемые компетенции и индикаторы их достижения: ОПК-1 (ОПК-1.1 и ОПК-1.2), и ПК-4 (ПК-4.3).

Выступления студентов на семинарских занятиях позволяют контролировать формирование у них способности использовать полученные знания в преподавании социологии, формировать практические навыки и умения для участия в научных и научно-прикладных исследованиях, аналитической и консалтинговой деятельности. Максимально за одно выступление можно получить 1 балл.

Подготовка реферата

Формируемые компетенции и индикаторы их достижения: ОПК-1 (ОПК-1.1 и ОПК-1.2), и ПК-4 (ПК-4.3).

Требования к содержанию реферата. Реферат должен отражать основные подходы к изучаемому объекту, а также содержать авторскую оценку описываемого объекта. Объем работы: 10 стр. печатного текста.

Оформление работы

Шрифт TimesNewRoman, кегль 14, поля 3 см слева, 2 сверху и внизу, 1,5 справа. Красная строка (абзацный отступ) 1,25 см.

Фамилия автора – распределение справа, название статьи и фамилия автора статьи по центру, основной текст – по ширине. Реферат оценивается максимум в 17 баллов.

Тематика рефератов:

1. Сравнительный анализ языков программирования для задач анализа данных: Python vs R.
2. История развития языка Python: от любительского проекта до мирового стандарта Data Science.
3. Экосистема Python для научных вычислений: обзор ключевых библиотек (NumPy, SciPy, Pandas, Matplotlib, Scikit-learn).

4. Методы очистки и предобработки «грязных» социологических данных с помощью Pandas.
5. Разведочный анализ данных (EDA) на Python: инструменты и визуализации.
6. Применение библиотеки NumPy для обработки многомерных массивов в социальных исследованиях.
7. Техники визуализации социологических данных: сравнительный анализ Matplotlib и Seaborn.
8. Автоматизация сбора данных: веб-скрапинг vs API (этические и технические аспекты).
9. Использование открытых API (ВКонтакте, Twitter, Telegram) для сбора данных в социологических целях.
10. Методы обработки текстовой информации в Python: регулярные выражения и библиотеки для NLP (на примере анализа открытых вопросов).
11. Программирование на Python как инструмент воспроизводимости социологических исследований (reproducible research).
12. Интерактивные отчеты и дашборды: использование Jupyter Notebook в работе социолога-аналитика.
13. Основы объектно-ориентированного программирования (ООП) в Python и их применимость в социологических проектах.
14. Парадигмы программирования: структурный, функциональный и объектно-ориентированный подходы на примере Python.
15. Тестирование кода в Python: модуль unittest и pytest для обеспечения качества анализа.
16. Работа с базами данных из Python: библиотека SQLAlchemy и Pandas для загрузки/выгрузки данных.
17. Применение Python для анализа социальных сетей (Social Network Analysis): библиотека NetworkX.
18. Введение в машинное обучение на Python: библиотека Scikit-learn для классификации и кластеризации социальных групп.
19. Этические проблемы применения искусственного интеллекта и алгоритмов в социальных исследованиях.
20. Цифровые следы как данные: методы сбора и анализа информации из социальных медиа на Python.
21. Геопространственный анализ в социологии: библиотеки GeoPandas и Folium.

22. Оптимизация производительности Python-кода для работы с большими массивами данных.
23. Системы контроля версий Git в работе аналитика данных: основы и лучшие практики.
24. Разработка телеграм-ботов на Python для проведения опросов и сбора данных.
25. Визуализация временных рядов в социологии: анализ трендов и сезонности на Python.
26. Работа с форматами данных (JSON, XML, Parquet) в Python: импорт и экспорт.
27. Применение регулярных выражений (re) для извлечения структурированной информации из текстов.
28. Создание пользовательских функций и библиотек для автоматизации повторяющихся социологических расчетов.
29. Основы параллельных вычислений в Python (multiprocessing) для ускорения обработки данных.
30. Будущее Python в социологии: тренды, новые библиотеки и вызовы.

Вопросы для самопроверки:

1. **Какой тип данных в Python используется для хранения целых чисел?**
 - A) float
 - B) int (*верно*)
 - C) str
 - D) bool
2. **Какая функция выводит информацию на экран?**
 - A) input()
 - B) print() (*верно*)
 - C) len()
 - D) type()
3. **Какой тип данных вернет выражение `type(3.14)`?**
 - A) int
 - B) float (*верно*)
 - C) str
 - D) complex
4. **Что будет результатом выражения `"5" + "3"` в Python?**
 - A) 8
 - B) 53

- C) "53" *(верно)*
- D) Ошибка, так как нельзя складывать строки
5. **Какой индекс имеет первый элемент списка?**
- A) 1
- B) 0 *(верно)*
- C) -1
- D) Любой, задается при создании
6. **Какой метод добавляет элемент в конец списка?**
- A) append() *(верно)*
- B) insert()
- C) add()
- D) push()
7. **Что такое словарь (dict) в Python?**
- A) Упорядоченная коллекция элементов с доступом по индексу
- B) Неупорядоченная коллекция уникальных элементов
- C) Коллекция типа «ключ-значение» *(верно)*
- D) Неизменяемая последовательность
8. **Какое ключевое слово используется для определения функции?**
- A) func
- B) define
- C) def *(верно)*
- D) function
9. **Что вернет выражение len([1, 2, 3, 4, 5])?**
- A) 1
- B) 5 *(верно)*
- C) [1,2,3,4,5]
- D) Ошибку
10. **Какой оператор проверяет равенство двух значений?**
- A) =
- B) == *(верно)*
- C) !=
- D) equals
11. **Какой цикл выполняется, пока условие истинно?**
- A) for
- B) while *(верно)*

- C) if
 - D) loop
12. **Что делает инструкция break внутри цикла?**
- A) Пропускает текущую итерацию
 - B) Завершает цикл досрочно (*верно*)
 - C) Начинает цикл заново
 - D) Вызывает ошибку
13. **Как получить срез списка со второго по пятый элемент (включительно)?**
- A) list[2:5]
 - B) list[1:5] (*верно, т.к. индексация с 0*)
 - C) list[2:6]
 - D) list[1:6]
14. **Какая библиотека Python является основной для табличных вычислений и анализа данных?**
- A) Matplotlib
 - B) NumPy
 - C) Pandas (*верно*)
 - D) Requests
15. **Какой командой устанавливаются сторонние библиотеки?**
- A) install pip
 - B) python get lib
 - C) pip install (*верно*)
 - D) conda install (тоже верно, но pip — стандарт)
16. **Какой метод Pandas используется для чтения CSV-файла?**
- A) read_excel()
 - B) read_csv() (*верно*)
 - C) load_csv()
 - D) open_csv()
17. **Что возвращает метод head() в Pandas?**
- A) Последние 5 строк
 - B) Первые 5 строк (*верно*)
 - C) Случайные 5 строк
 - D) Количество строк
18. **Как выбрать столбец age в DataFrame df?**
- A) df[age]

