

Волинський національний університет імені Лесі Українки

Кафедра теоретичної та комп'ютерної фізики
імені А. В. Свідзинського

Замуруєва О. В., Сахнюк В.Є., Федосов С.А.

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

Лабораторний практикум

ЛУЦЬК 2021

УДК 004.451(075.8)

З-26

Рекомендовано до друку науково-методичною радою Волинського національного університету імені Лесі Українки (протокол № __ від __ листопада 2021 р.).

Рецензенти:

Муляр В. П. – кандидат педагогічних наук, доцент кафедри експериментальної фізики, інформаційних та освітніх технологій, ВНУ імені Лесі Українки;

Сачук Ю.В. – кандидат фізико-математичних наук, доцент кафедри комп'ютерних наук та кібербезпеки ВНУ імені Лесі Українки.

З Замуруєва О. В., Сахнюк В.Є., Федосов С.А. **Інформаційні технології та системи** : лабораторний практикум. Луцьк : Вежа-Друк, 2021. – 38 с.

Методичні вказівки до лабораторних робіт з навчальних дисциплін «Інформаційні технології та системи», «Бази даних на мові SQL», «Алгоритми і структури даних» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійними програмами 014.08 Середня освіта (Фізика), 104 Фізика та астрономія, 105 Прикладна фізика та наноматеріали.

УДК 004.451(075.8)

ЗМІСТ

МОДУЛЬ 1 «БАЗИ ДАНИХ».....	4
Лабораторна робота №1 СТВОРЕННЯ БАЗИ ДАНИХ.....	4
Лабораторна робота № 2 КОРИГУВАННЯ БАЗИ ДАНИХ.....	7
Лабораторна робота №3 ПРОСТІ ЗАПИТИ. ГРУПОВІ ОПЕРАЦІЇ. ВИКОРИСТАННЯ АГРЕГАТНИХ ФУНКЦІЙ.....	8
Лабораторна робота № 4 БАГАТОТАБЛИЧНІ ЗАПИТИ. ВКЛАДЕНІ ЗАПИТИ. ПРЕДСТАВЛЕННЯ.....	10
Лабораторна робота № 5 ГЕНЕРАТОРИ. ТРИГЕРИ. КОНСТРУКЦІЇ МОВИ SQL.....	12
Лабораторна робота № 6 ЗБЕРЕЖЕНІ ПРОЦЕДУРИ.....	13
Лабораторна робота № 7 БЕЗПЕКА БАЗИ ДАНИХ. КОРИСТУВАЧІ, РОЛІ, ПРАВА.....	14
МОДУЛЬ 2 «ПРОГРАМУВАННЯ МОВОЮ PYTHON».....	15
Лабораторна робота №1 ВИКОРИСТАННЯ БАЗОВИХ ТИПІВ ТА ЗАСОБІВ МОВИ ПРОГРАМУВАННЯ PYTHON. СЕРЕДОВИЩА ТА ІНСТРУМЕНТАРІЙ РОЗРОБЛЕННЯ ПРОГРАМ МОВОЮ PYTHON.....	15
Лабораторна робота №2 УПРАВЛЯЮЧІ КОНСТРУКЦІЇ ТА МАСИВИ У МОВІ PYTHON.....	18
Лабораторна робота №3 ОБРОБКА ПОСЛІДОВНОСТЕЙ ПРИ ПРОГРАМУВАННІ НА МОВІ PYTHON.....	20
Лабораторна робота №4 РОЗРОБКА ПРОГРАМ З ВИКОРИСТАННЯМ ПРОЦЕДУР І ФУНКЦІЙ.....	25
Лабораторна робота №5 РОЗРОБКА ПРОГРАМ З ВИКОРИСТАННЯМ КЛАСІВ В PYTHON.....	30
Лабораторна робота №6 РОБОТА З БАЗОЮ ДАНИХ ІЗ PYTHON- ПРОГРАМИ.....	33
Лабораторна робота №7 ПОБУДОВА ГРАФІКІВ МАТЕМАТИЧНИХ ФУНКЦІЙ У МОВІ PYTHON.....	35

МОДУЛЬ 1 «БАЗИ ДАНИХ»

Лабораторна робота №1 СТВОРЕННЯ БАЗИ ДАНИХ

Завдання до виконання:

1. Завантажте програму *IBExpert*.
2. За допомогою команди **Database ⇒ New Database** створіть на диску D:\Students у своїй папці базу даних Univer.gdb для користувача **SYSDBA** з паролем **masterkey**. Розмір сторінки задайте 4096, кодування WIN1251, **Server – Remote**, **Server name – localhost**. Після чого відбувається перенаправлення на сторінку реєстрації, де вказуються **Server Version – Firebird 2.5** і псевдонім бази даних. (Для використання створеної бази даних, наприклад на іншому комп'ютері, її необхідно зареєструвати за допомогою команди **Database ⇒ Register Database**).
3. За допомогою команди **Database ⇒ Connect to Database** відкрийте базу даних **UNIVER**.
4. Створіть домен **NAM** для текстових даних, що містять не більше 20 символів, задавши кодування і порядок сортування WIN1251.
5. За допомогою команди **Database ⇒ New Table** створіть таблицю **STUDENTS** з полями

Таблиця **STUDENTS**

SNOM	SFAM	SIMA	SOTCH	STIP	GRUP	FORM	FOTO
200101	Іванчук	Сергій	Петрович	75,00	ПМ-32	платна	<null>

SNOM – номер залікової книжки студента є первинним ключем, задаються значення Not Null і PK. Поля **SFAM** – прізвище, **SIMA** – ім'я, **SOTCH** – по батькові студента, для яких задайте тип даних у вигляді домену **NAM**. Поле **GRUP** – група, в якій навчається студент, для нього задайте значенням за замовчуванням назву своєї групи в одинарних лапках. Поле **STIP** – стипендія – має тип даних **NUMERIC** або **DECIMAL** з 2 знаками після коми. Для поля **FOTO** – фото студента – вибирається тип даних **BLOB**

SUB_TYPE BYNARY SIZE 2048. Для поля **FORM** – форма навчання – задайте обмеження, що допускають ввід тільки двох значень 'платна' або 'бюджет'.

6. За допомогою команди **Database** ⇒ **New Table** створіть таблицю

USPISH з полями

Таблиця **USPISH**

NOM	DATA	SNOM	PNOM	OCINKA
1	13.06.2005	200101	301	5

NOM є первинним ключем, для поля **DATA** – дата здачі іспиту – тип даних **DATE**. **OCINKA** – оцінка за іспит є цілим. Для поля **OCINKA** допускається ввід цілих значень від 1 до 5 включно або **NULL**. За замовчуванням встановлюється значення **NULL** (не визначено).

