

**ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО «ВИШИЙ НАВЧАЛЬНИЙ
ЗАКЛАД «МІЖРЕГІОНАЛЬНА АКАДЕМІЯ УПРАВЛІННЯ
ПЕРСОНАЛОМ»**

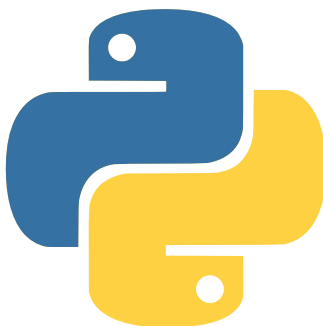
МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

щодо виконання лабораторних робіт

з навчальної дисципліни «Основи програмування»

для студентів усіх форм навчання

**спеціальностей 121 «Інженерія програмного
забезпечення», 122 «Комп'ютерні науки», 125
«Кібербезпека та захист інформації»**



Київ, 2024

Методичні рекомендації щодо виконання лабораторних робіт з навчальної дисципліни «Основи програмування» для студентів усіх форм навчання спеціальностей 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 125 «Кібербезпека та захист інформації».

Розробники:

ГОРДІЄНКО Олександр Миколайович, кандидат технічних наук, доцент кафедри комп'ютерних інформаційних систем та технологій Інститут комп'ютерно-інформаційних технологій та дизайну ПрАТ «ВНЗ «Міжрегіональна Академія управління персоналом»

КОВАЛЬ Аліна Олександрівна, викладач кафедри комп'ютерних інформаційних систем та технологій Інститут комп'ютерно-інформаційних технологій та дизайну ПрАТ «ВНЗ «Міжрегіональна Академія управління персоналом»

Розглянуто та ухвалено на засіданні кафедри комп'ютерних інформаційних систем та технологій Інституту комп'ютерно-інформаційних технологій та дизайну ПрАТ «ВНЗ «Міжрегіональна Академія управління персоналом»

Протокол №__ від «__» _____ 2024 р.

Завідувач кафедри _____ КАВУН Сергій Віталійович

Гордієнко О.М., Коваль А.О. Методичні рекомендації щодо виконання лабораторних робіт з навчальної дисципліни «Основи програмування» для студентів усіх форм навчання спеціальностей 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки», 125 «Кібербезпека та захист інформації» – К.: МАУП, 2024. – 47 с.

Міжрегіональна Академія управління персоналом (МАУП) 2024 ©

ЗМІСТ

Вступ	5
Лабораторна робота 1. Підготовка робочого середовища..	11
Завдання для виконання лабораторної роботи	11
1 Встановлення і перевірка роботи ПО	11
2 Надання результатів виконання домашньої роботи	14
Лабораторна робота 2. Перший проект.....	15
Завдання для виконання лабораторної роботи	15
2 Виконання роботи.....	16
3 Результат виконання лабораторної роботи.....	17
Лабораторна робота 3. Депозитний калькулятор	19
Завдання для виконання лабораторної роботи	19
1 Постановка задачі	19
2 Підготовка до виконання лабораторної роботи	20
3 Виконання роботи.....	21
4 Результат виконання лабораторної роботи.....	23

Лабораторна робота 4. Вибір депозитної програми	25
Завдання для виконання лабораторної роботи	25
1 Постановка задачі	25
2 Підготовка до виконання лабораторної роботи	27
3 Обрати варіант	28
4 Виконання роботи.....	29
5 Результат виконання лабораторної роботи.....	35
Список літератури	37
Додаток 1. Титульна сторінка. Вимоги до оформлення	39
Додаток 2. Варіанти виразів для виконання лабораторної роботи №1	44
Додаток 3. Приклад депозитного розрахунку з урахуванням щорічної капіталізації	46

ВСТУП

Методичні рекомендації щодо виконання лабораторних робіт з навчальної дисципліни «Основи програмування» призначені для студентів усіх форм навчання спеціальностей 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки» та 125 «Кібербезпека та захист інформації».

Дана навчальна дисципліна є базовою у формуванні компетентностей майбутніх фахівців у сфері розробки програмного забезпечення, інформаційних технологій, кібербезпеки та захисту інформації. Дисципліна базується на викладанні основ програмування мови Python, як сучасної, універсальної і простої у використанні мови програмування, яка забезпечує ефективне засвоєння основ алгоритмізації, структурного, об'єктно-орієнтованого та функціонального програмування.

Мета методичних рекомендацій – сприяти розвитку практичних навичок програмування, необхідних для вирішення задач різного рівня складності, а також

формуванню логічного мислення та навичок проектування алгоритмів.

У документі наведено постановки завдань для студентів першого курсу навчання, алгоритми виконання завдань, приклади розв'язків ключових елементів коду, а також рекомендації щодо оформлення звітів. Методичні рекомендації дозволяють студентам:

- поступово освоїти практичні основи мови Python
- розвинути розуміння принципів написання ефективного, структурованого та читаємого коду
- застосовувати отримані знання у практичній діяльності, зокрема в розв'язанні задач з використанням базових алгоритмів, обробки даних та автоматизації процесів

Рекомендації адаптовані для студентів із різним рівнем підготовки, що дозволяє враховувати індивідуальні особливості навчання та темпи засвоєння матеріалу.

Методичні рекомендації також спрямовані на розвиток самостійності у навчанні, що є важливим

компонентом професійного становлення майбутніх фахівців.

Особливу увагу приділено виконанню лабораторних робіт, які є ключовим елементом практичного засвоєння матеріалу. Лабораторні завдання спрямовані на опанування таких тем:

1. Базові поняття
 - Історична довідка
 - Перший запуск
 - Лексеми та ідентифікатори
 - Прості операції вводу/виводу
 - Синтаксис
 - Змінні
 - Типи даних. Типізація
 - Прості типи
 - Явні/неявні перетворення
 - Оператори
 - Логічні вирази та оператори
2. Рядковий тип да операції з ним
 - Керуючі послідовності
 - Операції з рядками

- Строкові методи та функції
 - Форматування строк
3. Складні структури даних
- Списки
 - Кортеж
 - Словники
 - Діапазони
 - Множини
4. Умовні оператори. Цикли. Функції
- Умовні оператори
 - Цикли `while`, `for`. Додаткові методики написання циклів
 - Функції та декоратори. Генератори. Лямда функції
5. Дата та час. Файли. Виняткові ситуації. Модулі та пакети
- Робота з датою та часом
 - Робота з файлами
 - Обробка виняткових ситуацій
 - Підключення сторонніх та написання власних пакетів

Для успішного використання методичних рекомендацій студенти повинні володіти базовими навичками роботи з комп'ютером та розумінням принципів функціонування програмного забезпечення. Крім того, важливою умовою є готовність до постійного самовдосконалення, пошуку нових знань та активного використання сучасних інформаційних технологій у навчанні.

