Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«СибирскИЙ государственнЫЙ Университет

геоСИСТЕМ И ТЕХНОЛОГИЙ»

(СГУГИТ)

ОТЧЕТ

ЛАБОРАТОРНАЯ РАБОТА 5. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ НА PYTHON

Выполнил обучающийся

Группы БИ-33

Ахиярова И.М.

Проверил

Воронкин Е.Ю.

Новосибирск – 2021

**Цель работы:** познакомиться с принципами объектно-ориентированного программирования на языке программирования python.

**Задачи работы**

1. Изучить основные конструкции, используемые при построении классов и объектов;
2. Приобрести навыки практического применения ООП.

**Перечень обеспечивающих средств**

Для выполнения работы необходимо иметь компьютер с установленной операционной системой семейства Windows, установленным python и IDE PyCharm Professional.

**Задание**

Вариант 4. Разработать приложение «Органайзер». Приложение предназначено для записи, хранения и поиска адресов и телефонов физических лиц и организаций, а также расписания, встреч и др. Приложение предназначено для любых пользователей компьютера.

**ОПИСАНИЕ ВЫПОЛНЕННЫХ РАБОТ**

Опишем схему работы приложения. Органайзер реализуется в виде объекта соответствующего класса и папки с файлами на диске. При создании объекта, программой создается папка с заданным пользователем названием и двумя текстовыми файлами в ней: списками «Люди» и «События». В эти файлы записываются данные, вводимые пользователем. Класс «Органайзер» предусматривает определенные методы: чтение списков, добавление записей в списки, поиск в списках.

Начнем работу над программой. В начале добавим модуль os для программного создания папки.



Рисунок 1 – Добавление модуля os

Создадим основной класс «Organizer» и его конструктор. При создании объекта ему задается имя, для него создается папка и два файла со списками. В данной реализации используется конструкция «try … except». Эта конструкция используется для выявления ошибки «FileExistsError», если органайзер уже существует, то процесс создания пропускается.

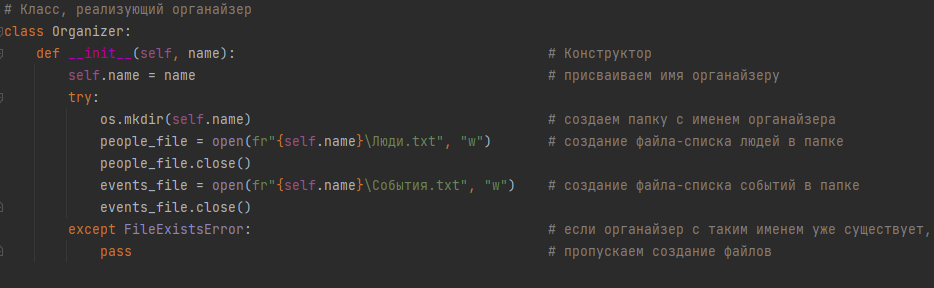


Рисунок 2 – Конструктор класса

Далее реализуем основные методы класса.

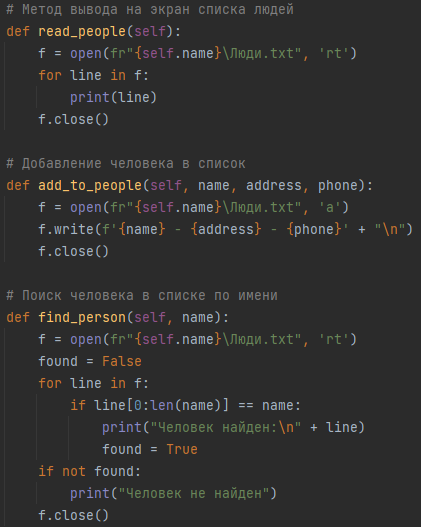


Рисунок 3 – Методы класса (1/2)

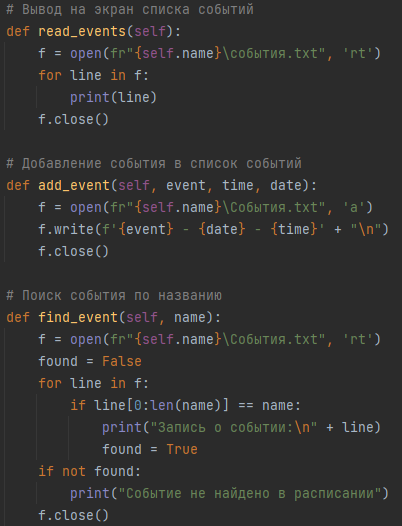


Рисунок 4 – Методы класса (2/2)

Методы для вывода списков на экран представляют собой построчное чтение текстовых файлов. Методы добавления записей в списки запрашивают необходимые аргументы и добавляют соответствующие записи. Методы поиска реализованы очень упрощенно – метод принимает искомое имя в качестве аргумента и проверяет каждую строку на совпадение первых символов искомому имени.

