МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Брестский государственный технический университет»

Кафедра «Интеллектуальные информационные технологии»

Лабораторная работа №4

По дисциплине «Современные системы программирования»

За 7 семестр

Тема: «Объектно-ориентированное проектирование»

Выполнила:  
студентка 4 курса  
группы АС-56  
Карпенко М.В.

Проверил:

Войцехович Г.Ю.

Брест 2022

*Цель работы:* Приобрести практические навыки в области объектно-ориентированного проектирования.

ВАРИАНТ 4

*Задание 1.* Создать класс Зачетная Книжка с внутренним классом, с помощью объектов которого можно хранить информацию о сессиях, зачетах, экзаменах.

*Код программы:*

class Ze4etnaiaKnijka:  
 ob = "Python-inputs"  
 info = None  
 city = None  
  
 def \_\_init\_\_(self):  
 print("vvedite FIO studenta", end="")  
 self.setZe4etnaiaKnijka(input())  
 self.info = self.Info  
  
 def getZe4etnaiaKnijka(self):  
 return self.city  
  
 def setZe4etnaiaKnijka(self, city):  
 self.city = city  
  
 class Info:  
 pr = 0  
 st = 0  
 sq = 0  
  
 def \_\_init\_\_(self):  
 print("sessiya No:", end="")  
 self.setPr(input())  
 print("kolichestvo ne sdanny`x zachyotov:", end="")  
 self.setSt(input())  
 print("kol-vo e`kzamenov na sl. sessiyu:", end="")  
 self.setSq(input())  
  
 def getPr(self):  
 return self.pr  
  
 def setPr(self, pr):  
 self.pr = pr  
  
 def getSt(self):  
 return self.st  
  
 def setSt(self, st):  
 self.st = st  
  
 def getSq(self):  
 return self.sq  
  
 def setSq(self, sq):  
 self.sq = sq  
  
 def prints(self):  
 print("Student po familii" + " " + self.getZe4etnaiaKnijka() + " sdal" + " " + str(  
 self.getPr()) + "sessiyu" + " " + str(self.getSt()) + "zachetov" + " i " + str(  
 self.getSq()) + "e`kzamenov", end="")  
  
 @staticmethod  
 def main(args):  
 ct = Ze4etnaiaKnijka()  
 ct.Info.prints()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 Ze4etnaiaKnijka.main([])

*Задание 2.* Создать класс Текст, используя классы Страница, Слово.

*Код программы:*

class Blinov4A1\_TextAsObject:  
 @staticmethod  
 def main(args):  
 part1 = Word("Hello")  
 part2 = Punctuation(",")  
 part3 = Word("world")  
 part4 = Punctuation("!!!")  
 sentence1 = Sentence(part1, part2, part3, part4)  
 sentence2 = Sentence(Word("Why"), Word("should"), Word("we"), Word("always"), Word("use"), Word("the"),  
 Word("same"), Word("phrase"), Punctuation("?"))  
 sentence1.showSentence()  
 print()  
 sentence2.showSentence()  
 print()  
 text = Text()  
 text.addSentence(sentence1)  
 text.addSentence(sentence2)  
 text.showText()  
  
  
class Text:  
 text = []  
  
 def addSentence(self, sentence):  
 self.text.append(sentence)  
  
 def showText(self):  
 for sentence in self.text:  
 sentence.showSentence()  
 print(" ", end="")  
  
  
class Sentence:  
 sentence = []  
  
 def \_\_init\_\_(self, partsOfSentence):  
 i = 0  
 while (i < len(partsOfSentence)):  
 if (not partsOfSentence[i].getString() == "," and i > 0):  
 self.sentence.append(Punctuation(" "))  
 self.sentence.append(partsOfSentence[i])  
 i += 1  
  
 def addPartOfSentence(self, partOfSentence):  
 self.sentence.append(partOfSentence)  
  
 def showSentence(self):  
 for part in self.sentence:  
 print(part.getString(), end="")  
  
  
class PartOfSentence:  
 def getString(self):  
 pass  
  
  
class Word(PartOfSentence):  
 word = None  
  
 def \_\_init\_\_(self, word):  
 self.word = word  
  
 def getString(self):  
 return self.word  
  
  
class Punctuation(PartOfSentence):  
 punctuation = None  
  
 def \_\_init\_\_(self, punctuation):  
 self.punctuation = punctuation  
  
 def getString(self):  
 return self.punctuation  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 Blinov4A1\_TextAsObject.main([])

*Задание 3.* Система Вступительные экзамены. Абитуриент регистрируется на Факультет, сдает Экзамены. Преподаватель выставляет Оценку. Система подсчитывает средний балл и определяет Абитуриентов, зачисленных в учебное заведение.

*Код программы:*

class Main:  
 @staticmethod  
 def main(args):  
 a1 = Abiturient("Vasya", "Pupkin", "Izi", 18, "Moskva")  
 b1 = Faculty("Avtomatika i vy`chislitel`naya texnika")  
 b1.showFacult(a1.Name, a1.Surname)  
  
  
class Faculty:  
 Namefacult = None  
 Budgetplace = None  
 NameAbiturient = None  
 SurNameAbiturient = None  
  
 def \_\_init\_\_(self, namefacult):  
 self.Namefacult = namefacult  
  
 def showFacult(self, nameabiturient, surnameabiturient):  
 self.NameAbiturient = nameabiturient  
 self.SurNameAbiturient = surnameabiturient  
 print(  
 "Abiturient " + self.NameAbiturient + " " + self.SurNameAbiturient + " zaregistrirovalsya na fakul`tet: " + self.Namefacult)  
  
  
class Abiturient:  
 Name = None  
 Surname = None  
 Otch = None  
 Age = 0  
 City = None  
  
 def \_\_init\_\_(self, name, surname, otch, age, city):  
 self.Name = name  
 self.Surname = surname  
 self.Otch = otch  
 self.Age = age  
 self.City = city  
  
 def show(self):  
 print("Abiturient: " + self.Surname + " " + self.Name + " " + self.Otch + " || Vozrast: " + str(  
 self.Age) + " || Gorod: " + self.City)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 Main.main([])

*Результат работы программы:*



*Вывод:* Приобрела практические навыки в области объектно-ориентированного проектирования.