Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра АСУ



**Звіт**

до виконаної лабораторної роботи №5

з дисципліни “Об’єктно-орієнтоване програмування”

на тему:

**“ Поліморфізм та неявна типізація”**

Виконав

студент групи КН-118

Cтельмах Віталій

Прийняв

Цимбал Ю.В.

Львів – 2021

## Лабораторна робота № 5

*Тема роботи:* **Поліморфізм та неявна типізація**

***Мета роботи:*** ознайомитись з поняттям поліморфізму, його застосуванням та вивчити механізм його реалізації за допомогою неявної типізації.

***Послідовність роботи:***

1. Для заданого варіанта завдання написати програму з використанням поліморфізму на рівні об’єктів і рівні методів та продемонструвати дію механізму неявної типізації (duck typing).

2. Оформити звіт про роботу за такою структурою:

* назва роботи;
* мета роботи;
* послідовність роботи;
* індивідуальне завдання;
* текст програми;
* контрольний приклад та результати роботи програми;
* висновки.

20. Створити базовий клас ОЛІМПІАДНІ ЗАВДАННЯ (дані про учасника, кількість тестових прикладів, кількість пройдених тестів). Створити похідні класи ЗАВДАННЯ «ЧИМ БІЛЬШЕ, ТИМ КРАЩЕ» (задається максимальна кількість балів за завдання (бали учаснику виставляються пропорційно відсотку пройдених тестів) та ЗАВДАННЯ «ЧИМ ШВИДШЕ, ТИМ КРАЩЕ» (задається час учасника на розв’язання, найкращий час усіх учасників, максимальна кількість балів за завдання, відсоток зниження балу за хвилину відставання від найкращого часу). Для заданих прикладів завдань, які розв’язували учасники, впорядкувати учасників за зростанням набраних балів і визначити сумарну кількість балів, набраних учасниками олімпіади. Для перевірки використати дії над списком, у якому розміщено об’єкти різних похідних класів.

**Код програми**

mass=[]

class OlimpTask:

def \_\_init\_\_(self):

self.name=0

self.count\_all=0

self.count\_complate=0

def get\_name(self):

return self.name

def get\_count\_all(self):

return self.count\_all

def get\_count\_complate(self):

return self.count\_complate

def set\_name(self,nm):

self.name=nm

def set\_count\_all(self,all):

self.count\_all=all

def set\_count\_complate(self,comp):

self.count\_complate=comp

class TheMoreTheBetter(OlimpTask):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.max\_point=0

def get\_max\_point(self):

return self.max\_point

def set\_max\_point(self,max):

self.max\_point=max

def CalcPoint(self):

return ((self.get\_count\_complate()\*self.get\_max\_point())/self.get\_count\_all())

def \_\_str\_\_(self):

return "Ім'я - {} " \

"Кількість прикладів = {} " \

"Кількість вирішених = {} " \

"Максимальна кількість балів = {} " \

"Отримані бали = {}".format(mass[i].get\_name(),

mass[i].get\_count\_all(),

mass[i].get\_count\_complate(),

mass[i].get\_max\_point(),

round(mass[i].CalcPoint(),2))

class TheFasterTheBetter(OlimpTask):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.user\_time=0

self.best\_time=0

self.percent=0

self.max\_point=0

def get\_user\_time(self):

return self.user\_time

def get\_best\_time(self):

return self.best\_time

def get\_percent(self):

return self.percent

def get\_max\_point(self):

return self.max\_point

def set\_user\_time(self,time):

self.user\_time=time

def set\_best\_time(self,best):

self.best\_time=best

def set\_percent(self,perc):

self.percent = perc

def set\_max\_point(self, max):

self.max\_point = max

def CalcPoint(self):

if self.get\_max\_point() - ((self.get\_user\_time()-self.get\_best\_time())\*self.get\_percent())>0:

return (self.get\_max\_point() - ((self.get\_user\_time()-self.get\_best\_time())\*self.get\_percent()))

else:

return 0

def \_\_str\_\_(self):

return "Ім'я - {} " \

"Час виконання = {} " \

"Найкращий час = {} " \

"Відсоток зниження за хв = {} " \

"Максимальна кількість балів = {} " \

"Отримані бали = {}".format(mass[i].get\_name(),

mass[i].get\_user\_time(),

mass[i].get\_best\_time(),

mass[i].get\_percent(),

mass[i].get\_max\_point(),

round(mass[i].CalcPoint(), 2))

max = int(input("Максимальна кількість балів для учасників олімпіади\n"))

perc = float(input("Відсоток зниження за хвилину відставання\n"))

best\_time=9999999

menu = 1

while menu == 1:

a = 1

n = int(input("Виберіть дію \n"

"1-додати об'єкт класу TheMoreTheBetter\n"

"2-додати об'єкт класу TheFasterTheBetter\n"

"3-вивести список \n"

"4-відсортувати по сумі балів\n"

"5-сума балів, які здобули всі учасники\n"