Теперь реализуем программу, использующую данный класс.

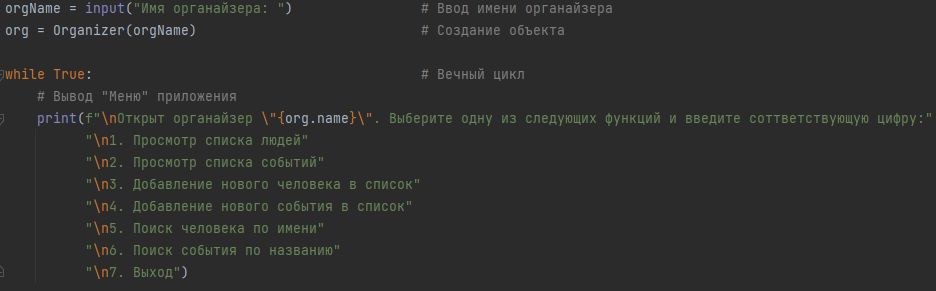


Рисунок 5 – Основная программа (1/2)

В начале своей работы, программа запрашивает у пользователя название органайзера и создает соответствующий объект. Затем программа переходит к основному циклу. Перед пользователем появляется «меню», в котором он может выбирать необходимую функцию.

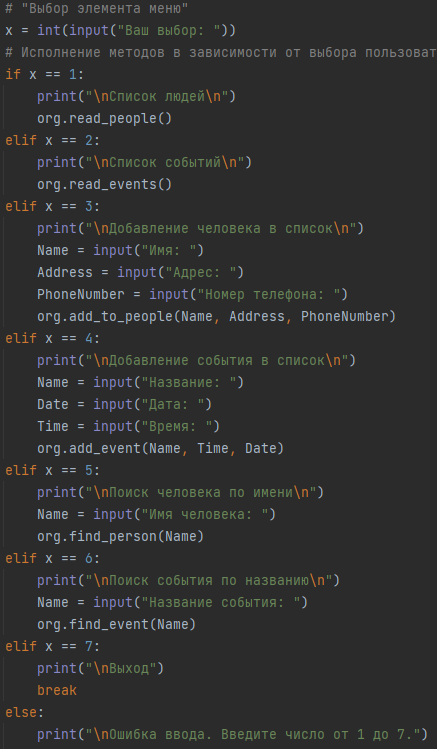


Рисунок 6 – Основная программа (2/2)

Пользователь вводит цифру меню, после чего запускается соответствующий блок кода, включающий определенный метод класса. При выборе варианта «Выход» основной цикл программы прерывается. Если введен неправильный «код», то программа выводит сообщение об ошибке.

После выполнения любого из вариантов кроме варианта «Выход» программа возвращается в «Меню».

Ниже приведен полный код программы.

