Министерство образования Республики Беларусь

УО «Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №6

По дисциплине: “Алгоритмы и структуры данных”

Тема: «Классы. Инкапсуляция.Наследование»Вариант №1

**Выполнил**:

студент 2 курса группы ПО-7 Белев С.В.

**Проверила:**

Дряпко А.В.

Брест 2021

**Постановка задачи 1:**

Для своего варианта 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.

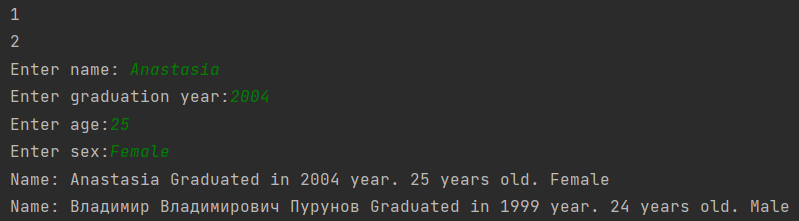
**Вариант задания:**

Выпускник ВУЗа

**Код программы:**

class graduate:  
 count = 0  
 name = str()  
 date = int()  
 age = int()  
 sex = str()  
  
 def \_\_init\_\_(self):  
 graduate.count += 1  
 print(graduate.count)  
  
 def \_\_init\_\_(self, name="", date=0, age=0, sex="undefined"):  
 graduate.count += 1  
 print(graduate.count)  
 self.name = name  
 self.date = date  
 self.age = age  
 self.sex = sex  
  
 def set\_name(self, nm):  
 self.name = nm  
  
 def set\_date(self, dt):  
 self.date = dt  
  
 def set\_age(self, ae):  
 self.age = ae  
  
 def set\_sex(self, sx):  
 self.sex = sx  
  
 def get\_name(self):  
 return self.name  
  
 def get\_date(self):  
 return self.date  
  
 def get\_age(self):  
 return self.age  
  
 def get\_sex(self):  
 return self.sex  
  
 def read(self):  
 self.name = str(input("Enter name: "))  
 self.date = int(input("Enter graduation year:"))  
 self.age = int(input("Enter age:"))  
 self.sex = str(input("Enter sex:"))  
  
 def show(self):  
 print("Name: " + self.name + " Graduated in " + str(self.date) + " year. " + str(self.age) + " years old. " + self.sex)  
  
frstobj = graduate()  
scndobj = graduate("Владимир Владимирович Пурунов", 1999, 24, "Male")  
frstobj.read()  
frstobj.show()  
scndobj.show()

**Результат программы:**



**Постановка задачи 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

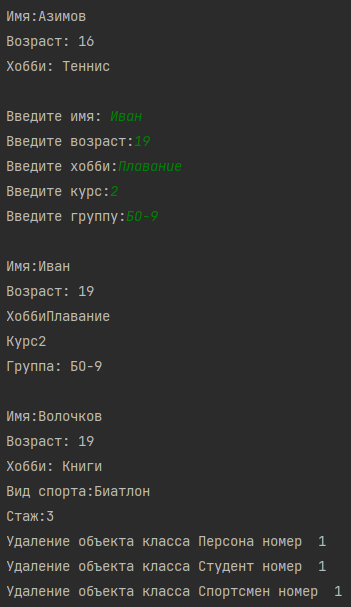
**Вариант задания:**

Студент– Спортсмен – Персона

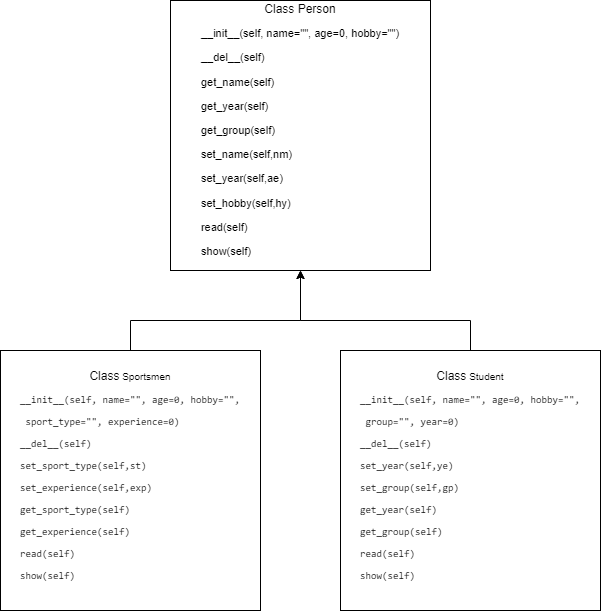
**Код программы:**

class Person:  
 count = 0  
 name = str()  
 age = int()  
 hobby = str()  
  
