

Міністерство освіти та науки України
Карпатський національний університет імені Василя Стефаника

Семаньків М.В.

**МЕТОДИЧНІ ВКАЗІВКИ ДО ЛАБОРАТОРНИХ РОБІТ
З ДИСЦИПЛІНИ “МОВА ПРОГРАМУВАННЯ PYTHON”**

Івано-Франківськ, 2025

УДК 004.43
ББК 32.973-018.1

Рекомендовано до друку Вченою радою факультету математики та інформатики
Карпатського національного університету імені Василя Стефаника
(протокол № 9 від 22.10.2025р.).

Рецензенти :

Горєлов В.О., кандидат технічних наук, доцент (Карпатський національний університет імені Василя Стефаника)

Петришин М.Л., кандидат технічних наук (Карпатський національний університет імені Василя Стефаника)

Семаньків М.В. Методичні вказівки до лабораторних робіт з дисципліни
“Мова програмування Python”. – Івано-Франківськ: Карпатський
національний університет імені Василя Стефаника, 2025. – 32 с.

Наведено завдання до лабораторних робіт з мови програмування Python.
Призначено для проведення лабораторних занять з курсу “Мова
програмування Python”.

Для студентів спеціальностей Комп’ютерні науки та Інформаційні системи та
технології.

© М.В.Семаньків, 2025

ЗМІСТ:

ЛАБОРАТОРНА РОБОТА №1 Синтаксис мови. Введення, виведення, перетворення типів даних	4
ЛАБОРАТОРНА РОБОТА №2 Списки	5
ЛАБОРАТОРНА РОБОТА №3 Вкладені списки. Генератори	7
ЛАБОРАТОРНА РОБОТА №4 Множини. Текст. КORTEЖ.	10
ЛАБОРАТОРНА РОБОТА №5 Класи	11
Завдання для самостійного виконання	13
ЛАБОРАТОРНА РОБОТА №1 PyQt. Designer	15
ЛАБОРАТОРНА РОБОТА №2 Робота з відметами	18
ЛАБОРАТОРНА РОБОТА №3 Діалогові вікна	21
ЛАБОРАТОРНА РОБОТА №4 Створення меню	23
ЛАБОРАТОРНА РОБОТА №5 Створення калькулятора	25
ЛАБОРАТОРНА РОБОТА №6 Розробка програмного продукту для тестового контролю знань	25
ЛАБОРАТОРНА РОБОТА №7 Панель інструментів	27
ЛАБОРАТОРНА РОБОТА №8 Побудова фракталів	29
Література	32

ЛАБОРАТОРНА РОБОТА №1

Синтаксис мови

Введення, виведення, перетворення типів даних

Теорія		Завдання
int(x [,ase])	Перетворення x в ціле число. Приклад, int(12.4) -> 12	1. Ввести число x з клавіатури, задавши тип ціле число 2. Вивести число. 3. Додати 0.8. вивести число 4. Задати цілий тип для x. вивести 5. Задати тип рядок для x. Вивести x*3
float(x)	Перетворення x в число с плаваючою комою. Приклад float(10) - 10.0	
complex(real [,img])	Створення комплексного числа. Приклад complex(20) -> (20+0j)	
str(x)	Перетворення x в текст. Приклад str(10) -> '10'	

Розгалуження

if Умова: Блок 1 else: Блок 2	if Умова: Блок 1 elif Умова 2: Блок 2 else: Блок 3	6. Для даного числа n<100 закінчіть фразу "Zalik zdalo..." реченням "n studentiv", "n studenty", "n student", правильно використовуючи закінчення
--	---	--

Бібліотека math

import math		from math import *		7. Вхідні дані: 2 невід'ємних дійсних числа a та b. b не дорівнює 0. Вихідні дані: дійсне число -- результат обчислення формули $x = \frac{\sqrt{ab}}{e^a * b} + ae^{\frac{2a}{b}}$ Приклад Вхідні дані: 0 1 Результат: 0.0 Вхідні дані: 0.5 10 Результат: 0.688209837593													
y = math.sin(x) print(math.sin(math.pi/2))		y = sin(x) print(sin(pi/2))															
<table><tr><td>round(x)</td><td>Округлення до ближчого числа</td></tr><tr><td>floor(x)</td><td>Округлення «вниз»</td></tr><tr><td>ceil(x)</td><td>Округлення «вверх»</td></tr><tr><td>abs(x)</td><td>модуль</td></tr><tr><td>sqrt(x)</td><td>Корінь</td></tr><tr><td>pow(a, b)</td><td>a^b.</td></tr><tr><td>exp(x)</td><td>експонента</td></tr></table>		round(x)	Округлення до ближчого числа	floor(x)	Округлення «вниз»	ceil(x)	Округлення «вверх»	abs(x)	модуль	sqrt(x)	Корінь	pow(a, b)	a ^b .	exp(x)	експонента		
round(x)	Округлення до ближчого числа																
floor(x)	Округлення «вниз»																
ceil(x)	Округлення «вверх»																
abs(x)	модуль																
sqrt(x)	Корінь																
pow(a, b)	a ^b .																
exp(x)	експонента																

Цикли

for i in range(n): Тіло циклу range(x) -- генерує список цілих значень від 0 до x-1. range(x, y) -- генерує список цілих значень від x до y-1. range(x, y, z) -- генерує список цілих значень від x до y-1 з кроком z.	8. Не використовуючи формулу суми геометричної прогресії, обчислити $1+x+x^2+...+x^n$. Програма отримує на вхід ціле число n і дійсне число x. Вхідні дані: 4 0.1 Вхідні дані: 1.1111 9. Дано число, перевірити чи воно є простим 10. Квиток на одну поїздки в метро коштує 15 гривень, квиток на 10 поїздок коштує 125 гривень, квиток на 60 поїздок коштує 440 гривень. Пасажир має намір здійснити n поїздок. Для заданої кількості поїздок, порахувати яку найменшу суму може потратити пасажир, скориставшись максимально знижками. Вивести суму 11. Дано цілі невід'ємні числа a, b, c, d, при цьому $0 \leq c < d$. Виведіть у порядку зростання всі числа від a до b, які дають остачу c при діленні на d. Примітка У цьому завданні не можна використовувати інструкцію if.						
<table border="1"> <tr><td>%</td><td>остача</td></tr> <tr><td>**</td><td>Піднести до степеня</td></tr> <tr><td>//</td><td>Відкидає дробову частину</td></tr> </table>	%	остача	**	Піднести до степеня	//	Відкидає дробову частину	
%	остача						
**	Піднести до степеня						
//	Відкидає дробову частину						

<pre>word = "child" # рядок word bag = ["knife", "wallet", "pen", "notebook"] # список bag countries = { "Swiss": "Bern", "Ukraine": "Kiev", "Italy": "Rome", "Australia": "Canberra", "Japan": "Tokyo" } # словник countries</pre>	for letter in word: print letter # друкує по одній букві word for item in bag: print item # друкує по одному елементу bag for country in countries: print "The capital of %s is %s" % (country, countries[country])
while умова: блок інструкцій	12. З клавіатури вводять число. Вивести всі його цифри

Рядки

<pre>([] и [:]) Оператор плюс (+) - з'єднання рядків (*) оператор повторення. len() [::-1] ,букви з кінця</pre>	13. Дано слово . Вивести всі букви 14. Дано рядок, що складається з двох слів, що розділені пропуском. Поміняти місцями слова. Вивести результат. 15. Дано рядок, в якому буква h зустрічається як мінімум два рази. Розвернути послідовність символів, що знаходяться між першим і останнім h в протилежний бік. 16. Визначити чи слово є паліндромом 17. Дано текст і слово. Якщо дане слово зустрічається в тексті, то замінити всі символи даного слова в тексті на «*»
---	--

Функції

<pre>def назва(змінні): блок дій return ____ def unknown(*args): for argument in args: print argument unknown("hello","world") unknown(1,2,3,4,5)</pre>	18. Написати функцію phib(n), яка для заданого цілого невід'ного числа n повертає n-е число з ряду Фібоначчі (1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233,...). Вхідні дані: 6 Вихідні дані: 8
<pre>def f(): global a a = 1 print (a) a = 0 f() print(a)</pre>	

ЛАБОРАТОРНА РОБОТА №2

СПИСКИ

Задання і введення елементів списку <pre>a = [] for i in range(int(input())): a.append(int(input())) print(a)</pre> len() – довжина списку del a[i] знищення елементу списку list.append(x) – додавання елементу в кінець списку list.extend(L) – розширює список list, додає в кінець всі елементи зі списку L list.insert(i, x) - Вставляє на i-ий елемент значення x list.remove(x) – Знищує перший елемент в списку, що має	1. Створити порожній список. Додати елементи з клавіатури. Вивести список. Вивести довжину списку. Знищити перший елемент 2. Згенерувати список a = [i ** 2 for i in range(n)]. Вивести 3. Задати список, що створений випадковими числами від 1 до 9
---	--