7. За допомогою **SQL Editor** в командному режимі створіть таблицю **VYKLAD** з полями

Таблиця **VYKLAD**

VNOM	VFAM	VIMA	VOTCH	KAF	POSADA	OKLAD
10001	Бачишина	Лариса	Дмитрівна	ПМ	Ст.викладач	5710,40

VNOM – табельний номер викладача, є цілим, не допускає невизначених значень, задається як первинний ключ таблиці. **VFAM** – прізвище, **VIMA** – ім'я, **VOTCH** – по батькові викладача. Для прізвища, імені, по батькові задайте тип даних у вигляді домену **NAM**, **KAF** – місце роботи. **POSADA** – посада викладача, **OKLAD** – оклад викладача. Для окладу викладача виберіть тип даних **NUMERIC** або **DECIMAL**, вказавши 2 знаки після коми.

8. Створіть таблицю **PREDMET** з полями

Таблиця **PREDMET**

PNOM	PNAME	VNOM	GOD	SEMESTR
301	Бази даних	101	100	5

PNOM – номер предмета в навчальному плані є цілим і є первинним ключем. **PNAME** – назва предмета. **GOD** – кількість годин по даному предмету. **SEMESTR** – семестр, в якому читається предмет. Для семестру задайте обмеження від 1 до 10.

9. На вкладці **Constraints** вікна **Table** для таблиці **USPISH** задайте зовнішній ключ для зв'язку з таблицею **STUDENTS** по полю **SNOM**. Задайте умови цілісності, що задовольняють умовам каскадування. При створенні таблиці **PREDMET** задайте зв'язок у вигляді зовнішнього ключа з таблицею **VYKLAD** по полю **VNOM**. Задайте умови цілісності: при знищенні – **SET NULL**, при зміні – **CASCADE**.

Лабораторна робота № 2

КОРИГУВАННЯ БАЗИ ДАНИХ

Завдання до виконання:

1. Завантажте програму *IBExpert*.
2. Відкрийте базу даних **Univer**.
3. За допомогою команди **Tools** ⇒ **SQL Editor** відкрийте вікно для вводу SQL-команд і виконайте наступні дії:
 - 1) задайте обмеження для поля **SEMESTR** таблиці **PREDMET** так, щоб значення даного поля були >0 і ≤ 10 ;
 - 2) встановіть зв'язок таблиці **USPISH** з таблицею **PREDMET** по полю **PNOM**. Задайте умови цілісності, що задовольняють умовам каскадування;
 - 3) додайте у таблицю **PREDMET** поле **POLE** типу **FLOAT**;
 - 4) змініть назву поля **POLE** на **POLE1**;
 - 5) змініть тип даних на **VARCHAR(15)**;
 - 6) розташуйте дане поле третім по порядку;
 - 7) видаліть поле **POLE1**;
 - 8) введіть дані у таблиці **VYKLAD** і **PREDMET** (не менше 6);
 - 9) видаліть запис про викладача, прізвище якого **Бачишина**;
 - 10) модифікуйте у таблиці **PREDMET** запис про предмет Баз даних, задавши семестр 6 і кількість годин 108;
 - 11) створіть розрахункове поле **OPLATA**, в якому вираховується податок 13% від зарплати;
 - 12) видаліть поле **OPLATA**;
 - 13) закрийте *SQL Editor*.
4. В *IBExpert* відкрийте таблицю **STUDENTS**, введіть 7 значень у дану таблицю. Закрийте таблицю.
5. Відкрийте таблицю **USPISH**, введіть 15 значень у дану таблицю. Закрийте таблицю.

Лабораторна робота №3
ПРОСТІ ЗАПИТИ. ГРУПОВІ ОПЕРАЦІЇ. ВИКОРИСТАННЯ
АГРЕГАТНИХ ФУНКЦІЙ.

Завдання до виконання:

1. Завантажте програму *IBExpert*.
2. Відкрийте базу даних **Univer**.
3. Використовуючи команду **SELECT** у командному вікні *SQL Editor*, виконайте наступні запити:
 - 1) Вивести на екран прізвища, імена та розмір стипендії всіх студентів. Дані відсортувати в алфавітному порядку по прізвищах.
 - 2) Вивести розрахунок стипендії після підвищення її у два рази для кожного студента. Задати назву для нового стовпця **NSTIP**.
 - 3) Вивести на екран прізвища та імена студентів, стипендія яких дорівнює 75.
 - 4) Вивести на екран номери залікових книжок студентів, які не мають трійок по певному предмету.
 - 5) Вивести інформацію про студентів, які мають 4 і 5 по певному предмету.
 - 6) За допомогою оператора **IN** вивести прізвища студентів, імена яких Анатолій або Володимир.
 - 7) Вивести інформацію про студентів, прізвища яких починаються з літер із діапазону „А” – „М”.
 - 8) Вивести номери залікових книжок студентів, які складали іспити від 1.06.2005 до 15.06.2005.
 - 9) Вивести номери залікових книжок студентів, які складали іспити в червні.
 - 10) Вивести номери залікових книжок студентів, які не принесли фото.
 - 11) Вивести номери залікових книжок студентів, які не здали іспити (отримали незадовільні оцінки). Дані не повинні повторюватись.

- 12) Вивести список викладачів, прізвища яких починаються на літеру „Б”.
- 13) Вивести список дисциплін, в назві яких є слово „бази”.
- 14) Підрахувати кількість записів у таблиці **STUDENTS**.
- 15) Підрахувати суму стипендій, яка виплачується на групу
- 16) Підрахувати кількість студентів, які отримують стипендію.
- 17) Вивести середній бал успішності по таблиці **USPISH**.
- 18) Вивести мінімальну і максимальну оцінки для кожного студента по результатам сесії.
- 19) Вивести мінімальну позитивну оцінку для кожного студента.
- 20) Підрахувати кількість студентів, які склали іспит з кожного предмета (на будь-яку оцінку).
- 21) Підрахувати кількість студентів, які склали іспит з кожного предмета позитивно.
- 22) Вивести середній бал кожного студента.
- 23) Вивести список студентів (SNOM), середній бал яких $<4,3$.
- 24) Вивести кількість студентів які склали 3 іспити.
- 25) Підрахувати кількість студентів, які навчаються на бюджеті і платній формі.

Лабораторна робота № 4
БАГАТОТАБЛИЧНІ ЗАПИТИ. ВКЛАДЕНІ ЗАПИТИ.
ПРЕДСТАВЛЕННЯ.

