

**Задания к практическим занятиям по дисциплине**  
**"Алгоритмизация и программирование. Основы языка Python"**  
**весна 2024**

**Практическое занятие 1. Настройка PyCharm IDE. Создание первого проекта на Python. Работа с Git. Создание репозитория на GitHub, загрузка своего проекта.**

1.1) Откройте PyCharm IDE. Изучите основные возможности среды разработки и ее основные настройки.

1.2) Напишите первую программу, выводящую сообщение «Привет, мир!».

1.3) Изучите команды работы с Git.

1.4) Создайте свой репозиторий на GitHub.

1.5) Загрузите туда свой первый проект.

**Практическое занятие 2. Условные операторы.**

Решите следующие задачи, используя условные операторы.

2.1) При регистрации на сайтах требуется вводить пароль дважды. Это сделано для безопасности, поскольку такой подход уменьшает возможность неверного ввода пароля. Напишите программу, которая сравнивает пароль и его подтверждение. Если они совпадают, то программа выводит: «Пароль принят», иначе: «Пароль не принят».

2.2) Напишите программу, которая определяет, какой тип места в плацкартном вагоне (верхнее или нижнее, в купе или боковое) по заданному номеру места.

2.3) Год является високосным, если его номер кратен 4, но не кратен 100, или если он кратен 400. Напишите функцию, которая определяет, является ли год с данным номером високосным. Если год является високосным, то выведите «Год ... - високосный», где вместо многоточия выведите год, иначе выведите «Это год не високосный».

2.4) Красный, синий и желтый называются основными цветами, потому что их нельзя получить путем смешения других цветов. При смешивании двух основных цветов получается вторичный цвет:

- если смешать красный и синий, то получится фиолетовый;
- если смешать красный и желтый, то получится оранжевый;
- если смешать синий и желтый, то получится зеленый.

Напишите программу, которая считывает названия двух основных цветов для смешивания. Если пользователь вводит что-нибудь помимо названий «красный», «синий» или «желтый», то программа должна

вывести сообщение об ошибке. В противном случае программа должна вывести название вторичного цвета, который получится в результате.

### **Практическое занятие 3. Циклические алгоритмы.**

Решите следующие задачи, используя операторы циклов.

3.1) С клавиатуры вводятся поочередно N слов. Напишите программу, которая соединяет эти слова в одну длинную строку, разделяя слова пробелами. Используйте операторы цикла.

3.2) Модифицируйте предыдущую программу так, чтобы число вводимых слов не было задано, а программа работала до того момента, как пользователь введет слово «stop».

3.3) В игре в слова ценится наличие редких букв в словах. Считается, что буква "ф" встречается в русском языке реже всех остальных. Напишите программу, которая позволяет пользователям вводить какие-либо слова и проверяет, можно считать это слово редким или нет. Редкими будем считать слова, которые содержат букву "ф". Пусть эта программа выводит на экран одну из двух фраз: "Ого! Это редкое слово!", если в слове есть буква "ф", или "Эх, это не очень редкое слово...", если в нём этой буквы нет.

3.4) Напишите программу-игру «Математика для детей»: компьютер выводит выражение-сумму двух чисел, например:  $3 + 5 =$

Пользователь должен ввести ответ (курсор должен оставаться в одной строке с заданным выражением). Если ответ правильный, то вывести сообщение после результата «Правильно!», если ответ неправильный, то вывести сообщение «Ответ неверный».

Игра продолжается в цикле до тех пор, пока пользователь не сделает 3 ошибки. После этого вывести сообщение: «Игра окончена. Правильных ответов: ...» - вместо многоточия вывести количество правильных ответов.

### **Практическое занятие 4. Функции в Python.**

4.1) Напишите функцию, которая проверяет, делится ли введенное число на 3, или нет.

4.2) Напишите программу деления числа 100 на введенное пользователем число. Деление реализуйте с помощью функции. Предусмотрите возможные исключения (ValueError, возникающее в случае, если пользователь введет не число, а строку, и ZeroDivisionError – если будет введено число 0 и остальные).

4.3) Напишите функцию, которая возвращает True, если введенная пользователем дата является магической, и False в обратном случае. Магической считается дата, в которой произведение дня и месяца равно двум последним цифрам года, например: 02.11.2022.

4.4) "Счастливым" называют билет с номером, в котором сумма первой половины цифр равна сумме второй половины цифр. Номера могут быть произвольной длины, с единственным условием, что количество цифр всегда чётно, например: 33 или 2341 и так далее.

Билет с номером 385916 — счастливый, так как  $3 + 8 + 5 = 9 + 1 + 6$ . Билет с номером 231002 не является счастливым, так как  $2 + 3 + 1 \neq 0 + 0 + 2$ .

Реализуйте функцию, проверяющую является ли номер счастливым (номер — всегда строка).

### **Практическое занятие 5. Списки и кортежи.**

Решите следующие задачи.