значення <code>x</code>. <code>ValueError</code>, якщо такого елементу не існує <code>list.pop([i])</code> - Знищує <code>i</code>-ий елемент і повертає його. Якщо індекс не вказаний, знищує останній елемент <code>list.index(x, [start [, end]])</code> – повертання індекс першого елемента зі значенням <code>x</code> (при цьому пошук пошук здійснюється від <code>start</code> до <code>end</code>) <code>list.count(x)</code> -повертає кількість елементів зі значенням <code>x</code> <code>list.sort([key=функція])</code> - Сортує список на основі функції <code>list.reverse()</code> - Реверсне список Додавання списків і множення списку на число: <code>a = [1, 2, 3]</code> <code>b = [4, 5]</code> <code>c = a + b</code> <code>d = b * 3</code>	<pre>from random import randrange n = int(input()) a = [randrange(1, 10) for i in range(n)]</pre> Вивести. Додати його в кінець попереднього списку 4. Ввести з клавіатури слово. Перетворити його в список
---	---

ВИКЛЮЧЕННЯ ПОМИЛОК

<pre>try: блок команд except Error: # Якщо не спрацював try і оголошена помилка Error finally: # виконується в будь-якому випадку</pre>	1. З клавіатури вводиться число <code>x</code>. Обчислити дріб <code>1/x</code>. Можливість виникнення помилки «ділення на нуль» визначити блоком, що виводить «Ділити на нуль не можна» <pre>try: ... except ZeroDivisionError:</pre>
2. Вивести текст: My menu 1: create list 2: add element 3: preview 4: exit 3. Задати цикл <code>while True:</code> , в якому з клавіатури вводиться число. Якщо ввели число 1, то задати команду створення порожнього списку і вивести «Done» Якщо 2 – вивести “Element:” і очікувати введення з клавіатури елемента, додати його до списку Якщо 3 – вивести весь список Якщо 4 вийти з циклу - <code>break</code> 4. Передбачити виникнення помилок: Якщо вводять число більше 4, то вивести повідомлення, що дане число не входить у пункти меню Якщо вводять не вірне значення <code>x</code> (наприклад, текст), задати конструкцію <pre>try: ... except ValueError: вивести повідомлення, що необхідно задати число</pre> Якщо додають елемент через меню команди 2, а список ще не створений, то задати конструкцію для команд другого пункту меню <pre>try: ... except: і вивести повідомлення, що необхідно спочатку створити список</pre> 5. В меню додати пункт, що задати створення автоматичного списку натуральних чисел визначеної довжини: <code>list(range(y))</code> 6. Додати пункт меню для знищення елемента списку за його номером	

СЛОВНИКИ

Кожний елемент словника складається з двох об'єктів: ключа і значення. Доступ до елементів за ключем, а не за індексом dict() — створення словника; len() — повертає кількість пар; clear() — знищує всі значення зі словника; copy() — створює псевдокопію словника; deepcopy() — створює повну копію словника; fromkeys() — створення словника; get() — отримання значення за ключем; has_key() — перевірка значення за ключем; items() — повертає список значень; iteritems() — повертає ітератор; keys() — повертає список ключів; iterkeys() — повертає ітератор ключів; pop() — витягує значення за ключем; update() — змінює словник; values() — повертає список значень; in — оператор, що перевіряє наявність значення за ключем; del — оператор, що знищує пару за ключем; dict() — конструктор словника з допомогою послідовності.	1. Дано словник з назвами міст та країн, де вони знаходяться. "Berlin": "Germany" З клавіатури вводиться назва міста, вивести чи є дане місто в словнику і з якої країни воно З назви міст утворити список Вивести кількість елементів словника (len()) З клавіатури вводиться назва міста, якщо таке місто є в словнику, то знищити його Зробити зі значень словника список (list(dict.values())) З клавіатури ввести ще один елемент словника
Перебір елементів <pre>for key in A: print(key, A[key])</pre> Знищення елементу за ключем <pre>key = 'ac' if key in A: del A[key] try: del A[key] except KeyError: print("There is no element with key " + key + " in dict") print(A)</pre>	

ЛАБОРАТОРНА РОБОТА №3

Вкладені списки

Вкладені списки	
<pre>a = [["b", "c"], ["d", "e"], ["f", "g", "h"]] for v in Nlist: for item in v: print (item)</pre>	1. Задати список з назвами товарів та їх цінами (вкладені списки). Мінімум 5 товарів <pre>food=[["Apple",3.4],["Banane",12.4], ...]</pre> 2. Вивести дані про перший товар 3. Вивести список товарів — кожен товар в окремому рядку 4. Вивести кількість товарів
<pre>print(a[1]) print(a[0][2])</pre>	

довжина списку - len()	
	5. З клавіатури ввести назву товару, наявного у списку (тобто його купуємо) і кількість (шт. даного товару). Знайти його ціну в списку, помножити на введену кількість і вивести результат. for i in range(len(food)): використовувати (food[i][0]) і (food[i][1])
	6. Задати створення чеку : з клавіатури вводиться декілька найменувань товарів, що купляють, з їхніми кількостями, створити список, де записуються назви куплених товарів і суми, що потрібно потратити при купівлі заданої кількості цього товару. Виводиться новостворений список купленого товару з сумами і вкінці «Total:» з результуючою сумою

Наприклад, один з варіантів чеку:

```
['Banane', '12.4x2=', 24.8]
['Apple', '3.4x5=', 17.0]
-----
Total 41.8
```

Генератори

a = [0] * 5	1. Створити список з 5 нулів
	2. Створити список з десяти одиниць
a = [i for i in range(1,15)]	3. Створити список з натуральних чисел на проміжку від 1 до 14
range(1,15,2)	4. Створити список з квадратів чисел проміжку (0,10)
{i:i**2 for i in a}	5. Створити список з парних натуральних чисел на проміжку від 1 до 14
	6. Дано список a=[1, 2, 3, 4, 5]. Створити словник, де ключ –це елемент списку, а значення ключа це квадрат цього числа
	7. Дано список [45,67,24,45]. Створити словник, де ключ –це елемент списку, а значення ключа це остача від ділення цього числа на 10
	8. Дано список ['dfg', 'sggsdh', 'sdgsdgsd']. Створити словник, де ключ –це слово зі списку, а значення ключа це довжина цього слова
	9. Дано список чисел. З допомогою генератора списку утворити новий список з додатніх елементів попереднього
a=[5,0,4,8] B=[i for i in a if i!=0]	10. Дано список чисел. З допомогою генератора списку утворити новий список відкинувши всі нулі з попереднього списку
	11. Дано список чисел. З допомогою генератора списку утворити новий список з додатніх елементів попереднього списку
[c*2 for c in 'ghtjrklist']	12. Дано текст. Створити список, де кожену букву подвоїти
[c * 3 for c in 'list' if c!='s']	13. Потроїти всі букви крім s.
a = {1:10, 2:20, 3:30} b = [i*a[i] for i in a] підправити під умову	14. Дано словник: прізвище та оцінка. Створити список студентів, що здали предмет (оцінка >=50)
Перевірка умови if '0'<=i<='9'	15. Створити список тільки з цифр, які містяться в текстовому полі: Наприклад текст 'grt743gfcg0r3' - Відповідь [7, 4, 3, 0, 3]. Розв'язати одним рядком

<pre>from random import randrange n = 10 a = [randrange(1, 10) for i in range(n)] підправити під умову</pre>	16. Задати 10 випадкових чисел від 0 до 9.
<pre>a=[[0] * 5 for i in range(4)] for i in a: print (i) або a=[[0 for j in range(m)] for i in range(n)]</pre>	17. Створити двовимірний масив 4 на 5 , який заповнений нулями
<pre>[[i * j for j in range(5)] for i in range(4)]</pre>	18. Задати, щоб в двовимірному масиві елементи будувались наступним чином: $A[i][j]=i*j$
	19. Утворити наступні двовимірні списки <pre>0 0 0 0 0 0 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4</pre> <pre>0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5 0 1 2 3 4 5</pre> <pre>[0, 1, 2, 3, 4] [1, 2, 3, 4, 5] [2, 3, 4, 5, 6] [3, 4, 5, 6, 7]</pre>
<pre>for i in range(4): for j in range(5): if (i+j)%2==0: a[i][j]=1 else: a[i][j]=0</pre>	20. Утворити <pre>[1, 0, 1, 0, 1] [0, 1, 0, 1, 0] [1, 0, 1, 0, 1] [0, 1, 0, 1, 0]</pre>
<pre>s=0 for i in range(4): for j in range(5): a[i][j]=s s=s+1</pre>	<pre>[0, 1, 2, 3, 4] [5, 6, 7, 8, 9] [10, 11, 12, 13, 14] [15, 16, 17, 18, 19]</pre>
	<pre>[0, 4, 8, 12, 16] [1, 5, 9, 13, 17] [2, 6, 10, 14, 18] [3, 7, 11, 15, 19]</pre>

ЛАБОРАТОРНА РОБОТА №4

Тип колекції	Изменяемость	Индексированность	Уникальность	Как создаём
Список (list)	+	+	-	<code>[]</code> <code>list()</code>
Кортеж (tuple)	-	+	-	<code>()</code> , <code>tuple()</code>
Строка (string)	-	+	-	<code>" "</code> <code>" "</code>
Множество (set)	+	-	+	<code>{elm1, elm2}</code> <code>set()</code>
Неизменяемое множество (frozenset)	-	-	+	<code>frozenset()</code>
Словарь (dict)	+ элементы - ключи + значения	-	+ элементы + ключи - значения	<code>{}</code> <code>{key: value,}</code> <code>dict()</code>