Завдання до виконання:

1. Завантажте програму *IBExpert*.
2. Відкрийте базу даних **Univer**.
3. Використовуючи команду **SELECT** у командному вікні *SQL Editor*, виконайте наступні запити:

- 1) Вивести прізвища, імена, по батькові студентів і викладачів, відсортувати їх в алфавітному порядку по прізвищу.
- 2) До запиту 1 додати вивід стовпця **POSADA**, який включає значення 'студент' і 'викладач' відповідно для кожного запису.
- 3) Вивести прізвища викладачів і назви предметів, які вони викладають.
- 4) Вивести прізвища викладачів, назви предметів, які вони викладають і кількість годин по кожному предмету, якщо кількість годин >100.
- 5) За допомогою команди **INSERT** доповнити таблицю **STUDENTS** записами про двох нових студентів. Вивести прізвища усіх студентів і номери предметів, з яких вони склали іспити. (лівостороннє об'єднання).
- 6) Записати попередній запит з використанням правостороннього об'єднання.
- 7) Вивести прізвища студентів і назви предметів, іспити з яких вони повинні скласти, тобто розклад іспитів (операція декартового добутку).
- 8) Вивести прізвища студентів і назви предметів, іспити з яких вони склали (задати зв'язок між таблицями).
- 9) Використовуючи вкладені запити, вивести інформацію про предмети, які викладає викладач **Бачишина**.

10) Використовуючи вкладені запити, вивести прізвища студентів, які отримують максимальну стипендію (розмір максимальної стипендії наперед невідомий).

11) Використовуючи оператори EXISTS, ANY, ALL, SOME, сформувані запити, що реалізують наступні завдання:

- a) Вивести дані про студентів, що мають незадовільні оцінки
- b) Вивести прізвища тих студентів, що мають лише одну „2” .
- c) Вивести дані про студентів, що здали всі іспити.
- d) Вивести прізвища тих викладачів, що читають більше ніж один предмет.
- e) Вивести прізвища всіх студентів, що здавали іспит у вказаний день.
- f) Вивести назви тих дисциплін, для вивчення яких відведена однакова кількість годин.
- g) Вивести прізвища тих студентів, що мають ім'я, яке співпадає із заданим.
- h) Вивести прізвища всіх студентів, які здавали іспити (без повторень).

Лабораторна робота № 5

ГЕНЕРАТОРИ. ТРИГЕРИ. КОНСТРУКЦІЇ МОВИ SQL.

Завдання до виконання:

1. Завантажте програму *IBExpert*.
2. Відкрийте базу даних **Univer**.
3. В *SQL Editor* створіть генератор **GEN1** з початковим значенням, рівним 20. За допомогою команди **INSERT** додайте запис у таблицю **VYKLAD**, використовуючи функцію **GEN_ID**.
4. За допомогою засобів *IBExpert* створіть генератор **GEN2**. Задайте за допомогою тригера лічильник для поля **NOM** для додавання нових записів у таблицю **USPISH**, починаючи з 30.
5. Створіть тригер, який перетворює назви предметів таблиці **PREDMET** у великі прописні літери при модифікації чи додаванні нових записів у дану таблицю.
6. Створіть тригер, який при додаванні нового запису у таблицю **STUDENTS** нараховує стипендію 180 грн. студентам державної форми навчання.
7. Створіть виключення **Forma**, задавши текст повідомлення: «Введіть правильно форму навчання: платна або бюджет». Створіть тригер, який при вводі неправильної форми навчання видає дане повідомлення.
8. Перевірте виконання тригерів.

Лабораторна робота № 6

ЗБЕРЕЖЕНІ ПРОЦЕДУРИ.

Завдання до виконання:

1. Завантажте програму *IBExpert*.
2. Відкрийте базу даних **Univer**.
3. Створіть процедуру, яка визначає мінімальну стипендію групи. Виведіть результат у вихідний параметр **Min_stip**.
4. Створіть процедуру, що визначає прізвища студентів, які отримують максимальну стипендію в групі.
5. Створіть процедуру, яка індексує стипендію на 0,25. (Коефіцієнт індексації може змінюватись). Задайте коефіцієнт індексації як вхідний параметр.
6. Створіть процедуру, що виводить прізвища студентів, які отримують вказану стипендію. Задайте розмір стипендії в програмі як локальну змінну.
7. Створіть процедуру, яка обчислює сумарну кількість годин, прочитану кожним викладачем.
8. Створіть процедуру, яка нараховує стипендію студента за результатами сесії. Якщо середній бал рівний 5 , то стипендія =200, якщо більше або рівний 4 і менший 5, то стипендія =100. В іншому випадку стипендія рівна 0.
9. Створіть процедуру, яка знищує з таблиці **STUDENTS** запис по прізвищу *Іванчук*.

Лабораторна робота № 7

БЕЗПЕКА БАЗИ ДАНИХ. КОРИСТУВАЧІ, РОЛІ, ПРАВА.

Завдання до виконання:

1. Завантажте програму *IBExpert*.
2. Відкрийте базу даних **Univer**.
3. Використовуючи *User Manager*, створити користувача *ST* з паролем *ST*.
4. Використовуючи *Grant Manager*, надайте йому права на модифікацію полів **SEMESTR**, **GOD** таблиці **PREDMET**.
5. Закрийте базу даних **Univer**. Відкрийте її заново як користувач *ST*. Перевірте, чи можна коригувати поля **SEMESTR**, **GOD** таблиці **PREDMET**, інші поля даної таблиці.
6. Відкрийте базу даних **Univer** як користувач **SYSDBA**.
7. Створіть роль **READER**, що дозволяє виконувати запити до таблиці **STUDENTS**.
8. Відкрийте базу даних **Univer** заново як користувач *ST* з роллю **READER**. Переконайтесь у дії даної ролі, створивши простий запит до таблиці **STUDENTS**.
9. Відмініть права, надані користувачу *ST*, і роль **READER**.
10. Видаліть їх.
11. Самостійно створіть користувачів і ролі, надавши їм певні права. Переконайтесь у правильній роботі даних користувачів по відношенню до різних об'єктів бази даних.
12. Виконайте повторно п.п. 3-10 у командному режимі, використовуючи *SQL Editor*.

МОДУЛЬ 2 «ПРОГРАМУВАННЯ МОВОЮ PYTHON»

Лабораторна робота №1

ВИКОРИСТАННЯ БАЗОВИХ ТИПІВ ТА ЗАСОБІВ МОВИ ПРОГРАМУВАННЯ PYTHON. СЕРЕДОВИЩА ТА ІНСТРУМЕНТАРІЙ РОЗРОБЛЕННЯ ПРОГРАМ МОВОЮ PYTHON.

Мета роботи: Познайомитись з середовищами розробки Python і отримати головні навички розробки програмного забезпечення мовою Python.