Виконання лабораторних робіт потребує від студентів уважності та дисциплінованості, а також здатності застосовувати теоретичні знання на практиці. Для підвищення ефективності роботи пропонується дотримуватися наступних рекомендацій:

1. Детально ознайомтеся з поставленим завданням. Перед початком роботи важливо зрозуміти умови задачі та визначити алгоритм її виконання.

2. Ретельно плануйте час. Ефективне планування дозволить уникнути затримок у виконанні завдань.

3. Використовуйте додаткові джерела інформації. У разі труднощів із виконанням завдання корисним може бути

звернення до спеціалізованої літератури, онлайн-документації Python або консультації з викладачем.

4. Тестуйте свої рішення. Регулярна перевірка коду на помилки забезпечить його правильну роботу.

5. Дотримуйтесь вимог до оформлення. Правильне оформлення звітів є невід'ємною частиною навчального процесу.

ЛАБОРАТОРНА РОБОТА 1. ПІДГОТОВКА РОБОЧОГО СЕРЕДОВИЩА

Мета роботи: встановити на робочому ПК набір програмного забезпечення необхідний для виконання наступних лабораторних робіт. Перевірка готовності робочого середовища.

Завдання для виконання лабораторної роботи

1 Встановлення і перевірка роботи ПО

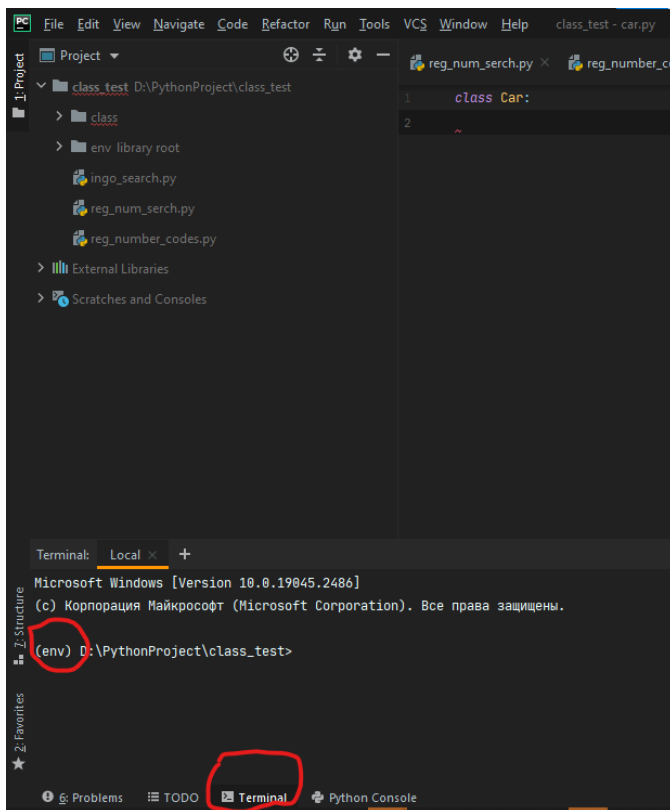
1.1 Завантажити Python за посиланням нижче та встановити його

<https://www.python.org/downloads/>

1.2 Завантажити PyCharm за посиланням нижче та встановити його

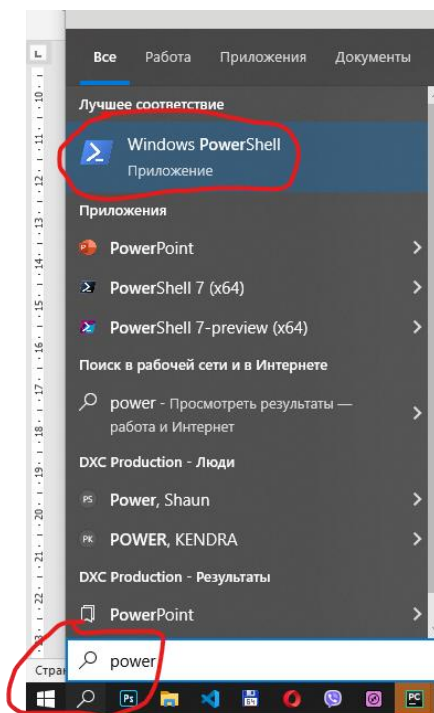
<https://www.jetbrains.com/ru-ru/pycharm/download/>

1.3 Запустити PyCharm, створити новий проект. Пересвідчитись, що у вікні термінала активоване віртуальне оточення (присутній надпис `env` або `venv`):



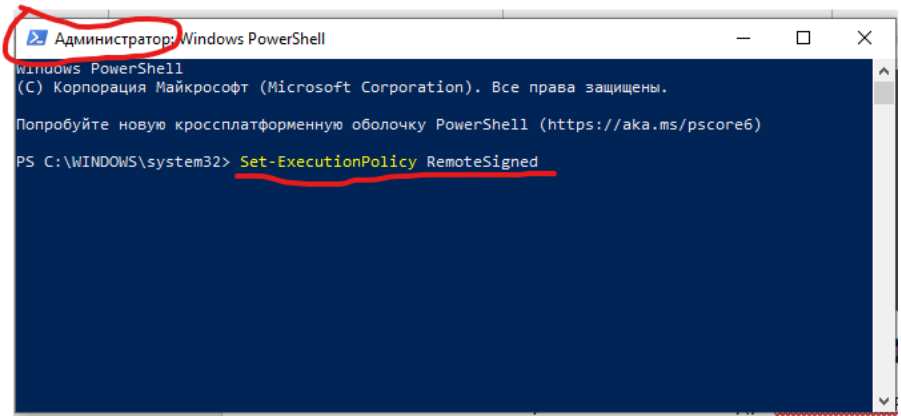
1.4 У випадку, якщо віртуальне оточення не активоване (відсутня надпис `env` або `venv`) зробити наступне. Натисніть

кнопку «Пошук» або «Windows» в операційній системі Windows та знайти PowerShell (іконка обов'язково має бути синього кольору). Натиснути праву кнопку миші на іконці та вибрати «Запуск від імені Адміністратора»



1.5 У вікні яке запустилось ввести команду «Set-ExecutionPolicy RemoteSigned», нажимаємо «Enter». Вікно можна закрити. Закрити та відкрити знову PyCharm.

