Лабораторна робота №3

Програмування алгоритмів розгалуженої структури. Програма для розв'язування алгебраїчних нерівностей.

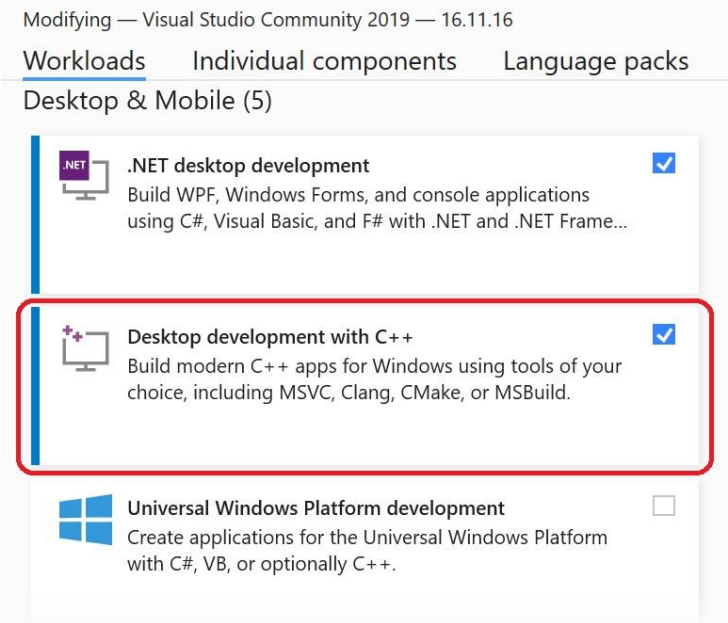
***Мета роботи**: Отримання практичних навичок програмування алгоритмів розгалуженої структури.*

Короткі теоретичні відомості

Розглянемо можливість використання до розробки на ***Python*** інтегрованого середовища ***Microsoft Visual Studio*** від компанії ***Microsoft***. Застосування цієї IDE найбільш популярне серед розробників н ***C++*** та C#(.net). Для професійного (комерційного) використання цього програмного продукту необхідно придбати ліцензію, але для навчальних цілей Microsoft розповсюджує «полегшену» версію – ***Community Edition***. Для використання Microsoft Visual Studio Community Edition достатньо мати лише ***обліковий запис Microsoft***, його можна зареєструвати безкоштовно, або скористатися раніше створеним, наприклад, корпоративним Microsoft-профілем, що надають студентам навчальні заклади.

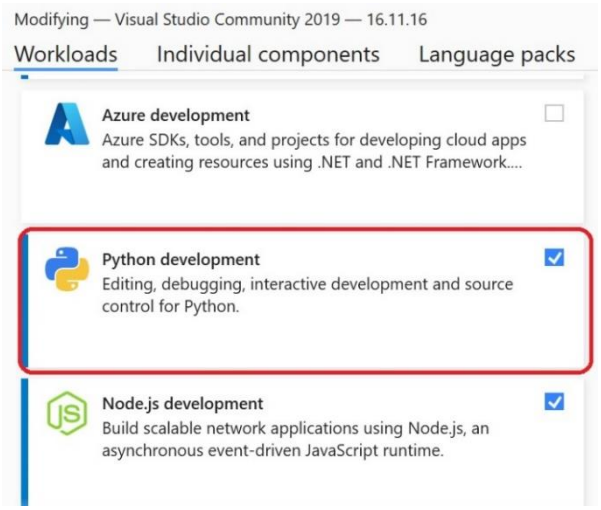
В поточному моменті корпорація Microsoft надає дві версій цього програмного продукту ***MS Visual Studio 2019*** та ***MS Visual Studio 2022***. Поданий далі огляд зроблено на основі ***MS Visual Studio 2019.***

MS Visual Studio підтримує розробку програм декількома мовами, в тому числі ***C++***, ***C#*** та ***Python***. Керування засобами для розробки, а також оновленнями самого програмного продукту здійснюється за допомогою спеціальної програми ***Visual Studio Installer***, що встановлюється разом із ***Visual Studio***. Ця програма дозволяє встановлювати так звані «робочі навантаження», що містять інструменти для використання тієї, чи іншої технології. Наприклад, для того, щоб працювати з класичними проєктами на C++ потрібно встановити Desktop development with C++.