5.1) Создайте список из пяти любых чисел. Спросите у пользователя число. Проверьте, есть ли данное число в списке. Выведите исходный список, число пользователя и соответствующие сообщение ("Поздравляю, Вы угадали число!" или "Нет такого числа!").

5.2) Создайте любой список. Определите, есть ли в списке повторяющиеся элементы, если да, то вывести на экран это значение.

5.3) Задан кортеж с перечнем названий дней недели. Спросить у пользователя, сколько выходных на неделе он хочет и вывести два списка:

"Ваши выходные дни: ..." - перечислить здесь столько дней недели с конца кортежа, сколько введено пользователем.

"Ваши рабочие дни: ..." - перечислить здесь оставшиеся дни недели.

5.4) Создайте два списка: один из 10 фамилий студентов Вашей группы, другой из 10 фамилий студентов другой группы.

а. Создайте спортивную команду (объедините в один кортеж) по 5 любых студентов из каждой группы.

б. Выведите на экран исходные списки групп и новый полученный кортеж.

в. Выведите его длину.

г. Отсортируйте кортеж по алфавиту.

д. Определите, входит ли в полученную команду студент "Иванов". И сколько раз встречается эта фамилия в кортеже.

### **Практическое занятие 6. Словари. Множества.**

Решите следующие задачи.

6.1) Создайте словарь, содержащий перечень стран и их столиц.

а) Выведите на экран все пары ключ-значение.

б) Выведите на экран столицу для определенной страны.

в) Отсортируйте и выведите на экран содержимое словаря в алфавитном порядке названий стран.

6.2) В игре в слова «Эрудит» каждая **буква** имеет определенную ценность:

А, В, Е, И, Н, О, Р, С, Т – 1 очко;

Д, К, Л, М, П, У – 2 очка;

Б, Г, Ё, Ъ, Я – 3 очка;

Й, Ы – 4 очка;

Ж, З, Х, Ц, Ч – 5 очков;

Ш, Э, Ю – 8 очков;

Ф, Щ, Ъ – 10 очков.

Напишите программу, которая вычисляет стоимость введенного пользователем **слова**.

6.3) \*Доп.задание на множество.

Есть множество студентов. Каждый из них знает некоторое количество языков. Нужно определить сколько различных языков знают студенты. Выведите отсортированный список этих языков. Выведите список студентов, которые знают китайский язык.

### **Практическое занятие 7. Обработка изображений.**

Решите следующие задачи.

7.1) Подготовьте любой графический файл для выполнения практической работы. Напишите программу, которая открывает и выводит этот файл на экран. Получите и выведите в консоль информацию о размере изображения, его формате, его цветовой модели.

7.2) Напишите программу, которая создаёт уменьшенную в три раза копию изображения. Получите горизонтальный и вертикальный зеркальный образ изображения. Сохраните изображения в текущую папку под новым именем.

7.3) Подготовьте 5 графических файлов с именами 1.jpg, 2.jpg, 3.jpg, 4.jpg, 5.jpg. Напишите программу, которая применит ко всем этим файлам сразу любой фильтр (кроме размытия, т.к. он рассматривался на лекции). Сохраните изображения в новую папку под новыми именами.

7.4) Напишите программу, которая добавляет на изображение водяной знак. Можно тоже применять сразу к нескольким изображениям.

### **Практическое занятие 8. Обработка изображений, продолжение.**

8.1) Скачайте любую открытку из интернета, определите область, которую Вам нужно вырезать из данного изображения (обрезать текст, часть фото и т.д.). Напишите программу, которая выполнит эту операцию. Сохраните изображения в текущую папку под новым именем.

**8.2)** Создайте словарь, содержащий перечень пары «Название праздника – имя\_файла с открыткой к нему». Спросите у пользователя, к какому празднику ему нужна открытка и выведите нужную открытку на экран.

**8.3)** Модифицируйте задачу 8.1 так: спросите еще у пользователя, имя того, кого он хочет поздравить, добавьте на заданную открытку текст «...., поздравляю!», где вместо .... вставьте полученное имя (выведите его разным цветом и шрифтами, посередине вверху или внизу фото). Найдите в сети интернет решение, как сделать надпись жирным текстом (по умолчанию, такого параметра нет). Сохраните новую открытку в файл с расширением png.

## **Практическое занятие 9. Работа с файлами.**

**9.1)** Модифицируйте программу из практики 7.3 (7 лабораторная работа ) или создайте заново: обработать любой операцией все картинки в заданной папке, используя для обхода файлов в папке модуль os (или Pathlib). При этом каталог для итоговых (обработанных) изображений должен тоже создаваться с помощью модуля os или Pathlib.

**9.2)** Модифицировать программу из практики 9.1, добавив проверку типа (расширения) файла, если в папке хранятся разные типы файлов, а вам нужно обработать только заданные (jpg, png).

