Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра АСУ



**Звіт**

до виконаної лабораторної роботи №6

з дисципліни “Об’єктно-орієнтоване програмування”

на тему:

**“Алгоритми розв’язання лабіринтних та ігрових задач”**

Виконав

студент групи КН-118

Cтельмах Віталій

Прийняв

Цимбал Ю.В.

Львів – 2021

*Лабораторна робота № 6*

*Тема роботи:* **Алгоритми розв’язання лабіринтних та ігрових задач**

***Мета роботи:*** ознайомитись з алгоритмічними стратегіями та навчитись розробляти ефективні алгоритми розв’язання лабіринтних та ігрових задач, оволодіти практичними прийомами написання об’єктно-орієнтованих програм мовою Python на основі таких алгоритмів.

***Послідовність роботи:***

1. Ознайомитись з теоретичним матеріалом про стратегії та алгоритми розв’язання лабіринтних та ігрових задач.

2. Для заданого варіанту завдання розробити алгоритм та написати програму розв’язання задачі, використовуючи засоби об’єктно-орієнтованого програмування:

* описати класи (не менше 2-х) для об’єктів з умови задачі, реалізувати конструктори, методи доступу, перевантажені операції для подання об’єкта у вигляді символьного рядка;
* в одному з класів реалізувати метод для знаходження розв’язку задачі.

Ваш алгоритм може бути більш ефективним, ніж повний перебір усіх можливих варіантів розв’язків задачі, повинен давати розв’язок для різних прикладів вхідних даних (не лише для наведеного у завданні) і допускати масштабованість задачі, наприклад, збільшення розміру вхідних даних.

3. Передбачити можливості:

* зчитування вхідних даних із заданого файлу;
* виведення результатів на екран або у файл.

4. Визначити часові характеристики роботи програми (сумарний час на виконання обчислень, не враховуючи введення з клавіатури та виведення на екран).

5. Зробити висновок про ефективність та універсальність застосованих алгоритмів та засобів об’єктно-орієнтованого програмування при розв’язанні даної задачі.

6. Оформити звіт про роботу за такою структурою:

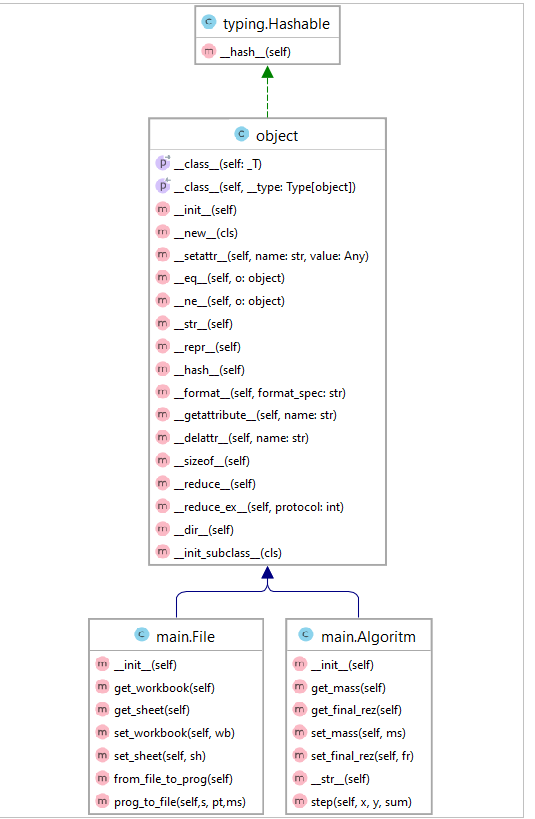
* назва роботи;
* мета роботи;
* послідовність роботи;
* індивідуальне завдання;
* діаграма класів;
* алгоритм розв’язання задачі та оцінка його часової складності;
* текст програми;
* формат файлу з вхідними даними та інструкція користувачу;
* контрольний приклад та результати роботи програми, часові характеристики;
* висновки.