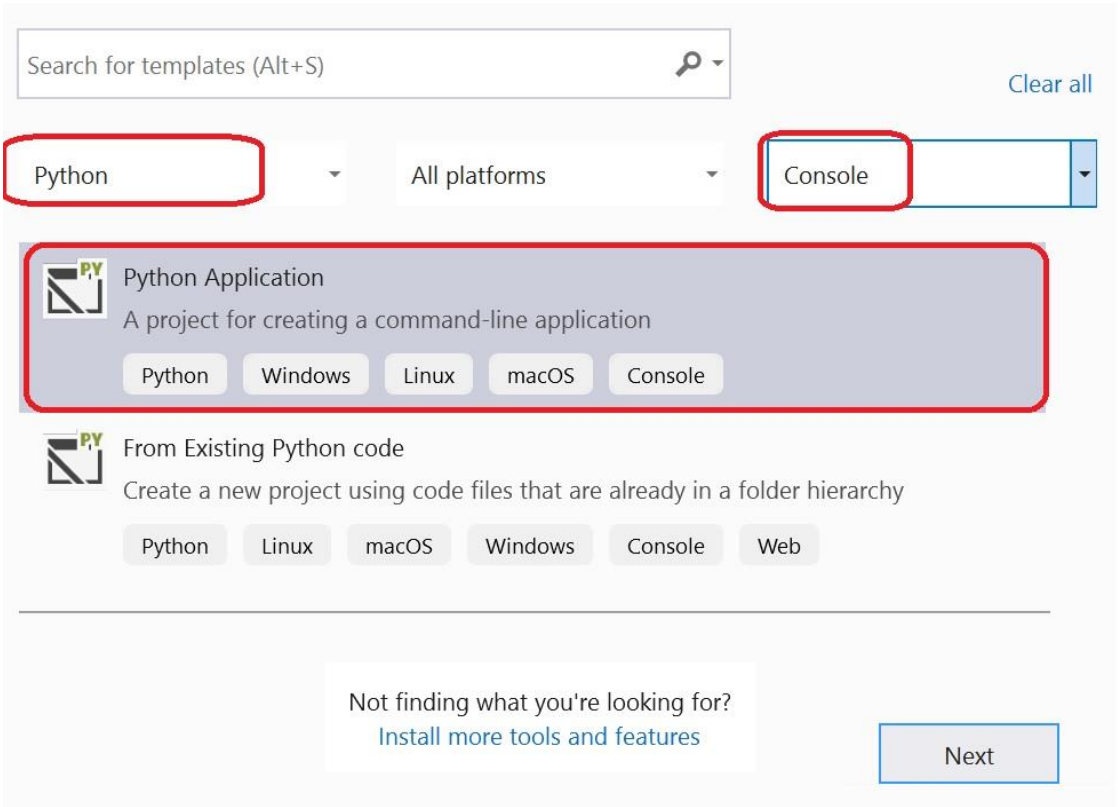


Visual Studio структурує код у вигляді проектів. ***Проєкт*** це, по суті, окрема програма, яку можна виконати. Проєкт може складатися з декількох файлів з кодом. MS Visual Studio зберігає проєкти в так званих «рішеннях», рішення може містити один або декілька проєктів.

Мова ***Python*** не належить мов платформи ***.Net***, як ***C++***. Тому для виконання програм на ***Python*** у ***Visual Studio*** потрібно додати інтерпретатор цієї мови. Найновіші версії інтерпретатора ***Python*** можна завантажити з офіційного сайту та встановити на ПК окремо, проте поточна версія ***Visual Studio*** може і не підтримувати роботу з найновішою версією ***Python***. Кращий підхід у цьому випадку – встановлення інтерпретатора з ***Visual Studio***.