МНОЖИНИ

Перетворити в множини: set()		1. Дано два списки. Знайти їх - Об'єднання - перетин – ті елементи які є в обох списках, і зберегти їх до кортежу. Наприклад $X = [1, 4, 3, 5]$ і $Y = [-2, 1, 3]$. Тоді $X \cap Y = (1, 3)$ 2. Дано два списки. Наяти їх симетричну різницю – ті елементи з першого списку яких нема у другому та ті елементи з другого списку яких нема у першому, і зберегти їх до кортежу. Наприклад $X = [1, 4, -3, 5]$ і $Y = [-2, 1, 3]$. Тоді $X \Delta Y = (4, -2, 3, -3, 5)$
$A \cup B$ <code>A.union(B)</code>	Об'єднання множин A і B.	
$A \cup= B$ <code>A.update(B)</code>	Додає в множину A всі елементи з множини B.	
$A \cap B$ <code>A.intersection(B)</code>	Перетин множин A і B.	
$A \cap= B$ <code>A.intersection_update(B)</code>	Залишає в множині A тільки ті елементи, які є в множині B.	
$A - B$ <code>A.difference(B)</code>	Різниця множин A і B (елементи, що входять в A, але не входять в B).	
$A -= B$ <code>A.difference_update(B)</code>	Знищує з множини A всі елементи, що входять в B.	
$A \Delta B$ <code>A.symmetric_difference(B)</code>	Симетрична різниця множин A і B (елементи входять в A або в B, але не входять до них обох одночасно).	
$A \Delta= B$ <code>A.symmetric_difference_update(B)</code>	Записує в A симетричну різницю множин A і B.	

ТЕКСТ

Кількість розрядів отримати перетворенням числа в іншу структуру	1. Дано число . Визначити суму цифр
<code>split()</code> : в якості розділювача використовується пропуск <code>split(delimiter)</code> : в якості розділювача використовується <code>delimiter</code> <code>splitted_text = text.split(",")</code> метод <code>join()</code> : об'єднання списку рядків. <code>words = ["Let", "me", "speak", "from", "my", "heart", "in", "English"]</code> # розділювач - пропуск <code>sentence = " ".join(words)</code> # розділювач - вертикальна лінія <code>sentence = " ".join(words)</code>	2. Дано текст. Вивести тільки перше слово 3. Порахувати кількість слів 4. Дано текст. Задати список, де елементи - кожне слово окремо з тексту 5. Забрати всі пропуски з тексту

КОРТЕЖ

Кортеж з одного елементу: <code>a = ('s',)</code> Змінювати елементу кортежу не можна	6. Дано текст 'hello, world!'. Перетворити в кортеж. Вивести.
	7. Створити кортеж з 10 випадкових чисел 8. Об'єднати обидва кортежі в новий

	кортеж 9. Порахувати кількість елементів останнього кортежу
--	--

ФУНКЦІЇ

def add(x, y): return x + y	1. Створити функцію додавання двох елементів
print(add(2,3)) print(add('rgdrf','ggdf'))	
	2. Задано два числа – сторони прямокутника. Задати функцію для обчислення периметру прямокутника
	3. Функцію для обчислення площі
	4. Функцію для обчислення довжини діагоналі
	5. Створити функцію для обчислення факторіала числа
	6. Створити функцію для знаходження для списку середнього значення його елементів
	7. Створити функцію, яка для числа виводить його цифри

ЛАБОРАТОРНА РОБОТА №5 КЛАСИ

Конструктор класу

`__init__()` - вона викликається при створенні об'єкта класу
При виклику класу в круглих дужках передаються значення, які будуть присвоєні параметрам методу `__init__()`. Перший його параметр – `self` – посилання на сам щойно створений об'єкт.

class Complex: def __init__(self, realpart, imagpart): self.r = realpart self.i = imagpart y = Complex(3.0, -4.5) print(y.r, y.i)	class Person: def __init__(self, n, s): self.name = n self.surname = s p1 = Person("Sam", "Baker") print(p1.name, p1.surname)
1. Творити клас Коло, конструктор класу, що визначає радіус кола і два методи: - Обчислення площі - Обчислення довжини кола	
Якщо ми хочемо для функції <code>print()</code> задати виведення іншої корисної інформації, то в клас додають спеціальний метод <code>__str__()</code> . Даний метод повинен містити рядок, який буде виводити функція <code>print()</code> :	class A: def __init__(self, v1, v2): self.field1 = v1 self.field2 = v2 def __str__(self): return str(self.field1) + " " + str(self.field2) a = A(3, 4) print(a)
class Rectangle: def __init__(self, width, height, sign): self.w = int(width) self.h = int(height) self.s = str(sign) def __str__(self): rect = [] for i in range(self.h): # кількість рядків	

<pre> rect.append(self.s * self.w) # rect = '\n'.join(rect) # список в рядок return rect b = Rectangle(10, 3, '*') print(b) </pre>	
Перезавантаження операторів <code>__add__(self, other)</code> - додавання. $x + y$ викликається <code>x.__add__(y)</code>. <code>__sub__(self, other)</code> - віднімання ($x - y$). <code>__mul__(self, other)</code> - множення ($x * y$).	
<pre> class Tochka: def __init__(self, x, y): # конструктор класу self.x = x self.y = y def __str__(self): # перетворення об'єкту на рядок return '(' + str(self.x) + ', ' + str(self.y) + ')' def __eq__(self, other): # перевантажений "==" if self.x == other.x and self.y == other.y: return True else: return False def __add__(self, other): # перевантажений "+" return Tochka(self.x + other.x, self.y + other.y) a = Tochka(1,2) b = Tochka(3,1) c = a + b print (a == b) print (c) </pre>	2. Створити клас Vector, який мав би: • Конструктор, що задає дві координати, які визначають вектор на площині • метод <code>__str__</code> для перетворення на рядок для використання у функції <code>print</code> • метод <code>__eq__</code> для перевантаження «рівності» • метод <code>__add__</code> для перевантаження «суми» • метод для знаходження довжини вектора • метод для знаходження скалярного добутку векторів • метод для перевірки чи вектори перпендикулярні (скалярний добуток дорівнює нулю)

1) Створити class Студент (Student),:

- конструктор, що задає прізвище та ім'я, а також словник, де у якості ключа буде назва предмету, значеннями буде список оцінок за цей предмет
- метод `__str__` для перетворення на рядок для використання у функції `print()`
- метод, що виводить, які предмети вчить студент
- метод, який би повертав список оцінок за вказаний предмет
- метод для знаходження середнього балу за певний предмет
- метод для обчислення середнього балу за всі предмети
- метод для виведення відповіді чи отримав студент залік за певний предмет (якщо середній бал більший 4)
- метод, який би додавав новий предмет з оцінками
- метод, який би додавав до заданого предмету ще одну оцінку
- метод створення списку з середніх балів з предметів. Вивести прізвище, потім назва предмету і його середній бал, і наступний предмет і сер. бал і т.д.

Завдання для самостійного виконання

1. Створити генератор-вираз створення послідовностей:

- a. Чисел, що ділять на 3 на проміжку [1:100]
- b. $1, \frac{1}{2}, \frac{1}{3}, \dots, \frac{1}{10}$
- c. Квадрати чисел з проміжку [1:100], які без остачі ділять на 5
- d. квадрати непарних елементів із заданого списку
- e. із заданого списку вибрати елементи які більші 10 але менші 50
- f. подвоїти всі букви у тексті
- g. 15 випадкових чисел

2. Створити функції:

- a. Для заданого числа-місяця з клавіатури (від 1 до 12) вивести пору року
- b. Для заданого з клавіатури числа визначити чи рік з даним числом є високосним
- c. Для сторони квадрата обчислити за допомогою функції 3 значення (з допомогою кортежу) - периметр квадрата, площа квадрата і діагональ квадрата.
- d. Щасливим називають таке шестизначне число, у якого сума перших трьох дорівнює сумі останніх трьох. Знайти всі щасливі числа. Описати функцію для визначення суми цифр тризначного числа.
- e. Створити функції для знаходження кількості додатніх елементів списку
- f. Задати функцію яка визначає чи число є простим
- g. Задати функцію, що здійснює циклічний зсув елементів списку на одну позицію вліво

3. Розв'язати задачі:

- a. Дано натуральне чотирицифрове число n. З'ясувати, чи є воно паліндромом (таким, що читається однаково зліва направо та справа наліво, наприклад 1221).
- b. 12) Написати програму, яка серед двоцифрових чисел вибирає числа, рівні сумі своїх цифр.
- c. Написати програму виведення всіх чисел від 1 до N, які закінчуються цифрою 3.
- d. Знайти мінімальне та максимальне значення у списку довільної довжини та обміняти їх місцями. У результаті має бути кортеж, який містить список з обмінаними місцями максимальним та мінімальним елементами
- e. Дано список який містить набір результати деяких вимірювань температури (як додатні, так і від'ємні значення). Знайти значення температури та її індекс, яка була ближча за все до 0. У результаті вивести кортеж, який містить значення температури найближчої до 0 та її індекс у списку.

