Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних технологій, автоматики та метрології

Кафедра ЕОМ



**Звіт**

**До лабораторної роботи №9**

# З дисципліни: «Кросплатформні засоби програмування»

На тему «ОСНОВИ ОБ’ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ У PYTHON»

# Варіант №17

Виконав: ст. гр. КІ-305

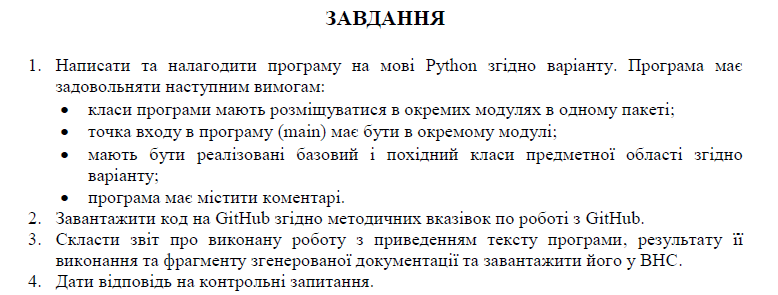
Легкобит Н.В

Прийняв:

Олексів М.В

Львів – 2023

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





**Виконання:**

**Лістинг програми:**

class Button:  
 def \_\_init\_\_(self, action):  
 self.action = action  
  
 @property  
 def get\_action(self):  
 return self.action  
  
 @property  
 def set\_action(self, action):  
 self.action = action  
  
 def \_\_str\_\_(self):  
 return f'Button{{ action=\'{self.action}\'}}'

class HardDisk:  
 def \_\_init\_\_(self, capacity, producer):  
 self.capacity = capacity  
 self.producer = producer  
  
 @property  
 def get\_capacity(self):  
 return self.capacity  
  
 @property  
 def get\_producer(self):  
 return self.producer  
  
 @property  
 def set\_capacity(self, capacity):  
 self.capacity = capacity  
  
 @property  
 def set\_producer(self, producer):  
 self.producer = producer  
  
 def \_\_str\_\_(self):  
 return "HardDisk{ " + "capacity = " + str(self.capacity) + " mb." + ", producer = '" + self.producer + '\'' + '}'

from Screen import Screen  
from HardDisk import HardDisk  
from VideocassetteRecorder import VideocassetteRecorder  
  
def main():  
 screen\_instance = Screen(7.8, "720x1980")  
 hard\_disk\_instance = HardDisk(1000, "Harman")  
  
 vcr\_instance = VideocassetteRecorder(screen\_instance, hard\_disk\_instance, "VHS")  
  
 vcr\_instance.add\_video("My birthday video")  
 vcr\_instance.add\_video("First time in France")  
 vcr\_instance.add\_video("Lviv")  
 vcr\_instance.add\_video("Party")  
 vcr\_instance.add\_video("Karate competition")  
  
 vcr\_instance.turn\_on\_next\_video()  
 vcr\_instance.turn\_on\_next\_video()  
 vcr\_instance.turn\_on\_next\_video()  
  
 vcr\_instance.turn\_on\_prev\_video()  
 vcr\_instance.turn\_on\_prev\_video()  
 vcr\_instance.turn\_on\_prev\_video()  
 vcr\_instance.turn\_on\_prev\_video()  
  
 print(vcr\_instance)  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

class Screen:  
 def \_\_init\_\_(self, diagonal, expansion):  
 self.diagonal = diagonal  
 self.expansion = expansion  
  
 @property  
 def get\_diagonal(self):  
 return self.diagonal  
  
 @property  
 def get\_expansion(self):  
 return self.expansion  
  
 @property  
 def set\_diagonal(self, diagonal):  
 self.diagonal = diagonal  
  
 @property  
 def set\_expansion(self, expansion):  
 self.expansion = expansion  
  
 def \_\_str\_\_(self):  
 return f'Screen{{ diagonal = {self.diagonal}, expansion = \'{self.expansion}\'}}'

from VideoPlayer import VideoPlayer  
  
class VideocassetteRecorder(VideoPlayer):  
 def \_\_init\_\_(self, screen, hard\_disk, cassette\_type):  
 super().\_\_init\_\_(screen, hard\_disk)  
 self.cassette\_type = cassette\_type  
  
 def insert\_cassette(self):  
 print(f"Inserting {self.cassette\_type} cassette into the videocassette recorder")  
  
 def eject\_cassette(self):  
 print(f"Ejecting {self.cassette\_type} cassette from the videocassette recorder")  
  
 def record\_video\_to\_cassette(self, video):  
 print(f"Recording video to {self.cassette\_type} cassette: {video}")

from Button import Button  
import logging  
  
class VideoPlayer:  
 def \_\_init\_\_(self, screen, hard\_disk):  
 self.next\_video = Button("next video")  
 self.prev\_video = Button("prev video")  
 self.pause = Button("pause")  
 self.logger = logging.getLogger("logs.txt")  
 self.screen = screen  
 self.hard\_disk = hard\_disk  
 self.videos = []  
 self.cur\_video = 0  
  
 self.logger.info("VideoPlayer constructor called")  
  
 def add\_video(self, video):  
 self.videos.append(video)  
 print(f"{video} was added to video player")  
 self.logger.info("VideoPlayer add\_video method was called")  
  
 def turn\_on\_next\_video(self):  
 self.logger.info("turn\_on\_next\_video VideoPlayer method was called")  
 if self.cur\_video == len(self.videos) - 1:  
 print(f"You push button {self.next\_video.get\_action}")  
 print(f"Now showing {self.videos[self.cur\_video]}")  
 self.cur\_video = 0  
 elif 0 <= self.cur\_video < len(self.videos) - 1:  
 print(f"You push button {self.next\_video.get\_action}")  
 print(f"Now showing {self.videos[self.cur\_video]}")  
 self.cur\_video += 1  
  
 def turn\_on\_prev\_video(self):  
 self.logger.info("turn\_on\_prev\_video VideoPlayer method was called")  
 if self.cur\_video == 0:  
 print(f"You push button {self.prev\_video.get\_action}")  
 print(f"Now showing {self.videos[self.cur\_video]}")  
 self.cur\_video = len(self.videos) - 1  
 elif 0 < self.cur\_video <= len(self.videos) - 1:  
 print(f"You push button {self.prev\_video.get\_action}")  
 print(f"Now showing {self.videos[self.cur\_video]}")  
 self.cur\_video -= 1  
  
 def \_\_str\_\_(self):  
 return f'VideoPlayer{{ screen={self.screen}, hard\_disk={self.hard\_disk}, videos={self.videos}, cur\_video={self.cur\_video}}}'

**Результати:**

****

**Висновок:** ознайомився з основами ООП на мові програмування Python.