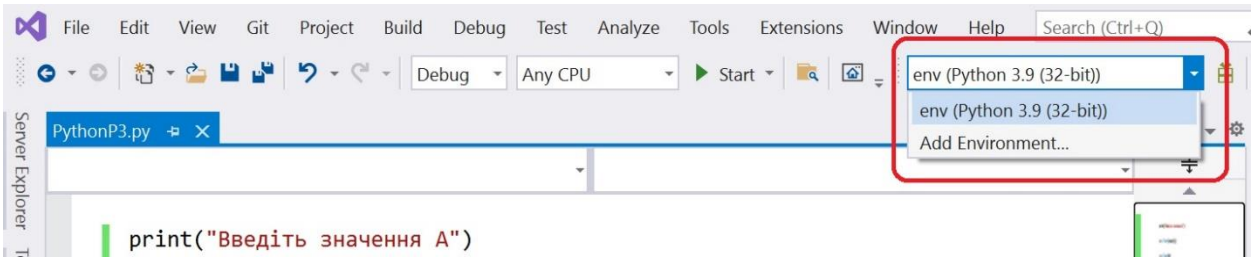
Для того, щоб мати можливість створювати проєкти на Python у Visual Studio потрібно встановити відповідне робоче навантаження за допомогою Visual Studio Installer.



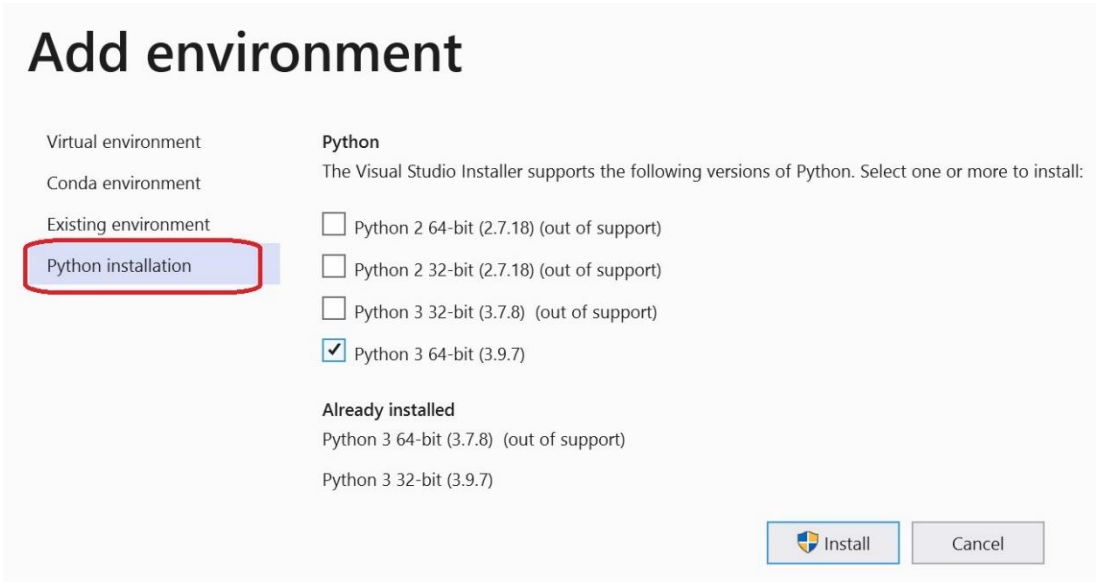
Для створення проєкту потрібно:

- *запустити Visual Studio та створити проєкт через меню File=> New=> Project, або Create a new project на стартовому екрані програми;*

- *задати для пошуку мову Python та тип проєкту Console;*
- *обрати тип Python Application та створити проєкт;*
- *після створення нового проекту на Python потрібно знайти на панелі інструментів Visual Studio список середовищ виконання і обрати Add Environment;*



- *для додавання нового інтерпретатора у вікні Add Environment переходимо до пункту Python installation;*
- *тут вже можна обирати останню версію, – у цьому списку присутні лише версії Python, сумісні з нашою версією Visual Studio. Далі залишилося почекати поки Visual Studio Installer завантажить потрібні дистрибутиви і встановить їх.*



Завдання.

