Федеральное государственное бюджетное образовательное

учреждение высшего образования

Московский государственный университет имени М.В. Ломоносова

|  |
| --- |
| **УТВЕРЖДАЮ**  **Ректор МГУ имени М.В. Ломоносова**  **В.А. Садовничий**  **«\_\_» \_\_\_\_\_\_\_\_ 2023 г.** |

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)**

**Наименование дисциплины (модуля):**

**Межфакультетский курс «Основы программирования и анализа данных на Python»**

*наименование дисциплины (модуля)*

**Уровень высшего образования:**

***Бакалавриат, магистратура, специалитет***

*бакалавриат, магистратура, специалитет*

**Направление подготовки / специальность:**

**По всем направлениям подготовки / специальностям**

*(код и название направления/специальности)*

**Направленность (профиль)/специализация ОПОП:**

**По всем направленностям (профилям)/специальностям ОПОП**

*(если дисциплина реализуется в рамках направленности (профиля))*

Форма обучения:

Очная

*очная, очно-заочная*

Рабочая программа рассмотрена и одобрена

*Методической комиссией МГУ, на заседании Ученого совета МГУ*

\_\_.\_\_.2023

Москва 2023

Рабочая программа дисциплины (модуля) разработана в соответствии с самостоятельно установленными МГУ образовательными стандартами (ОС МГУ) для реализуемых основных профессиональных образовательных программ высшего образования (*программы бакалавриата, магистратуры, реализуемых последовательно по схеме интегрированной подготовки; программы специалитета; программы магистратуры)*

**1.** Место дисциплины (модуля) в структуре ОПОП ВО:

*Межфакультетские учебные курсы относятся к вариативной части и являются дисциплиной по выбору (элективной).*

**2.** Цели и задачи дисциплины (модуля):

Межфакультетский курс посвящен основам программирования на языке Python. Слушатели курса познакомятся с основными конструкциями языка Python и парадигмами программирования, а также с инструментами для анализа и визуализации данных.

Цель дисциплины — изучение основ работы с текстовыми и числовыми данными на языке Python.

Задачи дисциплины:

— изучение базового синтаксиса языка Python 3 и коллекций «золотого фонда» Python 3;

— изучение базового функционала библиотек NumPy, Pandas и Matplotlib;

— знакомство с принципами анализа данных и методами визуализации данных.

**3.** Входные требования для освоения дисциплины (модуля), предварительные условия (если есть):

Изучение данной дисциплины базируется на следующих курсах:

— МФК “Введение в программирование на основе языка Python” (начальные навыки по программированию на Python упростят освоение программы дисциплины).

**4.** Результаты обучения по дисциплине (модулю):

|  |
| --- |
| **Планируемые результаты обучения по дисциплине (модулю)** |
| В результате освоения дисциплины обучающиеся должны:  **знать:**  - основной синтаксис алгоритмического языка программирования Python;  - основные типы объектов языка Python и основные операции с ними;  - способы работы с текстовыми файлами в различных форматах;  - возможности библиотеки Matplotlib по визуализации данных;  - как использовать массивы из библиотеки NumPy;  - как анализировать данные при помощи библиотеки Pandas;  - как получают и предобрабатывают данные для анализа;  - парадигмы программирования, применяемые для структурирования программного кода.  **уметь:**  - запускать скрипты на языке Python на локальном компьютере;  - читать программный код на Python и разбираться в его содержании;  - писать программный код на Python с использованием стандартных библиотек;  - пользоваться документацией языка Python и системой встроенной помощи;  - считывать данные из файлов в форматах CSV, XLS, XLSX, а также из сырого текста;  - выбирать оптимальные контейнеры Python 3 для хранения данных в зависимости от ситуации;  - строить диаграммы и графики при помощи Matplotlib;  - создавать массивы NumPy и осуществлять операции с ними;  - использовать библиотеку Pandas для простого анализа данных.  **владеть:**  - средами Jupyter Notebook, Jupyter Lab, Google Colab;  - навыком работы с классическими объектами языка Python. |

**5.** Объем дисциплины составляет 1 з.е. (36 часов, включая 24 часа на занятия лекционного типа и 12 часов на самостоятельную работу обучающихся).

