Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный университет”

Кафедра ИИТ

Лабораторная работа №6

По дисциплине “Языки программирования”

Вариант №8

Выполнил:

Крук Д.Н. (ПО-7,2)

Проверил:

Дряпко. А. В.

Дата выполнения:

16.10.21

Брест 2021

**Цель**:

Ознакомиться со способами создания классов и принципами ООП в Python

**Задание 1**:

Для своего варианта (см.ниже) выполнить следующее:

1. Определить пользовательский класс в соответствии с вариантом задания.

2. Определить счетчик

3. Определить в классе конструкторы с параметрами и без. Конструктор должен выводить сообщение о количестве объектов.

 4. Определить в классе внешние компоненты-функции для получения и установки полей данных.

5. Написать демонстрационную программу, в которой объекты пользовательского класса создаются с помощью неявного использования конструкторов без параметров.

6. Показать в программе явное использование конструкторов с параметрами.

**Предметная область**:

Компания

**Код программы**:

class company:  
 count = 0  
 \_\_name = str()  
 \_\_workers = int()  
 \_\_year = int()  
  
 def \_\_init\_\_(self):  
 company.count += 1  
 print(company.count)  
  
 def \_\_init\_\_(self, nm="", wrs=0, ye=0):  
 company.count += 1  
 print(company.count)  
 try:  
 self.\_\_name = nm  
 self.\_\_workers = wrs  
 self.\_\_year = ye  
 except ValueError as e:  
 print(e)  
  
 def setname(self, nm):  
 self.\_\_name = nm  
  
 def setworkers(self, wrs):  
 self.\_\_workers = wrs  
  
 def setdate(self, ye):  
 self.\_\_year = ye  
  
 def getname(self):  
 return self.\_\_name  
  
 def getworkers(self):  
 return self.\_\_workers  
  
 def getyears(self):  
 return self.\_\_year  
  
 def write(self):  
 try:  
 self.\_\_name = str(input("Введите название компании: "))  
 self.\_\_workers = int(input("Введите количество сотрудников: "))  
 self.\_\_year = int(input("Введите год основания компании: "))  
 except ValueError:  
 print("Не верный ввод!")  
  
 def show(self):  
 print(  
 "\nНазвание компании: " + self.\_\_name + "\nКоличество сотрудников: " + str(  
 self.\_\_workers) + "\nГод основания: " + str(self.\_\_year))

**Задание 2**:

Для своего варианта выполнить следующее:

1. Построить модель предметной области в соответствии со своим вариантом (см. ниже)

 2. Для каждого класса создать конструктор и деструктор, выдающий сообщение о своей работе

3. Для каждого класса создать внешние функции установки и получения полей данных

4. Для каждого класса разработать функции, позволяющие представить на экране значения полей данных

5. Для каждого класса разработать функции, позволяющие вводить с консоли значения полей данных

6. Написать демонстрационную программу, иллюстрирующую поочередный вызов конструкторов и деструкторов базового и производного классов

7. Построить диаграмму классов

**Предметная область**:

Фильм-Телефильм-Мультфильм

**Код программы**:

**Файл films.py:**

class Films:  
 count = 0  
  
 def \_\_init\_\_(self, nm="", wrs=0, ye=0):  
 Films.count += 1  
 print(Films.count)  
 try:  
 self.\_name = nm  
 self.\_fees = wrs  
 self.\_year = ye  
 except ValueError as e:  
 print(e)  
 print('Конструктор с параметрами класса films')  
  
 def \_\_del\_\_(self):  
 print('Дестрктор класса films')  
 Films.count -= 1  
  
 def setname(self, nm):  
 self.\_name = nm  
  
 def setworkers(self, wrs):  
 self.\_fees = wrs  
  
 def setdate(self, ye):  
 self.\_year = ye  
  
 def getname(self):  
 return self.\_name  
  
 def getworkers(self):  
 return self.\_fees  
  
 def getyears(self):  
 return self.\_year  
  
 def write(self):  
 try:  
 self.\_name = str(input("Введите название фильма: "))  
 self.\_fees = int(input("Введите количества сборов фильма: "))  
 self.\_year = int(input("Введите год выхода фильма: "))  
 except ValueError:  
 print("Не верный ввод!")  
  
 def show(self):  
 print(  
 "\nНазвание фильма: " + self.\_name + "\nСборы: " + str(  
 self.\_fees) + "\nГод выхода: " + str(self.\_year))

**Файл TV.py:**

from films import Films  
  
  
class TV(Films):  
 \_channel = str()  
  
 def \_\_init\_\_(self, nm="", wrs=0, ye=0, ch=""):  
 super(TV, self).\_\_init\_\_(nm, wrs, ye)  
 self.\_channel = ch  
 print('Конструктор класса TV')  
  
 def \_\_del\_\_(self):  
 print('Дестрктор класса TV')  
  
 def setchannel(self, ch):  
 self.\_channel = ch  
  
 def getchannel(self):  
 return self.\_channel  
  
 def write(self):  
 try:  
 self.\_name = str(input("Введите название телефильма: "))  
 self.\_fees = int(input("Введите количества сборов телефильма: "))  
 self.\_year = int(input("Введите год выхода телефильма: "))  
 self.\_channel = str(input("Введите канал вещания телефильма: "))  
 except ValueError:  
 print("Не верный ввод!")  
  
 def show(self):  
 print(  
 "\nНазвание фильма: " + self.\_name + "\nСборы: " + str(  
 self.\_fees) + "\nГод выхода: " + str(self.\_year) + "\nКанал вещания: " + self.\_channel)

**Файл cartoons.py:**  
from films import Films  
  
class cartoons(Films):  
 \_studio = str()  
  
 def \_\_init\_\_(self, nm="", wrs=0, ye=0, st=""):  
 super(cartoons, self).\_\_init\_\_(nm, wrs, ye)  
 self.\_studio = st  
 print('Конструктор класса cartoons')  
  
 def \_\_del\_\_(self):  
 print('Дестрктор класса cartoons')  
  
 def setchannel(self, sc):  
 self.\_studio = sc  
  
 def getchannel(self):  
 return self.\_studio  
  
 def write(self):  
 try:  
 self.\_name = str(input("Введите название мультфильма: "))  
 self.\_fees = int(input("Введите количества сборов мультфильма: "))  
 self.\_year = int(input("Введите год выхода мультфильма: "))  
 self.\_studio = str(input("Введите студию мультфильма: "))  
 except ValueError:  
 print("Не верный ввод!")  
  
 def show(self):  
 print(  
 "\nНазвание фильма: " + self.\_name + "\nСборы: " + str(  
 self.\_fees) + "\nГод выхода: " + str(self.\_year) + "Студия мультфильма: " + self.\_studio)

**Файл main.py:**

from TV import TV  
from films import Films  
from cartoons import cartoons  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 tv = TV('name', 234, 234, 'df')  
 cartoons = cartoons('fdsf', 12, 123)  
 films = Films('qwe', 12, 122)  
  
 print(str(tv))  
 print(str(cartoons))  
 print(str(films))

**Иерархия классов**:

**class** Films:

\_name: str

\_fees: flout

\_years: int

**class** Cartoon**s**:

\_name: str

\_fees: flout

\_years: int

\_\_studio: str

**class** TV:

\_name: str

\_fees: flout

\_years: int

\_channel = str()

**Вывод**:

Ознакомился с принципом наследования в Python