Скласти програму для розв'язування алгебраїчної нерівності (системи нерівностей) згідно варіанту. Програма повинна для заданих значень числових коефіцієнтів a, b та c визначати розв'язок у вигляді проміжків на числовій осі. Виконати програму для різних значень коефіцієнтів a, b і c так, щоб продемонструвати її роботу за усіма можливими вітками розгалуження. У звіт подати:

- умову задачі варіанта;
- словесний опис алгоритму розв'язування задачі;
- код програми на Python;
- результат виконання програми покожній вітці розгалуження для відповідного набору вхідних даних.

Варіанти завдань

№ вар.	Завдання
1	$\frac{a}{x^2+bx+c} \le 0$
2	$ax^2 - bx > c$
3	$\frac{x^2}{ax+b} < c$
4	$\frac{x-a}{x^2+bx+c} \ge 0$
5	$\frac{x^2-ax+b}{x+c} \le 0$
6	$(x-a)(x^2+bx+c) < 0$
7	$\begin{cases} x-a<0, \\ x^2+bx+c>0 \end{cases}$
8	$\begin{cases} x-a\ge 0, \\ x^2+bx+c<0 \end{cases}$

№ вар.	Завдання
9	$\frac{a}{x^2+bx+c} > 0$
10	$ax^2 - bx \le c$
11	$\frac{x^2}{ax+b} \ge c$
12	$\frac{x-a}{x^2+bx+c} < 0$
13	$\frac{x^2-ax+b}{x+c} > 0$
14	$(x+a)(x^2-bx+c) > 0$
15	$\begin{cases} x+a>0, \\ x^2+bx+c\le 0 \end{cases}$
16	$\begin{cases} x-a\ge 0, \\ x^2+bx+c>0 \end{cases}$

Приклад виконання завдання

Варіант № **16.** Скласти програму для розв'язування системи нерівностей.

$$\begin{cases} x - a \geq 0, \\ x^2 + bx + c < 0. \end{cases}$$

Програма повинна для заданих значень числових коефіцієнтів a, b та c визначати розв'язок у вигляді проміжків на числовій осі. Виконати програму для різних значень коефіцієнтів a, b і c так, щоб продемонструвати її роботу за усіма можливими вітками розгалуження.

Головною запорукою правильної роботи цієї програми є ґрунтовний аналіз можливих варіантів процесу розв'язування нерівності, залежно від значень коефіцієнтів a, b та c.

Перша нерівність має розв'язок $x \geq a$, тобто проміжок $[a; +\infty)$. Розв'язок системи залежить від того яким є розв'язок другої нерівності, та як відносно цього розв'язку розташоване на числовій осі число a.

Тобто програма спочатку повинна обчислити значення дискримінанта D квадратного тричлена в лівій частині другої нерівності та встановити наявність-відсутність його коренів.

- *Якщо дискримінант від'ємний, то квадратний тричлен не має коренів, а оскільки коефіцієнт при x^2 додатний, то друга нерівність матиме розв'язком усю множину дійсних чисел R. Отже розв'язком системи буде проміжок* $[a; +\infty)$.

При D=0 квадратний тричлен в другій нерівності має коренем число $x_0 = \frac{-b}{2}$, а сама нерівність – розв'язок $(-\infty; x_0) \cup (x_0; +\infty)$. Тоді розв'язок системи залежить від взаємного розташування на числовій прямій чисел a та x_0:

- *при $x_0 < a$ розв'язком системи залишиться проміжок* $[a; +\infty)$*;*
- *при $x_0 = a$ число a не є розв'язком системи і результатом буде* $x \in (a; +\infty)$ *(те ж що і* $x \in (x_0; +\infty)$*);*
- *при $x_0 > a$ число x_0 слід виключити з проміжка* $[a; +\infty)$*, отримаємо результат* $x \in [a; x_0) \cup (x_0; +\infty)$