|  |  |
| --- | --- |
| **Вид учебной работы** | **Всего часов** |
| **Аудиторные занятия (всего)** | **24** |
| В том числе: |  |
| Лекции (Л) | 24 |
| Практические занятия (ПЗ) | - |
| Семинары (С) | - |
| Лабораторные работы (ЛР) | - |
| **Самостоятельная работа (всего)** | **12** |
| В том числе: |  |
| Подготовка к семинарским занятиям | - |
| Подготовка домашних заданий | 10 |
| Решение профессиональных задач и т.д. | - |
| Подготовка к аттестации | 2 |
| Вид промежуточной аттестации (зачет, экзамен) | зачет |
| Общая трудоемкость: часы  зачетные единицы | **36** |
| **1** |

**6.** Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и виды учебных занятий.

**Содержательный план дисциплины**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование раздела дисциплины**  **(элемента модуля)** | **Лекции** | **Практические**  **занятия** | **Семинары** | **Самостоятельная работа** | **Всего** |
| 1. | **Тема 1. Алгоритмы и язык Python**  — Язык Python — универсальный, кросплатформенный, красивый  — Демонстрация того, что можно сделать на языке Python  — Что такое алгоритм и почему нельзя описать его словами  — Основы программирования с исполнителем turtle  — Последовательное исполнение  — Циклы и вложенные циклы  — Простые функции-«процедуры» без параметров  — Гуманитарная составляющая программирования — названия объектов  — Юридическая составляющая программирования — контракты функций  — Физическая составляющая программирования — моделирование движения  — Эстетическая составляющая программирования — фракталы, графика | **2** |  |  | **1** | **3** |
| 2. | **Тема 2. Текстовые данные**  — Работа в интерактивном режиме ipython  — Среды Jupyter Lab, Google Colab  — Ввод и вывод текстовых данных. Функции input и print  — Операции + и \* со строками  — Ошибки и как к ним правильно относиться  — Интроспекция в Python: dir, type, id  — Где искать помощь? Метод help. Документация на сайте python.org  — Строка как объект. Тип str. Методы объекта строки: lower, upper, replace и т. д.  — Автоматическая сборка мусора  — Цикл for для посимвольной обработки строки  — Срезы строк с одним, двумя и тремя индексами: s[start:stop:step]  — Текстовые файлы: как открыть и прочитать. Как и зачем закрывать  — Менеджер контекста with для файлов  — Цикл for для перебора строк файла  — Вложенные циклы for | **2** |  |  | **1** | **3** |
| 3. | **Тема 3. Итерируемые объекты**  — Кортеж имён. Разбиение строки на слова методами split и splitlines  — Тип list. Конструирование списка  — Список строк. Срезы списка words[start:stop:step]  — Генераторы и списковые включения («генераторы списков»)  — Функция map. Функция как параметр функции (объект 1-го рода)  — Безымянные функции lambda.  — Фильтрация при помощи filter  — Библиотека itertools. Комбинации, перестановки, декартово произведение  — Цепочки. Повторение и зацикливание. Срез последовательности islice  — Функция zip  — Запись в файл. Сборка списка строк методом join | **2** |  |  | **1** | **3** |
| 4. | **Тема 4. Числовые данные**  — Представление данных в памяти компьютера. Двоичная система счисления  — Числовые данные в памяти ПК  — Арифметическая прогрессия range(start, stop, step)  — «Переменные» в Python. Подсчёт ситуаций, суммирование чисел  — Цикл while. Генерация геометрической прогрессии  — Расчет таблицы долга по кредиту  — Ветвление хода исполнения программы. Каскадная конструкция if, elif, else  — Создание вручную файла CSV | **2** |  |  | **1** | **3** |
| 5. | **Тема 5. Сортировка данных**  — Изменяемость списка list в Python  — Срезы списков. Присваивание в срез  — Сортировка списка. Использование ключа сортировки lambda  — Методы списка. Копирование объектов list.  — Ссылочная модель данных. Операторы == и is  — Двумерные массивы (списки списков). Вложенная генерация | **2** |  |  | **1** | **3** |
| 6. | **Тема 6. Частотный анализ и группировка данных**  — Частотный анализ. Почему не эффективно делать просто s.count(a)?  — Тип set. Множества и работа с ними  — Тип dict. Словарь (ассоциативный массив) и операции с ним  — Dict comprehensions: генерация множеств и словарей  — Частотный анализ (метод подсчёта) со словарём  — Функция itertools.group. Ключ группировки | **2** |  |  | **1** | **3** |
| 7. | **Тема 7. Построение графиков в Matplotlib**  — Дробные числа. Числа с плавающей точкой и ошибки вычислений.  — Модуль math. Виды округлений: floor, ceil, round.  — Стандартные математические функции: тригонометрические, экспонента, логарифм.  — Установка библиотек NumPy и Matplotlib.  — Построение графика функции. Описание def пользовательской функции.  — Вывод экспериментальных данных с ошибками.  — Зачем нужны массивы np.array вместо списков list?  — np.arange и математические функции в NumPy.  — Срезы массивов в NumPy. | **2** |  |  | **1** | **3** |
| 8. | **Тема 8. Библиотека NumPy**  — Генерация случайных чисел  — Статистика выборки: среднее арифметическое, среднеквадратическое отклонение, медиана, максимум, минимум  — Случайные распределения  — Откуда возникают матрицы и какие операции с ними возможны  — Основы работы с библиотекой NumPy | **2** |  |  | **1** | **3** |
| 9. | **Тема 9. Библиотека Pandas**  — Логический тип bool. Условные выражения. Функции all и any  — Индексы и срезы Datasheet  — Фильтрация Pandas Datasheet | **2** |  |  | **1** | **3** |
| 10. | **Тема 10. Библиотека Pandas**  — Сложные моменты при работе с данными в Pandas  — Предобработка данных  — Объединение данных из нескольких таблиц | **2** |  |  | **1** | **3** |
| 11. | **Тема 11. Источники данных**  — Получение готовых данных для анализа из открытых источников  — Базы данных | **2** |  |  | **1** | **3** |
| 12. | **Тема 12. Парадигмы программирования**  — Повторное использование программного кода и почему это важно  — Структурная парадигма  — Модульная парадигма  — Объектно-ориентированное программирование в Python | **2** |  |  | **1** | **3** |
|  | **Итого по курсу:** | **24** |  |  | **12** | **36** |

