Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра “Електронних обчислювальних машин”



**Звіт**

до лабораторної роботи № 9

з дисципліни «Кросплатформні засоби програмування ”

на тему:

“ ОСНОВИ ОБ’ЄКТНО-ОРІЄНТОВАНОГО

ПРОГРАМУВАННЯ У PYTHON”

**Виконав:**

КІ-307

Магора О.Т.

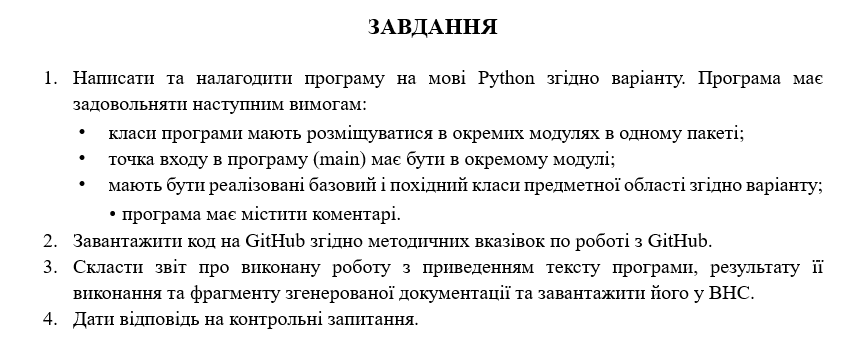
**Перевірив:**

Іванов Ю.С

**2023**

**Мета роботи:** оволодіти навиками реалізації парадигм об’єктно-орієнтованого

програмування використовуючи засоби мови Python.



**Варіант:**

Базовий клас:



Похідний клас:

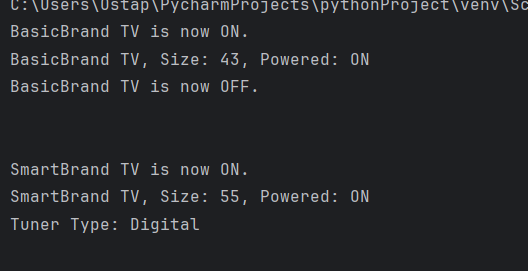


**Код:**

# main.py  
  
# Імпорт класу TV з модуля tv.  
from tv import TV  
  
# Імпорт класу SmartTV з модуля smart\_tv.  
from smart\_tv import SmartTV  
  
# Основна функція програми.  
def main():  
 # Створення екземпляра базового телевізора (TV) з параметрами "BasicBrand" та 43 дюйми.  
 basic\_tv = TV("BasicBrand", 43)  
 # Увімкнення базового телевізора.  
 basic\_tv.turn\_on()  
 # Виведення інформації про базовий телевізор.  
 basic\_tv.show\_info()  
 # Вимкнення базового телевізора.  
 basic\_tv.turn\_off()  
  
 # Вивід порожнього рядка для розділення виведених результатів.  
 print("\n")  
  
 # Створення екземпляра смарт-телевізора (SmartTV) з параметрами "SmartBrand", 55 дюймів і "Digital".  
 smart\_tv = SmartTV("SmartBrand", 55, "Digital")  
 # Увімкнення смарт-телевізора.  
 smart\_tv.turn\_on()  
 # Виведення інформації про смарт-телевізор.  
 smart\_tv.show\_info()  
 # Вимкнення смарт-телевізора.  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # Виклик основної функції, якщо сценарій виконується безпосередньо (не імпортується як модуль).  
 main()

# smart\_tv.py  
  
# Імпорт базового класу TV з модуля tv.  
from tv import TV  
  
# Визначення класу SmartTV, який успадковує властивості та методи класу TV.  
class SmartTV(TV):  
 # Конструктор класу SmartTV, який викликає конструктор класу TV та додає новий параметр tuner\_type.  
 def \_\_init\_\_(self, brand, size, tuner\_type):  
 super().\_\_init\_\_(brand, size)  
 self.tuner\_type = tuner\_type  
  
 # Перевизначення методу show\_info класу TV для додавання інформації про тип тюнера смарт-телевізора.  
 def show\_info(self):  
 # Виклик методу show\_info класу TV за допомогою super().  
 super().show\_info()  
 # Виведення інформації про тип тюнера смарт-телевізора.  
 print(f"Tuner Type: {self.tuner\_type}")

# Визначення класу TV для представлення основного телевізора.  
class TV:  
 # Конструктор класу, ініціалізує бренд, розмір та властивість powered\_on.  
 def \_\_init\_\_(self, brand, size):  
 self.brand = brand  
 self.size = size  
 self.powered\_on = False  
  
 # Метод для увімкнення телевізора.  
 def turn\_on(self):  
 self.powered\_on = True  
 print(f"{self.brand} TV is now ON.")  
  
 # Метод для вимкнення телевізора.  
 def turn\_off(self):  
 self.powered\_on = False  
 print(f"{self.brand} TV is now OFF.")  
  
 # Метод для виведення інформації про телевізор, включаючи бренд, розмір та стан ввімкнення/вимкнення.  
 def show\_info(self):  
 print(f"{self.brand} TV, Size: {self.size}, Powered: {'ON' if self.powered\_on else 'OFF'}")



**Висновок:** Під час виконання даної лабораторної роботи я оволодів навикам реалізації парадигм об’єктно-орієнтованогопрограмування використовуючи засоби мови Python.