- B) `df.age` (*верно*)
 - C) `df{"age"}`
 - D) `df.get(age)`
19. **Какая функция Pandas показывает описательную статистику по числовым столбцам?**
- A) `info()`
 - B) `describe()` (*верно*)
 - C) `stats()`
 - D) `summary()`
20. **Как отфильтровать строки DataFrame df, где значение в столбце city равно "Москва"?**
- A) `df[city == "Москва"]`
 - B) `df[df.city == "Москва"]` (*верно*)
 - C) `df.filter(city="Москва")`
 - D) `df.loc["Москва"]`
21. **Какая библиотека используется для визуализации данных?**
- A) Pandas
 - B) Matplotlib (*верно*)
 - C) NumPy
 - D) BeautifulSoup
22. **Какой тип графика лучше всего подходит для отображения распределения одной числовой переменной?**
- A) Гистограмма (`hist`) (*верно*)
 - B) Точечная диаграмма (`scatter`)
 - C) Линейный график (`plot`)
 - D) Круговая диаграмма (`pie`)
23. **Что такое API?**
- A) Язык программирования
 - B) Интерфейс для взаимодействия программ (*верно*)
 - C) Тип базы данных
 - D) Протокол передачи данных
24. **Какая библиотека используется для отправки HTTP-запросов?**
- A) BeautifulSoup
 - B) Requests (*верно*)

- C) urllib (тоже верно, но Requests — стандарт ответа)
D) http
25. **Как открыть файл data.txt для чтения?**
A) open("data.txt", "r") (*верно*)
B) open("data.txt", "w")
C) open("data.txt", "a")
D) open("data.txt", read)
26. **Какой оператор используется для проверки условия «и» (and)?**
A) &&
B) and (*верно*)
C) &
D) || (это от в некоторых языках)
27. **Что такое f-строка в Python?**
A) Функция для работы с файлами
B) Способ форматирования строк с подстановкой переменных (*верно*)
C) Тип данных для хранения дробей
D) Строка только для чтения
28. **Какая структура данных подходит для хранения уникальных значений?**
A) list
B) tuple
C) set (*верно*)
D) dict
29. **Как получить количество элементов в списке my_list?**
A) my_list.len()
B) len(my_list) (*верно*)
C) my_list.size()
D) count(my_list)
30. **Что делает метод lower() для строк?**
A) Удаляет пробелы в начале и конце
B) Заменяет строчные буквы на прописные
C) Заменяет прописные буквы на строчные (*верно*)
D) Разбивает строку на слова
31. **Какой тип цикла используется для перебора элементов последовательности?**
A) while
B) for (*верно*)

- C) loop
 - D) foreach
32. **Как объявить переменную x со значением 10?**
- A) var x = 10
 - B) x = 10 (*верно*)
 - C) int x = 10
 - D) let x = 10
33. **Что означает ошибка IndexError?**
- A) Обращение к несуществующему ключу словаря
 - B) Обращение к несуществующему индексу списка (*верно*)
 - C) Деление на ноль
 - D) Неверный тип данных
34. **Какая функция возвращает тип объекта?**
- A) id()
 - B) type() (*верно*)
 - C) str()
 - D) print()
35. **Как создать список [1, 2, 3]?**
- A) {1, 2, 3}
 - B) (1, 2, 3)
 - C) [1, 2, 3] (*верно*)
 - D) list(1, 2, 3)
36. **Как получить значение из словаря d по ключу "name"?**
- A) d.name
 - B) d["name"] (*верно*)
 - C) d.get("name") (*тоже верно, но B — основной способ*)
 - D) d.key("name")
37. **Как в Python обозначается комментарий?**
- A) // это комментарий
 - B) # это комментарий (*верно*)
 - C) /* это комментарий */
 - D) <!-- это комментарий -->
38. **Какой метод удаляет элемент из списка по его значению?**
- A) pop()
 - B) delete()

- C) `remove()` (*верно*)
 - D) `discard()`
39. **Что делает функция `range(5)`?**
- A) Создает список `[0,1,2,3,4,5]`
 - B) Создает последовательность чисел `0,1,2,3,4` (*верно*)
 - C) Создает список `[5]`
 - D) Вызывает ошибку
40. **Какая библиотека используется для парсинга HTML?**
- A) `Requests`
 - B) `BeautifulSoup` (*верно*)
 - C) `Scrapy`
 - D) `html.parser` (тоже верно, но `BeautifulSoup` — популярный ответ)
41. **Что такое Jupyter Notebook?**
- A) Среда разработки для Python
 - B) Интерактивная веб-среда для создания отчетов с кодом и графиками (*верно*)
 - C) Библиотека для визуализации
 - D) Менеджер пакетов
42. **Какой оператор используется для возведения в степень?**
- A) `^`
 - B) `**` (*верно*)
 - C) `^^`
 - D) `pow()`
43. **Как преобразовать строку "123" в целое число?**
- A) `int("123")` (*верно*)
 - B) `str(123)`
 - C) `"123" + 0`
 - D) `float("123")`
44. **Что делает конструкция `try...except`?**
- A) Определяет функцию
 - B) Обработывает исключения (ошибки) (*верно*)
 - C) Создает цикл
 - D) Импортирует модуль
45. **Какой метод Pandas удаляет строки с пропущенными значениями?**
- A) `fillna()`
 - B) `dropna()` (*верно*)

- C) `remove_na()`
 - D) `clean()`
46. Как объединить два DataFrame по строкам (вертикально)?
- A) `merge()`
 - B) `join()`
 - C) `concat()` (верно)
 - D) `append()` (устаревший метод, но тоже вариант)
47. Как создать точечную диаграмму (scatter plot) в Matplotlib?
- A) `plt.plot(x, y)`
 - B) `plt.bar(x, y)`
 - C) `plt.scatter(x, y)` (верно)
 - D) `plt.hist(x)`
48. Как получить справку по функции print в интерпретаторе?
- A) `?print`
 - B) `help(print)` (верно)
 - C) `man print`
 - D) `info(print)`
49. Что такое PEP 8?
- A) Библиотека Python
 - B) Стандарт оформления кода на Python (верно)
 - C) Версия Python
 - D) Название интегрированной среды разработки
50. Как сохранить график в файл plot.png в Matplotlib?
- A) `plt.show("plot.png")`
 - B) `plt.savefig("plot.png")` (верно)
 - C) `plt.export("plot.png")`
 - D) `plt.write("plot.png")`

Открытые вопросы (на размышление):

1. Предложите структуру данных (список словарей, DataFrame) для хранения результатов опроса 1000 респондентов. Обоснуйте свой выбор.
2. Напишите алгоритм (псевдокод или последовательность действий) для подсчета среднего возраста респондентов, которые ответили «Да» на вопрос о поддержке пенсионной реформы.