**9.3)** Имеется файл с данными в формате csv:

Продукт,Количество,Цена

Молоко,2,80

Сыр,1,500

Хлеб,2,70

Напишите программу, которая считывает данные из этого файла, подсчитывает итоговую сумму расходов и выводит данные в виде:

Нужно купить:

Молоко - 2 шт. за 80 руб.

Сыр - 1 шт. за 500 руб.

Хлеб - 2 шт. за 70 руб.

Итоговая сумма: 800 руб.

## Практическое занятие 10. Работа с файлами. Продолжение

**10.1)** Имеется файл JSON с информацией о продуктах:

```
{
  "products": [
    {
      "name": "Шоколад",
      "price": 50,
      "available": true,
      "weight": 100
    },
    {
      "name": "Кофе",
      "price": 100,
      "available": false,
      "weight": 250
    },
    {
      "name": "Чай",
      "price": 70,
      "available": true,
      "weight": 50
    }
  ]
}
```

Напишите программу, которая считывает информацию из этого файла и выводит ее на экран в виде:

Название: Шоколад

Цена: 50

Вес: 100

В наличии

Название: Кофе

Цена: 100

Вес: 250

Нет в наличии!

Название: Чай

Цена: 70

Вес: 50

В наличии

**10.2)** Модифицируйте программу 10.1 – добавьте в нее код, который добавляет данные в файл JSON (спрашивает их у пользователя) и потом также выводит содержимое итогового файла на экран.

**10.3)** Создание русско-английского словаря.

Имеется файл en-ru.txt, в котором находятся строки **англо-русского словаря** в таком формате:

cat - кошка

dog - собака

home - домашняя папка, дом

mouse - мышь, манипулятор мышь

to do - делать, изготавливать

to make – изготавливать

и т.п.

Требуется создать **русско-английский словарь** и вывести его в файл ru-en.txt в таком формате:

делать – to do

дом – home

домашняя папка – home

изготавливать – to do, to make

кошка – cat

манипулятор мышь – mouse

мышь – mouse

собака – dog

Обратите внимание, что строки в выходном файле нужно отсортировать по алфавиту!

## **Практическое занятие 11. Объектно-ориентированное программирование**

**11.1)** Задача «Ресторан»: создайте класс с именем Restaurant. Метод `__init__()` класса Restaurant должен содержать два атрибута: `restaurant_name` (название ресторана) и `cuisine_type` (тип кухни). Создайте метод `describe_restaurant()`, который выводит два атрибута, и метод `open_restaurant()`, который выводит сообщение о том, что ресторан открыт. Создайте на основе своего класса экземпляр с именем `newRestaurant`. Выведите два атрибута по отдельности, затем вызовите оба метода.

**11.2)** На основе ранее созданного класса Restaurant из задания 11.1 создайте три разных экземпляра (три ресторана), вызовите для каждого экземпляра метод `describe_restaurant()`.

**11.3)** Добавьте в созданный класс Restaurant атрибут, задающий начальный рейтинг ресторана и метод, который получает на вход новое значение рейтинга и обновляет его.

## **Практическое занятие 12. ООП. Продолжение**

**12.1)** На основе ранее созданного класса Restaurant из прошлого задания создайте разновидность ресторана – «Кафе-мороженое». Напишите класс `IceCreamStand`, наследующий от класса Restaurant. Добавьте атрибут с именем `flavors` для хранения списка сортов мороженого. Напишите метод, который выводит этот список. Создайте экземпляр `IceCreamStand` и вызовите этот метод.

**12.2)** Модифицировать ранее созданный класс `IceCreamStand`:

1. Добавить атрибуты для описания локации и времени работы кафе-мороженого.



2. Реализовать методы для добавления и удаления сортов мороженого из списка flavors.
3. Реализовать метод для проверки наличия определенного сорта мороженого в списке flavors.
4. Добавить методы для разных типов мороженого (например, мороженое на палочке, мягкое мороженое, и т.д.) и реализовать методы для работы с каждым типом.

**12.3)** Создать класс IceCreamStand для кафе мороженого и используя библиотеку tkinter построить простой графический интерфейс, показанный на рисунке, приложенном к задаче.

### **Практическое занятие 13. Создание приложений с GUI. TKinter**

13.1 Создать приложение, аналогичное рассмотренному в примере «Приложение для отслеживания погоды» (см. файл «Практика 13\_Создание приложений с GUI. TKinter.pdf»).

13.2 Найдите в сети интернет данные об источниках с открытыми API и создайте приложение, которое выводит любые другие данные из этих источников, например, 2 исторических факта (достопримечательности) в заданном городе и т.п.

### **Практическое занятие 14. Создание приложений с GUI. PyQt**

14.1 Создать приложение, аналогичное рассмотренному в примере «Конвертор валют» (см. файл «Практика 14\_Создание приложений с GUI. PyQt.pdf»).

14.2 Просмотреть в интернете дополнительные уроки по работе с графическими библиотеками. Создать свое любое приложение с GUI (не игру!).