Відкрити створений попереднього разу проект, або створити новий



The image shows a screenshot of a Windows PowerShell terminal window. The title bar at the top reads "Администратор: Windows PowerShell". The window content displays the following text: "Windows PowerShell", "(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.", "Попробуйте новую кроссплатформенную оболочку PowerShell (<https://aka.ms/pscore6>)", and the command prompt "PS C:\WINDOWS\system32> Set-ExecutionPolicy RemoteSigned". The command "Set-ExecutionPolicy RemoteSigned" is underlined with a red line.

2 Надання результатів виконання домашньої роботи

2.1 В кабінет необхідно завантажити скріншот запущеного PyCharm з відкритою закладкою «Terminal»

ЛАБОРАТОРНА РОБОТА 2. ПЕРШИЙ ПРОЕКТ

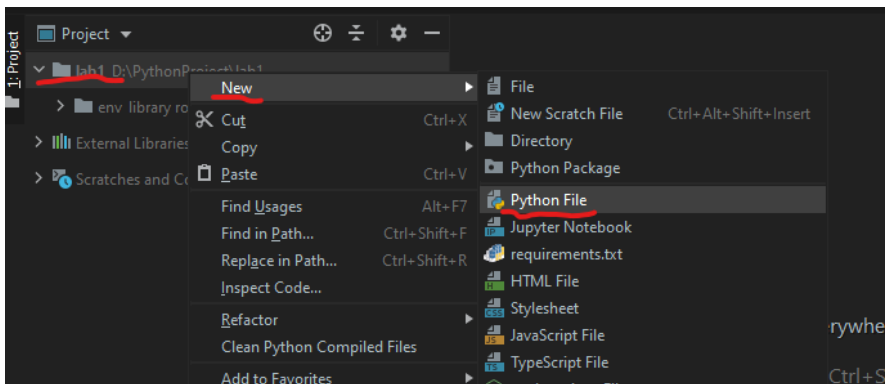
Мета роботи: Ознайомлення з середовищем розробки PyCharm. Створення і запуск першого проекту.
Ознайомлення з математичними функціями модуля math.
Освоєння роботи з редактором формул в MS Word.

Завдання для виконання лабораторної роботи

- 1 Підготовка до виконання лабораторної роботи
 - 1.1 Виберіть свій варіант виразу з Додатку 2
 - 1.2 Ознайомитись з переліком функцій доступних в модулі math за посиланням

<https://docs.python.org/uk/3/library/math.html>

- 1.3 Створіть новий проект у Pycharm
- 1.4 Додайте новий файл до проекту



2 Виконання роботи

2.1 Необхідно зчитати з клавіатури значення «x» та зробити його явне перетворення до типу float, для цього необхідно скористатись функцією **float()**. Зверніть увагу, що у випадку коли користувач введе крім числового значення інші символи Python видасть помилку. В роботі додаткових перевірок виконувати не треба – вважаємо що користувач коректно вводить числове значення.

2.2 Обчислити значення «y» згідно виразу, скориставшись відповідними функціями модуля `math`, для цього необхідно ввести функцію для обчислення значення виразу, коли цей вираз буде підкреслено PyCharm червоною хвилястою – натиснути комбінацію «Alt-Enter» вибрати пункт «Import

this name» та обрати відповідну функцію з модуля «math»ю
В результаті PyCharm автоматично добавить імпорт
вибраної функції з модуля math у першу строку вашого коду
у вигляді «**from math import cos**», де «cos» функцію яку
необхідно додати.

2.3 Вивести на екран обчислене значення за допомогою
функції «**print**».

3 Результат виконання лабораторної роботи

3.1 В Class Room необхідно додати файл у форматі MS Word
в який необхідно додати вираз, який обчислюється за
допомогою редактора формул та додати принтскрін екрану з
Pycharm де має бути видно реалізований код та консоль з
введеним значенням та результатом обчислення (приклад
нижче)

3.2 Приклад файла у форматі MS Word з результатами
виконання роботи:

$$y = \left(\frac{e^{-x} - 9.3}{\lg x - \cos(x^2)} \right)^{-0.3}$$

The image shows a Python IDE interface with a project named 'lab1' located at 'D:\PythonProject\lab1'. The file 'lab1.py' is open in the editor, containing the following code:

```
1 from math import cos, exp, log10
2
3 x = float(input('Введіть значення x: '))
4 y = ((exp(-x)-9.3)/(log10(x)-cos(x**2))) ** -0.3
5 print('Результат обчислення: ', y)
```

The Run console shows the execution of the script:

```
Run: lab1. x
↑ D:\PythonProject\lab1\env\Scripts\python.exe D:/PythonProject/lab1/lab1.py
↓ Введіть значення x: 0.125
Результат обчислення: 0.640139246929789
Process finished with exit code 0
```

ЛАБОРАТОРНА РОБОТА 3. ДЕПОЗИТНИЙ КАЛЬКУЛЯТОР

Мета роботи: закріпити на практиці матеріал, який стосується роботи зі строковими типами в Python, операціями вводу/виводу, циклами, умовними операторами та функціями. Запрограмувати принцип капіталізації з фінансової сфери.

Завдання для виконання лабораторної роботи

1 Постановка задачі

Один з комерційних банків об'явив акцію по депозитам для населення.

Згідно умов акції банк приймає гроші у національній валюті при наступних умовах:

1. Мінімальна сума вкладу 1000 грн. Сума кратна 1 гривні (банк копійки не приймає).
2. Строк вкладу від 3 до 5 років.
3. Банк нараховує 20% річних з щорічною капіталізацією.