3. Как с помощью Python можно автоматизировать процесс обработки 50 файлов Excel с идентичной структурой, полученных от разных региональных отделений?
4. Объясните, почему при анализе больших массивов данных использование циклов for для поэлементной обработки может быть неэффективным. Какие инструменты Python предлагаются вместо этого?
5. Представьте, что при импорте CSV-файла в Pandas в столбце «Возраст» обнаружались пропуски (NaN) и некорректные значения (строки вместо чисел). Предложите стратегию очистки этого столбца.
6. Какой тип визуализации вы выберете для сравнения распределения доходов среди мужчин и женщин, и почему?
7. В чем разница между библиотеками Matplotlib и Seaborn? В каких случаях предпочтительнее использовать каждую из них?
8. Опишите этические ограничения при сборе данных из социальных сетей с помощью парсинга или API. На что нужно обратить внимание?
9. Что такое «отладка» и какие методы (инструменты) вы знаете для поиска ошибок в коде на Python?
10. Предложите структуру итогового проекта, который включал бы загрузку данных, их очистку, анализ, визуализацию и формулировку выводов. Какую социальную проблему вы бы выбрали?

Промежуточная контрольная работа

Формируемые компетенции и индикаторы их достижения: ОПК-1 (ОПК-1.1 и ОПК-1.2), и ПК-4 (ПК-4.3).

Промежуточная контрольная работа проводится в середине семестра с целью сделать «срез» знаний по классической части курса «Программирование на Python».

Тематика вопросов для промежуточной контрольной:

1. Понятие алгоритма и программы. Роль Python в современной социологии.
2. Установка и настройка среды разработки (IDE) для Python.
3. Переменные и базовые типы данных (int, float, str, bool). Правила именования.
4. Арифметические операции и операции сравнения в Python.
5. Логические операторы (and, or, not). Таблицы истинности.
6. Строковый тип данных: методы форматирования (f-strings) и основные операции.
7. Индексация и срезы строк.
8. Список (list): создание, индексация, добавление и удаление элементов.
9. Методы списков: append, extend, insert, remove, pop, index.

10. Срезы списков и их использование для выборки данных.
11. Кортежи (tuple): неизменяемость и область применения.
12. Словари (dict): создание, доступ по ключу, методы keys, values, items.
13. Множества (set): уникальность элементов, операции над множествами.
14. Условный оператор if. Простое и составное условие.
15. Конструкция if-elif-else: множественный выбор.
16. Цикл while: синтаксис и примеры использования.
17. Цикл for: итерация по последовательности.
18. Функция range() и ее применение в циклах.
19. Управление циклом: break, continue.
20. Вложенные циклы: обработка двумерных структур.
21. Понятие функции. Определение и вызов функции.
22. Аргументы функции: позиционные, именованные, значения по умолчанию.
23. Возврат значения из функции (return).
24. Области видимости переменных: локальные и глобальные.
25. Создание собственных функций для типовых расчетов.
26. Работа с текстовыми файлами: открытие, чтение, запись.
27. Менеджер контекста with...as для работы с файлами.
28. Формат CSV: структура, чтение с помощью модуля csv.
29. Запись результатов в CSV-файл.

Тематика вопросов для итоговой контрольной работы:

Формируемые компетенции и индикаторы их достижения: ОПК-1 (ОПК-1.1 и ОПК-1.2), и ПК-4 (ПК-4.3).

1. Понятие библиотеки (модуля) в Python. Установка сторонних библиотек с помощью pip.
2. Библиотека NumPy: преимущества перед стандартными списками.
3. Создание массивов NumPy (ndarray) из списка и с помощью функций (arange, zeros, ones).
4. Индексация и срезы в многомерных массивах NumPy.
5. Универсальные функции (ufunc) для поэлементных операций в NumPy.
6. Агрегирующие функции NumPy: sum, mean, std, min, max.
7. Библиотека Pandas: структуры Series и DataFrame.
8. Чтение данных из CSV-файла в DataFrame (read_csv). Основные параметры.

9. Первичный анализ DataFrame: атрибуты shape, columns, dtypes; методы info(), describe(), head(), tail().
10. Выборка данных в Pandas: обращение по столбцам, loc, iloc.
11. Фильтрация строк в DataFrame по условию.
12. Добавление и удаление столбцов в DataFrame.
13. Работа с пропущенными значениями (NaN): обнаружение (isna), удаление (dropna), заполнение (fillna).
14. Сортировка данных в Pandas: sort_values.
15. Группировка данных: метод groupby и его возможности.
16. Агрегация данных после группировки: agg с несколькими функциями.
17. Анализ частот: value_counts. Таблицы сопряженности: crosstab.
18. Объединение DataFrame: merge (по ключу) и concat (по осям).
19. Библиотека Matplotlib: базовая структура графика (figure, axes).
20. Построение линейных графиков (plot), гистограмм (hist), ящиков с усами (boxplot).
21. Построение точечных диаграмм (scatter) для выявления корреляции.
22. Настройка графиков: заголовки, подписи осей, легенда, сохранение (savefig).
23. Библиотека Seaborn: особенности, стили, преимущества перед Matplotlib.
24. Построение статистических графиков в Seaborn: histplot, boxplot, violinplot.
25. Визуализация взаимосвязей: lmpplot, regplot, heatmap (корреляционная матрица).
26. Основы веб-скрапинга: библиотека Requests (получение страницы).
27. Парсинг HTML с помощью BeautifulSoup: поиск элементов (find, find_all).
28. Извлечение текста и атрибутов из HTML-элементов.
29. Понятие API. Отправка GET-запросов через requests.get().
30. Обработка JSON-ответов от API. Преобразование в DataFrame.

Контрольные вопросы для зачета/экзамена

Формируемые компетенции и индикаторы их достижения: ОПК-1 (ОПК-1.1 и ОПК-1.2), и ПК-4 (ПК-4.3).

1. Роль и место языков программирования (в частности Python) в работе современного цифрового социолога.
2. Базовые типы данных в Python: числовые (int, float), логический (bool), строковый (str). Операции с ними.
3. Строки в Python: методы для работы с текстом (split, join, replace, lower, upper). Форматирование строк (f-строки).