Приклад 1. Базові типи даних (вказуються неявно)

```
a = 5 # int
a = 5 # int
b = 7.0 # float
c = 2 > 4 # boolean
d = "World" # string
e = 1.5 + 0.5j # complex
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
print(a, b, c, d, e.real, e.imag)
```

Приклад 2. Основні арифметичні операції

```
a = 5 # int
b = 7.0 # float
c = 1 + 2 # 3
d = 5 - 3 # 2
e = a * b # 35.0
f = 3.0 / 2 # 1.5
g = 3 / 2 # 1
h = 5 % 3 # 2
j = 10 ** 7.3 # 19952623.1497
print(type(a))
print(type(b))
print(type(c))
print(type(d))
print(type(e))
print(type(f))
```

```

print(type(g))
print(type(h))
print(type(j))
print(a, b, c, d, e, f, g, h, j)

```

Приклад 3. Вбудовані математичні функції (необхідне підключення бібліотеки math)

```

from math import *
a = 1
b = 2
x = sqrt (a*b) / (exp(a)*b) + a*exp( (2*a) / b)
print(x)

```

Завдання

В завданні 1 кожного варіанту необхідно обчислити значення виразу та вивести його на екран.

Варіант	Завдання
1.	$y = \frac{\sqrt{x-1} - \operatorname{tg}(x+1)}{\arccos x + \ln x} + 2,75$
2.	$g = \frac{\sin x^2 - \cos^4(x-1)^2}{\operatorname{arctg}(x+2,6) + \sqrt[3]{\ln x}}$
3.	$p = \frac{e^{-3x} + \operatorname{tg}(3x-3)}{ \sin x + \sqrt[4]{\cos x + \cos 2x}}$
4.	$y = x + \frac{\sqrt{\arcsin x + \operatorname{arctg} \pi x}}{\lg(13.4x + \pi)}$
5.	$y = \operatorname{arctg} x + \frac{e^{0,6x-1} - \sqrt{(x+6,1)^3}}{\ln x + \operatorname{tg}^2 x}$
6.	$u = 0,3 \cdot \lg e^{-x} + \frac{\operatorname{arctg} x - \sin^2 x}{4 \cdot \sqrt{\ln x-1 }}$
7.	$y = \sin \left(\frac{x + 2,3 \cdot \lg(x+1)}{\sqrt{2 \ln x + \cos x}} \right)$

8.	$y = \operatorname{tg} \left(\frac{x + 3 \cdot \lg(x+1)}{\sqrt{\ln 4x + \cos x}} \right)$
9.	$c = \operatorname{tg} x - \frac{\sqrt{\ln 2 + \ln x}}{\sqrt[3]{\operatorname{tg} x} + \sqrt[4]{\cos x^{-1}}}$
10.	$c = \operatorname{arctg} x - \frac{\sqrt{\ln 4 + \ln x}}{\sqrt[3]{\lg 2.4} + \sqrt[5]{\cos x^{-1}}}$

Лабораторна робота №2

УПРАВЛЯЮЧІ КОНСТРУКЦІЇ ТА МАСИВИ У МОВІ PYTHON

Мета роботи Навчитися створювати найпростіші програми на мові Python, використовуючи оператори вибору і циклів, арифметичні вирази.

Приклад 1. Використання умовного оператора

```
name = ''
while True :
    print ( ' Who are you? ' )
    name = input ( )
    if name != ' Joe ' :
        continue
    print ( ' Hello, Joe. What is the password? (It is a fish.) ' )
    password = input ( )
    if password == ' swordfish ' :
        break
print ( ' Access granted. ' )
```

Приклад 2. Використання циклу

```
while name = None # спочатку ми не знаємо імені користувача # нескінчений цикл
while True :
    print ( ' Меню: ' )
    print ( ' 1. Ввести ім'я ' )
    print ( ' 2. Вивести привітання ' )
    print ( ' 3. Вийти ' )
    response = input ( ' Виберіть пункт: ' )
    print ( )
    if response == ' 1 ' :
        name = input ( ' Введіть ваше ім'я: ' )
    elif response == ' 2 ' :
        if name: #вітаємося з користувачем, якщо ім'я вже введено
            print ( ' Привіт, ', name, '!', sep= '' )
        else :
            print ( ' Я не знаю вашого імені. ' )
    elif response == ' 3 ' :
        # оператор break завершує виконання циклу
        break #якщо користувач вибрав 3, то виходимо з циклу
    else :
        print ( ' Неправильне введення. ' )
```

print ()

Завдання

В завданні 1 кожного варіанту для його реалізації слід застосувати розгалуження та цикли.

Варіант	Завдання
1.	$y = \frac{\cos^2 x}{x^2 + 1}$, $3,8 \leq x \leq 7,6$, $\Delta x = 0,6$;
2.	$y = \frac{\operatorname{tg} 0,5x}{x^3 + 7,5}$, $0,1 \leq x \leq 1,2$, $\Delta x = 0,1$;
3.	$y = \frac{e^{2x} - 8}{x + 3}$, $-1 \leq x \leq 2,3$, $\Delta x = 0,7$;
4.	$y = \frac{x + \cos 2x}{3x}$, $2,3 \leq x \leq 5,4$, $\Delta x = 0,8$;
5.	$y = \frac{x + \cos 2x}{x + 2}$, $0,2 \leq x \leq 10$, $\Delta x = 0,8$;
6.	$y = \frac{\cos^3 t^2}{1,5t + 2}$, $2,3 \leq t \leq 7,2$, $\Delta t = 0,8$;
7.	$z = \frac{x^3 + 2x}{3 \cos \sqrt{x+1}}$, $0 \leq x \leq 2$, $\Delta x = 0,4$;
8.	$z = \frac{t + \sin 2t}{t^2 - 3}$, $2,4 \leq t \leq 6,9$, $\Delta t = 0,4$;
9.	$y = \frac{x^3 - 2}{3 \ln x}$, $4,5 \leq x \leq 16,4$, $\Delta x = 2,2$;
10.	$z = \frac{2,3t + 8}{2 \cos t + 1}$, $0 \leq t \leq 6,5$, $\Delta t = 1,1$;

Лабораторна робота №3

ОБРОБКА ПОСЛІДОВНОСТЕЙ ПРИ ПРОГРАМУВАННІ НА МОВІ PYTHON

Мета роботи: Ознайомитися з особливостями визначення та використання одновимірних та двовимірних масивів, структурною організацією масивів та способів доступу до їх елементів.