3. Дано словник, що складається з прізвищ студентів та масивів з двох оцінок за контрольні. Створити меню для вибору наступних пунктів:

- a. «хочу отримати залік», в якому перепитують Ваше прізвище і за ним повідомляють чи ви маєте автоматом залік (сума за контрольні більше 50) чи перездача
- b. «хочу перездати контрольну», в якому перепитують прізвище та повідомляють яку контрольну треба перездати та додають ваше прізвище до нового списку тих, що прийде на перздачу
- c. «список студентів, що приходить на перездачу»: друк списку
- d. «перездачі»: список студентів, що мають заборгованість.
- e. Знайти максимальну та мінімальну оцінки
- f. Дано текст. Вивести слова, які зустрічаються в тексті тільки по одному разу.

4. Створити клас:

- a. Клас Працівник з полем **посада, зарплата, премія**. Задати методи:
 - Конструктор,
 - метод для виведення,
 - метод для нарахування зарплати за рік,
 - метод для підвищення зарплати на 10%,
 - для виведення суми заробітку за місяць(зарплата+премія),
 - метод зняття премії працівника
- b. Клас сесія з полями оцінок за **лаби, кр1, кр2 і тести**. Задати методи:
 - контруктор
 - Метод для виведення
 - метод для обчислення середнього значення оцінок
 - Метод для визначення чи студент отримав залік на основі заданих оцінок
 - метод для задання перездачі якщо хоча б одна оцінка <20
- c. Клас Собака з полями **кличка, порода, вага, колір**. Задати методи:
 - Конструктор
 - Метод для виведення
 - Метод для порівняння яка собака з двох заданих важча
 - Метод визначення чи породи у двох заданих собак однакові чи ні

5. Скласти програми

- a) обчислення кількості різних (значущих) цифр у десятковому записі натурального числа n;
- б) друку у порядку зростання всіх цифр, які не входять до десяткового запису натурального числа n.

6. Відомості про учня складаються з його унікального номера учнівського квитка, імені, прізвища та назви класу (рік навчання та літери), в якому він вчиться. Дано словник, який містить відомості про учнів школи. Скласти програми, які дозволяють а) визначити, чи є в школі учні з однаковим прізвищем; б) визначити, чи є учні з однаковим прізвищем в якихось паралельних класах; в) визначити, чи є учні з однаковим прізвищем у якомусь класі; г) відповісти на питання а)-в) по відношенню до учнів, у яких співпадають і ім'я, і прізвище; д) визначити, в яких класах нараховується більше. ніж 35 учнів;

7. Перевірити, чи правильно в заданому тексті розставлені круглі дужки (тобто, чи знаходиться справа від кожної відкриваючої дужки відповідна їй закриваюча дужка, а зліва від кожної закриваючої - відповідна їй відкриваюча). Відповідь - "так" або "ні"

8. Дана непорожня послідовність слів; між сусідніми словами - пропуск, за останнім символом - крапка. Надрукувати:
 1. всі голосні літери, які входять до кожного слова;
 2. всі голосні літери, які не входять до жодного слова;
 3. перші входження літер до рядка, зберігаючи початковий взаємний порядок;
 4. всі літери,що входять до рядка не менше двох раз;
 5. всі літери, що входять до рядка по одному разу

9. T13.1. Клас Person описано наступним чином:

```

class Person:                                #Клас Особа
    def __init__(self):
        self.name = None                    #прізвище
        self.byear = None                  #рік народження

    def input(self):                          #ввести особу
        self.name = input('Прізвище: ')
        self.byear = input('Рік народження: ')

    def print(self):                          #вивести особу
        print(self.name, self.byear, end = ' ')

```

Описати клас Знайомий на базі класу Person. У цьому класі повинно бути як мінімум одне додаткове поле «номер телефону» а також методи введення та виведення інформації про знайомого.

Використати цей клас для побудови телефонного довідника. Передбачити дії: створення довідника, додавання запису про знайомого, пошуку номера телефону за прізвищем та заміни номера телефону.

Телефонний довідник зберігати у файлі та читати з файлу.

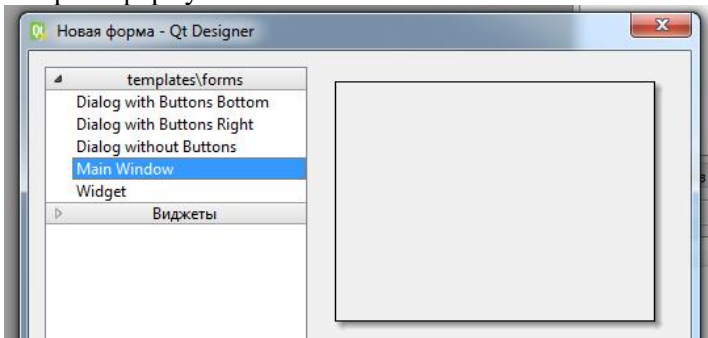
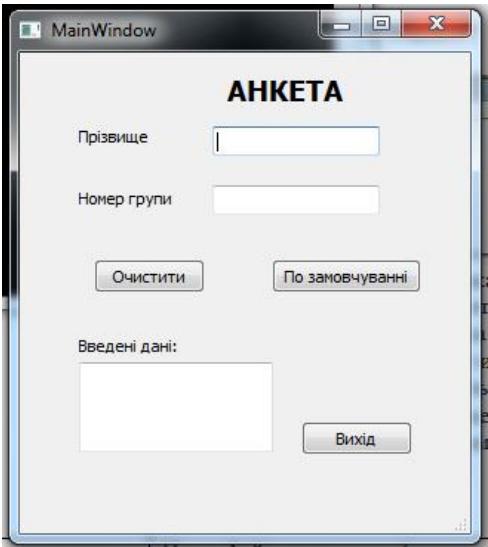
T13.2. Описати клас Співробітник на базі класу Person (див. завдання T13.1).

У цьому класі повинно бути як мінімум два додаткових поля: «табельний номер» та «заробітна платня», а також методи введення та виведення інформації про співробітника та розрахунку заробітної платні за кількістю відпрацьованих годин та загальною кількістю годин на місяць.

Використати цей клас для розрахунку заробітної платні за місяць, якщо співробітник працює на умовах погодинної оплати. Для розрахунку вводити відпрацьовані години по днях місяця (табель). Вважається, що відома кількість годин, яку повинен відпрацювати співробітник кожного дня, для отримання 100% платні.

Передбачити дії: введення табелю для співробітника, введення табелів всіх співробітників, розрахунку заробітної платні всіх співробітників. Табель зберігати у файлі та читати з файлу.

ЛАБОРАТОРНА РОБОТА №1 PYTHON. PYQT. DESIGNER

Качаємо останню версію Python з офіційного сайту https://www.python.org/downloads/windows/0 На першому екрані установки обов'язково поставте галочку "Додати в PATH"	
Тепер нам потрібно поставити PyQT - бібліотеку для створення на Python віконних додатків. Спершу в командному рядку Windows даємо команду pip install pyqt5 .	
Программа Designer Після цього качаємо і встановлюємо .exe інсталяційний файл звід https://sourceforge.net/projects/pyqt/files/	
ПРОЦЕС СТВОРЕННЯ ПРОГРАМИ В PYQT СКЛАДАЄТЬСЯ З ТРЬОХ ЕТАПІВ:	
• • малюється інтерфейс в програмі Designer, зберігається як .ui файл	Ми побачимо перед собою середовище візуальної розробки інтерфейсу. Вибрати форму Main window  Посередині знаходиться заготовка вікна нашої майбутньої програми. На неї потрібно перетягувати мишкою елементи інтерфейсу зі списку зліва (кнопки, поля введення і т.д.) Після того, як елемент інтерфейсу розташований в потрібному місці, можна виділити його клацанням миші і налаштувати його властивості в списку праворуч. Зберегти створений інтерфейс anketa.ui (наприклад D:\test\ anketa.ui) Створіть наступну анкету 
• .ui файл конвертується в файл .py за допомогою утиліти pyuic	Інтерфейс програми (знаходиться в D: \ test \) потрібно конвертувати в файл Python з розширенням .py. Для цього в командному рядку Windows даємо команду D: \ Python36 \ Scripts \ pyuic5.exe D: \ test \ anketa.ui -o D: \ test \ anketa.py Зверніть увагу що в цій команді у вас буде інший шлях до папки Python, так як у мене Python встановлено в папку Python36 на диску D, а у вас це швидше за все не так. Також не забудьте про опцію -o яка є в цьому рядку. Якщо в папку Python36 \ Scripts \ шлях прописаний в PATH, тоді заходите в папку з інтерфейсом програми і задаєте команду pyuic5.exe anketa.ui -o anketa.py