4. Списки (list): создание, индексация, срезы, основные методы (append, extend, remove, pop). Использование списков для хранения данных.
5. Кортежи (tuple) и множества (set): отличия от списков, область применения в социологических задачах.
6. Словари (dict): структура «ключ-значение», методы работы, применение для хранения структурированной информации.
7. Условный оператор if-elif-else. Логические операторы (and, or, not). Примеры сегментации респондентов.
8. Цикл while: синтаксис, применение для обработки данных с неизвестным количеством итераций.
9. Цикл for: перебор последовательностей, функция range(). Вложенные циклы.
10. Операторы break и continue: управление выполнением цикла.
11. Функции в Python: определение (def), аргументы, возврат значения (return). Области видимости переменных.
12. Создание собственных функций для типовых задач анализа социологических данных.
13. Работа с текстовыми файлами: чтение и запись. Режимы открытия файлов.
14. Работа с файлами формата CSV: модуль csv, чтение и запись табличных данных.
15. Обработка исключений: конструкция try-except-else-finally. Повышение устойчивости программ.
16. Библиотека NumPy: массивы (ndarray), создание, индексация, универсальные и агрегирующие функции.
17. Библиотека Pandas: структура DataFrame. Чтение данных из CSV (read_csv). Первичный анализ (info, describe, head).
18. Выборка данных в Pandas: по меткам (loc) и по позициям (iloc). Фильтрация данных по условию.
19. Обработка пропущенных значений в Pandas: dropna, fillna.
20. Группировка и агрегация данных в Pandas: groupby, agg, применение нескольких функций.
21. Анализ частот и таблицы сопряженности: value_counts, crosstab.
22. Библиотека Matplotlib: построение базовых графиков (plot, hist, boxplot, scatter). Настройка параметров.
23. Библиотека Seaborn: статистическая визуализация (histplot, boxplot, lplot, heatmap).
24. Основы веб-скрапинга: библиотеки Requests и BeautifulSoup. Этические аспекты.

25. Основы работы с API: понятие, отправка GET-запросов, обработка JSON-ответов.
26. Итераторы и генераторы: понятие, применение для обработки потоковых данных.
27. List comprehension (генераторы списков) как эффективный способ создания списков.
28. Основы отладки кода: использование print, отладчик IDE, логирование.
29. Стандарты оформления кода PEP 8: читаемость и поддерживаемость программ.
30. Полный цикл анализа данных на Python: от загрузки до визуализации (на примере конкретной задачи).

6. Учебно-методическое и информационное обеспечение дисциплины

6.1 Список источников и литературы

Основная литература

1. Лутц, М. Изучаем Python. — 5-е изд. — М.: Вильямс, 2020. — 1232 с. (Фундаментальный учебник по языку).
2. Маккинни, У. Python и анализ данных. — 3-е изд. — М.: Вильямс, 2023. — 576 с. (Библия Pandas от создателя библиотеки).
3. Гэддис, Т. Начинаем программировать на Python. — 5-е изд. — СПб.: БХВ-Петербург, 2022. — 800 с. (Отличный учебник для начинающих).
4. Свейгарт, Э. Автоматизация рутинных задач с помощью Python. — 2-е изд. — М.: Вильямс, 2021. — 592 с. (Практические примеры для автоматизации).
5. Вандер Плас, Дж. Python для сложных задач: наука о данных и машинное обучение. — СПб.: Питер, 2021. — 576 с. (NumPy, Pandas, Matplotlib, Scikit-learn).

Дополнительная литература

1. Мэтиз, Э. Изучаем Python: программирование игр, визуализация данных, веб-приложения. — 3-е изд. — СПб.: Питер, 2023. — 512 с.
2. Любанович, Б. Простой Python. Современный стиль программирования. — 2-е изд. — СПб.: Питер, 2021. — 592 с.
3. Седер, Н. Python для анализа данных. — 2-е изд. — М.: ДМК Пресс, 2022. — 540 с.
4. Хейдт, М. Изучаем Pandas. — М.: ДМК Пресс, 2023. — 432 с.
5. Моллер, А. Введение в машинное обучение с использованием Python. — СПб.: БХВ-Петербург, 2022. — 480 с.
6. Бизли, Д. Python. Подробный справочник. — 4-е изд. — СПб.: Символ-Плюс, 2021. — 1152 с.
7. Рамал, М. Python для сложных сетей. Анализ и визуализация данных. — М.: ДМК Пресс, 2022. — 400 с.
8. VanderPlas, J. Python Data Science Handbook. — 2nd ed. — O'Reilly Media, 2023. — 600 p.
9. McKinney, W. Python for Data Analysis. — 3rd ed. — O'Reilly Media, 2022. — 580 p.
10. Grus, J. Data Science from Scratch. — 2nd ed. — O'Reilly Media, 2019. — 406 p. (Актуальность сохраняется).
11. Matthes, E. Python Crash Course. — 3rd ed. — No Starch Press, 2023. — 552 p.
12. Severance, C. Python for Everybody: Exploring Data in Python 3. — CreateSpace, 2020. — 242 p.

13. Downey, A. B. Think Python: How to Think Like a Computer Scientist. — 3rd ed. — O'Reilly Media, 2024. — 400 p.
14. Lutz, M. Learning Python. — 6th ed. — O'Reilly Media, 2024. — 1500 p.
15. Hill, C. Python for Data Science For Dummies. — 3rd ed. — For Dummies, 2023. — 496 p.
16. Mueller, J. P. Beginning Programming with Python For Dummies. — 3rd ed. — For Dummies, 2023. — 416 p.
17. Beazley, D., Jones, B. K. Python Cookbook. — 4th ed. — O'Reilly Media, 2022. — 850 p.
18. Summerfield, M. Programming in Python 3. — 2nd ed. — Addison-Wesley, 2021. — 656 p.
19. Zinoviev, D. Data Science Essentials in Python. — Pragmatic Bookshelf, 2021. — 350 p.
20. Klein, B. Python for Social Scientists. — Sage Publications, 2022. — 320 p. (Специализированное издание для социологов).
21. Жуковский, О. И. Программирование на Python для сбора и анализа данных. — М.: Юрайт, 2023. — 250 с.
22. Симдянов, И. В. Python. Мастер-класс. — СПб.: БХВ-Петербург, 2022. — 656 с.
23. Прохоренок, Н. А., Дронов, В. А. Python 3 и PyQt 6. Разработка приложений. — СПб.: БХВ-Петербург, 2023. — 832 с. (Для интересующихся созданием интерфейсов).
24. Копец, Д. Классические задачи Computer Science на языке Python. — СПб.: Питер, 2021. — 256 с.
25. Ларман, К. Применение UML и шаблонов проектирования. — 3-е изд. — М.: Вильямс, 2022. — 736 с. (Для понимания архитектуры ПО).
26. Хант, Э., Томас, Д. Программист-прагматик. Путь от подмастерья к мастеру. — М.: Вильямс, 2021. — 352 с. (Философия программирования).
27. Федоров, Д. Ю. Программирование на Python для решения инженерных и научных задач. — М.: ДМК Пресс, 2022. — 340 с.
28. Доусон, М. Програмируем на Python. — СПб.: Питер, 2021. — 416 с.
29. Лучано, Р. Python. К вершинам мастерства. — 2-е изд. — М.: ДМК Пресс, 2022. — 896 с.
30. Седжвик, Р., Уэйн, К. Программирование на Python. Том 1-2. — М.: Вильямс, 2023. — 1360 с.

