Крос-платформне програмування

ЛАБОРАТОРНА РОБОТА № 2

Програмування алгоритмів розгалуженої структури.

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторної роботи
.

Львів

2024

Мета роботи: вдосконалення навичок написання програмного коду на Java та Kotlin. Ознайомлення з алгоритмічною конструкцією розгалуження та її використанням в програмах.

Завдання до лабораторної роботи.

1. Для обидвох задач (Завдання 1 та Завдання 2), відповідно до варіанта, розробити алгоритм розв'язування, та зобразити його графічно у вигляді блок-схеми.
2. Запрограмувати розроблений алгоритм мовою ***Java***.
3. Запрограмувати розроблений алгоритм мовою ***Kotlin***.
4. Оформити письмовий звіт про виконання роботи.

Звіт повинен містити:

- титульний аркуш:
- для кожного завдання:
  - ✓ умову задачі відповідного варіанту;
  - ✓ блок-схему алгоритму;
  - ✓ програму на Java;
  - ✓ результати виконання програми для деякого набору вхідних даних;
  - ✓ програму на Kotlin;
  - ✓ результати її виконання для того ж набору вхідних даних.

Короткі теоретичні відомості

Прості (примітивні) типи даних

В Java є вісім основних (примітивних) типів даних. П'ять із них — цілочисельні (включаючи символьний тип char), два — дійсні (float, double) і один логічний (булевий) тип даних.

Тип	Довжина (в байтах)	Діапазон або набір значень
boolean	не визначено	true, false
byte	1	$-128..127$
char	2	$0..2^{16}-1$, або 0..65535
short	2	$-2^{15}..2^{15}-1$, або −32768..32767
int	4	$-2^{31}..2^{31}-1$, або −2147483648..2147483647
long	8	$-2^{63}..2^{63}-1$, або приблизно $-9.2\cdot10^{18}..9.2\cdot10^{18}$
float	4	$-(2-2^{-23})\cdot2^{127}..(2-2^{-23})\cdot2^{127}$, або приблизно $-3.4\cdot10^{38}..3.4\cdot10^{38}$, а також $-\infty$, ∞, NaN
double	8	$-(2-2^{-52})\cdot2^{1023}..(2-2^{-52})\cdot2^{1023}$, або приблизно $-1.8\cdot10^{308}..1.8\cdot10^{308}$, а також $-\infty$, ∞, NaN

Оператори

Оператор (*англ. operator*) - це спеціальний символ, який повідомляє транслятору про те, що ви хочете виконати операцію з деякими операндами (наприклад, ,-, ,). азвичай, мови програмування визначають набір опера торів, подібних до операторів в математиці.

Арифметичні оператори використовуються в математичних виразах так само як і в алгебрі і представлені в таблиці.

Оператор	Операція
+	Додавання
+ =	Додаваня з присвоєнням
-	віднімання (а також унарний мінус)
- =	Віднімання з присвоєнням
*	Множення
* =	Множення з присвоєнням
/	Ділення

/ =	Ділення з присвоєнням
%	залишок ділення по модулю
%=	залишок ділення по модулю з присвоєнням
++	Інкремент (збільшення на 1)
--	Декремент (зменшення на 1)

Алгоритмічна конструкція розгалуження

Умовна інструкція в Java (в Kotlin так само) має форму:

```
if (логічний вираз) інструкція;
```

Якщо логічний вираз істинний (true) то буде виконана інструкція за умовою, інакше вона не буде виконана. Якщо необхідно виконати декілька інструкцій, то їх розміщають у блоці:

```
if (умова){
інструкція 1;
   …..
   інструкція n;
}
```

Якщо ж необхідно здійснити певну дію в разі хибності логічного виразу, то застосовують умовну інструкцію наступного виду:

```
if (умова) інструкція1;
else
інструкція2;
```

Логічні операції та вирази

Для побудови простих умов в розгалуженнях чи циклах використовують оператори відношення (чи операторами порівняння) які наведені в таблиці.