• •сконвертований файл підключається до основного модуля програми, пишеться код програми	Використовуємо стандартну заготовку (скелет) основної програми (main.py зберегти в тій же папці де файл anketa.py): <pre> from PyQt5 import QtCore, QtGui, QtWidgets import sys # Импортируем наш модуль интерфейса anketa.py from anketa import * class MyWin(QtWidgets.QMainWindow): def __init__(self, parent=None): QtWidgets.QWidget.__init__(self, parent) self.ui = Ui_MainWindow() self.ui.setupUi(self) # Тут прописуємо події натискання на кнопку self.ui.pushButton.clicked.connect(self.buttonpress) # функція яка виконується # при натисканні на кнопку def buttonpress(self): # тут розміщуються команди для виконання pass if __name__ == "__main__": app = QtWidgets.QApplication(sys.argv) myapp = MyWin() myapp.show() sys.exit(app.exec_()) </pre> звернути увагу на ім'я свого модуля (розробленого інтерфейсу) і підключити саме його
Запустити main.py Підправити розроблений макет у файлі main.py так, щоб всі кнопки працювали	
Кнопка очистити	Дані з двох текстових рядків занести у текстове поле під «Введені дані» та очищуються текстові рядки a=self.ui.lineEdit.text() взяти значення з текстового рядка c=self.ui.textEdit.toPlainText() взяти значення з текстового поля self.ui.textEdit.setText() задати значення
Кнопка по замовчуванні	В двох текстових рядках записується Іванов Іван
Кнопка виходу	<pre> from PyQt5.QtCore import QApplication self.ui.pushButton_3.clicked.connect(QCoreApplication.instance().quit) </pre>

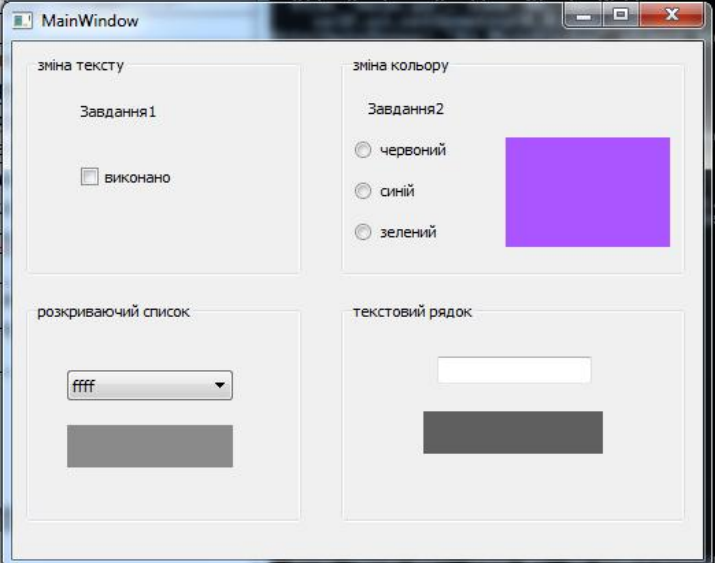
Домашнє завдання

Створити заготовку інтерфейсу кредитного калькулятора, де вводиться сума кредиту, термін кредитування, та процентна ставка банку. Вивести щомісячну оплату та загальну суму, яку потрібно заплатити за весь період

ЛАБОРАТОРНА РОБОТА №2

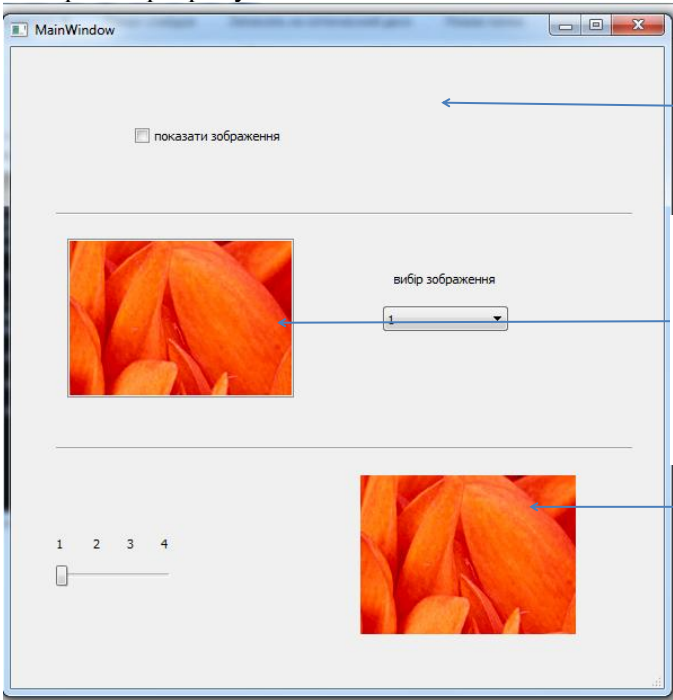
Робота з віджетами

Частина 1

Створити наступний інтерфейс в Designer Спочатку поставити GroupBox и тільки тоді на ньому об'єкти	
Перше завдання CheckBox	
Задати запуск функції при заданні чи знятті галочки	<pre>self.ui.checkBox.stateChanged.connect(.....) stateChanged.connect - зміна статусу галочки</pre>
Задати функцію, яка здійснюється заміною надпису мітки з текстом «Завдання1» на текст «виконано»	<pre>if self.ui.checkBox.isChecked() == True: setText('виконано')</pre>
Задати при знятті галочки, щоб знов був текст «Завдання1»	
Друге завдання RadioButton	
Задати зміна кольору фону фрейму в залежності від вибору радіокнопки	Задати колір фону через функцію lambda: колір – це значення функції <pre>self.ui.radioButton.toggled.connect(lambda:self.col('red'))</pre>
Функція змінює фон фрейму на червоний	Задати функцію з параметром <pre>def col(self,a):</pre> функція змінюватиме колір фону фрейму згідно введеного значення: <pre>self.ui.frame.setStyleSheet("background-color: %s;" %a)</pre>
Задати зміну фрейму на синій та зелений колір для інших радіокнопки	
Третє завдання Розкриваючий список	
Задати значення для розкриваючого списку і функцію запуску	<pre>self.ui.comboBox.addItem(['перший курс', 'другий курс', 'третій курс'])</pre> функція при клацанні <pre>self.ui.comboBox.currentIndexChanged.connect(.....)</pre>
Функція повинна брати значення з розкриваючого списку і ставити у мітку-надпис сірого кольору	<pre>a=self.ui.comboBox.currentText() setText() поставити текст у мітку-надпис</pre> <pre>self.ui.comboBox.setCurrentIndex(1) –поставити перше</pre>

	значення при запуску для розкриваючого списку
Четверте завдання Текстовий рядок	
Задати текстовий рядок і надпис-label з сірим фоном. Для текстового поля задати запуск функції Zmin при будь-якому введенні тексту	<code>self.ui.lineEdit.textChanged[str].connect(self.Zmin)</code>
Задати функцію Zmin, яка бере текст з текстового поля і ставить текст у label В результаті при введенні тексту він автоматично появлятиметься у сірій мітці-label	Взяти з текстового поля – попередня лабораторна

Частина 2. Графічні зображення

Створити програму	
	 Порожній квадратний label graphicsView з фоновим зображенням setStyleSheet- edit Resource background-image- new resource—add image label з фоновим зображенням setStyleSheet
При потребі у документі-заготовці інтерфейсу виверти команду-рядок: <code>import l_rc</code> проглянути шлях до файлів	
Перше завдання	
Задати для checkbox показ зображення в першому порожньому label, а сам checkbox стає невидимим	<code>self.ui.checkBox.stateChanged.connect(self.____)</code> <code>setVisible(False)</code>
	Перший порожній label заповнити малюнком за допомогою класу QPixmap : прописати підключення from PyQt5.QtGui import QIcon, QPixmap вставити зображення в мітку-надпис <code>pixmap = QPixmap('_____.jpg')</code> <code>self.ui.label.setPixmap(pixmap)</code>
Здійснити масштабування зображення	<code>self.ui.label.setScaledContents(True)</code>

Друге завдання	
Задати значення розкриваючого списку 1,2,3,4	
В кореневу папку скопіювати ще три зображення і перейменувати в 2.jpg і т.д.	Зображення повинні бути маленькі
При виборі значення зі списку - зображення graphicsView змінюється у відповідності до назви	<code>a=self.ui.comboBox.currentText()+'.jpg'</code> <code>self.ui.graphicsView.setStyleSheet("background-image: url(%s);" %a)</code>
Третє завдання	
При русі повзунка змінюється зображення нижнього label	Запуск функції при русі повзунка <code>self.ui.horizontalSlider.valueChanged[int].connect(_____)</code>
Приклад функції через setPixmap. (Можна задати через <code>setStyleSheet("background-image:.....")</code>)	Наприклад: <pre>def f(self,value): if value >= 0 and value <= 25: self.ui.label.setPixmap(QPixmap('1.png')) elif value > 25 and value <= 50: self.ui.label.setPixmap(QPixmap('_____')) elif</pre>