Приклад 1. Функція range

Параметр i приймає значення в діапазоні [0, 10)

```
for i in range ( 10 ) :  
    print ( ' i = ', i )
```

Параметр i приймає значення в діапазоні [5, 10)

```
for i in range ( 5, 10 ) :  
    print ( ' i = ', i )
```

Параметр i приймає значення в діапазоні [5, 10) з кроком 2

```
for i in range ( 5, 10, 2 ) :  
    print ( ' i = ', i )
```

Цикл буде повторюватися 3 рази, якщо користувач не завершить його раніше

```
for i in range ( 3 ) :  
    response = input ( ' Введіть stop, щоб зупинити цикл ( інакше що завгодно ) : ' )  
    if response == ' stop ' :  
        break  
    else :  
        # цю гілку виконують тільки якщо цикл не був перерваний  
        print ( ' Цикл сам був завершений ' )  
print ( ' Кінець програми ' )
```

Функція reversed дозволяє обходити послідовність в зворотному напрямку

```
for i in reversed ( range ( 5 ) ) :  
    print ( i )
```

Приклад 2. Зріз списку – отримання групи елементів за їхніми індексами

Створення списку чисел
my_list = [5, 7, 9, 1, 1, 2]

```

# Отримання передостаннього значення
pre_last = my_list [ -2 ] # pre_last == «1»
print ( pre_last )
# Обчислення суми першого і останнього значень
result = my_list [ 0 ] + my_list [ -1 ] ; print ( result )
my_list = [ 5, 7, 9, 1, 1, 2 ]
# Створення списку чисел
# Отримання зрізу списку від нульового (першого) елемента
# (включаючи його) до третього (четвертого) (не включаючи)
sub_list = my_list [ 0 : 3 ]
print ( sub_list )
# Виведення отриманого списку
# Виведення елементів списку від другого до передостаннього
print ( my_list [ 2 : -2 ] )
# Виведення ел-тів списку від 4-ого (5-ого) до 5-ого (6-ого)
print ( my_list [ 4 : 5 ] )

my_list = [ 5, 7, 9, 1, 1, 2 ]
# Вибір кожного другого ел-та списку (починаючи з першого),
# не включаючи останній елемент
sub_list = my_list [ 0 : -1 : 2 ]
print ( sub_list )
# виведення отриманого списку
# Виведення елементів від 2-ого (3-ього) до передостаннього
# з кроком 2
print ( my_list [ 2 : -2 : 2 ] )
# Виведення ел-тів списку, крім першого, в зворотному порядку
print ( my_list [ -1 : 0 : -1 ] )

my_list = [ 5, 7, 9, 1, 1, 2 ]
# Виведення елементів списку від 2-ого (3-ього) значення до кінця
print ( my_list [ 2 : ] )
# Виведення всіх ел-тів списку від початку до передостаннього ел-та
print ( my_list [ : -2 ] )
# Виведення всіх елементів списку в зворотному порядку
print ( my_list [ : : -1 ] )

```

Приклад 3. Нехай згенеровано список із цілих випадкових чисел та нулів [a1, ..., an]. Написати програму визначення елементів, розміщених після першого нульового. Вивести на екран початковий та отриманий списки

```
import random
```

```

arr = random.sample ( range ( -6, 6 ), 12 )
print ( " our random list : ", arr )
flag = 0
while flag == 0 :
    if 0 in arr : # перевіряємо чи є 0 в списку
        first_zero_index = arr.index ( 0 ) # індекс першого 0
        flag = 1 ;
    else :
        print ( " we have not at list one zero in list: " )
arr1 = [ ]
if flag == 1:
    for i in arr [ first_zero_index : ] :
        arr1.append ( i )
print ( arr1 )

```

Результат:

```

our random list: [ -4, -6, 4, 0, 2, 5, 3, -1, -3, -5, 1, -2]
[0, 2, 5, 3, -1, -3, -5, 1, -2 ]

```

Завдання 1.

Одновимірні масиви (вектори)

Нехай задано список різних випадкових чисел $[a_1, \dots, a_n]$, значення n визначає користувач програми. Використовуючи генератор випадкових чисел, заповнити список $[a_1, \dots, a_n]$ елементами:

- а) дійсними числами, які лежать в діапазоні від 0 до 1;
- б) цілими додатними та від'ємними числами, які лежать в діапазоні від -10 до 10 включно;
- в) цілими додатними числами, які лежать в діапазоні від 0 до 50 включно.

Варіант:

1. Задано список (б). Написати програму формування іншого списку, в якому усі елементи, які передують найбільшому від'ємному елементу, замінити на значення їх квадратів.

2. Задано список (б). Написати програму формування іншого списку, в якому, якщо елементи заданого списку не утворюють послідовності, яка зменшується, то замінити його від'ємні елементи одиницями.
3. Задано список (б). Написати програму формування іншого списку, в якому переставити елементи таким чином, щоб спочатку були розташовані всі невід'ємні елементи, а вкінці - від'ємні елементи.
4. Задано список (а). Написати програму формування іншого списку, в якому елементи сформовані таким чином $[a_1, a_{n+1}, a_2, a_{n+2}, \dots, a_n, a_{2n}]$.
5. Задано список (б). Перевірити чи утворюють елементи заданого масиву послідовність, яка чітко зменшується або збільшується. Вивести відповідне повідомлення.
6. Задано список (а). За заданими дійсними числами a_0, a_1, \dots, a_n, t обчислити значення багаточлена $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ в точці t .
7. Задано список (б). Написати програму визначення суми всіх елементів, розміщених до останнього додатного елемента включно.
8. Задано список (в). Написати програму визначення суми лише тих елементів, які є непарними числами.
9. Задано список (б). Написати програму визначення добутку елементів, розміщених між максимальним за модулем та мінімальним за модулем елементами.
10. Задано список (б). Написати програму формування іншого списку, в якому елементи сформовані таким чином, що нульові елементи перенесено у хвіст списку.

Завдання 2.

Двовимірні масиви (матриці) Використовуючи генератор випадкових чисел, заповнити список $[[a_{11}, \dots, a_{1n}], \dots, [a_{m1}, \dots, a_{mn}]]$, де n – кількість літер в імені, m – кількість літер в прізвищі, елементами:

- а) дійсними числами, які лежать в діапазоні від 0 до 1;
- б) цілими додатними та від'ємними числами, які лежать в діапазоні від -10 до 10 включно;

в) цілими додатними числами, які лежать в діапазоні від 0 до 20 включно;

г) дійсними числами, які лежать в діапазоні від -10 до 10 ; Елементи головної діагоналі розташовані з лівого верхнього кута матриці до правого нижнього; друга, зворотна діагональ матриці є побічною.

1. Задано список (б). Написати програму, яка змінить місцями два стовпчики: стовпчик, який містить максимальний від'ємний елемент, і стовпчик, який містить мінімальний додатний елемент матриці.

2. Задано список (в). Написати програму, яка змінить місцями перший рядок з рядком, що містить максимальний елемент матриці.

3. Задано список (б). Написати програму, яка змінить місцями останній рядок з рядком, який містить мінімальний додатний елемент матриці.