Необхідно розробити програму, яка буде:

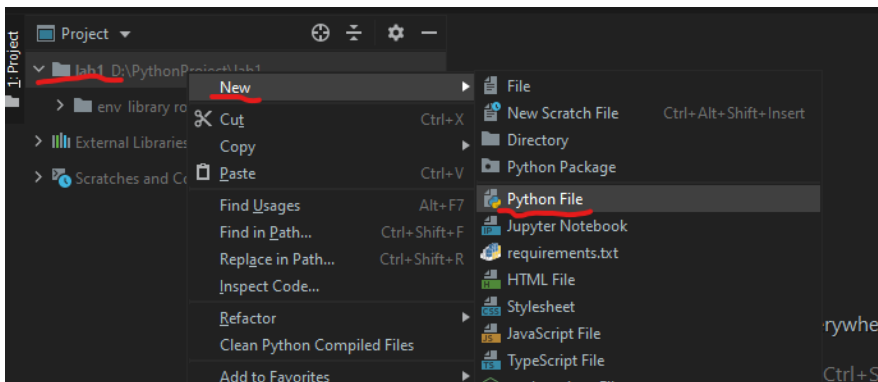
- зчитувати суму вкладу та строк, робити перевірку коректності введених параметрів
- формувати щомісячний графік нарахувань з врахуванням капіталізації
- виведе на екран графік та загальну суму грошей яку отримає вкладник по закінченню строку

При написанні програми необхідно використовувати: форматування строк, цикли, умовні оператори та функції.

2 Підготовка до виконання лабораторної роботи

2.1 Створіть новий проект у Pycharm

2.2 Додайте новий файл до проекту



3 Виконання роботи

3.1 Необхідно зчитати з клавіатури значення «сума вкладу» та зробити його явне перетворення до типу **int**, для цього необхідно скористатись функцією **int()**. Зверніть увагу, що у випадку коли користувач введе крім числового значення інші символи Python видасть помилку – перед явним перетворенням необхідно зробити перевірку на коректність за допомогою метода **.isdigit()**. Якщо користувач ввів дані некоректно – то вивести на екран повідомлення, що дані введені некоректно, та вказати якого формату вони мають бути, вивести на екран запит знов та повторювати спробу до тих пір, поки користувач не введе

коректні дані. Зчитування та перевірку на коректність оформити у вигляді функції.

3.2 Перевірити «сума вкладу» на допустиме значення. Якщо сума вкладу менша за визначену умовами – необхідно вивести повідомлення та повторювати процедуру вводу до тих пір, поки користувач не введе коректне значення, яке задовольняє умовам банку.

3.3 Необхідно зчитати з клавіатури значення **«строк вкладу»** (ціле число, в місяцях). Перевірку введеного значення зробити по аналогії з полем «сума вкладу», оформити окремою функцією.

3.4 Створити функцію яка має наступні вхідні параметри: «сума вкладу», «строк вкладу». Як результат функція має повернути структуру, яка має містити: місяць, сума нарахувань за місяць, загальна сума нарахувань. В функції необхідно передбачити, що нарахування розпочинаються з наступного дня від поточної дати. Щомісячна нарахована сума: $(\text{відсоткова щорічна ставка} / 12) * \text{сума вкладу}$. Реалізувати алгоритм, який дозволить врахувати щорічну капіталізацію вкладу, згідно прикладу в Додатку 2.

3.5 В основній програмі створити цикл, в якому необхідно вивести на екран результати нарахувань у вигляді таблиці. В кінці виводу таблиці вивести загальну суму яку вкладник має отримати та суму яку він заробив з депозитного вкладу.

3.6 Суми в графіку нарахувань мають бути округлені до копійок.

4 Результат виконання лабораторної роботи

4.1 В Class Room необхідно додати файл у *.ру з кодом програми та принт-скріни з результатами її роботи: ввод даних та результат обчислення.

4.2 Обов'язково надати принт-скріни з:

- некоректно введеними значеннями параметрів «сума вкладу» та «строк вкладу»
- з коректно введеними значеннями параметрів «сума вкладу» та «строк вкладу», але які не відповідають умовам банку
- коректно введеними параметрами «сума вкладу», «строк вкладу», графіком нарахувань та загальну

суму яку вкладник має отримати і суму яку він заробив з депозитного вкладу.

ЛАБОРАТОРНА РОБОТА 4. ВИБІР ДЕПОЗИТНОЇ ПРОГРАМИ

Мета роботи: закріпити на практиці матеріал, який стосується роботи зі строковими типами в Python, операціями вводу/виводу, циклами, умовними операторами та функціями. Запрограмувати принцип капіталізації з фінансової сфери.

Завдання для виконання лабораторної роботи

1 Постановка задачі

Фінансово-консультативна компанія надає послуги своїм Клієнтам. Одна з послуг – підбір депозитних програм Банків для розміщення коштів на термін, визначений Клієнтом. Для автоматизації роботи такої компанії та швидкого надання послуги необхідно створити програму, в яку можна додавати необмежену кількість Банків з налаштуванням їх депозитних програм. Вважається, що один Банк надає одну депозитну пропозицію.

Програма має працювати за наступним алгоритмом:

1. Зчитати інформацію щодо суми депозитного вкладу та терміну його розміщення які цікавлять Клієнта, зробити перевірку коректності введених даних.
2. Зчитати інформацію щодо Банків які підключені до програми.
3. Визначити які Банки готові надати свої послуги, зважаючи на умови Клієнта, щодо суми та терміну. Сформувати та вивести на екран перелік таких Банків (ім'я) та їх депозитних умов (відсоткова ставка, сума для видачі або сума заробітку), які готові розмістити в себе депозитний вклад.
4. Користувач має обрати Банк, в якому він хоче розмістити свій депозит, вказавши номер виведеної строки на екрані.
5. Вивести на екран графік у вигляді форматованої таблиці, загальну суму грошей та суму заробітку яку отримає вкладник по закінченню строку.