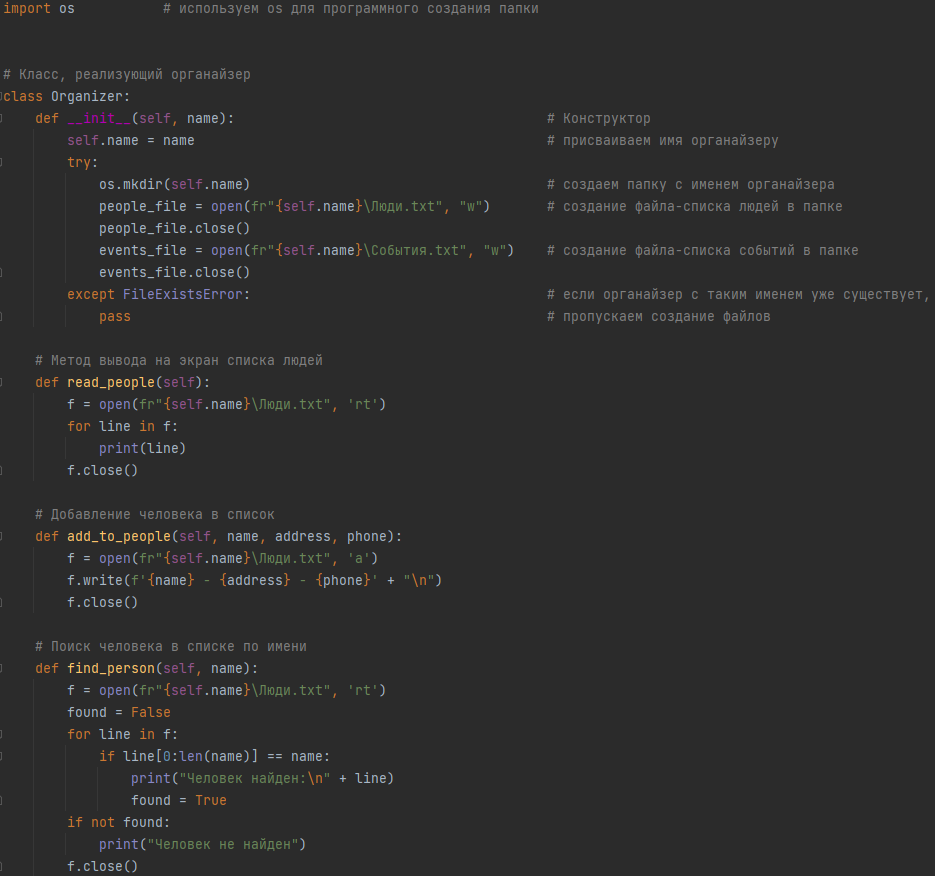


Рисунок 7 – Полный код программы (1/3)

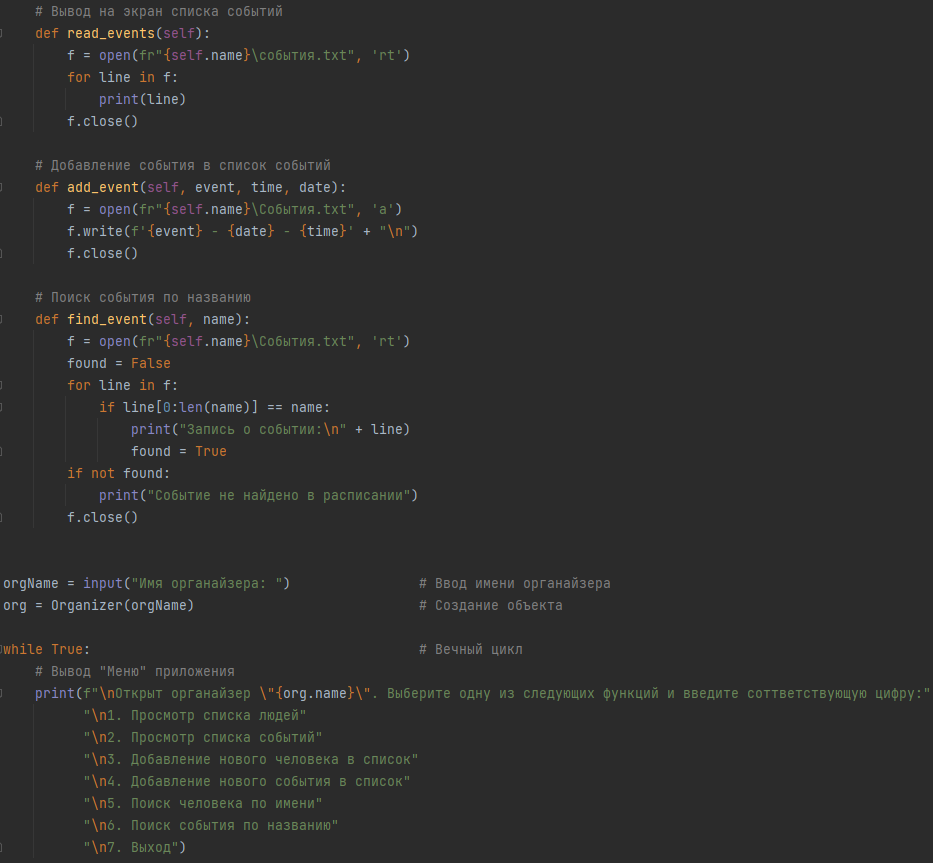


Рисунок 8 – Полный код программы (2/3)

