



ЗВІТ
з лабораторної роботи №3
з дисципліни 'Кросплатформні засоби програмування'

Виконала:
Ст.гр. КІ-303
Онишкевич Тетяна
Перевірів:
Майдан М.В

Львів - 2023

СПАДКУВАННЯ ТА ІНТЕРФЕЙСИ

Завдання:

1. Написати та налагодити програму на мові Java, що розширює клас, що реалізований у лабораторній роботі №2, для реалізації предметної області заданої варіантом. Суперклас, що реалізований у лабораторній роботі №2, зробити абстрактним. Розроблений підклас має забезпечувати механізми свого коректного функціонування та реалізовувати мінімум один інтерфейс. Програма має розміщуватися в пакеті Група.Прізвище.Lab3 та володіти коментарями, які дозволять автоматично згенерувати документацію до розробленого пакету.
2. Автоматично згенерувати документацію до розробленого пакету.
3. Завантажити код на GitHub згідно методичних вказівок по роботі з GitHub.
4. Скласти звіт про виконану роботу з приведенням тексту програми, результату її виконання та фрагменту згенерованої документації та завантажити його у ВНС.
5. Дати відповідь на контрольні запитання.

Варіант:

14. Телевізор з тюнером

Код програми:

```
from abc import ABC, abstractmethod

class TV(ABC):
    """Абстрактний клас"""

    def __init__(self, brand: str, price: float,
screen_size: int):
        self.COLOUR = ("neutral", "warm", "white",
"cool", "native")
        self.STATES = ("OFF", "ON")
        self.BASE_CHANNEL = 0
        self.BASE_VOLUME = 50

        self.brand = brand
        self.price = price
        self.screen_size = screen_size

        self.is_on = True
        self.volume = self.BASE_VOLUME
        self.scheme = self.COLOUR[0]
        self.current_channel = self.BASE_CHANNEL

    @abstractmethod
    def state(self) -> None: ...
```

```

        @abstractmethod
        def set_colour_scheme(self, scheme: str) ->
None: ...

        @abstractmethod
        def chanel_switch(self, channel: int) -> None: ...

        @abstractmethod
        def volume_control(self, volume: int) -> None: ...

        @abstractmethod
        def __str__(self) -> str: ...

class Tuner:
    def __init__(self):
        ...

class TvWithTuner(TV, ABC):
    """Перевизначення класу"""

    def __init__(self, brand: str, price: float,
screen_size: int, tuner: Tuner):
        super().__init__(brand, price, screen_size)

        self.tuner = tuner

    def state(self) -> None:
        print(f"Turning off the {self.brand} TV...")
if self.is_on else print(f"Turning on the {self.brand}
TV...")

        self.is_on = not self.is_on

    def set_colour_scheme(self, scheme: str) -> None:
        self.scheme = scheme if scheme in self.COLOUR
else self.COLOUR[0]

    def chanel_switch(self, channel: int) -> None:
        self.current_channel = channel if self.is_on
else self.BASE_CHANNEL

        print(f"Changed to channel
{self.current_channel}")

    def volume_control(self, volume: int) -> None:
        self.volume = volume if volume in range(100)
else self.BASE_VOLUME

        print(f"Changed on {self.volume} dB")

```

```

def __str__(self) -> str:
    return (f"Model: {self.brand}\n"
            f"Price: {self.price} $\n"
            f"Screen size: {self.screen_size}
Inch \n"
            f"Power: {self.STATES[self.is_on]}\n"
            f"Channel: {self.current_channel}\n"
            f"Volume: {self.volume}")

def main():
    tv = TvWithTuner("Samsung", 130, 45, Tuner())
    tv.chanel_switch(10)
    tv.set_colour_scheme("white")
    print(tv)

if __name__ == '__main__':
    main()

```

[Посилання на Github](#)

Результат:

```

C:\Users\tania\python-3.12.0-amd64.exe\Scripts
Changed to channel 10
Model: Samsung
Price: 130 $
Screen size: 45 Inch
Power: ON
Channel: 10
Volume: 50

Process finished with exit code 0

```