4. Задано список (б), $n=m$. Написати програму, яка визначить, чи є задана квадратна матриця симетричною відносно головної діагоналі.

5. Згенерувати список, $n=m$, який визначає квадратну матрицю – магічний квадрат (тобто така, в якій суми елементів у всіх рядках і стовпчиках є однаковими). Вивести на екран отриманий результат.

6. Задано список (б) (якщо треба згенерувати відповідний список). Написати програму, яка визначить добуток від'ємних елементів другого рядка та кількість елементів в другому стовпчику, які не кратні «5».

7. Задано список (б). Написати програму, яка визначить новий список, в якому кожен елемент матриці помножений на мінімальний за абсолютною величиною елемент у поточному стовпчику.

8. Задано список (г). Написати програму, яка змінить місцями останній стовпчик і стовпчик, який містить мінімальний додатний елемент матриці.

9. Задано список (б). Написати програму, яка визначити максимальний елемент третього стовпчика та суму непарних елементів першого рядка.

10. Задано список (г). Написати програму, яка визначить рядок, сума елементів якого мінімальна. Вивести на екран початкову матрицю та визначений рядок.

Лабораторна робота №4

РОЗРОБКА ПРОГРАМ З ВИКОРИСТАННЯМ ПРОЦЕДУР І ФУНКЦІЙ

Мета роботи Познайомитися з принципами побудови функцій користувача на мові Python, з використанням локальних і глобальних змінних.

Приклад 1. Локальна змінна

```
def function ( ) :  
    # визначення локальної змінної  
    var = ' локальна змінна '  
    # виведення значення локальної змінної на екран  
    print ( var )  
  
# визначення глобальної змінної  
var = ' глобальна змінна '  
function ( )  
# виведення на екран значення глобальної змінної  
print ( var )
```

Результат

локальна змінна

глобальна змінна

Службове слово *global* вказує на необхідність отримати доступ до глобальної змінної *var*, а не створювати нову локальну під час спроби щонебудь їй присвоїти:

```
def function():  
    global var  
    # виведення значення глобальної змінної на екран  
    print ( var )  
    # зміна глобальної змінної  
    var = ' нове значення '  
    # виведення значення глобальної змінної на екран  
    print ( var )  
var = ' глобальна змінна '  
function ( ) ; print ( var )
```

Результат

глобальна змінна

нове значення

```
def function ( c, d ):
# a, b – глобальні змінні; c, d -- локальні
global a, b
# зміна значення глобальної змінної
a = 5 ; =
b = 7
# зміна значення локальної змінної
c = 10
d = 12
a, b, c, d = 1, 2, 3, 4 # множинне присвоєння
print ( a, b, c, d ) # 1 2 3 4
function ( c, d )
print ( a, b, c, d ) # 5 7 3 4
```

Результат

1 2 3 4

5 7 3 4

Приклад 2. Описати функцію, яка обчислює множину значень $y = \sqrt[n]{x}$ ($x > 0, n > 0$), для цього використати рекурентну формулу Ньютона:

$$y_{k+1} = \frac{k-1}{k} y_k + \frac{x}{k \cdot y_k^{k-1}}, \quad k=0, 1, \dots, \quad y_0 = \frac{x+n-1}{2}$$
 яка має місце для $y_0 > 0$

Необхідну точність оцінюють співвідношенням $|y_{n+1} - y_n| \leq \varepsilon$.

```
import itertools
```

```
import math
```

```
def my_sqrt ( a, n, Epsilon ) :
    term1 = ( a + n-1 ) / 2
    term2 = ( 2 / 3 ) * term1 + ( a / ( 3 * term1 ** 2 ) )
    k = 1
    while ( abs ( term2 - term1 ) > Epsilon ) :
        term1 = term2 ; term2 = ( ( k-1 ) / k ) * term1 + a / ( k * ( term1 ** ( k-1 ) ) )
        k+=1
```

```
return term2
```

```
print ( " x ", " : ", " y ", " : ", " yt", " : ", " error " )  
print ( " ----- " )  
Epsilon = 0.0001  
a = 0.1 ; b = 1  
for i in itertools.count ( start = a, step = 0.2 ):  
    if i > b : break  
    n = 3  
    y = my_sqrt ( i, n, Epsilon )  
    y1 = i ** ( 1 / n ) # точне значення функції  
    error = abs ( y-y1 )  
    print ( ' %.2f ' %i, " : ", ' %.4f ' %y, " : ", ' %.4f ' %y1, " : ", ' %.4f ' %error )
```

Завдання 1

Розробіть функції для здійснення наступних Варіант:

1. Пошук елементів в словнику з значенням ключа.
2. Пошук елементу за значенням в списку.
3. Пошук послідовності елементів в списку.
4. Пошук перших п'яти мінімальних елементів в списку.
5. Пошук перших п'яти максимальних елементів в списку.
6. Пошук середнього арифметичного в списку.
7. Пошук усіх нульових елементів в списку.
8. Повернення списку, що сформований з початкового списку, але не містить повторів (залишається лише перший з однакових елементів).
9. Пошук послідовності елементів в рядку.
10. Розбиття речення на окремі слова (повертається список слів).

Завдання 2

Описати функцію обчислення значень функції (визначити m значень заданої функції $f(x)$ на відрізку $[a, b]$). Результат вивести на екран у вигляді табл. Значення a, b, ϵ ввести з клавіатури.

x_i	$f(x_i)$ точне значення	$f_{\text{набл}}(x_i)$ наближене значення	ϵ Точність	Кількість ітерацій
-------	-------------------------	---	---------------------	--------------------

$x_0 = a$				
$x_{m-1} = b$				

Стовпчики таблиці: 1 – значення x_i ; 2 – значення функції $f(x_i)$, яке обчислено з використанням функцій інтерпретатора (модуль *math*); 3 – значення функції, яке обчислено за допомогою розкладання в ряд із точністю ε ; 4 – точність обчислень; 5 – кількість ітерацій, необхідних для досягнення заданої точності.

Кількість точок на відрізку $[a, b]$ складає не менше 10 (розбиття може бути рівномірним). Необхідну точність оцінюють співвідношенням $|y_{n+1} - y_n| \leq \varepsilon$.

Використати для побудови розкладання елементарних функцій в ряд Тейлора.