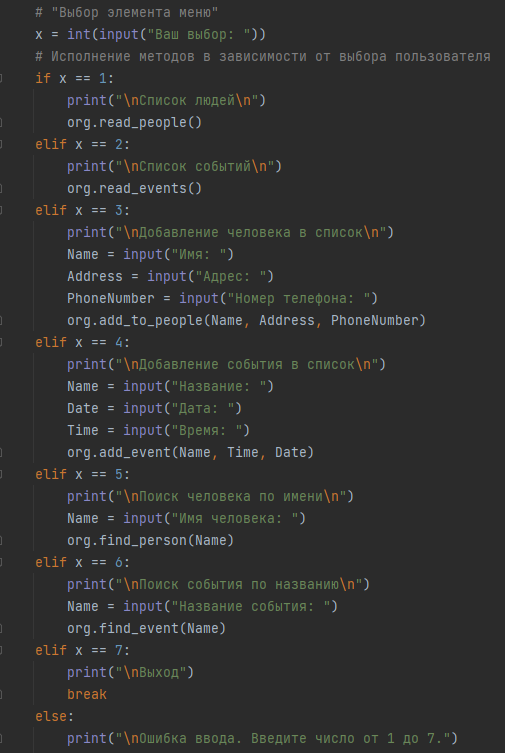


Рисунок 9 – Полный код программы (3/3)

Проверим работу программы.