"0-вийти з програми\n"))

if(n==1):

while a==1:

name = str(input("ім'я\n"))

all = int(input("кількість прикладів\n"))

comp = int(input("кількість вирішених\n"))

b=1

while b == 1:

if int(all)>=int(comp):

obj = TheMoreTheBetter()

obj.set\_name(name)

obj.set\_count\_all(all)

obj.set\_count\_complate(comp)

obj.set\_max\_point(max)

mass.append(obj)

b=0

else:

comp = input("кількість вирішених, не може бути більшою за загальну введіть ще раз\n")

b=1

a = int(input("Продовжити - 1\n"))

if (n == 2):

while a == 1:

name = str(input("ім'я\n"))

time = int(input("час в хвилинах\n"))

if time<best\_time:

best\_time=time

for i in range(len(mass)):

if type(mass[i]) == TheFasterTheBetter:

mass[i].set\_best\_time(best\_time)

obj = TheFasterTheBetter()

obj.set\_name(name)

obj.set\_user\_time(time)

obj.set\_best\_time(best\_time)

obj.set\_percent(perc)

obj.set\_max\_point(max)

mass.append(obj)

a = int(input("Продовжити - 1\n"))

if (n == 3):

for i in range (len(mass)):

print (mass[i])

if (n == 4):

N = len(mass)

for i in range(N - 1):

m = mass[i].CalcPoint()

p = i

for j in range(i + 1, N):

if m > mass[j].CalcPoint():

m = mass[j].CalcPoint()

p = j

if p != i:

t = mass[i]

mass[i] = mass[p]

mass[p] = t

print("Відсортовані учасники по балах")

for i in range (len(mass)):

print(mass[i])

if (n == 5):

print("Сума всіх балів набраних учасниками")

sumall=0

for i in range (len(mass)):

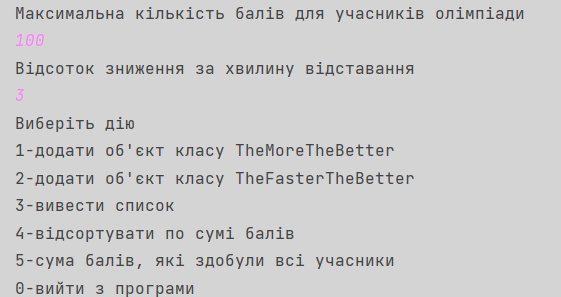
sumall+=mass[i].CalcPoint()

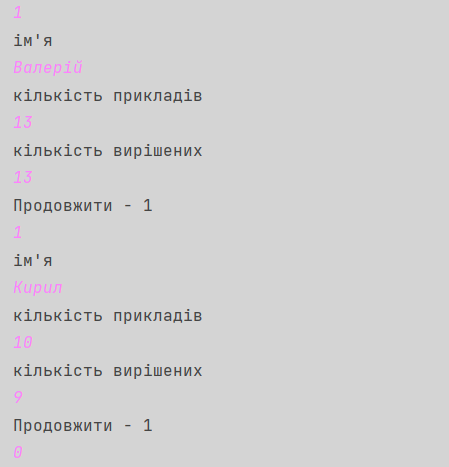
print(str(round(sumall,2)))

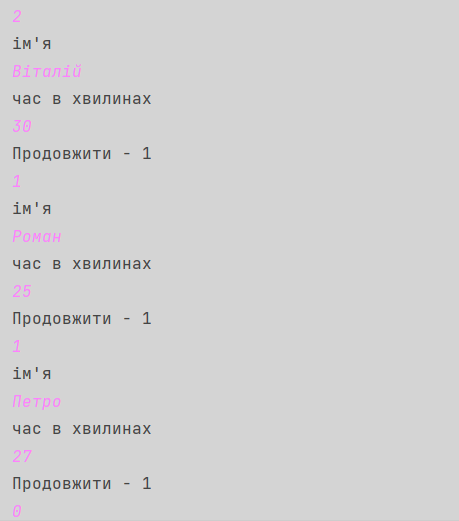
if (n == 0):

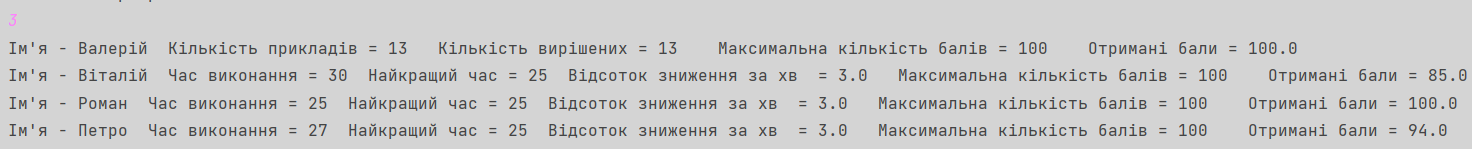
menu=0

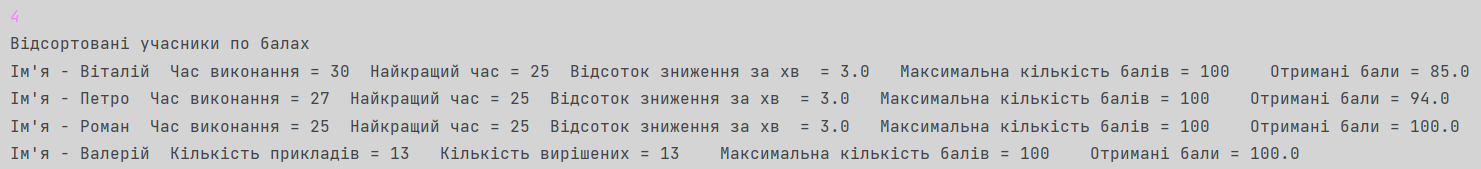
**Приклад роботи програми**

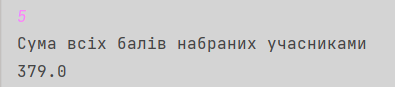












***Висновок:*** На даній лабораторній роботі я ознайомився з поняттям поліморфізму, його застосуванням та вивчив механізм його реалізації за допомогою неявної типізації.