**7.** Фонд оценочных средств (ФОС, оценочные и методические материалы) для оценивания результатов обучения по дисциплине.

**7.1.** Критерии оценивания

Поскольку межфакультетский курс является лекционным, а аттестация предусмотрена в форме зачета без оценки, аттестация слушателей будет проводиться исходя из выполнения ими не менее 70% задач из каждого домашнего задания, которые будут представлены в форме задач с автоматической проверкой.

**7.2.** Список вопросов к зачету:

1. Какие методы работы со строками Вам известны?
2. Дана строка: s=‘Hello, wOrLd!’ Какие операции нужно выполнить, чтобы из строки получить список: lst=[‘hello’, ‘world’]
3. В чём отличие списка и кортежа?
4. В чём отличие списков (list) и массивов (array) в NumPy?
5. Меняет ли метод sort() список, к которому он применяется?
6. Какое соотношение (<, > или ==) можно поставить между строками ‘111’ и ’21’, если сравнивать их в лексикографическом порядке.
7. Как удалить из списка все повторяющиеся элементы?
8. В чём отличие словаря и множества кортежей из двух элементов?
9. Что такое .csv-файл?
10. Что такое DataFrame в Pandas и какие операции с ним Вам известны?
11. Отличие циклов for и while. В каких ситуациях лучше использовать for, а в каких while?
12. В каких случаях необходимо использовать генераторы?
13. Перечислите способы чтения данных из файлов в Python.
14. Какие параметры есть у метода sort()?
15. Какие структуры данных существуют в Python? В чем их ключевые отличия?
16. Какие типы данных есть в Python? Приведите примеры для каждого типа.
17. Графики и диаграммы каких типов можно построить с помощью библиотеки Matplotlib? Приведите пример.
18. Придумайте пример программы, в которой будет использоваться конструкция if..elif...else.
19. Почему “опасно” копировать списки при помощи обычного присваивания?

**8.** Ресурсное обеспечение.

Основная литература

1. Марк Саммерфилд. Программирование на Python 3. Подробное руководство. Издательство: Символ-Плюс. ISBN 978-5-93286-161-5, 978-0-13-712929-4, 2009 г.

Дополнительная литература

1. Марк Лутц. Python. Карманный справочник. Марк Лутц. Издательство: Вильямс ISBN 978-5-8459-1965-6, 2014 г.
2. Майкл Доусон. Программируем на Python. Издательство: Питер ISBN 978-5-459-00314-7, 978-1435455009, 2012 г.

**9.** Преподаватели дисциплины: Хирьянов Т.Ф., Мигачёва О.А., Горохов О.Е.