Національний університет «Львівська Політехніка»

Інститут комп’ютерних технологій, автоматики та метрології

Кафедра електронних обчислювальних машин



Звіт

Про виконання лабораторної роботи №9

З дисципліни «Кросплатформлені засоби програмування»

**Виконав:**

студент групи КІ-306

Ориняк Ю.І.

**Перевірив:**

Доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

**Тема:** Основи об’єктно-орієнтованого програмування у Python.

**Мета:** Оволодіти навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python.

**Завдання:**

1. Написати та налагодити програму на мові Python згідно варіанту. Програма має задовольняти наступним вимогам:  
• класи програми мають розміщуватися в окремих модулях в одному пакеті;  
• точка входу в програму (main) має бути в окремому модулі;  
• мають бути реалізовані базовий і похідний класи предметної області згідно  
варіанту;  
• програма має містити коментарі.  
2. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.  
3. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.  
4. Дати відповідь на контрольні запитання.

**Завдання згідно варіанту:**

**Варіант: 12**

***Базовий клас:***



***Похідний клас:***



**Виконання:**

**Код програми:**

***main.py***

from my\_Sea import Sea  
  
# основна програма  
if \_\_name\_\_ == "\_\_main\_\_":  
 # об'єкт на основі класу море  
 black = Sea(27, 11, 21)  
  
 # використання методів класу за допомогою створеного об'єкта  
 black.setParametersSquare(11.5, 32)  
 black.Printer("Площа водойми:", black.getCalcSquare(), "m^2")  
 black.setParametersforNaturalWord("ламінарія, зелена водорость", "cудак, дельфін, камбала, бичок")  
 black.getInformation()  
 black.checkSalinity()  
 black.setParameterforFishing(1000)  
 black.ResultFishing()  
 black.RunStockapond()  
 black.SituationofWater()  
 black.Printer("Об'єм води становить: ",black.calculateVolume(), "litr")

***my\_NaturalWorld.py***

# клас природній світ водойми  
class NaturalWorld:  
 # конструктор  
 def \_\_init\_\_(self, Flora="default", Fauna="default"):  
 self.\_\_flora = Flora  
 self.\_\_fauna = Fauna  
 # метод повернення рослинності водойми  
 def getFlora(self):  
 return self.\_\_flora  
 # метод повернення тваринності водойми  
 def getFauna(self):  
 return self.\_\_fauna  
 # метод встановлення рослинності водойми  
 def setFlora(self, Flora ):  
 self.\_\_flora = Flora  
 # метод встановлення тваринності водойми  
 def setFauna(self, Fauna):  
 self.\_\_fauna = Fauna

***my\_FishingofWater.py***

# клас риболовля на водоймі  
class FishingofWater:  
 # конструктор  
 def \_\_init\_\_(self, Amountoffish=0):  
 self.\_\_amountoffish = Amountoffish  
 # метод повернення кількості риби  
 def getAmountofFish(self):  
 return self.\_\_amountoffish  
 # метод встановлення кількості риби  
 def setAmountofFish(self, Amountoffish):  
 self.\_\_amountoffish = Amountoffish  
 # метод Риболовля  
 def Fishing(self):  
 return self.\_\_amountoffish - 5  
  
 # метод Зариблення водойми  
 def Stockapond(self):  
 self.\_\_amountoffish=10000  
 return self.\_\_amountoffish

***my\_square.py***

# клас площа водойми  
class Square:  
#конструктор  
 def \_\_init\_\_(self, xwidth=0, xlength=0):  
 self.\_\_width = xwidth  
 self.\_\_length = xlength  
#метод повернення ширини  
 def getWidth(self):  
 return self.\_\_width  
#метод повернення довжини  
 def getLength(self):  
 return self.\_\_length  
#метод встановлення ширини  
 def setWidth(self, xwidth):  
 self.\_\_width = xwidth  
#метод встановлення довжини  
 def setLength(self, xlength):  
 self.\_\_length = xlength  
#метод для обрахунку площі водойми  
 def CalcSquare(self):  
 square = self.\_\_width \* self.\_\_length  
 return square

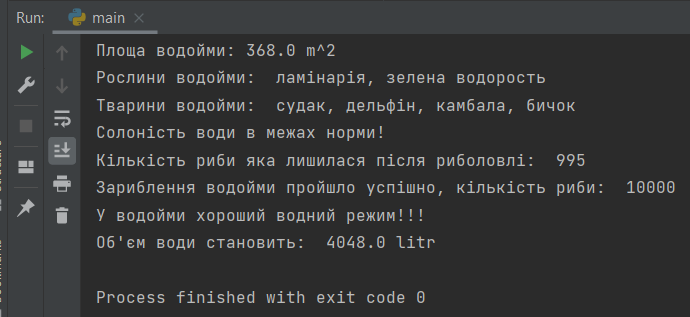
***my\_Water.py***

from my\_square import Square  
from my\_FishingofWater import FishingofWater  
from my\_NaturalWord import NaturalWorld  
  