Оператор	**Опис**
==	Дорівнює
!=	Не дорівнює
>	Більше
<	Менше
>=	Більше рівне
<=	Менше рівне

Результатом операції порівняння є результат типу boolean із значеннями true або false.

Завдання 1.

Написати програму для розв’язання задачі згідно варіанту. Всі дані для роботи програми вводити з консолі. Якщо тип даних в умові не вказано явно, то вважати дані дійсними числами типу **float**.

Варіанти завдань.

1. Задано три цілі числа. Знайти кількість додатних чисел в цьому наборі.

2. Задано три дійсні числа. Знайти кількість від’ємних чисел в цьому наборі.

3. Задано дві змінні дійсного типу: A, B. Перерозподілити значення даних змінних так, щоб A виявилось меншим із заданих значень, а B - більше. Вивести нові значення змінних A і B.

4. Задано дві змінні цілого типу: A і B. Якщо їхні значення не рівні, то присвоїти кожній змінній суму цих значень, а якщо рівні, то присвоїти змінним нульове значення. Вивести нові значення змінних.

5. Задано два цілих числа: A і B. Якщо обидва числа парні, то зменшити їх значення вдвічі, якщо обидва непарні, то присвоїти кожному півсуму їх значень. Вивести нові значення змінних.

6. У магазині при покупці товару на суму від S_1 грн діє знижка p_1% та p_2%, – при покупці на суму від S_2 грн ($S_1 < S_2$). На покупки сумою до S_1 знижка не поширюється. Скласти програму, яка для заданих значень S_1, S_2, p_1 та p_2, за введеною з клавіатури вартістю покупки розраховує суму до сплати з урахуванням знижки.

7. У платіжній системі комісія за переказ коштів складає 10 грн, якщо сума переказу менша за 1000 грн, 1%, – на перекази сумою від 1000 до 10000 грн включно і 0,5% на суми, більші за 10000 грн. Скласти програму, яка для заданої суми переказу S розраховує розмір комісії та суму до сплати з урахуванням комісії.

8. Задано три цілі числа. Знайти найменше з них.

9. Задано три цілі числа. Визначити скільки серед них парних.

10. Задано три дійсні числа. Знайти середнє з них (тобто число, розташоване між найменшим і найбільшим).

11. Задано координати точки, що не лежить на координатних осях OX та OY. Визначити номер координатної чверті, в якій знаходиться дана точка.

12. Задано цілочисельні координати трьох вершин прямокутника, сторони якого паралельні координатним осям. Знайти координати його четвертої вершини.

13. Перевірити чи задане шестицифрове натуральне число є паліндромом (число, утворене записом цифр заданого числа у зворотному порядку дорівнює самому числу).

14. Тіло масою m з об'ємом V занурюють в рідину густиною ρ. Дослідити чи тіло плаватиме в рідині і вивести відповідне повідомлення.

15. Для введених з клавіатури двох цілих чисел визначити найбільшу величину серед їх суми, різниці та добутку.

16. На ділянці автошляху з обмеженням швидкості M км/год автомобіль рухається зі швидкістю V км/год . Якщо швидкість автомобіля перевищує обмеження не більше, ніж на $p\%$, патрульний повинен зупинити автомобіль і зробити водієві авта усне попередження. Якщо ж перевищення швидкості більше за $p\%$ від обмеження, патрульний повинен накласти на водія штраф у сумі S грн. Написати програму, яка моделює алгоритм дій патрульного.

17. У торгівельній мережі для зменшення втрат пов'язаних з термінами зберігання продуктів запроваджено гнучку систему ціноутворення: на продукти, до кінця зберігання яких залишилося менше половини терміну придатності діє знижка у розмірі p%, а товари, термін придатності яких спливає наступної доби продають за ціною p% від початкової. Скласти програму, яка за заданою ціною товару, терміном його придатності та кількістю діб, що пройшла з дня його виготовлення визначатиме поточну його ціну, за вказаним вище правилом.