**Завдання**

20. Від верхнього лівого квадрата прокладіть такий шлях до нижнього правого квадрата, щоб сума чисел, записаних у проміжних квадратах, а також у стартовому та фінішному, складала 112. (Шуканий маршрут може перетинати лише сторони, але не вершини проміжних квадратів і проходити через кожен проміжний квадрат лише один раз).



**Діаграма класів**



**Код програми**

import easygui as eg

import xlrd

import xlsxwriter as xlwr

import time

class File:

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

self.workbook=""

self.sheet=""

def get\_workbook(self):

return self.workbook

def get\_sheet(self):

return self.sheet

def set\_workbook(self, wb):

self.workbook = wb

def set\_sheet(self, sh):

self.sheet = sh

def from\_file\_to\_prog(self):

book = xlrd.open\_workbook(self.get\_workbook())

sheet = book.sheet\_by\_index(self.get\_sheet())

mass\_from\_file = [[sheet.cell\_value(r, c) for c in range(1,sheet.ncols,1)] for r in range(1,sheet.nrows,1)]

fin\_rez = int(sheet.cell\_value(0, 0))

for i in range(len(mass\_from\_file)):

for j in range(len(mass\_from\_file[0])):

if mass\_from\_file[i][j]=='':

mass\_from\_file[i][j]=fin\_rez

mass\_from\_file[i][j] = int(mass\_from\_file[i][j])

return (mass\_from\_file,fin\_rez)

def prog\_to\_file(self,s, pt,ms):

pt[0][0]=True

for i in range(len(pt)):

for j in range(len(pt[0])):

if pt[i][j]==True:

pt[i][j]=ms[i][j]

else: pt[i][j]=0

with xlwr.Workbook(s) as workbook:

worksheet = workbook.add\_worksheet()

for row\_num, data in enumerate(pt):

worksheet.write\_row(row\_num, 0, data)

class Algoritm:

path = []

path.append([0, 0, 0, 0, 0])

path.append([0, 0, 0, 0, 0])

path.append([0, 0, 0, 0, 0])

path.append([0, 0, 0, 0, 0])

path.append([0, 0, 0, 0, 0])

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

self.mass=[]

self.final\_rez=0

def get\_mass(self):

return self.mass

def get\_final\_rez(self):

return self.final\_rez

def set\_mass(self, ms):

self.mass = ms

def set\_final\_rez(self, fr):

self.final\_rez = fr

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

vivod=''

for i in range(len(self.mass)):

for j in range(len(self.mass[0])):

if self.mass[i][j]==self.get\_final\_rez():

vivod += "X" + " "

else: vivod += str(self.mass[i][j]) + " "

vivod += "\n"

return (vivod)

def step(self, x, y, sum):

vivod = ""

sum+=self.mass[x][y]

if sum>self.get\_final\_rez():

return False

if x == len(self.mass)-1 and y== len(self.mass[0])-1:

if(sum!=self.get\_final\_rez()):

return False

self.path[x][y]=True

for i in range(len(self.mass)):

for j in range(len(self.mass[0])):

if self.path[i][j]==True:

vivod+=str(self.mass[i][j]) + " "

else: vivod+="-" + " "

vivod+="\n"

print(vivod)

self.path[x][y] = True

#up

if y > 0 and not(self.path[x][y-1]):

self.step(x,y-1,sum)

#down

if y < 4 and not (self.path[x][y + 1]):

self.step(x, y + 1, sum)

#left

if x > 0 and not (self.path[x-1][y]):

self.step(x-1, y, sum)

#right

if x < 4 and not (self.path[x+1][y]):

self.step(x+1, y, sum)

self.path[x][y] = False

return False

file = File()

a=eg.fileopenbox()

file.set\_workbook(a)

file.set\_sheet(0)

t = time.time\_ns()

alg = Algoritm()

alg.set\_mass(file.from\_file\_to\_prog()[0])

alg.set\_final\_rez(file.from\_file\_to\_prog()[1])

print(alg)

alg.step(0,0,0)

time\_alg = time.time\_ns() - t

print(f"Час виконання алгоритму = {time\_alg/10\*\*9} секунди")

qqq = int(input('Зберегти в файл?'))