 def \_\_init\_\_(self, name="", age=0, hobby=""):  
 Person.count += 1  
 self.name = name  
 self.age = age  
 self.hobby = hobby  
  
 def \_\_del\_\_(self):  
 print("Удаление объекта класса Персона номер ", self.count)  
 Person.count -= 1  
  
 def set\_name(self, nm):  
 self.name = nm  
  
 def set\_year(self, ae):  
 self.age = ae  
  
 def set\_group(self, hy):  
 self.hobby = hy  
  
 def get\_name(self):  
 return self.name  
  
 def get\_year(self):  
 return self.age  
  
 def get\_group(self):  
 return self.hobby  
  
 def read(self):  
 self.name = str(input("\nВведите имя: "))  
 self.age = int(input("Введите возраст:"))  
 self.hobby = str(input("Введите хобби:"))  
  
 def show(self):  
 print("\nИмя:" + self.name + "\nВозраст: " + str(self.age) + "\nХобби: " + self.hobby)  
  
  
class Sportsmen(Person):  
 count = 0  
 sport\_type = str()  
 experience = int()  
  
 def \_\_init\_\_(self, name="", age=0, hobby="", sport\_type="", experience=0):  
 Sportsmen.count += 1  
 self.name = name  
 self.age = age  
 self.hobby = hobby  
 self.sport\_type = sport\_type  
 self.experience = experience  
  
 def \_\_del\_\_(self):  
 print("Удаление объекта класса Спортсмен номер ", self.count)  
 Sportsmen.count -= 1  
  
 def set\_sport\_type(self, st):  
 self.sport\_type = st  
  
 def set\_experience(self, exp):  
 self.experience = exp  
  
 def get\_sport\_type(self):  
 return self.sport\_type  
  
 def get\_experience(self):  
 return self.experience  
  
 def read(self):  
 self.name = str(input("\nВведите имя: "))  
 self.age = int(input("Введите возраст:"))  
 self.hobby = str(input("Введите хобби:"))  
 self.sport\_type = str(input("Введите вид спорта:"))  
 self.experience = int(input("Введите стаж:"))  
  
 def show(self):  
 print("\nИмя:" + self.name + "\nВозраст: " + str(self.age) + "\nХобби: " + self.hobby + "\nВид спорта:" +  
 self.sport\_type + "\nСтаж:" + str(self.experience))  
  
  
class Student(Person):  
 count = 0  
 age = int()  
 hobby = str()  
  
 def \_\_init\_\_(self, name="", age=0, group="", year=0, hobby=""):  
 Student.count += 1  
 self.name = name  
 self.age = age  
 self.hobby = hobby  
 self.year = year  
 self.group = group  
  
 def \_\_del\_\_(self):  
 print("Удаление объекта класса Студент номер ", self.count)  
 Student.count -= 1  
  
 def set\_year(self, ye):  
 self.year = ye  
  
 def set\_group(self, gp):  
 self.group = gp  
  
 def get\_year(self):  
 return self.year  
  
 def get\_group(self):  
 return self.group  
  
 def read(self):  
 self.name = str(input("\nВведите имя: "))  
 self.age = int(input("Введите возраст:"))  
 self.hobby = str(input("Введите хобби:"))  
 self.year = int(input("Введите курс:"))  
 self.group = str(input("Введите группу:"))  
  
 def show(self):  
 print("\nИмя:" + self.name + "\nВозраст: " + str(self.age) + "\nХобби" + self.hobby + "\nКурс" + str(self.year) +  
 "\nГруппа: " + self.group)  
  
pers = Person("Азимов", 16, "Теннис")  
pers.show()  
stud = Student()  
stud.read()  
stud.show()  
smen = Sportsmen("Волочков", 19, "Книги", "Биатлон", 3)  
smen.show()

**Результат программы:**



**Иерархическая схема:**

****

**Вывод:**

Задания выполнены согласно варианту.