18. Для введених з клавіатури двох дійсних чисел визначити найбільшу величину серед їх суми, добутку та частки.

19. Задано три дійсних числа. Знайти суму двох менших з них.

20. Задано три цілі числа. Знайти добуток двох менших з них.

21. Оплата за телефонний дзвінок стягується за таким тарифом: p грн за першу хвилину розмови, незалежно від тривалості дзвінка, q грн за кожну наступну хвилину з посекундною тарифікацією, після 10 хв розмови плата не стягується взагалі. Скласти програму, яка для заданої тривалості дзвінка за заданими тарифами p та q розраховує його вартість.

22. Задано три цілі числа. Визначити скільки з них є двоцифровими.

23. Задано три дійсні числа. Знайти суму двох більших з них.

24. Підприємець сплачує фіксовану суму податку у розмірі X грн, якщо сума його місячного прибутку менша за S_1 грн. Якщо ні, то X грн плюс $p\%$ від суми, яка перевищує S_1. Якщо ж сума прибутків підприємця перевищує S_2 ($S_2 > S_1$), то фіксована сума не стягується, а нараховується $q\%$ податку від загальної суми прибутків. Скласти програму для обчислення розміру податку на вказану суму прибутку.

25. Фільми на диску зберігаються у трьох різних папках, залежно від їх тривалості. Фільми тривалістю до 35 хв записують у папку «короткометражні», фільми тривалістю від 35 хв до однієї години, – в папку «телефільми», а фільми, довші за 1 годину, в папку «кіно». Скласти програму, яка за введеною з

клавіатури тривалістю фільму повідомить назву папки, до якої слід завантажити фільм.

26. Задано три цілі числа. Вивести спочатку найменше, а потім найбільше з даних чисел.

27. На координатній площині побудоване коло радіуса R з центром в початку координат та точка $A(x; y)$. За заданим радіусом кола та координатами точки встановити де розміщена точка: всередині кола, поза колом, чи на колі.

28. На координатній площині задано пряму $y = kx + b$ $(k \neq 0)$ та точку $A(x; y)$. За заданими параметрами рівняння та координатами точки A встановити де розміщена точка: над прямою $y > kx + b$, під прямою $y < kx + b$, чи на прямій.

29. Задано три цілі числа. Знайти різницю між найбільшим і найменшим з них.

30. Задано натуральне число з діапазону 1 – 9999. Вивести його опис у вигляді: «одноцифрове число», «двоцифрове число» і т. д.