При изучении дисциплины «Программирование на Python» студенту следует использовать монографии, учебники, учебные пособия, справочники (в том числе энциклопедические), а также *новейшие* научные публикации в *научных изданиях* (материалах научных и практических конференций, тематических сборниках и т.д.), в том числе периодических (журналах). Для поиска необходимых материалов помимо фондов научной библиотеки РГГУ и других библиотек целесообразно обращаться к ресурсам интернета.

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

Профессиональные базы данных и информационно-справочные системы

1. Национальная электронная библиотека (НЭБ): <http://www.rusneb.ru>
2. [ELibrary.ru](http://www.elibrary.ru) Научная электронная библиотека: <http://www.elibrary.ru>
3. Официальный сайт Microsoft. Поддержка
Access: <https://support.microsoft.com/ru-ru/access>
4. Онлайн-учебник по SQL (W3Schools): <https://www.w3schools.com/sql/>
5. Stepik. Курс «Основы SQL»: <https://stepik.org/course/63054/promo>
6. Единый архив экономических и социологических данных (ЕАЭСД) НИУ ВШЭ: <http://sophist.hse.ru/>
7. Федеральная служба государственной статистики (Росстат): <https://rosstat.gov.ru/>
8. Европейское социальное исследование (ESS): <https://www.europeansocialsurvey.org/>
9. World Values Survey (WVS): <http://www.worldvaluessurvey.org/>
10. Официальная документация PostgreSQL: <https://www.postgresql.org/docs/>
11. Официальная документация MySQL: <https://dev.mysql.com/doc/>
12. [SQL-ex.ru](https://sql-ex.ru) (тренажер по SQL): <https://sql-ex.ru/>

Иные ресурсы, необходимые для усвоения дисциплины

1. Официальная документация Python: <https://docs.python.org/3/>
2. Официальная документация Pandas: <https://pandas.pydata.org/docs/>
3. Официальная документация NumPy: <https://numpy.org/doc/>
4. Официальная документация Matplotlib: <https://matplotlib.org/stable/contents.html>
5. Stepik. Курс «Программирование на Python»: <https://stepik.org/course/67/promo>
6. Stepik. Курс «Анализ данных на Python»: <https://stepik.org/course/724/promo>
7. W3Schools Python Tutorial: <https://www.w3schools.com/python/>
8. Real Python (туториалы и статьи): <https://realpython.com/>
9. Kaggle Learn Python: <https://www.kaggle.com/learn/python>
10. Stack Overflow (форум для вопросов): <https://stackoverflow.com/>
11. GitHub (репозитории с открытым кодом): <https://github.com/>
12. Единый архив экономических и социологических данных (ЕАЭСД) НИУ ВШЭ: <http://sophist.hse.ru/> (источники данных)

Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: <https://liber.rsu.ru/ru/bases>

Информационные справочные системы:

1. Консультант Плюс

2. Гарант

7. Материально-техническое обеспечение дисциплины

Для обеспечения дисциплины используется материально-техническая база образовательного учреждения: учебные аудитории, оснащённые компьютером и проектором для демонстрации учебных материалов.

Состав программного обеспечения:

1. Windows
2. Microsoft Office

Профессиональные полнотекстовые базы данных:

1. Национальная электронная библиотека (НЭБ) www.rusneb.ru
2. ELibrary.ru Научная электронная библиотека www.elibrary.ru
3. Электронная библиотека Grebennikon.ru www.grebennikon.ru
4. Cambridge University Press
5. ProQuest Dissertation & Theses Global
6. SAGE Journals
7. Taylor and Francis
8. JSTOR

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

1. для слепых и слабовидящих:
 - ~ лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - ~ письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
 - ~ обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
 - ~ для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
 - ~ письменные задания оформляются увеличенным шрифтом;

- ~ экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.
- 2. для глухих и слабослышащих:
 - ~ лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования;
 - ~ письменные задания выполняются на компьютере в письменной форме;
 - ~ экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.
- 3. для лиц с нарушениями опорно-двигательного аппарата:
 - ~ лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - ~ письменные задания выполняются на компьютере со специализированным программным обеспечением;
 - ~ экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

1. для слепых и слабовидящих:
 - ~ в печатной форме увеличенным шрифтом;
 - ~ в форме электронного документа;
 - ~ в форме аудиофайла.
2. для глухих и слабослышащих:
 - ~ в печатной форме;
 - ~ в форме электронного документа.

3. для обучающихся с нарушениями опорно-двигательного аппарата:
 - ~ в печатной форме;
 - ~ в форме электронного документа;
 - ~ в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

1. для слепых и слабовидящих:
 - ~ устройством для сканирования и чтения с камерой SARA CE;
 - ~ дисплеем Брайля PAC Mate 20;
 - ~ принтером Брайля EmBrailleViewPlus;
2. для глухих и слабослышащих:
 - ~ автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих;
 - ~ акустический усилитель и колонки;
3. для обучающихся с нарушениями опорно-двигательного аппарата:
 - ~ передвижными, регулируемые эргономическими партами СИ-1;
 - ~ компьютерной техникой со специальным программным обеспечением

9. Методические материалы

9.1 Планы практических занятий

Часть 1. Основы алгоритмизации и программирования на Python

Тема 1. Настройка среды разработки. Первая программа.

Вопросы для обсуждения и задания:

- ~ Установка интерпретатора Python и среды разработки PyCharm Community Edition / VS Code.
- ~ Создание первого проекта и Python-файла.
- ~ Команда `print()` для вывода текста.
- ~ Команда `input()` для ввода данных.
- ~ Написание программы, которая запрашивает имя пользователя и выводит приветствие.
- ~ Запуск скрипта разными способами (из IDE, из командной строки).

Литература: [3, осн.], [4, осн.], [1, доп.], Интернет-ресурс [1].

Тема 2. Переменные и базовые типы данных. Арифметические операции.

Вопросы для обсуждения и задания:

- ~ Создание переменных с различными типами данных (int, float, bool).
- ~ Выполнение арифметических операций (+, -, *, /, //, %, **).
- ~ Преобразование типов данных (int(), float(), str()).
- ~ Написание программы-калькулятора для расчета индекса массы тела (ИМТ) по формуле $\text{weight} / \text{height}^{**2}$.
- ~ Изучение функций type() и id().

Литература: [1, осн.], [3, осн.], [7, доп.].

Тема 3. Строки: индексация, срезы, методы.

Вопросы для обсуждения и задания:

- ~ Создание строковых переменных.
- ~ Индексация строк (положительная и отрицательная).
- ~ Извлечение подстрок с помощью срезов.
- ~ Применение методов: lower(), upper(), replace(), split(), join(), strip().
- ~ Написание программы, которая очищает введенный пользователем город от лишних пробелов и приводит к формату "Москва" (первая заглавная, остальные строчные).
- ~ Форматирование строк с помощью f-строк для вывода отчетов.

Литература: [1, осн.], [3, осн.], [2, доп.].

Тема 4. Управляющие конструкции: условный оператор if.