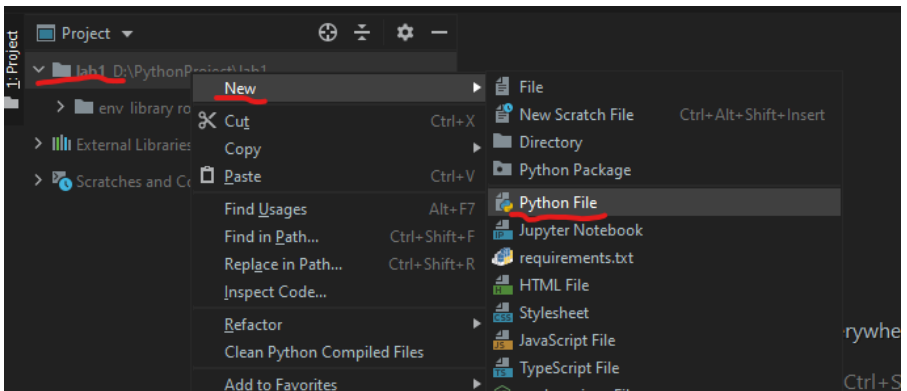
При написанні програми необхідно використовувати: форматування строк, цикли, умовні оператори та функції,

динамічне підключення модулів, отримання переліку файлів в каталозі, цикли, складні структури.

2 Підготовка до виконання лабораторної роботи

2.1 Створіть новий проект у Pycharm

2.2 Додайте новий файл «main.py» до проекту



2.3 Додайте до проекту папку, в якій необхідно розміщувати модулі. Один модуль = один Банк.

2.4 Створити тестовий файл «requirements.txt» в тій же папці що основний модуль «main.py». В файл додати модулі, які необхідно проінсталювати для роботи програми за допомогою команди: `pip install -r requirements.txt`.

3 Обрати варіант

Варіант	Рейтинг	Банк №1	Банк №2	Банк №3	Банк №4	Банк №5
1	П	26/1600/ 20/40	13/2600/ 30/80	25/1200/ 20/70	25/1100/ 40/50	27/1800/ 40/80
2	З	20/1300/ 30/70	27/2900/ 30/60	–	–	–
3	П	25/2500/ 40/70	16/2400/ 30/60	11/1400/ 40/80	11/2300/ 20/60	14/2500/ 20/80
4	З	17/2800/ 30/70	10/1800/ 20/60	14/2000/ 20/60	–	–
5	П	21/2100/ 20/30	25/2900/ 30/60	18/2500/ 20/80	19/1700/ 30/80	–
6	З	26/1400/ 30/70	16/2600/ 30/50	23/2300/ 30/80	12/1700/ 30/40	–
7	П	10/2300/ 30/80	20/1100/ 30/80	12/2200/ 20/70	30/2400/ 20/70	24/2000/ 40/70
8	З	30/2300/ 40/80	29/1700/ 20/50	–	–	–
9	П	20/2600/ 20/40	10/2900/ 20/40	10/1400/ 20/40	26/2900/ 20/80	–
10	З	21/1300/ 30/80	27/1200/ 20/60	–	–	–
11	П	10/1900/ 10/40	21/2200/ 30/50	19/2100/ 10/80	28/2600/ 30/60	–
12	З	11/1700/ 30/50	27/2600/ 40/70	15/2800/ 20/40	–	–
13	П	20/2700/ 20/40	18/2500/ 30/80	–	–	–
14	З	10/1400/ 40/60	30/1700/ 30/50	–	–	–
15	П	16/1600/ 10/20	20/3000/ 30/70	–	–	–

16	З	28/2800/ 30/80	19/2400/ 30/70	30/2700/ 30/70	27/2000/ 30/80	11/2500/ 10/70
17	П	13/2400/ 20/40	27/3000/ 30/50	15/2200/ 40/50	25/2800/ 10/80	–
18	З	15/2100/ 20/80	23/3000/ 20/60	25/2900/ 20/40	19/2200/ 20/50	–
19	П	10/1900/ 30/50	23/2900/ 30/70	17/1900/ 20/60	–	–
20	З	29/1800/ 10/50	28/2200/ 20/70	27/1000/ 40/60	28/2700/ 30/60	18/2700/ 10/40

де:

- для кожного Банку вказано відсоткову ставку/мінімальну суму у грн/мінімальний термін розміщення депозиту у місяцях/максимальний термін розміщення депозиту у місяцях
- «рейтинг»: П = «Сума повернення, грн.», З = «Заробіток, грн.»

4 Виконання роботи

4.1 В папці для зберігання модулів створити першій модуль, назва – має відповідати назві банку. Назви Банка обираються самостійно.

4.2 На початку модуля мають бути задані константи які мають зберігати: назву банка, мінімальну суму депозиту, мінімальний та максимальний період розміщення депозиту,

відсоткову ставку. Значення констант мають бути задані у відповідності до варіанту роботи.

4.3 В модулі мають бути присутні наступні функції:

4.3.1 check_deposit_sum(sum) – перевірка на відповідність суми умовам Банку. Вхідний параметр – сума депозиту, введена Клієнтом. Має бути використана константа «мінімальна сума депозиту». Функція має повертати або суму вкладу Користувача (sum), якщо вона відповідає умовам Банку, або **None** – у випадку, якщо не відповідає умовам Банку.

4.3.2 check_deposit_term(term) – перевірка на відповідність терміну умовам Банку. Вхідний параметр – термін для розміщення депозиту, введена Клієнтом. Мають бути використані константи «мінімальний період розміщення депозиту» та «максимальний період розміщення депозиту». Функція має повертати або термін вкладу Користувача (term), якщо вона відповідає умовам Банку, або **None** – у випадку, якщо не відповідає умовам Банку.

4.3.3 get_graf(sum, term) – функція має повернути структуру, яка має містити: місяць, дату нарахування, суму

нарахувань за місяць, загальну суму нарахувань на дату. В функції необхідно передбачити, що нарахування розпочинаються з наступного дня від поточної дати. Щомісячна нарахована сума: (відсоткова щорічна ставка / 12) * сума вкладу(sum). Суми в графіку нарахувань мають бути округлені до копійок. Функція має враховувати щорічну капіталізацію (по аналогії з лабораторною роботою №2).