Домашнє завдання

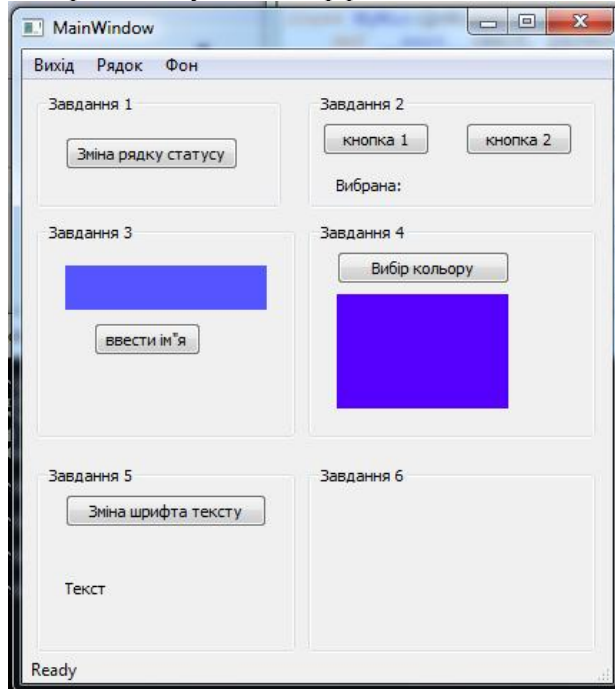
Створити валютний калькулятор довільного дизайну з розкриваючим списком, де можна вибрати валюту у яку треба перевести суму, яку введуть в текстовий рядок. Можна додати рядки для введення курсу валют і т.д. Вставити зображення, що показує символ-картинку вибраної валюти

Приклади:

ЛАБОРАТОРНА РОБОТА №3

Діалогові вікна

Створити наступний інтерфейс:



Завдання 1.

При натисненні кнопки запускається функція, що змінює значення статусного рядка

```
self.statusBar().showMessage('Ready')
```

Завдання 2 Обидві кнопки запускають одну функцію, якій бере текст кнопки і у label записується яка кнопка була натиснута

```
sender = self.sender()
self.ui.label.setText(sender.text() + ' was pressed')
```

Завдання 3

При натисненні кнопки появляється діалогове вікно, де записується ім'я і в label записується «Привіт **введене ім'я**»

Перевірте чи імпортовані у Вас діалогові вікна для введення тексту

```
import sys
from PyQt5.QtWidgets import (QInputDialog,
                              QApplication)
```

```
def showDialog(self):
    text,ok = QInputDialog.getText(self, 'Input
    Dialog', 'Enter your name:')
    if ok:
        self.ui.label_2.setText('Привіт '+str(text))
```

Завдання 4

Зміна фоновому кольору фрейму через діалогове вікно

Не забути прописати

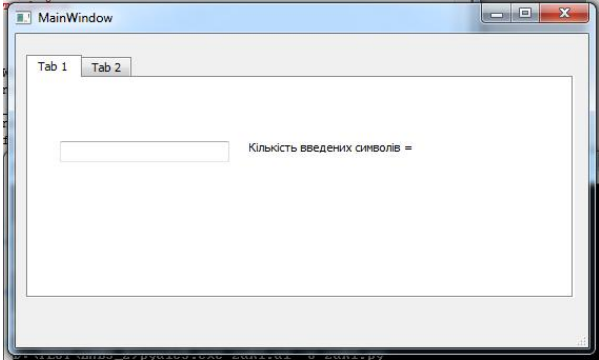
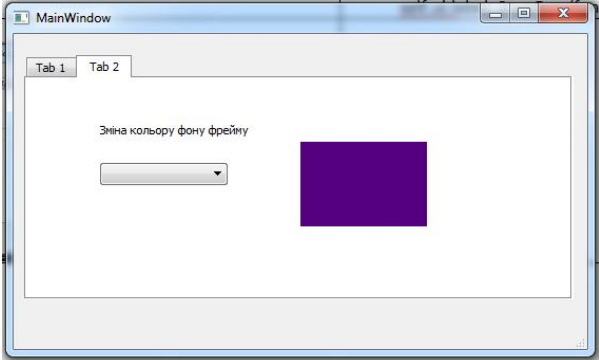
```
import sys
from PyQt5.QtWidgets import (QWidget,
                              QPushButton, QFrame, QColorDialog,
                              QApplication)
from PyQt5.QtGui import QColor

def showDialog_2(self):
    col = QColorDialog.getColor()
    if col.isValid():
        self.ui.frame.setStyleSheet("QWidget
        { background-color: %s }" % col.name())
```

Завдання 5

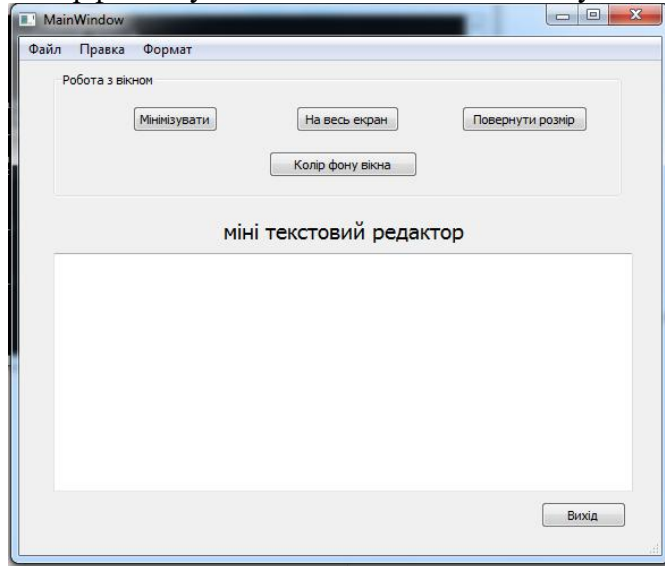
Зміна шрифту тексту через діалогове вікно

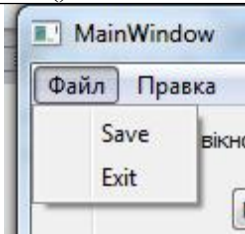
```
import sys
from PyQt5.QtWidgets import (QWidget,
                              QVBoxLayout, QPushButton, QSizePolicy,
                              QLabel, QFontDialog, QApplication)
```

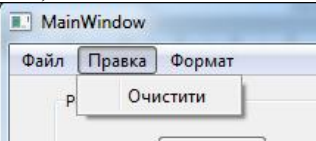
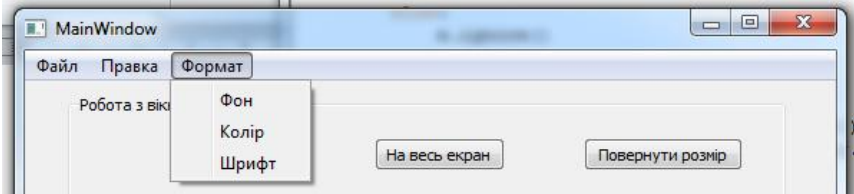
	<pre>def showDialog_3(self): font, ok = QFontDialog.getFont() if ok: self.ui.label_3.setFont(font)</pre>
Робота з закладками Створити наступний інтерфейс	
	
Задати для першої закладки: при веденні тексту у текстове поле підраховується і відображається кількість введених символів	
Зробити елементи другої закладки недоступними для роботи	<code>self.ui.tab_2.setEnabled(False)</code>
При введенні тексту «зроблено» у текстове поле повинен з'явитися надпис «Друга закладка доступна» і зробити її доступною (True)	
На другій закладці для розкриваючого списку задати значення red, blue і т.д.	
В виборі кольору з розкриваючого списку колір фону фрейму змінюється у відповідності до вибраного кольору	

ЛАБОРАТОРНА РОБОТА №4 СТВОРЕННЯ МЕНЮ

Створити наступний інтерфейс: пункти меню мають свої підпункти (далі в лабораторній)



Робота з вікном: Задати три кнопки і три функції, які мінятимуть розмір вікна	<pre>def on_min(self): self.showMinimized() def on_full(self): self.showFullScreen() def on_norm(self): self.showNormal()</pre>
Впливаючі підказки для кожної кнопки	<code>self.ui.pushButton_2.setToolTip('Мінімізація вікна')</code>
Підказка для самого вікна	<code>self.setToolTip('Робоче поле програми')</code>
Задати кнопку «Колір фону», що змінює фон всього вікна через діалогове вікно, де вибирають колір	<code>self.setStyleSheet...</code>
Кнопку «вихід» задати	<code>self.ui.pushButton.clicked.connect(self.close)</code>
Перепитування при виході з програми при натисненні «хрестика»	<pre>def closeEvent(self, e): result = QtWidgets.QMessageBox.question(self, "Confirm Dialog", "Really quit?", QtWidgets.QMessageBox.Yes QtWidgets.QMessageBox.No, QtWidgets.QMessageBox.No) if result == QtWidgets.QMessageBox.Yes: e.accept() else: e.ignore()</pre>
 Пункт меню Файл	
Пункт підменю вихід (у мене <code>action_4</code>)	<code>self.ui.action_4.triggered.connect(self.close)</code>
Пункт підменю зберегти для збереження тексту з текстового поля	<pre>def saveToFile(self): self.writeFile = open("text.txt", 'w', encoding='utf-8') self.writeFile.write(self.ui.textEdit.toPlainText()) self.writeFile.close()</pre> задати функцію для пункту підменю <code>self.ui.action_2.triggered.connect(self.saveToFile)</code>