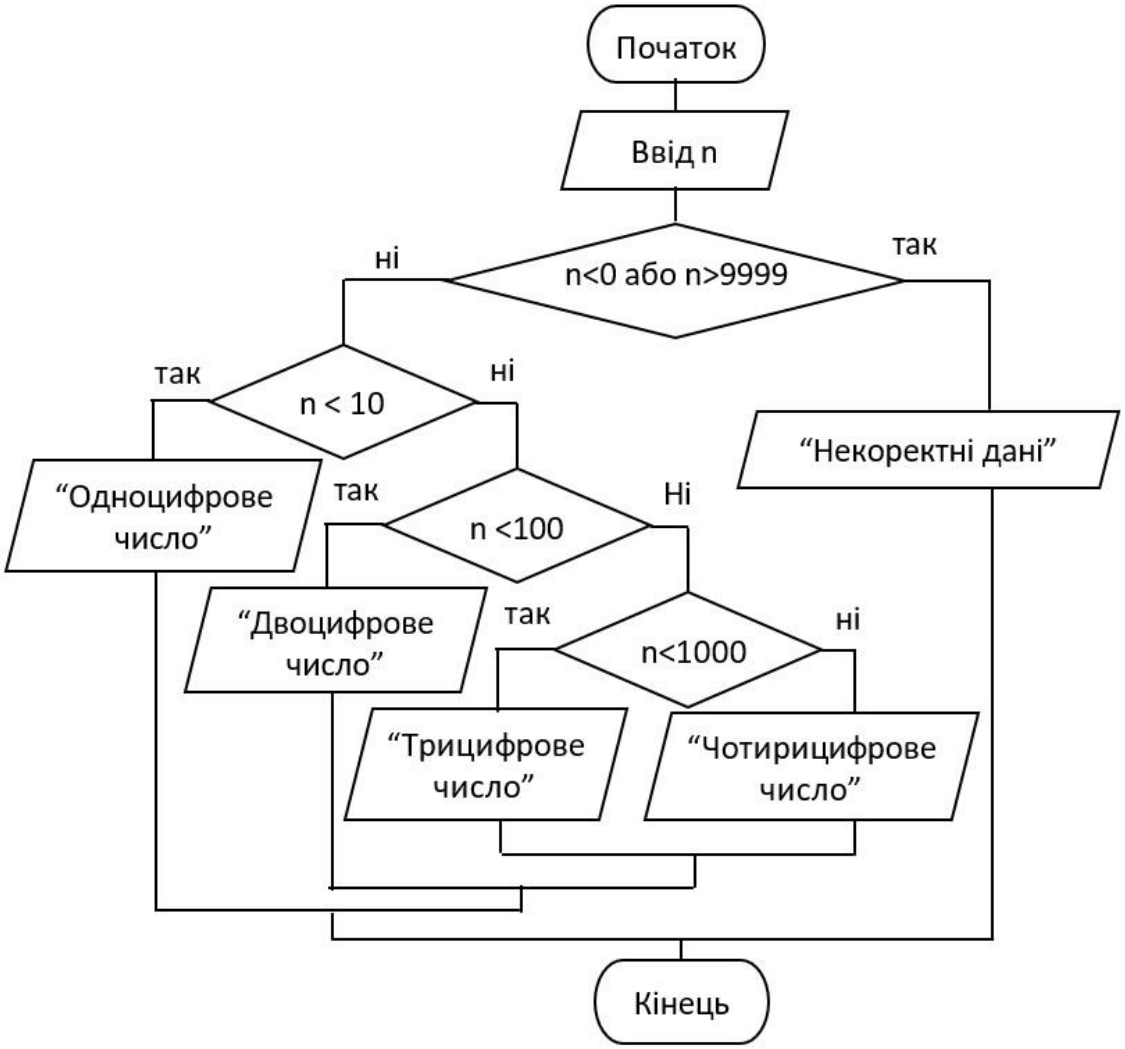
Приклад розв'язування задачі 30

30. Задано натуральне число з діапазону 1 – 9999. Вивести його опис у вигляді: «одноцифрове число», «двоцифрове число» і т. д.

Можна побудувати декілька варіантів алгоритму розв'язання сформульованої задачі. Зупинимося на варіанті, що використовує лише розгалуження у повній формі. Для коректної роботи такого алгоритму при довільних значеннях вхідних даних, спочатку слід перевірити чи введене ціле число відповідає умові задачі, тобто належить діапазону [1; 9999]. Якщо введене число відповідає умові, то його класифікацію можна реалізувати за схемою:

- *числа менші за 10 – одноцифрові,*
- *інші числа, але менші за 100, – двоцифрові,*
- *всі інші числа, менші за 1000 – трицифрові,*
- *решта чисел – чотирицифрові.*

Графічне зображення такого підходу подано нижче.



Програма на Java.

Як відомо, синтаксичні конструкції ***Java*** запозичені з ***C++***, де алгоритмічна конструкція повного розгалуження в C++ складається з двох операторів: ***if*** та ***else*** і, зазвичай, записується в такій формі:

```
if (<умова>) {
    <послідовність команд 1>;
}
else {
    <послідовність команд 2>;
}
```

