Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный университет”

Кафедра ИИТ

Лабораторная работа №6

По дисциплине: “ЯП”

Тема: “Классы. Инкапсуляция. Наследование”

Выполнил:

Студент 2-го курса

Группы ПО-7

Рекун И.В.

Проверил:

Бойко Д.О.

Брест, 2021

**Вариант 6**

**Цель работы:** Ознакомиться с принципом инкапсуляции.

Дополнительные цели:

1. Ознакомиться с конструкторами и деструкторами в Python

2. Ознакомиться с указателем self

3. Инкапсуляция в Python

4. Ознакомиться с использованием принципа наследования в Python

**Задание 1.**

1. Определить пользовательский класс – «Банк»

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.

**Задание 2.**

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.

**Код программы:**

class Bank:

\_\_counter = 0

\_\_name = str()

\_\_monetaryCapital = float()

\_\_numberOfEmployees = int()

def \_\_init\_\_(self):

Bank.\_\_counter += 1

print(Bank.\_\_counter)

def \_\_init\_\_(self, nm = "", mc = 0.0, nof = 0):

Bank.\_\_counter += 1

print(Bank.\_\_counter)

self.\_\_name = nm

self.\_\_monetaryCapital = float(mc)

self.\_\_numberOfEmployees = int(nof)

def set\_name(self, nm):

self.\_\_name = nm

def set\_capital(self, mc):

self.\_\_monetaryCapital = float(mc)

def set\_number(self, nof):

self.\_\_numberOfEmployees = int(nof)

def get\_name(self):

return self.\_\_name

def get\_capital(self):

return self.\_\_monetaryCapital

def get\_number(self):

return self.\_\_numberOfEmployees

def read(self):

self.\_\_name = str(input("Enter name: "))

self.\_\_monetaryCapital = float(input("Enter monetary capital: "))

self.\_\_numberOfEmployees = int(input("Enter number of employees: "))

def show(self):

print("Name: " + self.\_\_name)

print("Monetary Capital: " + str(self.\_\_monetaryCapital))

print("Number of employees: " + str(self.\_\_numberOfEmployees))

if \_\_name\_\_ == "\_\_main\_\_":

y = Bank()

y1 = Bank()

y.read()

y1.read()

y.show()

y1.show()

class Sportsman:

\_\_counter = 0

\_\_name = str()

\_\_sport\_type = str()

\_\_age = int()

def \_\_init\_\_(self):

Sportsman.\_\_counter += 1

print(Sportsman.\_\_counter)

def \_\_init\_\_(self, nm = "", st = "", age = 0):

Sportsman.\_\_counter += 1

print("Вызов конструктора №" + str(Sportsman.\_\_counter))

self.\_\_name = nm

self.\_\_sport\_type = st

self.\_\_age = int(age)

def \_\_del\_\_(self):

print("Вызов деструктора №" + str(Sportsman.\_\_counter))

Sportsman.\_\_counter -= 1

def set\_name(self, nm):

self.\_\_name = nm

def set\_st(self, st):

self.\_\_sport\_type = st

def set\_age(self, age):

self.\_\_age = age

def get\_name(self):

return self.\_\_name

def get\_st(self):

return self.\_\_sport\_type

def get\_age(self):

return self.\_\_age

def read(self):

self.\_\_name = input("Enter name: ")

self.\_\_sport\_type = input("Enter sport type: ")

self.\_\_age = int(input("Enter age: "))

def show(self):

print("Name: " + self.\_\_name)

print("Sport type: " + self.\_\_sport\_type)

print("Age: " + str(self.\_\_age))

class Person(Sportsman):

\_\_counter = 0

\_\_height = float()

\_\_weight = float()

def \_\_init\_\_(self):

pass

def \_\_init\_\_(self, nm = "", st = "", age = 0, ht = 0.0, wt = 0.0):

super(Person, self).\_\_init\_\_(nm, st, age)

self.\_\_height = ht

self.\_\_weight = wt

def set\_ht(self, ht):

self.\_\_height = ht

def set\_wt(self, wt):

self.\_\_weight = wt

def get\_ht(self):

return self.\_\_height

def get\_wt(self):

return self.\_\_weight

def read(self):

self.\_\_name = input("Enter name: ")

self.\_\_sport\_type = input("Enter sport type: ")

self.\_\_age = int(input("Enter age: "))

self.\_\_height = float(input("Enter height: "))

self.\_\_weight = float(input("Enter weight: "))

def show(self):

print("Name: " + str(self.\_\_name))

print("Sport type: " + self.\_\_sport\_type)

print("Age: " + str(self.\_\_age))

print("Height: " + str(self.\_\_height))

print("weight: " + str(self.\_\_weight))

class Prize\_Winner(Person):

\_\_counter = 0

\_\_world\_champion = bool()

\_\_world\_ranking = int()

def \_\_init\_\_(self):

pass

def \_\_init\_\_(self, nm = "", st = "", age = 0, ht = 0.0, wt = 0.0, wc = True, wr = 0):

super(Prize\_Winner, self).\_\_init\_\_(nm, st, age, ht, wt)

self.\_\_world\_champion = wc

self.\_\_world\_ranking = wr

def set\_wc(self, wc):

self.\_\_world\_champion = wc

def set\_wr(self, wr):

self.\_\_world\_ranking = wr

def get\_wc(self):

return self.\_\_world\_champion

def get\_wr(self):

return self.\_\_world\_ranking

def read(self):

self.\_\_name = input("Enter name: ")

self.\_\_sport\_type = input("Enter sport type: ")

self.\_\_age = int(input("Enter age: "))

self.\_\_height = float(input("Enter height: "))

self.\_\_weight = float(input("Enter weight: "))

check = str(input('Enter is there any world champion: '))

if check.startswith('True'.lower()):

self.\_\_world\_champion = True

elif check.startswith('False'.lower()):

self.\_\_world\_champion = False

self.\_\_world\_ranking = int(input("Enter world ranking: "))

def show(self):

print("Name: " + self.\_\_name)

print("Sport type: " + self.\_\_sport\_type)

print("Age: " + str(self.\_\_age))

print("Height: " + str(self.\_\_height) + " сантиметров")

print("weight: " + str(self.\_\_weight) + " килограмм")

print("World champion: " + str(self.\_\_world\_champion))

print("World ranking: " + str(self.\_\_world\_ranking))

if \_\_name\_\_ == "\_\_main\_\_":

P0 = Sportsman()

P1 = Person()

P2 = Prize\_Winner()

P0.read()

P1.read()

P2.read()

P0.show()

P1.show()

P2.show()

**Вывод:** В ходе лабораторной работы освоил основные способы работы с классами. Познакомился с понятиями инкапсуляция и наследование, а также реализовал их на практике.