	перевірити чи створиться текстовий файлу папці, де є програмний продукт, який розробляється
Підправити попередній підпункт меню, щоб появлялось діалогове вікно	QFileDialog прописати імпортування <pre>def saveToFile(self): options = QtWidgets.QFileDialog.Options() self.fileName, _ = QtWidgets.QFileDialog.getSaveFileName(self, "Save To File", "", "Text Files (*.txt)", options=options) if self.fileName: self.writeFile = open(self.fileName, 'w',encoding='utf-8') self.writeFile.write(self.ui.textEdit.toPlainText()) self.writeFile.close() self.ui.statusbar.showMessage('Saved to %s' % self.fileName)</pre>
Пункт меню Правка 	
Оформити підпункт Очистити, що текстове поле робить порожнім	
Пункт меню Формат 	
Підпункт Фон змінює колір фону всього вікна	
Пункт Колір змінює колір тексту через діалогове вікно вибору кольору текстового поля	
Пункт Шрифт змінює розмір шрифту текстового поля через діалогове вікно	

Домашнє завдання

Створити програмку з двома текстовими полями і надпис. Меню має містити два пункти: Дії та Правка. Пункт меню дії має мати наступні підпункти: Додати, Відняти, Помножити. Кожен з пунктів згідно зі своєї назви додає, віднімає, множить числа з текстовим полів і виводить результат.

Пункт Правка містить: Очистити поля, Очистити надпис.

Не забудьте, що в текстові поля замість чисел можуть бути введені букви. Задати показ надпису «Не можливо» при спробі виконання операцій над текстом (Try... except...)

ЛАБОРАТОРНА РОБОТА №5

Створення калькулятора



Створити калькулятор з **lcdnumber**.

Команда показати елемент: `self.ui.lcdNumber.display(c)`

Команда взяти значення: `self.ui.lcdNumber.value()`

Збільшувати розрядність чисел при введення нової цифри **a**:

def `f(self,a):`

`c=self.ui.lcdNumber.value()*10+a`

ЛАБОРАТОРНА РОБОТА №6

Розробка програмного продукту для тестового контролю знань

Задати наступні елементи:

Розкриваючий список `spinBox` з мінімумом 1 і максимумом 5

Поставити `ToolBox`

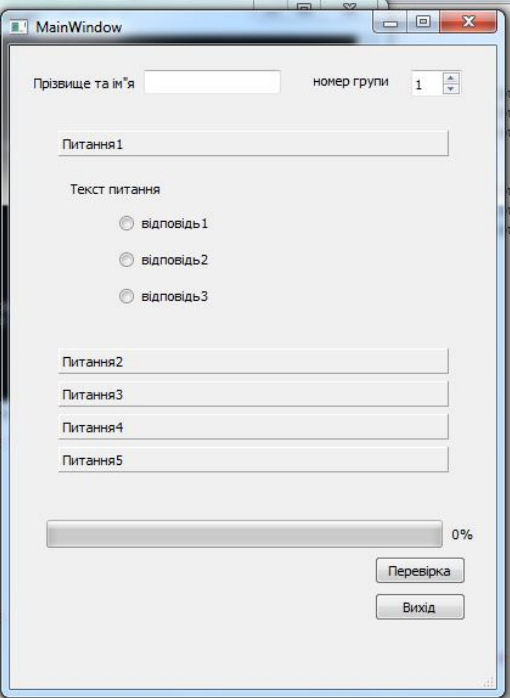
Додати ще сторінки: права кнопка «вставити сторінку»

Перейменувати: `currentItemText`

На кожній сторінці `ToolBox` поставити питання тесту і радіо кнопки для вибору відповіді

Вставити `progressBar`, поставити значення по замовчуванні `=0`

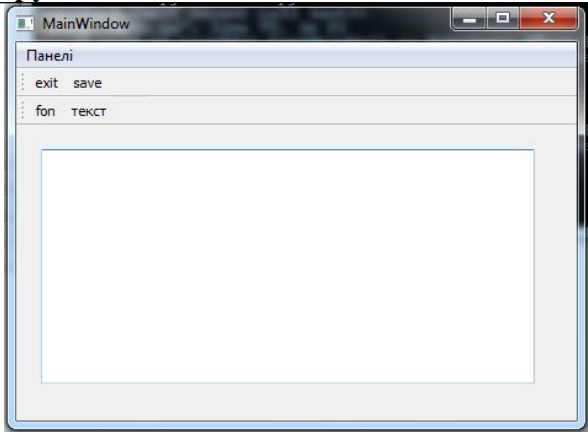
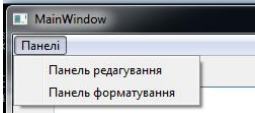
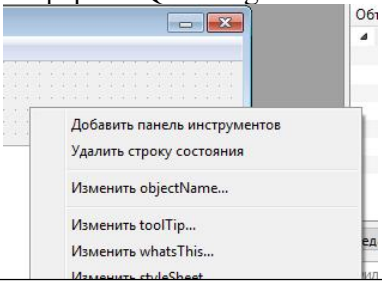
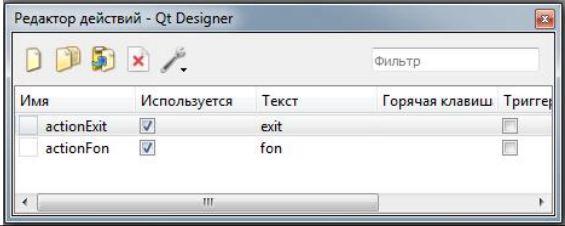
Внизу програми (зліва від кнопок) поставити порожній `Label`, куди буде розміщуватись оцінка за тест

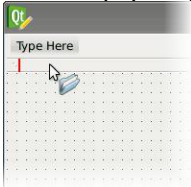
	
Для кнопки «Перевірка» задати функцію, яка демонструє як пробігає зелена смужка	<pre>def download(self): self.completed = 0 while self.completed < 100: self.completed += 0.0001 self.ui.progressBar.setValue(self.completed)</pre>
Розрахунок поділу наповненості смужки	$self.b=100/5$ (100% наповненості поділити на 5 питань)
Розробка першого питання: При виборі варіанту з радіо кнопки смуга наповненості збільшується (показує процент пройденого тесту) і вибір відповіді для даного питання стає недоступним	Функція запускається коли здійснити вибір відповіді <code>self.ui.radioButton_2.toggled.connect(self.buttonpress)</code> прогрес бар змінюється <code>self.completed += self.b</code> <code>self.ui.progressBar.setValue(self.completed)</code> всі радіо кнопки для цього питання повинні вже стати недоступними <code>self.ui.radioButton_2.setEnabled(False)</code>
Дл обчислення правильних відповідей задати змінну з початковим значенням 0	<code>self.k=0</code>
Підправити функцію при натисненні радіо кнопок, яка тепер передаватиме 1, якщо відповідь правильна і 0 якщо не правильна	<code>self.ui.radioButton_2.toggled.connect(lambda:self.buttonpress(1))</code> <code>self.ui.radioButton_3.toggled.connect(lambda:self.buttonpress(0))</code> def buttonpress(self,a) self.k+=a ...
Так ми підраховуватимемо кількість правильних відповідей	
Записати питання з варіантами відповідей на кожній сторінці ToolBox	
Закладки, на яких була вибрана відповідь перейменувати, щоб позначити на які питання вже	Нумерація починається з нуля <code>self.ui.toolbox.setItemText(0,'ok')</code>

дано відповідь	
Задати кнопку Вихід з програми	
Підправити кнопку Перевірка: Вона повинна брати прізвище з тестового рядка, номер групи і виводити все разом з кількістю правильних відповідей у label внизу програми «Ви отримали»	<code>self.ui.spinBox.value()</code> бере значення
	Модифікувати та вдосконалити тести

ЛАБОРАТОРНА РОБОТА №7

Панель інструментів

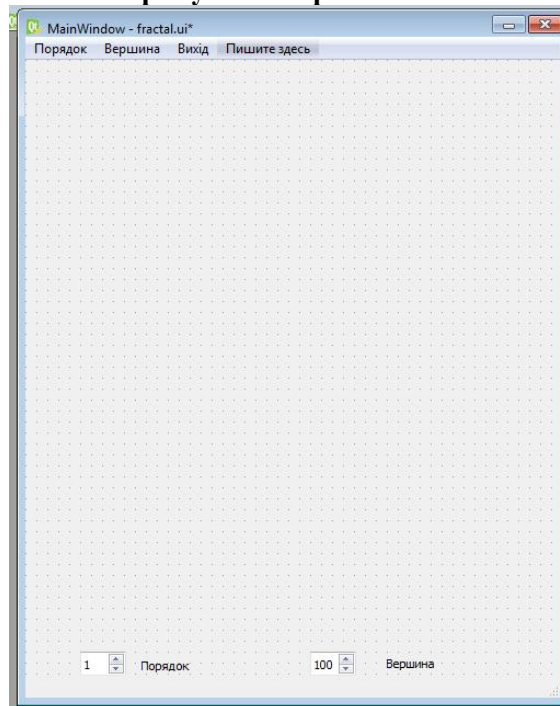
Створити наступне вікно	
Пункт меню «Панелі» повинен містити два підпункти: Панель редагування та панель форматування  Для кожного підпункту буде своя панель інструментів	
Додати дві панелі інструментів: Права кнопка на формі в QT Designer 	Одна з них буде для редагування Друга для форматування
Перейти на «редактор дій» і задайте дві дії 	Наприклад: вихід та зміна фону Кожній дії задається назва, що буде використовуватись в основній програмі (наприклад <code>actionExit</code> , <code>actionFon</code>)
З «редактора дій» одну дію перетягнути на одну панель інструментів, іншу – на другу	