Решта особливостей умовних конструкцій в Java пропонуємо розглянути в наступному коді:

```
package com.tasks.par2;
//пакет класу Scanner, використаного для вводу даних
import java.util.Scanner;
public class Task90 {
  public static void main(String[] args) {
    System.out.println("Введіть натуральне число "+
        "від 1 до 9999 включно");
    int n;
    Scanner input = new Scanner(System.in);
    n = input.nextInt();
    //первірка, чи користувач ввів коректне число
    if(n > 0 && n<10000){
      if(n < 10)System.out.printf(
          "%5d - одноцифрове число\n",n);
        //одну інструкцію після if можна ставити без {}
      else {
        if(n < 100) {
      System.out.printf("%5d - двоцифрове число\n",n);
        }
        else {
          if (n < 1000){
            System.out.printf(
                "%5d - трицифрове число\n", n);
          }
          else System.out.printf(
              "%5d - чотирицифрове число\n", n);
        }
      }
    }
    else System.out.println("Ви ввели число, " +
        "що не належить вказаному відрізку");
  }

}
```

Програма на Kotlin.

Контрукція умовного оператора в ***Kotlin*** така ж, як і в ***Java***. Замінивши Java-клас зі статичним методом ***main*** на ***Kotlin-функцію main*** та Java-об'єкти консольного вводу-виводу на відповідні функції ***Kotlin***, отримаємо наступний код:

```
package com.tasks.par2

fun main(args: Array<String>) {
    println("Введіть натуральне число від 1 до 9999" )
    val n: Int = readln().toInt()
    //первірка, чи користувач ввів коректне число
    // можна використовувати таку перевірку належності
    // числа діапазону замість n > 0 && n<10000
    if (n in 1..9999) {
        if (n < 10) {
            println("%5d - одноцифрове число".format(n))
        }else if (n < 100) {
            println("%5d - двоцифрове число".format(n))
        } else if (n < 1000) {
            println("%5d - трицифрове число".format(n))
        } else {
            println("%5d - чотирицифрове число".format(n))
        }
    } else {
        println("Ви ввели число, " +
                "що не належить вказаному відрізку")
    }
}
```

Завдання 2.

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 гілками розгалуження.

Варіанти завдань.

1. $\frac{ax}{x^2+bx+c} \leq 0;$

2. $\frac{ax}{x^2+bx+c} > 0;$

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

4. $\frac{x^2-bx-c}{ax} \leq 0;$

5. $\frac{x^2}{ax+b} < c;$

6. $\frac{x^2}{ax+b} \geq c;$

7. $\frac{x-a}{x^2+bx+c} \geq 0;$

8. $\frac{x-a}{x^2+bx+c} < 0;$

9. $\frac{x-a}{x^2+bx+c} > 0;$

10. $(x - a)(x^2 + bx + c) < 0;$

11. $\begin{cases} x + a > 0, \\ x^2 + bx + c \leq 0; \end{cases}$

12. $(x + a)(x^2 - bx + c) > 0;$

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

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

15. $\frac{x^2-ax+b}{x+c} > 0;$

16. $\frac{x^2}{ax-b} < c;$

17. $\frac{ax}{x^2-bx+c} > 0;$

18. $\frac{x-a}{x^2-bx+c} \geq 0;$

19. $\frac{x^2-bx-c}{ax} \leq 0;$

20. $\frac{x-a}{x^2+bx+c} > 0;$

21. $\frac{x^2}{ax+b} \geq c;$

22. $\begin{cases} x + a > 0, \\ x^2 + bx + c \leq 0; \end{cases}$

23. $\frac{ax}{x^2+bx+c} \leq 0;$

24. $\begin{cases} x + a \leq 0, \\ bx^2 - cx < 0; \end{cases}$

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

26. $(x + a)(x^2 + bx - c) < 0;$

27. $\frac{x^2}{ax+b} < c;$

28. $(x - a)(x^2 + bx + c) > 0;$

29. $\begin{cases} x + a > 0, \\ x^2 - bx + c > 0; \end{cases}$

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

Приклад розв'язування задачі 30

30. 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* $x_1 \le a \le x_2$ *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 програм.

Програма на Java.