Вопросы для обсуждения и задания:

- ~ Написание программы с простым условием (например, проверка возраста для допуска к голосованию).
- ~ Использование полной формы if-else.
- ~ Составление сложных условий с and, or, not.
- ~ Практикум: сегментация респондентов по возрасту (ребенок, подросток, взрослый, пенсионер).
- ~ Программа проверки пароля на сложность

Литература: [3, осн.], [4, осн.], [7, доп.].

Тема 5. Множественное ветвление if-elif-else.

Вопросы для обсуждения и задания:

- ~ Написание программы с несколькими условиями (оценка уровня дохода: низкий, средний, высокий).
- ~ Конвертация числовых оценок в буквенные (A, B, C, D, E).
- ~ Вложенные условные операторы (например, внутри группы по возрасту дополнительная сегментация по полу).
- ~ Разбор кейса: присвоение социального статуса на основе возраста, образования и занятости

Литература: [3, осн.], [4, осн.], [7, доп.].

Тема 6. Списки: создание, индексация, методы.

Вопросы для обсуждения и задания:

- ~ Создание списка ответов респондентов.
- ~ Доступ к элементам по индексу и с помощью срезов.
- ~ Добавление элементов (append, extend, insert).
- ~ Удаление элементов (remove, pop, del).
- ~ Поиск индекса элемента (index) и подсчет количества (count).
- ~ Написание программы для хранения и вывода списка участников фокус-группы

Литература: [1, осн.], [3, осн.], [2, доп.].

Тема 7. Кортежи и множества.

Вопросы для обсуждения и задания:

- ~ Создание кортежа с координатами или фиксированными характеристиками.
- ~ Попытка изменения кортежа (получение ошибки).
- ~ Создание множества уникальных интересов/хобби из списка (удаление дубликатов).
- ~ Операции над множествами: поиск общих интересов у двух групп (пересечение), объединение групп (объединение), различие.
- ~ Практикум: анализ пересечения аудиторий двух пабликов в соцсетях.

Литература: [1, осн.], [3, осн.], [3, доп.].

Тема 8. Словари.

Вопросы для обсуждения и задания:

- ~ Создание словаря для хранения профиля респондента (ключи: имя, возраст, город, ответы на вопросы).

- ~ Доступ к значениям через квадратные скобки и метод `get()` (обработка отсутствия ключа).
 - ~ Добавление и изменение пар ключ-значение.
 - ~ Перебор словаря: получение ключей, значений, пар.
 - ~ Создание списка словарей (база данных респондентов в памяти).
 - ~ Поиск респондента по имени в списке словарей.
- Литература:* [1, осн.], [2, осн.], [3, доп.].

Тема 9. Цикл `while`.

Вопросы для обсуждения и задания:

- ~ Написание программы с циклом `while` для ввода данных до тех пор, пока пользователь не введет "стоп".
 - ~ Программа для подсчета суммы вводимых чисел.
 - ~ Использование `break` для выхода при определенном условии.
 - ~ Использование `continue` для пропуска некорректных значений.
 - ~ Игра «Угадай число» (компьютер загадывает, пользователь угадывает).
- Литература:* [1, осн.], [3, осн.], [4, доп.].

Тема 10. Цикл `for` и функция `range`.

Вопросы для обсуждения и задания:

- ~ Вывод чисел от 1 до 10 с помощью `for` и `range`.
 - ~ Перебор элементов списка с подсчетом их количества.
 - ~ Вычисление суммы и среднего арифметического элементов списка.
 - ~ Вложенные циклы: создание таблицы умножения.
 - ~ Практикум: обработка матрицы 3x3 (например, три вопроса у трех респондентов).
- Литература:* [1, осн.], [3, осн.], [2, доп.].

Тема 11. Генераторы списков.

Вопросы для обсуждения и задания:

- ~ Создание списка квадратов чисел с помощью цикла `for` и генератора списка (сравнение).
- ~ Фильтрация списка с помощью условия в генераторе (оставить только четные числа).
- ~ Создание списка длин строк из другого списка.

- ~ Применение генератора для обработки данных: извлечение всех возрастов из списка словарей в отдельный список.
 - ~ Генераторы множеств и словарей.
- Литература:* [1, осн.], [3, осн.], [2, доп.].

Тема 12. Функции: определение, аргументы, возврат значений.

Вопросы для обсуждения и задания:

- ~ Написание функции без параметров (приветствие).
 - ~ Написание функции с параметрами (расчет индекса социального самочувствия).
 - ~ Функция, возвращающая значение (расчет возраста по году рождения).
 - ~ Функция с несколькими параметрами и возвратом кортежа (расчет среднего и медианы списка).
 - ~ Использование значений параметров по умолчанию.
- Литература:* [1, осн.], [3, осн.], [2, доп.].

Тема 13. Функции: области видимости и документация.

Вопросы для обсуждения и задания:

- ~ Эксперименты с локальными и глобальными переменными.
 - ~ Попытка изменения глобальной переменной внутри функции и использование `global`.
 - ~ Написание `docstring` для созданных ранее функций.
 - ~ Создание библиотеки (модуля) из нескольких функций для социологических расчетов в отдельном файле.
 - ~ Импорт собственного модуля в основной скрипт.
- Литература:* [1, осн.], [3, осн.], [2, доп.].

Тема 14. Работа с текстовыми файлами.

Вопросы для обсуждения и задания:

- ~ Запись результатов работы программы (списка респондентов) в текстовый файл.
 - ~ Чтение данных из текстового файла (построчно) и вывод на экран.
 - ~ Использование менеджера контекста `with`.
 - ~ Добавление новых данных в существующий файл (режим 'a').
 - ~ Чтение файла и подсчет количества строк, слов, символов.
- Литература:* [4, осн.], [5, осн.], [4, доп.].

Тема 15. Работа с CSV-файлами.

Вопросы для обсуждения и задания:

- ~ Чтение CSV-файла с результатами опроса с помощью модуля csv.
- ~ Извлечение заголовков столбцов.
- ~ Подсчет количества респондентов, удовлетворяющих условию (например, возраст > 30) при чтении файла.
- ~ Запись результатов фильтрации (список отобранных респондентов) в новый CSV-файл.
- ~ Создание CSV-файла с итоговой статистикой (средний возраст по группам).

Литература: [4, осн.], [5, осн.], [4, доп.].

Тема 16. Обработка исключений.

Вопросы для обсуждения и задания:

- ~ Добавление try-excerpt в программу калькулятора для обработки деления на ноль и ввода букв вместо чисел.
- ~ Обработка ошибки FileNotFoundError при попытке открыть несуществующий файл.
- ~ Использование нескольких блоков excerpt для разных типов ошибок.
- ~ Добавление блоков else и finally.
- ~ Практикум: повышение устойчивости скрипта загрузки данных из "грязного" CSV.

Литература: [1, осн.], [3, осн.], [7, доп.].

Часть 2. Библиотеки для анализа данных и автоматизации

Тема 17. Введение в NumPy: создание массивов.