панель форматування 	
Після запуску програми перевірити чи можна з допомогою мишки перемістити панель інструментів вниз вікна	
В основній програмі обидві панелі зробити невидимими	<code>self.ui.toolBar_2.setVisible(False)</code>
Для кожного підпункту меню задати дію включення своєї панелі (кожен підпункт робить свою панель інструментів видимою)	Задати змінні-індикатори для кожної панелі (по замовчуванню <code>self.k1=0</code> перша панель невидима) Задати функції для кожного підпункту меню, які включають та виключають видимість панелі в залежності від значення індикатора <pre> if self.k1==0: self.ui.toolBar.setVisible(True) self.k1=1 else: self.ui.toolBar.setVisible(False) self.k1=0 </pre>
Запрограмувати кнопки exit і фон панелей інструментів	В основній програмі кожній дії (назву задавали в «редакторе дій») задати запуск відповідної функції з діями: Вихід – здійснює вихід з програми (<code>self.ui.actionExit.triggered.connect(self.close)</code>) Фон – викликає діалогове вікно з вибором кольору фону
Перейти в . на панель редагування додати кнопку Зберегти. В основній програмі підключити дію збереження	Лаб4:імпортування у текстовий файл з діалоговим вікном
У «редакторе дій» додати до кнопок зображення, тобто щоб панель інструментів складалась із зображень	Знайти зображення, зберегти в кореневій папці У файлі із дизайном можливо потрібно буде витерти шлях до зображень
Запрограмувати кнопки курсив, підкреслений 	Курсив <code>self.ui.textEdit.setStyleSheet("font: italic;")</code> Підкреслений <code>self.ui.textEdit.setStyleSheet("text-decoration: underline;")</code>
колір тексту	Колір тексту задати через діалогове вікно і через атрибут color <code>self.ui.textEdit.setStyleSheet("QWidget { color: %s }" % col.name())</code>
Додати ще кнопки за бажанням	

ЛАБОРАТОРНА РОБОТА №8

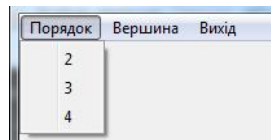
Побудова фракталів

Трикутник Серпінського



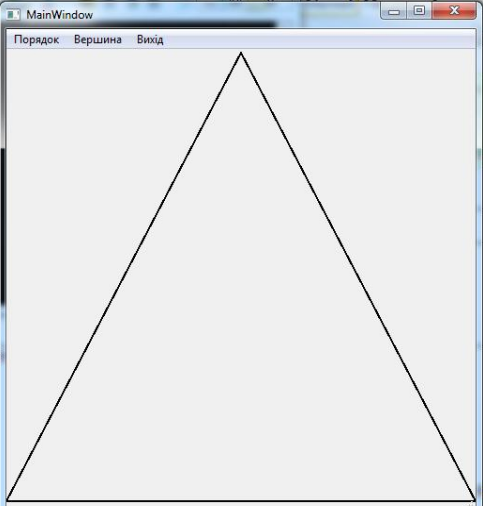
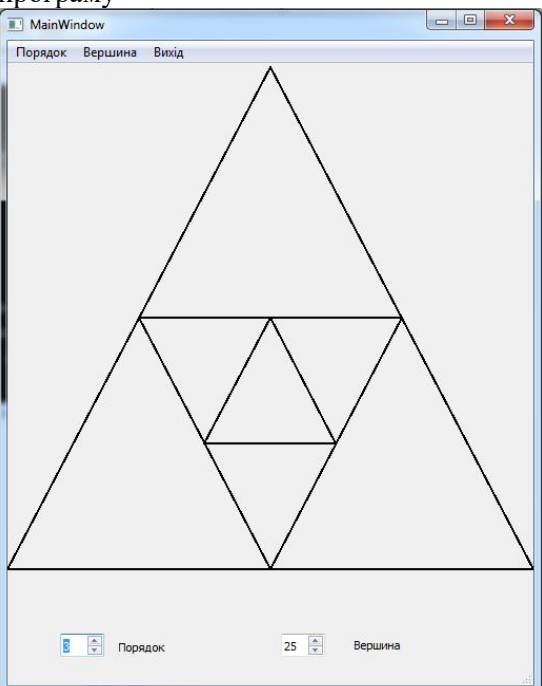
Перший spinBox min=1 max=9
spinBox_2 min=25 max=450

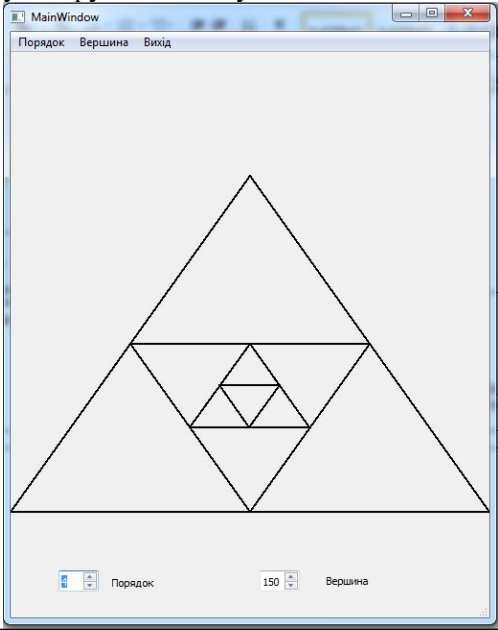
Меню:



Пункт Вершина із значеннями 100, 250

В дизайн-файлі загальний розмір вікна MainWindow.resize(500, 510)	Потрібні значення більші 500
Клас QPainter відповідає за все низкоуровневое рисование.	from PyQt5.QtGui import QPainter , QColor, QPen
Товщина і колір лінії Початкові значення і три побудови лінії	from PyQt5.QtCore import Qt
	функція, що задає подію малювання: def paintEvent(self, QPaintEvent): qp= QPainter(self) pen = QPen(Qt.black, 2, Qt.SolidLine) qp.setPen(pen) x1=0 y1=500 x2=500 y2=500 x3=250 y3=25

	<pre>qp.drawLine(x1, y1, x2, y2) qp.drawLine(x2, y2, x3, y3) qp.drawLine(x3, y3, x1, y1)</pre>
Задати цикл для задання порядку (кількість) замість останніх трьох команд малювання ліній	<pre>for i in range(self.n): qp.drawLine(x1, y1, x2, y2) qp.drawLine(x2, y2, x3, y3) qp.drawLine(x3, y3, x1, y1) x1n=(x1+x2)//2 y1n=(y1+y2)//2 x2n=(x3+x2)//2 y2n=(y3+y2)//2 x3n=(x3+x1)//2 y3n=(y3+y1)//2 x1=x1n y1=y1n x2=x2n y2=y2n y3=y3n x3=x3n</pre>
Порядок берем зі першого списку self.n spinBox	В основній програмі: self.n=self.ui.spinBox.value()
При зміні значень оновити self.n та оновити всю програму 	В основній програмі self.ui.spinBox.valueChanged.connect(self.up) і сама функція <pre>def up(self): self.n=self.ui.spinBox.value() self.update()</pre>

Підправити програму, яка бере значення вершини y_3 з другого списку 	Підправити $y_3 = \text{self.v}$ і задати в основній програмі $\text{self.v} = \dots$ при виборі значення задати оновлення даних (або нову програму або підправити $\text{up}()$)
Дооформити меню Порядок міняє значення першого списку $\text{self.ui.spinBox.setValue(2)}$ і оновити Вершина міняє значення другого списку	

ЛІТЕРАТУРА

1. Язык программирования Python / Г. Россум, Ф. Л. Дж. Дрейк, Д. С. Откидач та ін. 2001. – 454 с.
2. Бизли Д. Python. Подробный справочник / Д. Бизли. – СПб. : Символ-Плюс, 2010. – 864 с.
3. Хахаев И. А. Python. Практикум по алгоритмизации и программированию на Python / И. А. Хахаев. – М. : Альт Линукс, 2010. – 126 с.
4. Любанович Б. Python. Простой Python. Современный стиль программирования / Б. Любанович. – СПб. : Питер, 2016. – 480 с.
5. Федоров Д. Ю. Основы программирования на примере языка Python : учеб.пособие / Д. Ю. Федоров. – СПб. : Питер, 2016. – 176 с.
6. Лутц М. Изучаем Python, 4-е издание / М. Лутц. – СПб. : Символ-Плюс, 2011. – 1280 с.
7. Доусон М. Програмуємо на Python / М. Доусон. – СПб. : Питер, 2014. – 416 с.