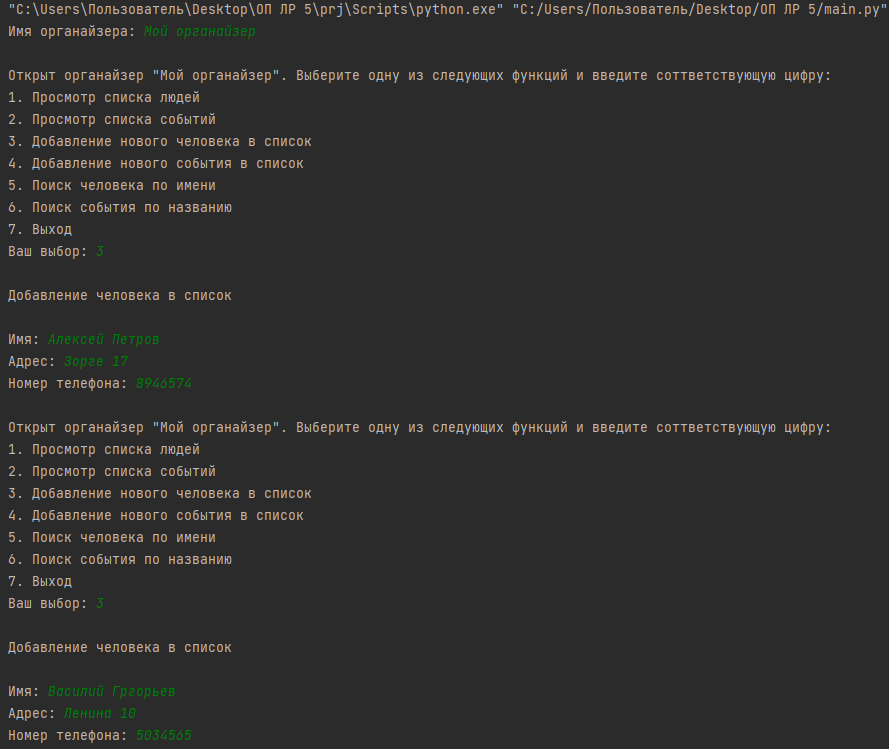


Рисунок 10 – Добавление людей в список

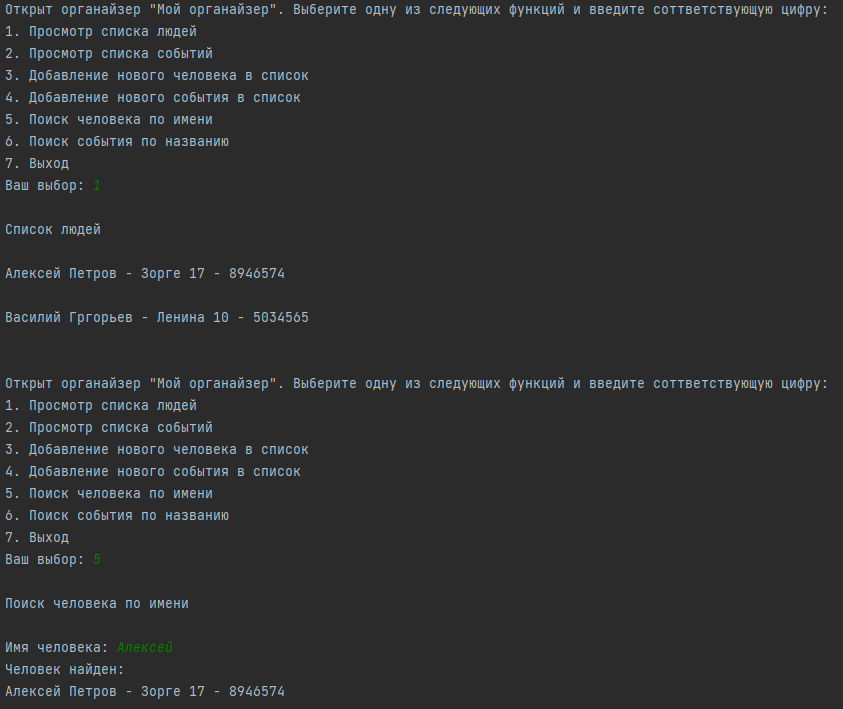


Рисунок 11 – Просмотр списка и поиск человека

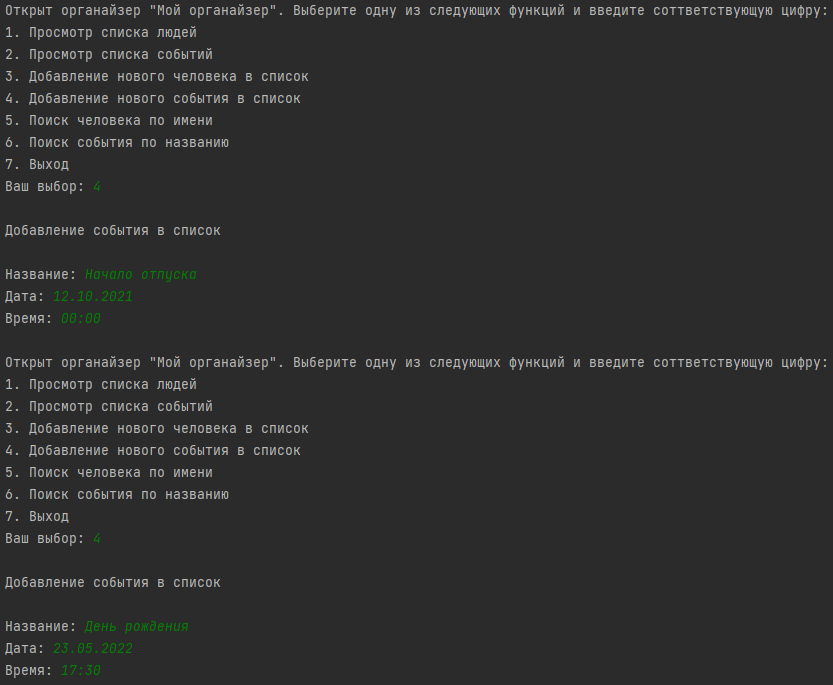


Рисунок 12 – Добавления событий в список

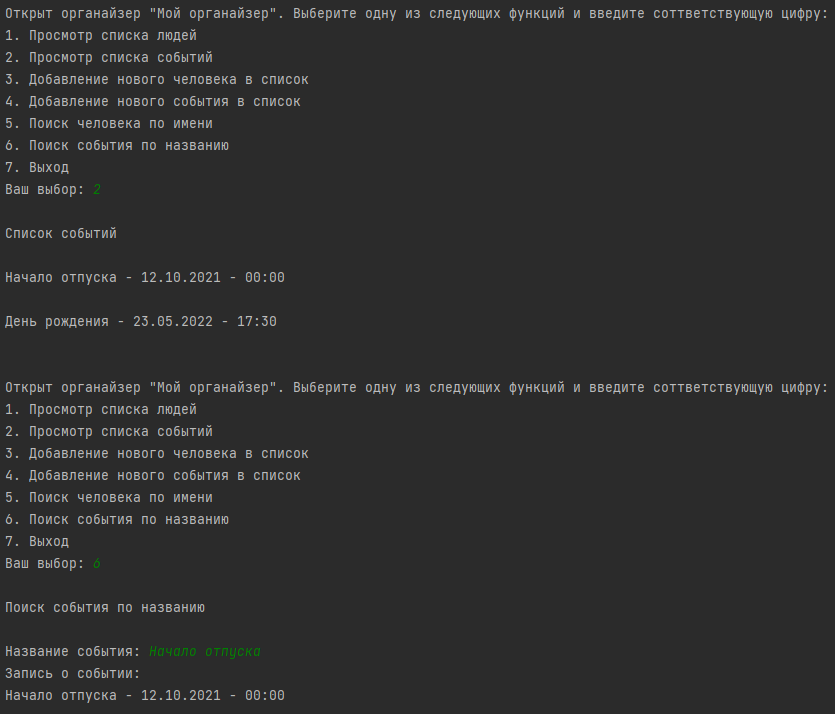


Рисунок 13 – Просмотр списка и поиск события

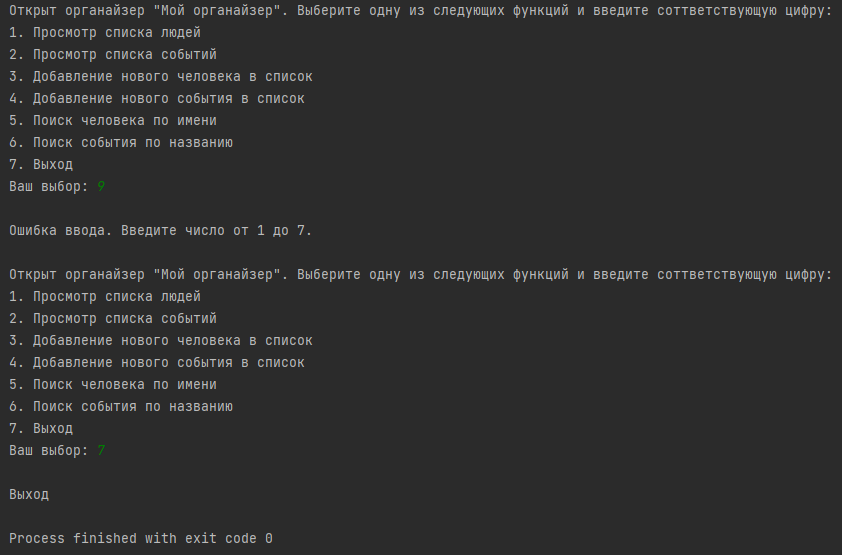


Рисунок 14 – Ошибка ввода и выход

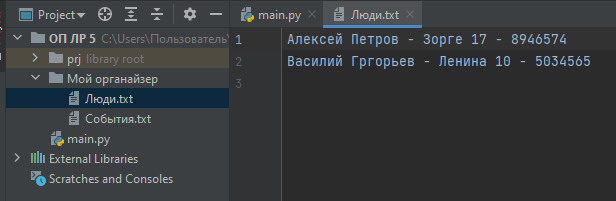


Рисунок 15 – Созданные файлы и папка

**Вывод:** я познакомилась с принципами объектно-ориентированного программирования на языке программирования python.

**Контрольные вопросы**

1. Что такое класс в Python? Каковы его основные характеристики?

Класс в Python – это логическая группа данных и функций. Он дает возможность создавать структуры данных, которые содержат произвольный контент и, следовательно, легко доступны. Классы могут иметь атрибуты (свойства, переменные) и методы (функции).

1. Опишите базовые принципы ООП.

Абстракция – отделение концепции от ее экземпляра;

Полиморфизм – реализация задач одной и той же идеи разными способами;

Наследование – способность объекта или класса базироваться на другом объекте или классе. Это главный механизм для повторного использования кода. Наследственное отношение классов четко определяет их иерархию;

Инкапсуляция – размещение одного объекта или класса внутри другого для разграничения доступа к ним.

1. Что такое экземпляр класса? Каким образом можно осуществить его создание?

В то время как класс является шаблоном, экземпляр является реализацией класса с фактическими значениями, буквально объектом, принадлежащим определенному классу.