1. $f(x) = e^{2x}$
2. $f(x) = e^{(1+x)}$
3. $f(x) = \frac{\sqrt{x}}{1-x^2}$
4. $f(x) = \frac{1}{\sqrt{1+x}}$
5. $f(x) = e^{4x}$
6. $f(x) = \frac{1}{1-x}$
7. $f(x) = \frac{1}{1+x}$
8. $f(x) = \ln(1+x)$
9. $f(x) = \ln(1-x)$
10. $f(x) = e^x$

Лабораторна робота №5

РОЗРОБКА ПРОГРАМ З ВИКОРИСТАННЯМ КЛАСІВ В PYTHON

Приклад 1. Програма з використанням класу. Обчислення середнього бала студента з трьох предметів.

```
class Student :

    def set_marks ( self, e1, e2, e3 ) :
        self.e1 = e1
        self.e2 = e2
        self.e3 = e3
    def set_name ( self, name ) :
        self.name = name

    def get_average_mark ( self ) :
        print ( self.name, ' - ', ( ( self.e1 + self.e2 + self.e3 ) / 3 ) )

s1 = Student ( )
s2 = Student ( )
s1.set_name ( ' Dmytro ' )
s1.set_marks ( 5, 4, 5 )
s1.get_average_mark ( )

s2.set_name ( ' Mykola ' )
s2.set_marks ( 4, 4, 4 )
s2.get_average_mark ( )
```

Завдання

1. Розробити клас "домашня бібліотека". Додати конструктор, який приймає ціле число – порядковий номер книги та словник, що містить інформацію про книгу наступному форматі: 1) автор; 2) назва; 3) видавництво; 4) жанр; 5) рік видання.

Реалізувати можливість роботи з довільним числом книг, пошуку по книгах за декількома параметрами (за автором, за роком видання, за жанром тощо), додавання книг у бібліотеку, видалення книг з неї, доступу до книги

за номером. Написати програму, що буде демонструвати всі розроблені елементи класу.

2. Розробити клас для представлення відомостей про успішність студента. Об'єкт класу має містити поля для збереження імені студента та балів, отриманих ним за виконання лабораторних робіт та індивідуального творчого завдання.

Забезпечити наступні методи класу: конструктор, який приймає рядок ім'я студента та словник, що містить налаштування курсу у наступному форматі: 1) максимально можлива кількість балів за задачу індивідуального творчого завдання; 2) максимально можлива кількість балів за задачу однієї лабораторної роботи; 3) кількість лабораторних робіт в курсі; метод, за допомогою якого вносяться оцінки за лабораторну роботу, який приймає параметри оцінку та номер лабораторної роботи; метод, за допомогою якого вносяться дані про оцінку за індивідуальне творче завдання; метод, який повертає дійсне число (суму балів студента за проходження курсу).

3. Розробити клас "інтернет-замовлення". Додати конструктор, який приймає ціле число – порядковий номер замовлення та словник, що містить інформацію про замовлення наступному форматі: 1) прізвище клієнта; 2) назва товару; 3) кількість; 4) вартість; 5) дата замовлення.

Реалізувати можливість роботи із замовленнями: пошук по декількома параметрами (за прізвищем замовника, за датою замовлення, за товаром тощо), додавання нових замовлень, видалення інформації про замовлення, доступу до інформації про замовлення за номером. Написати програму, що буде демонструвати всі розроблені елементи класу.

4. Розробити клас "Працівник". Додати конструктор, який приймає ціле рядок – прізвище та ім'я працівника та словник, що містить інформацію про працівника наступному форматі: 1) назва віділу; 2) посада; 3) рік народження; 4) стаж роботи.

Реалізувати можливість роботи із замовленнями: пошук по декількома параметрами (за прізвищем, за віком, за відділом тощо),

додавання нових працівників, видалення інформації про працівника, доступу до інформації про замовлення за прізвищем та ім'я Написати програму, що буде демонструвати всі розроблені елементи класу.

5. Розробити клас для представлення відомостей про успішність студента. Об'єкт класу має містити поля для збереження імені студента та балів, отриманих ним за виконання лабораторних робіт та лекційних занять.

Забезпечити наступні методи класу: конструктор, який приймає рядок ім'я студента та словник, що містить налаштування курсу у наступному форматі: 1) кількість лекційних занять в курсі; 2) максимально можлива кількість балів за здачу однієї лабораторної роботи; 3) кількість лабораторних робіт в курсі; метод, за допомогою якого вносяться оцінки за лабораторну роботу, який приймає параметри оцінку та номер лабораторної роботи; метод, за допомогою якого вносяться дані про кількість відвіданих лекцій; метод, який повертає дійсне число (суму балів студента за проходження курсу).

Лабораторна робота №6

РОБОТА З БАЗОЮ ДАНИХ ІЗ PYTHON-ПРОГРАМИ

Приклад 1. Збереження даних в базі даних

```
import sqlite3

# Клас для що описує працівників підприємства
class Worker ( ) :
    def __init__( self, **kwargs ) :
        self.name = kwargs.get ( ' name ' )
        self.position = kwargs.get ( ' position ' )
        self.work_from_date = kwargs.get ( ' work_from_date ' )
        self.birth_date = kwargs.get ( ' birth_date ' )

    def get_data ( self ) :
        return self.name, self.position,
self.work_from_date, self.birth_date

# створення екземпляра класу працівник з використанням непозиційних
параметрів у конструкторі
w1 = Worker ( name = ' Petrov Vadym ', position = ' manager ',
work_from_date = ' 09-12-2019 ', birth_date = ' 18-02-1990 ' )

conn = sqlite3.connect ( " staff_db.db " )
cursor = conn.cursor ( )

# Створення таблиці після першого запуску скрипта потрібно закоментувати
cursor.execute ( " " " CREATE TABLE staff ( name text, position text, work_from_date
text, birth_date text) " " " )

# Створення першого запису в таблиці "вручну"
cursor.execute ( " " " INSERT INTO staff VALUES ( ' Ivanov Ivan ', ' director ', ' 07-10-
2019 ', ' 2-12-1980 ' ) " " " )

# Створення першого запису в таблиці із екземпляра класу Worker
cursor.execute ( " " " INSERT INTO staff VALUES ( ?, ?, ?, ? ) " " " , w1.get_data ( ) )

# Зберігаємо зміни
conn.commit ( )

# Вставляємо список з трьох працівників
all_workers = [ ( ' Borysov Mykola ', ' ingeneer ', ' 23-11-1976 ', ' 04-12-2019 ' ),
( ' Pavlyik Inna ', ' secretary ', ' 12-09-1991 ', ' 22-01-2020 ' ),
```

```
        ( ' Kolodych Leonid ', ' ingeneer ', ' 16-08-1986 ', ' 13-01-2020 ' ) ]
cursor.executemany ( " INSERT INTO staff VALUES (?, ?, ?, ? ) ",
all_workers )
conn.commit ( )
```

```
# Виведення на екран всіх записів
```

```
print ( " Записи в таблиці бази даних у вигляді списку : " )
sql = " SELECT * FROM staff "
cursor.execute ( sql )
print ( cursor.fetchall ( ) )
```

```
# Редагування запису для конкретного працівника
```

```
sql = " " "
UPDATE staff
SET position = ' main ingeneer '
WHERE name = ' Kolodych Leonid '
" " " "
```

```
cursor.execute ( sql )
conn.commit ( )
```

```
# Виводимо список всіх інженерів
```

```
print ( " Список всіх ingeneer : " )
```

```
sql = " SELECT * FROM staff WHERE position = ? "
cursor.execute ( sql, [ ( " ingeneer " ) ] )
print ( cursor.fetchall ( ) )
```

```
# Виведення на екран всіх записів
```

```
print ( " Список всіх записів в таблиці : " )
for row in cursor.execute ( " SELECT rowid, * FROM staff ORDER BY name " ) :
    print ( row )
```

Завдання

Доповнити розроблену в лабораторній роботі 5 програму збереженням даних в базі даних. Додати до створеного класу методи для запису даних у базу даних, видалення записів із бази, пошуку записів по заданому критерію.