4.3.4 get_bank_conditions(sum, term) – функція має повертати кортеж у вигляді («ім'я Банку», «Відсоткова ставка», «Сума повернення, грн.» або «Заробіток, грн.», в залежності від варіанту) якщо і сума вкладу (sum) і термін (term) відповідають вимогам Банку, якщо ні – функція повертає None. Функція має використовувати функції: check_deposit_sum, check_deposit_term, get_graf.

4.3.5 В модулі має бути присутня конструкція «**if __name__ == '__main__':**», яка може мати наступний вигляд:

```
if __name__ == '__main__':
    sum = deposit_min
    term = period_min
    bank_cond = get_bank_conditions(sum, term)
    if bank_cond is None:
        print(f'Умови банківського депозиту для
{bank_name} не виконані')
    else:
```

```

        print(f'Банк {bank_cond[0]} пропонує
Відсоткову ставку
{round(bank_cond[1]*100,2):.2f}%')
        print(f'В кінці терміну буде видано
{bank_cond[2]:.2f} грн.')
```

```

        graf = get_graf(sum, term)
        print('-' * 82)
        print('{0:^3s} | {1:^16s} | {2:^30s} |
{3:^22s} |'.format('#', 'Дата', 'Ставка', 'Сума'))
        print('-' * 82)
        for x in graf:
            print(f'{x[0]:<3} | {x[1].strftime("%d-
%m-%Y"):<16} | {x[2]:>30} | {x[3]:>22.2f} |')
```

```

        print('-' * 82)
        print(f'Сума до видачі, грн.:
{x[3]:>15.2f}')
```

```

        print(f'Заробіток клієнта, грн.: {x[3] -
sum:>12.2f}')
```

де `sum` та `term` – мають відповідати мінімальній сумі (`deposit_min`) та терміну (`period_min`) для Банку; `bank_name` – назва Банку.

4.4 Коли перший модуль буде написаний і протестований – модуль необхідно продублювати стільки раз скільки Банків визначено варіантом. Мають бути змінені назви модулів, та змінені константи в модулі - умови депозитної програми, у відповідності до варіанту.

4.5 В основному модулі «`main.py`» мають бути присутні наступні елементи:

4.5.1 Функція або функції які послідовно мають вивести на екран запити до Користувача на ввід суми вкладу у грн., та

терміну його розміщення у місяцях. Користувач має ввести чисельні значення цілого типу. У випадку якщо введені дані не відповідають критерію – програма має повідомити користувача про це і знову запросити інформацію для вводу. Процес вводу кожного параметру має тривати до тих пір доки Користувач не введе коректно кожен з параметрів.

4.5.2 В основному модулі має бути присутня конструкція «`if __name__ == '__main__':`», в якій має запускатись основний процес програми.

4.5.3 Процес основного модуля має містити:

4.5.3.1 Динамічне підключення модулів, які розміщуються в відповідній папці та обов'язково містять дві функції «`get_bank_conditions`» та «`get_graf`»:

```
banks_list = []

for file in listdir(path):
    if file.endswith('.py'):
        module_name = file.replace('.py', '')
        m = __import__(f"{path}.{module_name}",
fromlist=[module_name])
        if hasattr(m, 'get_bank_conditions') and
hasattr(m, 'get_graf'):
            banks_list.append(m)
```

де `path` – константа, яка містить назву папки, в якій знаходяться модулі.

4.5.3.2 Програма має перевіряти, чи підключений хоча би один з модулів, якщо ні – має видати повідомлення користувачу та вийти з програми.

4.5.3.3 Програма має перевірити пропозиції від Банків на відповідність запиту Клієнта:

```
propositions = []

for m in banks_list:
    p = m.get_bank_conditions(sum, term)
    if p is not None:
        propositions.append(p)

if len(propositions) == 0:
    print('На ваших умовах не має депозиту в жодному з банків !')
    print('Змініть умови для депозиту')
```

де `sum` та `term` – коректно введені Клієнтом сума та термін вкладу. Якщо на умовах Клієнта жоден з банків не готовий надати депозитну програму – програма має про це сповістити та заставити Користувача повторно ввести значення суми та терміну вкладу. Цей процес має тривати до тих пір, поки введені дані не стануть відповідати умовам хоча б одного з Банків.

4.5.3.4 Вивести на екран пропозиції від усіх банків, які згодні надати депозитну програму Клієнту:

```
print('Для вас готові надати послуги наступні банки:')
for i, p in enumerate(propositions):
    print(f'[{i}]. Банк: {p[0]}. Відсоткова ставка: {p[1]}. Сума в кінці терміну: {p[2]:.2f} грн. ')
```

4.5.3.5 Зчитати з клавіатури код Банку (у наведеному прикладі, код виведено у «квадратних дужках») та вивести на екран графік нарахувань та суми до видачі:

```

bank = int(input('вкажіть номер банку для отримання
графіку: '))
print(f'Графік розрахунку для Банку
{propositions[bank][0]}:')
graf = banks_list[bank].get_graf(sum, term)
print('-' * 82)
print('{0:^3s} | {1:^16s} | {2:^30s} | {3:^22s}
|'.format('#', 'Дата', 'Ставка', 'Сума'))
print('-' * 82)
for x in graf:
    print(f'{x[0]:<3} | {x[1].strftime("%d-%m-
%Y"):<16} | {x[2]:>30} | {x[3]:>22.2f} |')

print('-' * 82)
print(f'Сума до видачі, грн.: {x[3]:>15.2f}')
print(f'Заробіток клієнта, грн.: {x[3] -
sum:>12.2f}')

```

4.5.3.6 Робота основного модуля завершена

5 Результат виконання лабораторної роботи

В архів для перевірки необхідно додати:

5.1 Архів, який має містити: main.py, requirements.txt та папку з модулями. Назва архіву має містити відомості про ПІБ студента, групу, та префікс «MW_».