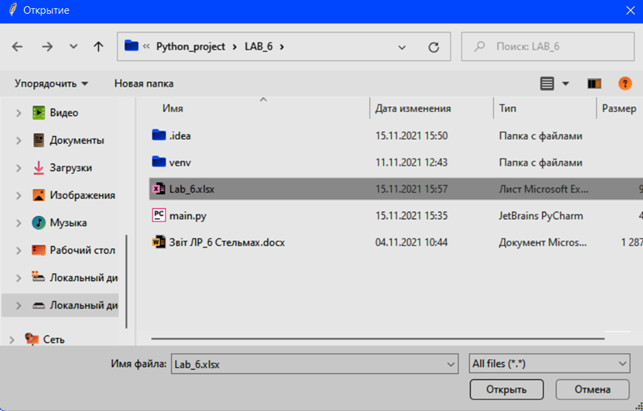
if qqq==1:

s=eg.filesavebox() + ".xlsx"

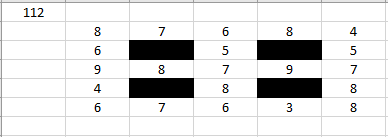
file.prog\_to\_file(s,alg.path,alg.mass)

**Приклад роботи програми**

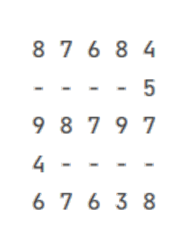
Вікно вибору вхідного файлу



Для внесення даних використовуємо excel у клітинці (0;0) міститься контрольна сума, починаючи з клітинки (1;1) вноситься квадратна матриця будь-якої розмірності

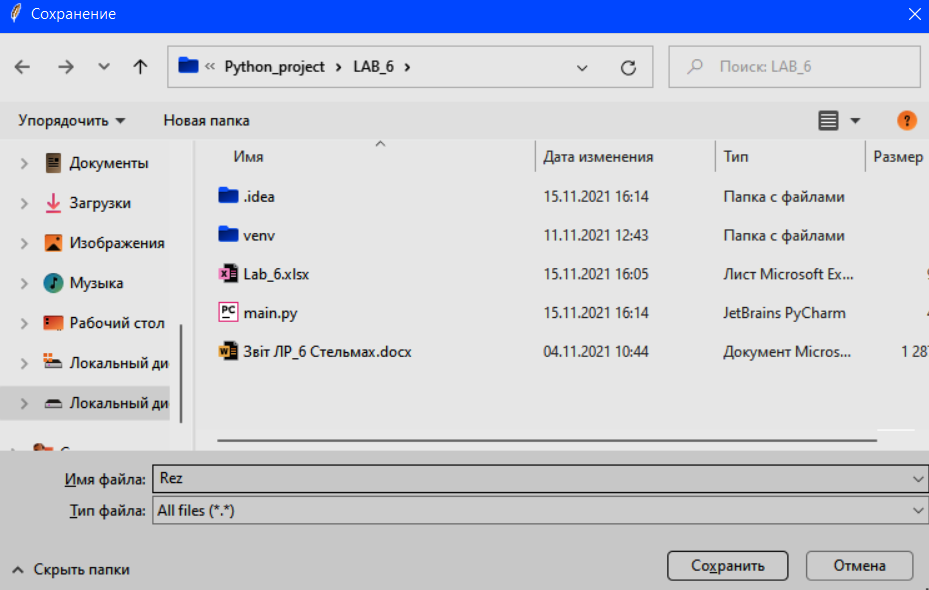


**Результат роботи**

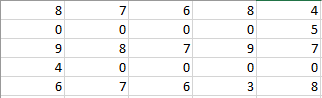
  



**Вікно збереження файлу**



**Вміст збереженого файлу**



***Висновок:*** На даній лабораторній роботі я ознайомився з алгоритмічними стратегіями та навчився розробляти ефективні алгоритми розв’язання лабіринтних та ігрових задач, оволодів практичними прийомами написання об’єктно-орієнтованих програм мовою Python на основі таких алгоритмів