#клас водойма  
class Water:  
 # конструктор  
 def \_\_init\_\_(self, xdepth=0.0):  
 self.\_\_squareofwater = Square()  
 self.\_\_naturalsituation = NaturalWorld()  
 self.\_\_fishingwater = FishingofWater()  
 self.\_\_depth = xdepth  
  
 # метод встановлення глибини  
 def SetDepthWater (self, xdepth):  
 self.\_\_depth = xdepth  
  
 # метод перевірки водного режиму  
 def SituationofWater(self):  
 if(self.\_\_depth < 2.0):  
 print("Водойма на стадії обміління!")  
 else:  
 print("У водойми хороший водний режим!!! ")  
  
 # метод для отримання площі водойми  
 def getCalcSquare(self):  
 return self.\_\_squareofwater.CalcSquare()  
  
 # метод для встановлення параметрів для обчислення площі  
 def setParametersSquare(self, xwidth, xlength):  
 self.\_\_squareofwater.setWidth(xwidth)  
 self.\_\_squareofwater.setLength(xlength)  
  
 # метод для поверненя ширини  
 def getWidthforWater(self):  
 return self.\_\_squareofwater.getWidth()  
  
 # метод для повернення довжини  
 def getLengthforWater(self):  
 return self.\_\_squareofwater.getLength()  
  
 # метод для встановлення параметрів риболовлі на водоймі  
 def setParameterforFishing(self, xAmountoffish):  
 self.\_\_fishingwater.setAmountofFish(xAmountoffish)  
  
 # метод для проведення риболовлі  
 def ResultFishing(self):  
 print("Кількість риби яка лишилася після риболовлі: ", self.\_\_fishingwater.Fishing())  
  
 # метод для зариблення водойми  
 def RunStockapond(self):  
 print("Зариблення водойми пройшло успішно, кількість риби: ", self.\_\_fishingwater.Stockapond())  
  
 # метод для встановлення інформації про природній світ водойми  
 def setParametersforNaturalWord(self, xFlora, xFauna):  
 self.\_\_naturalsituation.setFlora(xFlora)  
 self.\_\_naturalsituation.setFauna(xFauna)  
  
  
 # метод для отримання повної інформації про природній світ водойми  
 def getInformation(self):  
 print("Рослини водойми: ", self.\_\_naturalsituation.getFlora())  
 print("Тварини водойми: ", self.\_\_naturalsituation.getFauna())

***my\_Sea.py***

from my\_Water import Water  
  
#клас море  
class Sea(Water):  
  
 #конструктор  
 def \_\_init\_\_(self, xdepth=0.0, Xdepth=0.0, Xsalinity=0):  
 super().\_\_init\_\_(xdepth)  
 self.\_\_depth = Xdepth  
 self.\_\_salinity = Xsalinity  
  
 #метод повернення солоності  
 def getSalinity(self):  
 return self.\_\_salinity  
  
 #метод встановлення солоності  
 def setSalinity(self,Xsalinity):  
 self.\_\_salinity = Xsalinity  
  
 #метод повернення глибини  
 def getDepth(self):  
 return self.\_\_depth  
  
 #метод встановлення солоності  
 def setDepth(self, Xdepth):  
 self.\_\_depth = Xdepth  
  
 #метод обраховує сальдо судноплавства  
 def checkSalinity(self):  
 if (self.\_\_salinity < 1.0):  
 print("У морі недостатня солоність води!!!")  
 else:  
 print("Солоність води в межах норми!");  
  
 def calculateVolume(self):  
 volume = self.getWidthforWater() \* self.getLengthforWater() \* self.\_\_depth;  
 return volume  
 # метод виводить на екран властивості класу  
 def Printer(self, line,line1, line3):  
 print(line, line1, line3)

**Результат роботи програми у консолі:**

****

**Висновок:** на цій лабораторній роботі я оволодів навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python. Створив клас водойма, який наслідує деякі властивості класів площа водойми, поточна екологічна ситуація водойми та повна інформація про водойму. Також створив клас море, на основі класу водойми і розробив програму-драйвер, у якій створив об’єкт на основі цього класу та протестував його методи.