5.2 Звіт з виконання модульної роботи в форматі MS Word, який має містити наступну інформацію:

- титульний аркуш

- принтскрини роботи програми з некоректно введеними терміном та сумою вкладу
- принтскрін з коректно введеними значеннями параметрів «сума вкладу» та «строк вкладу», але значення не відповідають умовам жодного з Банків
- принтскрін роботи програми з відсутніми модулями в папці
- принтскрін з коректно введеними параметрами «сума вкладу», «строк вкладу», списком Банків, які готові надати депозитну програму, вибраним користувачем кодом Банку та графіком нарахувань за умовами вибраного Банку та загальну суму яку Клієнт має отримати і суму яку він заробив з депозитного вкладу.

СПИСОК ЛІТЕРАТУРИ

1. Баженов В.А., Венгерський П.С., Гарвона В.С. та ін. Інформатика. Комп'ютерна техніка. Комп'ютерні технології. - К.: Каравела, 2019. – 592 с.
2. В.М.Ткачук Алгоритми і структура даних: Навчальний посібник. - Видавництво Прикарпатського національного університету імені Василя Стефаника, 2018. – 286 с.
3. Кормен Т, Лейзерсон Ч., Рівест Р., Стайн К. Вступ до алгоритмів - К.: К.І.С., 2019. – 1288 с.
4. Пекарський Б.Г. Основи програмування: Навчальний посібник.- Кондор. 2018.–364 с.
5. Робсон Е., Фрімен Е., Head First. Патерни проєктування. - Фабула. 2020 – 672 с.
6. Булгакова О.С., Зосімов В.В., Ходякова Г.В. Алгоритмізація і програмування: теорія та практика: навч. посіб. для дист. навч. - Миколаїв: СПД Румянцева. 2021. – 138 с.

7. Костюченко А. О. Основи програмування мовою Python: навч. посіб. - Чернігів: ФОП Баликіна С. М. 2020. – 180 с.
8. Васильєв О. М. Програмування мовою Python. - Тернопіль: Навчальна книга – Богдан. 2019. – 504 с.
9. Крєневич А. П. Python у прикладах і задачах. Частина 1. Структурне програмування: навч. посіб. - Київ : ВПЦ "Київський Університет". 2017. – 206 с.
10. Stephenson B.: The Python Workbook: A Brief Introduction with Exercises and Solutions, 2nd ed. - Springer. 2019 – 219 p.
11. Kalb I.: Learn to Program with Python 3: A Step-by-Step Guide to Programming, 2nd ed. - Apress. 2018. – 371 p.
12. Browning J.B., Alchin M.: Pro Python 3: Features and Tools for Professional Development, 3 rd ed. - Apress. 2019 – 458 p.

ДОДАТОК 1. ТИТУЛЬНА СТОРІНКА. ВИМОГИ ДО ОФОРМЛЕННЯ

Вимоги до оформлення результатів виконання лабораторних робіт регулюються загальноприйнятими стандартами академічного письма, зокрема ДСТУ, та рекомендації Міністерства освіти і науки України. Ось основні вимоги до оформлення курсової роботи станом на 2024 рік:

1. Загальні вимоги

- **Мова виконання:** українська, якщо інше не визначено навчальним закладом.
- **Обсяг роботи:** 30–40 сторінок друкованого тексту (залежно від спеціальності та вимог викладача).
- **Шрифт:** Times New Roman.
- **Розмір шрифту:** 14 pt.
- **Міжрядковий інтервал:** 1.5.
- **Вирівнювання тексту:** по ширині.
- **Поля сторінки:**
 - Ліве — 30 мм;

- Праве — 10–15 мм;
- Верхнє та нижнє — 20 мм.
- **Нумерація сторінок:**
 - Починається з другої сторінки (титульна сторінка не нумерується).
 - Номер сторінки розташовується в правому нижньому куті.

2. Структура роботи

1. Титульна сторінка:

Оформлюється відповідно до вимог закладу.

Зазвичай містить:

- Назву закладу.
- Факультет, кафедру.
- Назву роботи.
- Прізвище, ім'я та по батькові студента.
- Науковий ступінь і звання керівника.
- Рік виконання.

2. Зміст (план):

- Перелік розділів із зазначенням сторінок, де вони починаються.
- Наприклад:

- Зміст
- Вступ 3
- 1. Теоретичні основи дослідження 5
- 2. Практична частина 15
- Висновки 30
- Список використаних джерел 33

3. Вступ:

- Обґрунтування актуальності теми.
- Мета і завдання роботи.
- Об'єкт і предмет дослідження.
- Методологія дослідження.
- Структура роботи.

4. Основна частина:

- Поділяється на розділи (2–3 основних розділи).
- **Перший розділ:** Теоретичний — огляд літератури, визначення основних понять.
- **Другий розділ:** Практичний — аналіз, розрахунки, експерименти, проектування тощо.

- **Третій розділ** (за потреби): Узагальнення, порівняння результатів або рекомендації.

5. Висновки:

- Стисле узагальнення основних результатів роботи.
- Висновки щодо виконання поставлених завдань.

6. Список використаних джерел:

- Оформлюється за стандартом ДСТУ 8302:2015.
- Джерела наводяться в алфавітному порядку.

7. Додатки (за потреби):

- Таблиці, графіки, рисунки, великі блоки коду тощо.

3. Оформлення ілюстрацій, таблиць, формул

• Рисунки:

- Підписуються під ілюстрацією (наприклад: *Рисунок 1. Схема системи*).
- Всі рисунки нумеруються послідовно.

• Таблиці:

- Заголовок розташовується над таблицею (наприклад: *Таблиця 2. Порівняння параметрів*).
- Таблиці також нумеруються послідовно.
- **Формули:**
 - Розташовуються по центру сторінки.
 - Нумеруються в круглих дужках праворуч (наприклад: (1)).

4. Оформлення списку використаних джерел

- **Кількість джерел:** Зазвичай 10–20, залежно від теми.
- **Формат цитування:** Оформлюється відповідно до ДСТУ 8302:2015:
 - Книги: *Автор. Назва книги. – Місто: Видавництво, Рік. – Кількість сторінок.*
 - Статті: *Автор. Назва статті // Назва журналу. – Рік. – Том, номер. – Сторінки.*

5. Електронна версія

Подача роботи в електронному вигляді. Необхідно дотримуватися зазначеного формату (зазвичай PDF).