Объект создается при помощи конструктора класса.

1. Дайте определение атрибута класса и опишите его основные особенности.

Атрибут класса – переменная, описание которой создает программист при создании класса. Все данные объекта содержатся в атрибутах. Атрибуты бывают статическими и динамическими. Доступ к атрибутам осуществляется по имени. Существует три уровня доступа: public, protected, private.

1. Что такое методы класса? Каковы особенности создания и вызова метода?

Метод класса – функция, принадлежащая какому-либо класса или объекту. Методы являются составляющей частью классов. Методы имеют набор параметров, первым из которых является параметр self. Методы вызываются из своего класса, из других классов, из основной программы. Методы имеют уровни доступа. Для вызова метода необходимо указать его имя и передать все необходимые аргументы.

1. Каковы отличия закрытых методов от обычных?

Закрытый метод можно вызывать только изнутри класса, в котором он определен.

1. В чем заключается преимущество использования конструктора \_\_init\_\_() при создании класса?

Конструктор может задавать необходимые свойства объекта и вызывать необходимые методы сразу при создании нового объекта. Конструктор не позволит создать объект без необходимых атрибутов.

1. Как осуществляется перегрузка специальных методов класса в Python?

Специальные методы выделяются двумя подчеркиваниями с обеих сторон и переопределяются в классе таким же образом, как и обычные методы.

1. Как реализуется принцип наследования в Python? Приведите примеры.

В python есть возможность создать класс-наследник, указав при создании класс-родитель в скобках. Наследование подразумевает то, что дочерний класс содержит все атрибуты родительского класса, при этом некоторые из них могут быть переопределены или добавлены в дочернем.

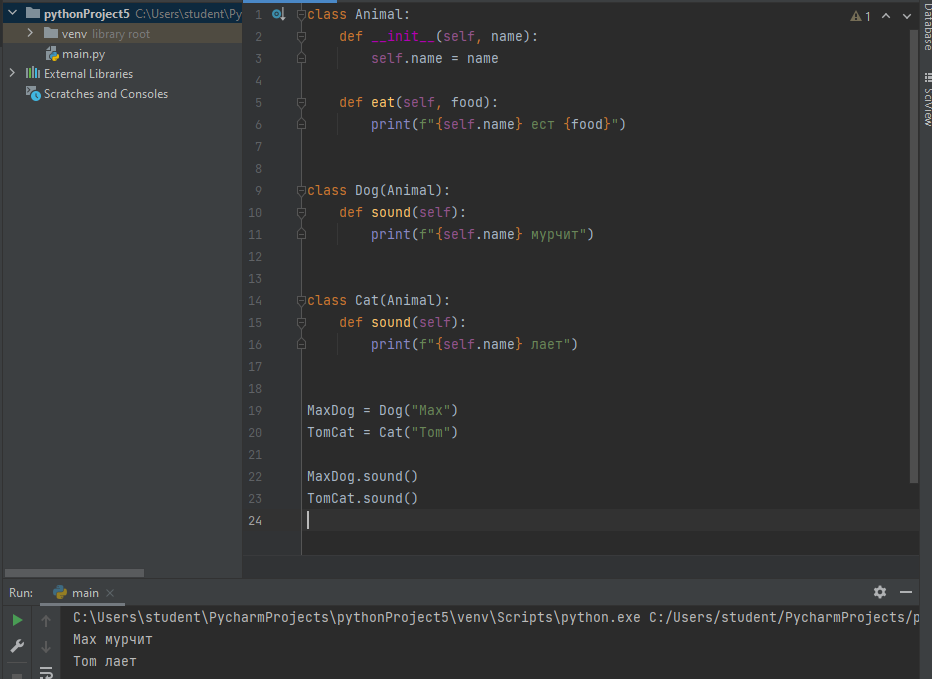


Рисунок 16 – Простейший пример наследования

1. В чем смысл использования абстрактного метода в Python?

Абстрактные методы не могут быть инстанциированы, от них нужно унаследовать, реализовать все их абстрактные методы и только тогда можно будет создать экземпляр такого класса. Подразумевается, что все наследуемые классы должные иметь подобный метод, но реализован он должен быть по-своему. Например, абстрактный класс «шахматная фигура» может иметь метод «ход», который по-своему реализуется для каждой шахматной фигуры.