При $D > 0$ знайдемо корені $x_1 = \frac{-b-\sqrt{D}}{2}$ та $x_2 = \frac{-b+\sqrt{D}}{2}$ квадратного тричлена в другій нерівності (очевидно, що $x_1 < x_2$). Друга нерівність матиме розв'язок $(-\infty; x_1) \cup (x_2; +\infty)$ а для розв'язку системи матимемо такі випадки:

- *при $a < x_1$ розв'язок системи $x \in [a; x_1) \cup (x_2; +\infty)$;*
- *при $x_1 \leq a \leq x_2$ розв'язок системи $x \in (x_2; +\infty)$;*
- *при $a > x_2$ розв'язком системи залишиться проміжок* $[a; +\infty)$.

Враховуючи велику кількість розгалужень в отриманому алгоритмі обійдемося без складання блок-схеми. Взявши за основу наведений вище словесний опис алгоритму розв'язування задачі перейдемо одразу до написання коду програми.

З огляду на велику кількість вкладених розгалужень, реалізація нашого алгоритму на ***Python*** також вимагає акуратного запису вкладених блоків коду:

```
import math
print('Введіть коефіцієнти нерівності')
a = float(input())
b = float(input())
c = float(input())
d = b * b - 4 * c;
if d < 0: #друга нерівність не має коренів
    print('x Є [%8.2f; +inf)' % a)
else :
    if d == 0 :
        x0 = -b/2
        if a < x0 :
            print('x Є [%8.3f; %8.3f) U (%8.3f; +inf)' \
            %(a, x0, x0))
        elif a == x0 :
            print('x Є (%8.3f; +inf)' % a)
        else :
            print('x Є [%8.3f; +inf)' % a)
    else: #вітка, що виконується лише при d > 0
        x1 = (-b - math.sqrt(d))/2;
        x2 = (-b + math.sqrt(d))/2;
        # формули для x1 та x2 обрані так, що x1 < x2
        if a < x1 :
            print('x Є [%8.3f; %8.3f) U (%8.3f; +inf)' \
            % (a, x1, x2))
        elif a <= x2 :
            print("x Є (%8.3f; +inf)", x2);
        else :
            print("x Є [%8.3f; +inf)", a);
```

Розробка програми, яка не має синтаксичних помилок в коді і запускається та програми, яка правильно розв'язує поставлену задачу – різні речі. Для того, щоб переконатися, що ви створили правильну програму для розв'язування задачі, її роботу потрібно перевірити. Такій превірці в сучасній розробці прогрного забезпечення присвячений цілий розділ – ***тестування***.

У тестуванні для перевірки якості програми використовують так звані ***тест-кейси***, тобто ситуації при яких результат роботи програми відомий і можна оцінити коректність її роботи. Нижче подано набори вхідних даних, кожен з яких змушує програму виконати одну із запрограмованих віток розгалуження. Отримавши відповідний результат при кожному з поданих

наборів значень ***a, b, c***, ми можемо вважати, що наша програма працює коректно.

Значення параметрів			Результат
A	B	C	Роботи
1	1	5	x Є [1.000; +inf]
1	-4	4	x Є [1.000; 2.000)U(2.000; +inf]
2	-4	4	x Є (2.000; +inf]
3	-4	4	x Є [3.000; +inf]
1	-5	6	x Є [1.000; 2.000)U(3.000; +inf]
2	-5	6	x Є (3.000; +inf]
5	-5	6	x Є [5.000; +inf]

Запитання для самоконтролю

1. Що потрібно для написання та виконання Python-коду за допомогою Microsoft Visual Studio?
2. Опишіть процес виконання команди повного розгалуження ***if-else***.
3. Яка синтаксична конструкція призначена для полегшення запису вкладених розгалужень ***if***?
4. В чому проявляються відмінності між короткою та повною формами розгалуження?
5. Які розділові знаки та синтаксичні особливості використовуються при написанні команд ***if-else*** в ***Python***?
6. Який сиснтаксис запису оператора ***if-else*** в одну стрічку?
7. Чи є в Python ***оператор вибору*** (перемикач)?