Вопросы для обсуждения и задания:

- ~ Установка библиотеки NumPy.
- ~ Создание одномерных и двумерных массивов из списков.
- ~ Использование функций arange, zeros, ones, linspace.
- ~ Изучение атрибутов массивов: shape, ndim, size, dtype.
- ~ Изменение формы массива (reshape).

Литература: [2, осн.], [5, осн.], [3, доп.], [8, доп.].

Тема 18. NumPy: индексация, срезы и базовые операции.

Вопросы для обсуждения и задания:

- ~ Индексация и срезы одномерных и двумерных массивов.
- ~ Поэлементные арифметические операции с массивами.
- ~ Универсальные функции (sqrt, exp, log).
- ~ Булева индексация для фильтрации массива (выбрать все элементы > 0).
- ~ Агрегирующие функции: sum, mean, std, min, max по всему массиву и по осям (axis=0, axis=1).

Литература: [2, осн.], [5, осн.], [3, доп.], [8, доп.].

Тема 19. Введение в Pandas: Series. *Вопросы для обсуждения и задания:*

- ~ Установка библиотеки Pandas.
- ~ Создание Series из списка и из словаря.
- ~ Доступ к элементам по позиции и по индексу-метке.
- ~ Векторизованные операции с Series.
- ~ Применение методов describe, unique, value_counts.
- ~ Обработка пропусков в Series (isna, fillna).

Литература: [2, осн.], [5, осн.], [4, доп.], [9, доп.].

Тема 20. Pandas: создание DataFrame.

Вопросы для обсуждения и задания:

- ~ Создание DataFrame из словаря списков.
- ~ Создание DataFrame из списка словарей (как из базы данных).
- ~ Просмотр данных: head, tail, sample.
- ~ Атрибуты: index, columns, values, shape.
- ~ Получение информации о данных: info, dtypes.
- ~ Генерация описательной статистики: describe.

Литература: [2, осн.], [5, осн.], [4, доп.], [9, доп.].

Тема 21. Pandas: загрузка данных из CSV.

Вопросы для обсуждения и задания:

- ~ Загрузка реального датасета (например, данные опроса в формате CSV) с помощью read_csv.
- ~ Настройка параметров загрузки: sep, encoding, header.
- ~ Сохранение DataFrame в CSV (to_csv).
- ~ Просмотр загруженных данных (первые строки, информация, описательная статистика).

- ~ Импорт данных из Excel (read_excel) — краткое знакомство.
- Литература:* [2, осн.], [5, осн.], [4, доп.], [9, доп.].

Тема 22. Pandas: выборка данных (loc, iloc).

Вопросы для обсуждения и задания:

- ~ Выбор одного столбца (два способа) — результат Series.
 - ~ Выбор нескольких столбцов — результат DataFrame.
 - ~ Выбор строк по индексу (iloc).
 - ~ Выбор строк по метке (loc).
 - ~ Смешанная выборка (строки и столбцы).
 - ~ Добавление новых столбцов на основе существующих.
- Литература:* [2, осн.], [5, осн.], [4, доп.], [9, доп.].

Тема 23. Pandas: фильтрация данных.

Вопросы для обсуждения и задания:

- ~ Булева индексация: отбор строк по одному условию (df[df.age > 30]).
 - ~ Составные условия с &, |, ~ (обязательно скобки вокруг каждого условия).
 - ~ Метод isin() для отбора по списку значений.
 - ~ Метод between() для отбора по диапазону.
 - ~ Фильтрация строк, содержащих определенный текст (str.contains).
 - ~ Практикум: выделение подвыборки респондентов (мужчины, 25-40 лет, с высшим образованием).
- Литература:* [2, осн.], [5, осн.], [4, доп.], [9, доп.].

Тема 24. Pandas: работа с пропущенными значениями.

Вопросы для обсуждения и задания:

- ~ Обнаружение пропусков: isna(), sum() для подсчета по столбцам.
 - ~ Визуализация пропусков (библиотека missingno — краткая демонстрация).
 - ~ Удаление строк с пропусками (dropna) — параметры thresh, subset.
 - ~ Удаление столбцов с пропусками.
 - ~ Заполнение пропусков константой (fillna).
 - ~ Заполнение пропусков средним/медианой по столбцу.
 - ~ Интерполяция временных рядов.
- Литература:* [2, осн.], [5, осн.], [4, доп.], [9, доп.].

Тема 25. Pandas: сортировка и группировка данных.

Вопросы для обсуждения и задания:

- ~ Сортировка DataFrame по одному столбцу (sort_values).
- ~ Сортировка по нескольким столбцам.
- ~ Группировка данных по категориальному признаку: groupby.
- ~ Применение агрегатных функций к группам: mean, sum, count.
- ~ Использование agg() для разных функций к разным столбцам.
- ~ Практикум: расчет среднего дохода по полу и уровню образования.

Литература: [2, осн.], [5, осн.], [4, доп.], [9, доп.].

Тема 26. Pandas: анализ частот и таблицы сопряженности.

Вопросы для обсуждения и задания:

- ~ Анализ частот категориальных переменных: value_counts(), normalize=True.
- ~ Построение таблиц сопряженности: pd.crosstab().
- ~ Добавление маргинальных итогов (normalize='index', margins=True).
- ~ Визуализация таблиц сопряженности с помощью тепловых карт Seaborn.
- ~ Практикум: анализ зависимости между удовлетворенностью жизнью и уровнем дохода.

Литература: [2, осн.], [5, осн.], [4, доп.], [9, доп.].

Тема 27. Pandas: объединение данных (merge, concat).

Вопросы для обсуждения и задания:

- ~ Вертикальное объединение двух DataFrame с одинаковыми столбцами (concat).
- ~ Горизонтальное объединение (concat по axis=1).
- ~ Соединение таблиц по ключу: inner merge (только совпадающие ключи).
- ~ Left merge (все из левой таблицы, только совпадающие из правой).
- ~ Практикум: объединение таблицы с демографическими данными и таблицы с ответами на вопросы по ID респондента.

Литература: [2, осн.], [5, осн.], [4, доп.], [9, доп.].

Тема 28. Введение в Matplotlib: линейные графики.

Вопросы для обсуждения и задания:

- ~ Установка Matplotlib.
- ~ Построение простого линейного графика (plot).
- ~ Построение нескольких линий на одном графике.
- ~ Добавление заголовка, подписей осей, легенды.

- ~ Настройка стилей линий и маркеров.
 - ~ Изменение размера графика (figsize).
 - ~ Сохранение графика в PNG.
- Литература:* [5, осн.], [8, доп.], [4, доп.].

Тема 29. Matplotlib: гистограммы и ящики с усами.