Лабораторна робота №7

ПОБУДОВА ГРАФІКІВ МАТЕМАТИЧНИХ ФУНКЦІЙ У МОВІ PYTHON

Мета роботи Набуття навичок роботи з бібліотекою Matplotlib для візуалізації даних

Для початку необхідно інсталиювати пакети numpy і matplotlib. Для цього зручно використовувати pip – це інстальатор пакетів для мови Python.

Установка pip

Завантажуємо Пітон скрипт get-pip.py (<https://bootstrap.pypa.io/get-pip.py>). При зберіганні переконайтесь, що зберегли файл із розширенням .py, а не .txt.

У командній стрічці (краще PowerShell) запустить даний файл з допомогою Python інтерпретатора:

```
python \path\to\get-pip.py
```

В даному випадку файл get-pip.py лежить в директорії D:/work.

Після запуску даної команди маєте отримати повідомлення про успішну інсталяцію pip пакет менеджера. Інакше – потрібно розбиратись: або шлях до файла вказано не правильно, або Python не у видимих для командної стрічки шляхах.

Для остаточного тесту, що **pip** встановлено правильно просто запускаєте в командній стрічці (PowerShell) команду pip. Без аргументів дана команда видасть вам документацію по використанню.

Тепер маючи Python і Pip черга за numpy і matplotlib:

```
pip install numpy
```

```
pip install matplotlib
```

```
pip install numpy
```

Приклад 1. Набір точок

```
import matplotlib.pyplot as plt
```

```
plt.plot ( [ 1, 3, 2, 4 ] )  
plt.show ( )
```

Функція `plot()` будує графік, а функція `show()` його показує. Аргумент, що приймається функцією `plot()` – це послідовність узначень. Інший, який ми опустили, що стоїть перед `y` – це послідовність `x`-значень. Оскільки його немає, графік генерується для чотирьох зазначених `y`, список з чотирьох `x`: `[0, 1, 2, 3]`.

Приклад 2. Функція

```
from numpy import * # для використання функцій exp та linspace  
import matplotlib.pyplot as plt
```

```
def f ( t ) :  
    return t ** 2 * exp ( -t ** 2 )
```

```
t = linspace ( 0, 3, 51 ) # 51 точка між 0 та 3  
y = f ( t )  
plt.plot ( t, y )  
plt.show ( )
```

Приклад 3. Налаштування вигляду графіків

```
import matplotlib.pyplot as plt
```

```
t = linspace ( 0, 3, 51 )  
y = t ** 2 * exp ( -t ** 2 )  
plt.plot ( t, y, 'g--', label = ' t ^ 2 * exp ( - t ^ 2 ) ' )  
plt.axis ( [ 0, 3, -0.05, 0.5 ] ) # [ xmin, xmax, ymin, ymax ]  
plt.xlabel ( ' t ' ) # позначення вісі абсцис  
plt.ylabel ( ' y ' ) # позначення е вісі ординат  
plt.title ( ' My first normal plot ' ) # назва графіка  
plt.legend ( ) # вставка легенди (тексту в label)
```

```
plt.show ( )
```

Приклад 4. Декілька кривих на одному графіку

```
from numpy import *  
import matplotlib.pyplot as plt
```

```

t = linspace ( 0, 3, 51 )
y1 = t ** 2 * exp ( -t ** 2 )
y2 = t ** 4 * exp ( -t ** 2 )
y3 = t ** 6 * exp ( -t ** 2 )
plt.plot ( t, y1, 'g ^', # маркери із зелених трикутників
           t, y2, 'b--', # синя штрихова
           t, y3, 'ro-' ) # червоні круглі маркери
# з'єднані суцільною лінією
plt.xlabel ( ' t ' )
plt.ylabel ( ' y ' )
plt.title ( ' Plotting with markers ' )
plt.legend ( [ ' t ^ 2 * exp ( -t ^ 2 ) ',
              ' t ^ 4 * exp ( -t ^ 2 ) ',
              ' t ^ 6 * exp ( -t ^ 2 ) ' ], # список легенди
            loc = ' upper left ' ) # положення легенди
plt.show ( )

```

Завдання

Зобразити 2d графік функції відповідно своєму варіанту та зберегти у .png файл.

1. $Y(x)=x \cdot \sin(5 \cdot x)$, $x=[-2 \dots 5]$
2. $Y(x)=1/x \cdot \sin(5 \cdot x)$, $x=[-5 \dots 5]$
3. $Y(x)=2^x \cdot \sin(10x)$, $x=[-3 \dots 3]$
4. $Y(x)=x^{(1/2)} \cdot \sin(10 \cdot x)$, $x=[0 \dots 5]$
5. $Y(x)=15 \cdot \sin(10 \cdot x) \cdot \cos(3 \cdot x)$, $x=[-3 \dots 3]$
6. $Y(x)=5 \cdot \sin(10 \cdot x) \cdot \sin(3 \cdot x)$, $x=[0 \dots 4]$
7. $Y(x)=\sin(10 \cdot x) \cdot \sin(3 \cdot x)/(x^2)$, $x=[0 \dots 4]$
8. $Y(x)=5 \cdot \sin(10 \cdot x) \cdot \sin(3 \cdot x)/(x^{(1/2)})$, $x=[1 \dots 7]$
9. $Y(x)=5 \cdot \cos(10 \cdot x) \cdot \sin(3 \cdot x)/(x^{(1/2)})$, $x=[0 \dots 5]$
10. $Y(x)=-5 \cdot \cos(10 \cdot x) \cdot \sin(3 \cdot x)/(x^{(1/2)})$, $x=[0 \dots 10]$

Навчальне видання

Замуруєва Оксана Валеріївна

Сахнюк Василь Євгенович

Федосов Сергій Анатолійович

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА СИСТЕМИ

Курс лекцій

Друкується в авторській редакції