Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра «Електронних обчислювальних машин»



**Звіт**

з лабораторної роботи № 9

з дисципліни: «Кросплатформенні засоби програмування»

**На тему:** «Основи Об’єктно-орієнтованого програмування у PYTHON»

**Виконав:**

студент групи КІ-307

Бажулін С.В.

**Прийняв:**

доцент кафедри ЕОМ

Іванов Ю. С.

Львів – 2023

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

**Завдання(варіант №2):**

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.

**Індивідуальне завдання:**





**Вихідний код програми:**

**Файл MainBazhulin.py**

from ReusableSpaceship import Mission, ReusableSpaceship

from Spaceship import Spaceship, Engine, ControlPanel, Door

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

    reusableSpaceship = ReusableSpaceship()

    reusableSpaceship.RefuelSpaceship(50)

    print(reusableSpaceship.getFuelStatus())

    reusableSpaceship.CloseDoor()

    reusableSpaceship.StartSpaceship()

    reusableSpaceship.SetSpeed(100)

    reusableSpaceship.TakeOff()

    reusableSpaceship.TurnRightSpaceship()

    direction = reusableSpaceship.getDirectionSpaceship()

    if direction == ControlPanel.Direction.FORWARD:

        print("Forward direction")

    elif direction == ControlPanel.Direction.LEFT:

        print("Left direction")

    else:

        print("Right direction")

    reusableSpaceship.Land()

    reusableSpaceship.SwitchOffSapceship()

    reusableSpaceship.OpenDoor()

    reusableSpaceship.dispose()

**Файл Spaceship.py**

import os

class Spaceship:

    def \_\_init\_\_(self, fuel=0):

        self.fuel = fuel

        self.engine = Engine()

        self.controlPanel = ControlPanel()

        self.door = Door()

        self.tmp = 675

        self.fout = open("Log.txt", "w")

    def RefuelSpaceship(self, fuel):

        pass

    def StartSpaceship(self):

        pass

    def SwitchOffSapceship(self):

        pass

    def getFuelStatus(self):

        fuel = self.engine.getFuel()

        return fuel

    def getEngineSpaceshipStatus(self):

        status = self.engine.getEngineStatus()

        return status

    def TurnLeftSpaceship(self):

        self.controlPanel.TurnLeft()

        print("Spaceship turned left\n")

        self.fout.write("Spaceship turned left\n")

    def TurnRightSpaceship(self):

        self.controlPanel.TurnRight()

        print("Spaceship turned right\n")

        self.fout.write("Spaceship turned right\n")

    def ForwardSpacehip(self):

        self.controlPanel.Forward()

        print("Spaceship forward\n")

        self.fout.write("Spaceship forward\n")

    def SetSpeed(self, speed):

        self.controlPanel.SpeedChange(speed)

        print("Spaceship speed set " + str(speed) + "\n")

        self.fout.write("Spaceship speed set " + str(speed) + "\n")

    def getSpeedSpaceship(self):

        return self.controlPanel.getSpeed()

    def OpenDoor(self):

        self.door.Open()

        print("Door of spaceship was opened\n")

        self.fout.write("Door of spaceship was opened\n")

    def CloseDoor(self):

        self.door.Close()

        print("Door of spaceship was closed\n")

        self.fout.write("Door of spaceship was closed\n")

    def getStatusDoor(self):

        return self.door.getDoorStatus()

    def getDirectionSpaceship(self):

        return self.controlPanel.getDirection()

    def \_\_str\_\_(self):

        try:

            file = open("randomFile.txt", "w+")

            value = str(self.tmp)

            file.write(value)

            return "Data writed successfuly"

        except IOError as ex:

            return ex.strerror

    def dispose(self):

        self.fout.close()

class Engine:

    def \_\_init\_\_(self):

        self.fuel = 0

        self.isStarted = False

    def StartEngine(self):

        if self.fuel > 0:

            self.isStarted = True

        else:

            print("Fuel is not enough for starting of engine")

    def SwitchOffEngine(self):

        self.isStarted = False

    def Refuel(self, fuel):

        self.fuel = fuel

    def getFuel(self):

        return self.fuel

    def getEngineStatus(self):

        return self.isStarted

class ControlPanel:

    class Direction:

        RIGHT = "RIGHT"

        LEFT = "LEFT"

        FORWARD = "FORWARD"

    def \_\_init\_\_(self):

        self.speed = 0

        self.direction = self.Direction.FORWARD

    def SpeedChange(self, speed):

        self.speed = speed

    def getSpeed(self):

        return self.speed

    def TurnRight(self):

        self.direction = self.Direction.RIGHT

    def TurnLeft(self):

        self.direction = self.Direction.LEFT

    def Forward(self):

        self.direction = self.Direction.FORWARD

    def getDirection(self):

        return self.direction

class Door:

    def \_\_init\_\_(self):

        self.isClosed = False

    def Open(self):

        self.isClosed = False

    def Close(self):

        self.isClosed = True

    def getDoorStatus(self):

        return self.isClosed

**Файл ReusableSpaceship.py**

import os

from Spaceship import Spaceship, Engine, ControlPanel, Door

class Mission:

    def TakeOff(self):

        pass

    def Land(self):

        pass

    def getStatus(self):

        pass

class ReusableSpaceship(Spaceship, Mission):

    def \_\_init\_\_(self, fuel=0, flightsCount=0):

        super().\_\_init\_\_(fuel)

        self.flightsCount = flightsCount

        self.inMission = False

    def setFlightsCount(self, count):

        self.flightsCount = count

    def getFlightsCount(self):

        return self.flightsCount

    def StartSpaceship(self):

        self.engine.StartEngine()

    def SwitchOffSapceship(self):

        self.engine.SwitchOffEngine()

    def RefuelSpaceship(self, fuel):

        self.engine.Refuel(fuel)

    def TakeOff(self):

        self.inMission = True

    def Land(self):

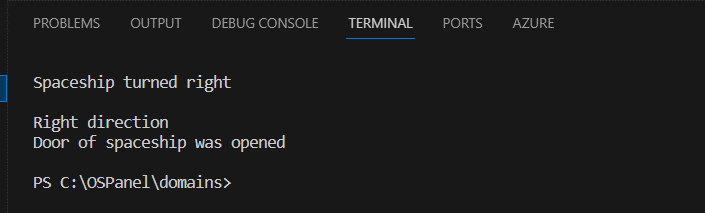
        self.inMission = False

        self.flightsCount += 1

    def getStatus(self):

        return self.inMission

**Результат роботи програми:**



**Відповідь на контрольні питання**

1. ***Що таке модулі?***

Модулі - це файли, які містять код для використання в інших програмах.

1. ***Як імпортувати модуль?***

Імпорт модуля здійснюється за допомогою ключового слова "import" та імені модуля.

1. ***Як оголосити клас?***

Клас оголошується ключовим словом "class", за яким слідує ім'я класу.

1. ***Що може міститися у класі?***

У класі можуть міститися атрибути (змінні) та методи (функції).

1. ***Як називається конструктор класу?***

Конструктор класу називається "init".

1. ***Як здійснити спадкування?***

Спадкування здійснюється за допомогою наслідування від іншого класу.

1. ***Які види спадкування існують?***

Існують одинарне та багатошарове спадкування.

1. ***Які небезпеки є при множинному спадкуванні, як їх уникнути?***

При множинному спадкуванні можуть виникати конфлікти та нерозбіжності, які слід уникнути шляхом коректної організації класів.

1. ***Що таке класи-домішки?***

Класи-домішки - це класи, які містять методи, які можна використовувати у інших класах.

1. ***Яка роль функції super() при спадкуванні?***

Функція super() використовується для виклику методів батьківського класу при спадкуванні.

**Висновок:** на цій лабораторній роботі, я оволодів навичками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python.