Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра ЕОМ



Звіт

з лабораторної роботи № 9

з дисципліни: «Кросплатформні засоби програмування »

на тему: «ФАЙЛИ ТА ВИКЛЮЧЕННЯ У PYTHON »

Варіант 19

Виконав:

ст. гр. КІ-301

Пазюк О.С.

Перевірив:

Майдан М.В.

Львів – 2023

Мета

Оволодіти навиками реалізації парадигм об’єктно-орієнтованого програмування використовуючи засоби мови Python

Завдання(Варіант 19)

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

Варіант завдання: Базовий клас: Прінтер, похідний: Багатофункціональний пристрій

Код програми:

MAIN.PY

from multifunctional\_device import Multifunctional\_device, Printer  
if \_\_name\_\_ == '\_\_main\_\_':  
 printer=Printer("Printer Multy","Sumsung", "A56Yt3", 1000)  
 print("--------------------------------------------")  
 printer.get\_info()  
 printer.number=0  
 printer.developer="Acer"  
 printer.get\_info()  
 print("--------------------------------------------")  
 md=Multifunctional\_device("MDP","Mycrosoft","23Ghj33",100,True,True)  
 md.get\_info()  
 print("--------------------------------------------")  
 md.model="333Fgdc2"  
 md.name="Kompikora"  
 md.get\_info()

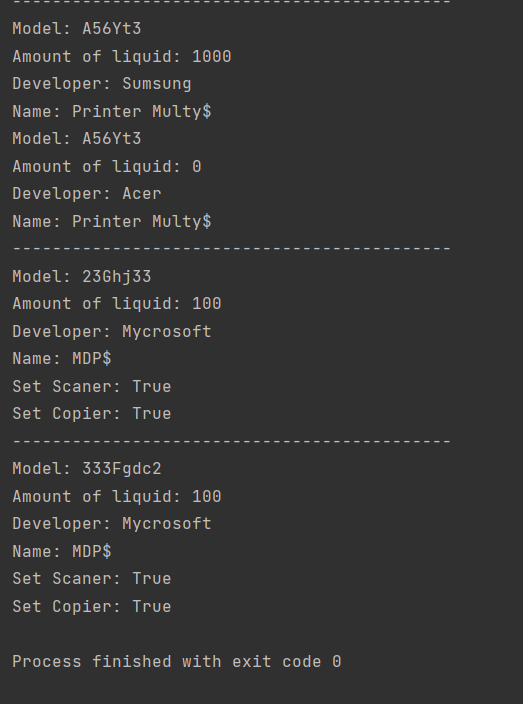
multifunctional\_device.PY

from printer import Printer  
class Multifunctional\_device(Printer):  
 def \_\_init\_\_(self, name, developer, model, n\_Liquid, is\_copier, is\_scaner):  
 super().\_\_init\_\_(name, developer, model, n\_Liquid)  
 self.\_is\_copier=is\_copier  
 self.\_is\_scaner=is\_scaner  
 # Getter for is\_copier  
 @property  
 def os(self):  
 return self.\_is\_copier  
  
 # Setter for is\_copier  
 @os.setter  
 def os(self, value):  
 self.\_is\_copier = value  
  
 # Getter for is\_scaner  
 @property  
 def camera\_px(self):  
 return self.\_is\_scaner  
  
 # Setter for is\_scaner  
 @camera\_px.setter  
 def camera\_px(self, value):  
 self.\_is\_scaner = value  
 def get\_info(self):  
 super().get\_info()  
 print(f"Set Scaner: {self.\_is\_scaner}")  
 print(f"Set Copier: {self.\_is\_copier}")

printer.PY

class Printer:  
 def \_\_init\_\_(self, name, developer, model, n\_Liquid):  
 self.\_name = name  
 self.\_developer = developer  
 self.\_model = model  
 self.\_n\_Liquid = n\_Liquid  
  
 # Getter for \_name  
 @property  
 def price(self):  
 return self.\_name  
  
 # Setter for \_name  
 @price.setter  
 def price(self, value):  
 self.\_name = value  
  
 # Getter for developer  
 @property  
 def developer(self):  
 return self.\_developer  
  
 # Setter for developer  
 @developer.setter  
 def developer(self, value):  
 self.\_developer = value  
  
 # Getter for model  
 @property  
 def model(self):  
 return self.\_model  
  
 # Setter for model  
 @model.setter  
 def model(self, value):  
 self.\_model = value  
  
 # Getter for \_n\_Liquid  
 @property  
 def number(self):  
 return self.\_n\_Liquid  
  
 # Setter for \_n\_Liquid  
 @number.setter  
 def number(self, value):  
 self.\_n\_Liquid = value  
  
 def get\_info(self):  
 print(f"Model: {self.\_model}")  
 print(f"Amount of liquid: {self.\_n\_Liquid}")  
 print(f"Developer: {self.\_developer}")  
 print(f"Name: {self.\_name}$")

Результати



Висновок

Я ознайомився з основними принципами мови програмування python та оволодів навиками застосування них. Закріпив теорію на практиці.