ДОДАТОК 2. ВАРІАНТИ ВИРАЗІВ ДЛЯ ВИКОНАННЯ
ЛАБОРАТОРНОЇ РОБОТИ №1

Варіант	Завдання
1	$y = \frac{\sqrt{x-1} - \operatorname{tg}(x+1)}{\arccos(x) + \ln(x)} + 2.42$
2	$y = \frac{\sin x^2 - \cos^4(x-1)^2}{\operatorname{arctg}(x+2.6) + \sqrt[3]{\ln(x)}}$
3	$y = \frac{e^{-3x} + \operatorname{tg}(3x-3)}{ \sin(x) + \sqrt[4]{\cos(x)} + \cos(2x)}$
4	$y = \frac{e^{-x} - 4x - \ln^3(x)}{\lg x+1 + \operatorname{ctg}(x^2-1)}$
5	$y = \operatorname{arcsin}\left(\frac{x \cdot \ln(x)}{1 + \cos(x)} + 1\right)$
6	$y = x + \frac{\sqrt{\operatorname{arcsin}(x) + \operatorname{arctg}(\pi x)}}{\lg(13.4x + \pi)}$
7	$y = \operatorname{arctg}(x) + \frac{e^{0.6x-1} - \sqrt{(x+6.1)^3}}{\ln(x) + \operatorname{tg}^2(x)}$
8	$y = 0.3 \cdot \lg(e^{-x}) + \frac{\operatorname{arctg}(x) - \sin^2(x)}{4 \cdot \sqrt{\ln x-1 }}$
9	$y = e^2 \cdot \lg(x^4) \cdot \frac{(x-0.5)^2 - \cos(x)}{\sqrt{ x+1 + x }}$

10	$y = \frac{e^{-x} - 4 \cdot \ln(x)}{\ln(x) - \cos x+1 }$
11	$y = \sin^2(x) - \frac{\sqrt{\lg 2-x + \lg(x)}}{\sqrt[3]{\lg(x)} + \sqrt{\cos^3(x)}}$
12	$y = \arccos\left(\frac{x - \lg(x)}{1 + \cos(3x)} + 1\right)$
13	$y = \arctg(x) - \frac{\sqrt{\ln(4) + \ln(x)}}{\sqrt[3]{\lg(2.40)} + \sqrt[5]{\cos(x^{-1})}}$
14	$y = \frac{e^{-3x} + \lg(4x-1)}{ \cos(x) + \sqrt{\cos(2x)}}$
15	$y = \frac{e^{-3x} + \ln^3(x-1)}{\ln x+1 + \lg(x^2-1)}$
16	$y = \sin\left(\frac{x + 2.3 \cdot \lg(x+1)}{\sqrt{2\ln(x)} + \cos(x)}\right)$
17	$y = \ln 1-x + \frac{\lg(x) - \sin^2(x)}{1 - \sqrt{\ln(x)}}$
18	$y = \frac{e^{-x} + \lg(x-1)}{ \ln(x) + \sqrt{\sin(x) + \cos(2x)}}$
19	$y = \lg x - \frac{\sqrt{\ln(2) + \ln(x)}}{\sqrt[3]{\lg(x)} + \sqrt[4]{\cos(x^{-1})}}$
20	$y = \frac{\sqrt{x+1} - \sin(x-\pi)}{\cos(x-3.1) + \ln^2(x)} + x \cdot \lg(x)$

ДОДАТОК 3. ПРИКЛАД ДЕПОЗИТНОГО РОЗРАХУНКУ З УРАХУВАННЯМ ЩОРІЧНОЇ КАПІТАЛІЗАЦІЇ

Капіталізація депозитного вкладу — це процес, за якого нараховані проценти додаються до основної суми депозиту, таким чином збільшуючи розмір вкладу. Це дозволяє в майбутньому нараховувати проценти не тільки на початкову суму, а й на раніше нараховані проценти.

Таким чином, за допомогою капіталізації, депозит "працює" на себе і з часом приносить більший дохід. Капіталізація може відбуватися різними періодами: щомісяця, щоквартально, раз на рік тощо, залежно від умов договору.

Сума вкладу	1000
Строк (міс)	36
Відсоткова річна ставка	20%
Відсоткова щомісячна ставка	1,67%

Дата оформлення	13.10.2023
Дата початку дії договору	14.10.2023

1	14.11.2023	16,67	1016,67	1000
2	14.12.2023	16,67	1033,33	1000
3	14.01.2024	16,67	1050,00	1000
4	14.02.2024	16,67	1066,67	1000
5	14.03.2024	16,67	1083,33	1000
6	14.04.2024	16,67	1100,00	1000

7	14.05.2024	16,67	1116,67	1000
8	14.06.2024	16,67	1133,33	1000
9	14.07.2024	16,67	1150,00	1000
10	14.08.2024	16,67	1166,67	1000
11	14.09.2024	16,67	1183,33	1000
12	14.10.2024	16,67	1200,00	1000
13	14.11.2024	20,00	1220,00	1200
14	14.12.2024	20,00	1240,00	1200
15	14.01.2025	20,00	1260,00	1200
16	14.02.2025	20,00	1280,00	1200
17	14.03.2025	20,00	1300,00	1200
18	14.04.2025	20,00	1320,00	1200
19	14.05.2025	20,00	1340,00	1200
20	14.06.2025	20,00	1360,00	1200
21	14.07.2025	20,00	1380,00	1200
22	14.08.2025	20,00	1400,00	1200
23	14.09.2025	20,00	1420,00	1200
24	14.10.2025	20,00	1440,00	1200
25	14.11.2025	24,00	1464,00	1440
26	14.12.2025	24,00	1488,00	1440
27	14.01.2026	24,00	1512,00	1440
28	14.02.2026	24,00	1536,00	1440
29	14.03.2026	24,00	1560,00	1440
30	14.04.2026	24,00	1584,00	1440
31	14.05.2026	24,00	1608,00	1440
32	14.06.2026	24,00	1632,00	1440
33	14.07.2026	24,00	1656,00	1440
34	14.08.2026	24,00	1680,00	1440
35	14.09.2026	24,00	1704,00	1440
36	14.10.2026	24,00	1728,00	1440
				1728

200
 <- капіталізація суми

240,00
 <- капіталізація суми

288

1000 728