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 14**

*Виконав:*

*ст. гр. КІ-302*

*Левицький В.О.*

*Перевірив:*

*Іванов Ю.С.*

Львів – 2024

**Мета**

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

**Завдання(Варіант 14)**

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: Телевізор , похідний: Телевізор з тюнером

Код програми:

MAIN.PY

from smartTelevision import Television, SmartTelevision  
if \_\_name\_\_ == "\_\_main\_\_":  
 # Creating an object of the Television class  
 my\_tv = Television(brand="Sony", model="Bravia", screen\_size=55, resolution="4K")  
  
 # Displaying information about the television  
 print("=== Television Information ===")  
 my\_tv.get\_info()  
 print("=============================")  
  
 # Changing some parameters  
 my\_tv.set\_screen\_size(65)  
 my\_tv.set\_resolution("8K")  
  
 # Displaying updated information about the television  
 print("\n=== Updated Television Information ===")  
 my\_tv.get\_info()  
 print("=============================")  
  
 # Creating an object of the SmartTelevision class  
 my\_smart\_tv = SmartTelevision(brand="Samsung", model="QLED", screen\_size=50, resolution="4K",  
 has\_tuner=True, tuner\_type="Digital")  
  
 # Displaying information about the smart television  
 print("\n=== Smart Television Information ===")  
 my\_smart\_tv.get\_info()  
 print("=============================")  
  
 # Changing some parameters  
 my\_smart\_tv.set\_has\_tuner(False)  
 my\_smart\_tv.set\_tuner\_type("Analog")  
  
 # Displaying updated information about the smart television  
 print("\n=== Updated Smart Television Information ===")  
 my\_smart\_tv.get\_info()  
 print("=============================")

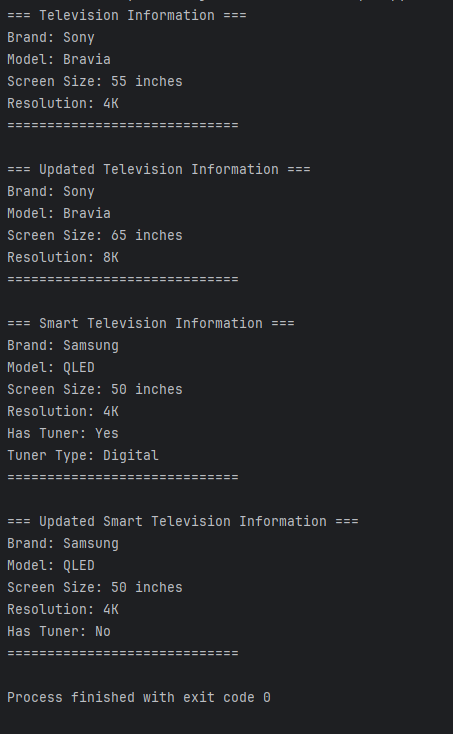
television.PY

class Television:  
 def \_\_init\_\_(self, brand, model, screen\_size, resolution):  
 self.\_brand = brand  
 self.\_model = model  
 self.\_screen\_size = screen\_size  
 self.\_resolution = resolution  
  
 # Getter methods  
 def get\_brand(self):  
 return self.\_brand  
  
 def get\_model(self):  
 return self.\_model  
  
 def get\_screen\_size(self):  
 return self.\_screen\_size  
  
 def get\_resolution(self):  
 return self.\_resolution  
  
 # Setter methods  
 def set\_brand(self, brand):  
 self.\_brand = brand  
  
 def set\_model(self, model):  
 self.\_model = model  
  
 def set\_screen\_size(self, screen\_size):  
 self.\_screen\_size = screen\_size  
  
 def set\_resolution(self, resolution):  
 self.\_resolution = resolution  
  
 # Method to get information about the television  
 def get\_info(self):  
 print("Brand:", self.\_brand)  
 print("Model:", self.\_model)  
 print("Screen Size:", self.\_screen\_size, "inches")  
 print("Resolution:", self.\_resolution)

smarttelevision.PY

from television import Television  
class SmartTelevision(Television):  
 def \_\_init\_\_(self, brand, model, screen\_size, resolution, has\_tuner, tuner\_type):  
 # Call the constructor of the base class Television  
 super().\_\_init\_\_(brand, model, screen\_size, resolution)  
 self.\_has\_tuner = has\_tuner  
 self.\_tuner\_type = tuner\_type  
  
 # Additional methods to get and set tuner-specific parameters  
 def has\_tuner(self):  
 return self.\_has\_tuner  
  
 def get\_tuner\_type(self):  
 return self.\_tuner\_type  
  
 def set\_has\_tuner(self, has\_tuner):  
 self.\_has\_tuner = has\_tuner  
  
 def set\_tuner\_type(self, tuner\_type):  
 self.\_tuner\_type = tuner\_type  
  
 # Overridden method to get information about the smart television  
 def get\_info(self):  
 # Call the base class method to get general television information  
 super().get\_info()  
 print("Has Tuner:", "Yes" if self.\_has\_tuner else "No")  
 if self.\_has\_tuner:  
 print("Tuner Type:", self.\_tuner\_type)

**Результати**

**\**

**Висновок**

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