При реалізації описаного алгоритму на ***Java*** потрібно слідкувати за коректним використанням блоків коду у кожній вітці розгалуження. Конструкція ***if-then-else*** тут, як і в ***C/C++*** формально складається з двох окремих операторів: ***if*** та ***else***, тому вкладені розгалуження повної форми прийнято брати у фігурні дужки. Сама ж ***Java***-програма, я завжди, повинна мати об'єктно-орієнтовану структуру у вигляді класу зі статичним методом ***main***.

```
package com.tasks.par2;
import java.util.Scanner;
public class Task105 {
  public static void main(String[] args) {
    double a,b,c,d;
    System.out.println("Введіть коефіцієнти нерівності");
    Scanner scanner = new Scanner(System.in);
    a = scanner.nextDouble();
    b = scanner.nextDouble();
    c = scanner.nextDouble();
    d = b * b - 4 * c;
    if (d < 0) { //друга нерівність не має коренів
      System.out.println("x Є [" + a + "; +inf)");
    }
    else {
      if (d == 0) {
```

```
        double x0 = -b/2;
        if (a < x0) { System.out.printf(
            "x Є [%.3f; %.3f) U (%.3f; +inf)\n",a, x0, x0);
        }
        else {
          if(a == x0) System.out.printf(
              "x Є (%.3f; +inf)\n", a);
          else System.out.printf("x Є [%.3f;+inf)\n",a);
        }
      }
      else {//вітка, що виконується лише коли d > 0
        double x1 = (-b - Math.sqrt(d))/2;
        double x2 = (-b + Math.sqrt(d))/2;
        //формули для x1 та x2 обрані так, що x1 < x2
        if (a < x1)  {
          System.out.printf(
          "x Є [%.3f; %.3f) U (%.3f; +inf)\n", a, x1, x2);
        }
        else{
          if (a <= x2)
            System.out.printf("x Є (%.3f; +inf)\n", x2);
          else
            System.out.printf("x Є [%.3f; +inf)\n", a);
        }
      }
    }
  }
}
```

Програма на Kotlin.

Перекладемо поданий вище Java-код на ***Kotlin***, замінивши Java-клас зі статичним методом ***main*** на ***Kotlin-функцію main*** та відповідні засоби консольного вводу-виводу на відповідні функції ***Kotlin***. Усі дані, що використовує програма (вхідні та результати обчислень) вважатимемо сталими (***val***). Отримаємо наступний код:

```
package com.tasks.par2


import kotlin.math.sqrt


fun main(args: Array<String>) {
  println("Введіть коефіцієнти a, b та c нерівності" +
    " по одному в рядоку,  без розділових знаків")
  // Сумістимо оголошення з вводом даних
  // коефіцієнти нерівності незмінні протягом
  // розв'язування задачі, тому оголошуємо їх
  // як сталі з специфікатором "val"
  val a: Double = readln().toDouble()
  val b: Double = readln().toDouble()
  val c: Double = readln().toDouble()
  val d = b * b - 4 * c
  if (d < 0) { //друга нерівність не має коренів
    println("x Є [$a; +inf)")
  } else if (d == 0.0) {
    val x0 = -b / 2
    if (a < x0) {
      println("x Є [%.3f; %.3f) U (%.3f; +inf)"
        .format(a, x0, x0))
    } else {
      if (a == x0)
        println("x Є (%.3f; +inf)".format(a))
      else println("x Є [%.3f;+inf)".format(a))
```

```
    }
  } else { //вітка, що виконується лише коли d > 0
    val x1 = (-b - sqrt(d)) / 2
    val x2 = (-b + sqrt(d)) / 2
    //формули для x1 та x2 обрані так, що x1 < x2
    if (a < x1) {
      println("x Є [%.3f; %.3f) U (%.3f; +inf)"
        .format(a, x1, x2))
    } else {
      if (a <= x2) println(
        "x Є (%.3f; +inf)".format(x2))
      else println("x Є [%.3f; +inf)".format(a))
    }
  }
}
```

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]