Вопросы для обсуждения и задания:

- ~ Построение гистограммы распределения возраста (hist).
 - ~ Эксперименты с количеством бинов (bins).
 - ~ Построение ящиков с усами (boxplot) для сравнения распределения дохода по группам (например, по образованию).
 - ~ Добавление подписей к группам.
 - ~ Создание нескольких подграфиков на одной фигуре (subplots).
- Литература:* [5, осн.], [8, доп.], [4, доп.].

Тема 30. Matplotlib и Seaborn: точечные диаграммы и корреляции.

Вопросы для обсуждения и задания:

- ~ Построение точечной диаграммы (scatter) для изучения связи между возрастом и доходом.
 - ~ Добавление регрессионной линии с помощью `np.polyfit` и `plot`.
 - ~ Использование Seaborn: построение `lmpplot` (точечная диаграмма с регрессией).
 - ~ Расчет корреляционной матрицы с помощью Pandas (`df.corr()`).
 - ~ Визуализация корреляционной матрицы с Seaborn (`heatmap`).
- Литература:* [5, осн.], [8, доп.], [4, доп.].

Тема 31. Введение в веб-скрапинг (BeautifulSoup).

Вопросы для обсуждения и задания:

- ~ Установка библиотек `requests` и `beautifulsoup4`.
 - ~ Получение HTML-страницы с помощью `requests.get()`.
 - ~ Создание объекта BeautifulSoup.
 - ~ Поиск элементов по тегу (`find`, `find_all`).
 - ~ Поиск по классу и `id`.
 - ~ Извлечение текста и ссылок.
 - ~ Практикум: сбор названий и цен книг с сайта-парсера (учебного).
- Литература:* [4, осн.], [20, доп.], Интернет-ресурс [9].

Тема 32. Работа с API. Итоговый проект.

Вопросы для обсуждения и задания:

- ~ Регистрация и получение API-ключа (для демо-API).
- ~ Отправка GET-запроса к API (например, JSONPlaceholder или публичное API ВКонтакте).
- ~ Обработка JSON-ответа (`response.json()`).
- ~ Извлечение нужных данных и создание DataFrame.
- ~ Начало работы над итоговым проектом: выбор темы, формулировка задачи, поиск источника данных (CSV, API, парсинг).

Литература: [4, осн.], [20, доп.], Интернет-ресурс [9].

9.2. Методические указания

1. Освоение дисциплины строится по принципу «от задачи — к коду»: прежде чем писать программу, сформулируйте исследовательскую задачу на естественном языке, затем разбейте её на последовательные шаги (алгоритм) и только после этого приступайте к реализации на Python.
2. На лекционных занятиях фиксируйте не только синтаксис конструкций, но и типовые сценарии их применения в социологических задачах: где именно в анализе данных пригодится словарь, а где — множества, для чего нужна группировка в Pandas и какой график выбрать для визуализации распределения.
3. Выполнение лабораторных работ обязательно сопровождайте экспериментированием: не ограничивайтесь решением по образцу, изменяйте параметры функций, добавляйте новые условия, намеренно создавайте ошибки и учитесь их исправлять — это единственный способ перейти от заучивания к пониманию.
4. При изучении новых библиотек (NumPy, Pandas, Matplotlib) используйте официальную документацию и примеры как основной источник информации. Умение читать документацию — важнейший профессиональный навык, который останется с вами даже после смены версий языка и библиотек.
5. Ведите личный репозиторий решений (GitHub или просто папку с скриптами), где сохраняйте удачные реализации типовых задач: загрузка CSV, очистка данных, расчет статистик, построение графиков. Это станет вашей персональной библиотекой кода для будущих исследований.
6. При подготовке к контрольным работам и экзамену акцентируйте внимание не на заучивании синтаксиса, а на понимании логики работы конструкций: объясните,

почему в данной ситуации нужно использовать цикл, а не генератор списка, почему здесь подходит словарь, а не список, почему выбран именно этот тип визуализации.

7. При работе над итоговым проектом строго соблюдайте принцип воспроизводимости исследования: ваш код должен быть самодостаточным, содержать комментарии, а выходные данные (графики, таблицы) должны генерироваться автоматически при запуске скрипта. Это обеспечит возможность проверки результатов и их повторного использования.
8. При возникновении ошибок и затруднений не пытайтесь запомнить все правила, а осваивайте методы поиска решений: формулируйте запрос для поисковых систем, обращайтесь к Stack Overflow, учите читать сообщения об ошибках (traceback) — именно эти навыки отличают профессионального разработчика от человека, просто выучившего синтаксис.

Приложение 1. Аннотация рабочей программы дисциплины

АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Дисциплина «Программирование на Python» реализуется на социологическом факультете кафедрой фундаментальной и прикладной математики.

Цель дисциплины: формирование у студентов системного представления о фундаментальных основах алгоритмизации и программирования на языке Python как универсальном инструменте решения широкого круга профессиональных задач цифрового социолога, а также развитие устойчивых практических навыков разработки программных средств для сбора, обработки, анализа и визуализации социальной информации.

Задачи:

1. Сформировать понимание роли программирования в структуре современного инструментария цифрового социолога и преимуществ Python для анализа данных.
2. Обучить фундаментальным концепциям: переменные, типы данных, управляющие конструкции, функции.
3. Привить навыки работы со структурами данных (списки, словари) для хранения и обработки социологической информации.
4. Обучить чтению и записи файлов (CSV, TXT) как основным источникам эмпирических данных.
5. Развить навыки использования специализированных библиотек: NumPy (числовые вычисления), Pandas (табличный анализ), Matplotlib/Seaborn (визуализация).
6. Дать начальные представления о веб-скрапинге и работе с API для сбора данных из интернет-источников.
7. Воспитать культуру написания читаемого и документированного кода (PEP 8).

Дисциплина направлена на формирование следующих компетенций студентов:

- ~ **ОПК-1.1** понимает содержание, структуру и принципы работы современных информационных технологий, применяемых для решения задач профессиональной деятельности;
- ~ **ОПК-1.2** использует современные информационные технологии при решении задач профессиональной деятельности.
- ~ **ПК-4.3** обрабатывает, анализирует и представляет собранные данные с применением информационных технологий

В результате освоения дисциплины обучающийся должен:

- ~ **Знать:** синтаксис и семантику базовых конструкций Python; встроенные типы данных и структуры; методы работы с файлами; возможности библиотек NumPy, Pandas, Matplotlib для анализа и визуализации данных; основы веб-скрапинга и работы с API.
- ~ **Уметь:** разрабатывать скрипты для автоматизации обработки данных; применять Pandas для фильтрации, группировки и агрегации данных; строить графики и диаграммы; интерпретировать результаты вычислений; собирать данные из открытых источников.

Владеть: навыками написания программ на Python в среде разработки; техниками отладки и обработки ошибок; методами анализа табличных данных с помощью Pandas; инструментами визуализации Matplotlib/Seaborn.

По дисциплине предусмотрена промежуточная аттестация в форме зачета. Общая трудоемкость освоения дисциплины составляет 5 зачетных единиц.