

Лабораторна робота №3

Програмування на Java алгоритмів розгалуженої структури

Мета роботи: вдосконалення навичок написання Java-коду, ознайомлення з алгоритмічною конструкцією розгалуження в мові Java та її використанням в програмах.

Короткі теоретичні відомості.

Структура Java-програми

Програма на мові Java – це проект (Project), побудований з класів та їх екземплярів (об'єктів). Об'єкти об'єднують в собі набір даних (властивостей), що характеризують їх стан та процедури для опрацювання цих даних (методи), що описують їх поведінку. Клас є загальним описом, на основі якого створюються окремі об'єкти (однакової структури). Клас містить опис назв та типів властивостей об'єктів та опис процесу виконання методів (реалізацію методів). Об'єкт, як представник класу, наділяється конкретними значеннями властивостей, та може опрацьовувати їх описаними в класі методами. Навіть найпростіший проект (Java-програма) повинен складатися принаймні з одного класу. Цей клас повинен обов'язково містити головний метод

```
public static void main(String[] args) {  
    // тут писати код методу  
}
```

Основні правила написання коду

Прості (примітивні) типи даних

В Java є вісім основних (примітивних) типів даних. П'ять із них — цілочисельні (включаючи символний 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 $-9.2 \cdot 10^{18}..9.2 \cdot 10^{18}$
float	4	$-(2 \cdot 2^{-23}) \cdot 2^{127}..(2 \cdot 2^{-23}) \cdot 2^{127}$, або приблизно $-3.4 \cdot 10^{38}..3.4 \cdot 10^{38}$, а також $-\infty, \infty, \text{NaN}$
double	8	$-(2 \cdot 2^{-52}) \cdot 2^{1023}..(2 \cdot 2^{-52}) \cdot 2^{1023}$, або приблизно $-1.8 \cdot 10^{308}..1.8 \cdot 10^{308}$, а також $-\infty, \infty, \text{NaN}$

Оператори

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

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

Оператор	Операція	Оператор	Операція
+	Додавання	+=	Додавання 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 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 розміщують 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
==	Рівно
!=	Не рівно
>	Більше
<	Менше
>=	Більше рівне
<=	Менше рівне

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

Завдання 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.

1. Дано три цілих числа. Знайти кількість додатних чисел в цьому наборі.
2. Дано три дійсних числа. Знайти кількість від'ємних чисел в цьому наборі.
3. Дано 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. Дано дві змінні цілого типу: A і B. Якщо їхні значення не рівні, то присвоїти кожній змінної суму цих значень, а якщо рівні, то присвоїти змінним нульові значення. Вивести нові значення змінних.
5. Дано три числа. Знайти найменше з них.
6. Дано три числа. Знайти середнє з них (тобто число, розташоване між найменшим і найбільшим).
7. Дано три числа. Вивести спочатку найменше, а потім найбільше з даних чисел.
8. Дано три числа. Знайти суму двох більших з них.
9. Дано координати точки, що не лежить на координатних осях OX та OY. Визначити номер координатної чверті, в якій знаходиться дана точка.
10. Дано цілочисельні координати трьох вершин прямокутника, сторони якого паралельні координатним осям. Знайти координати його четвертої вершини.
11. Тіло масою m з об'ємом V занурюють в рідину густиною ρ . Дослідити чи тіло плаватиме в рідині і вивести відповідне повідомлення.
12. Для введених двох чисел визначити найбільшу величину серед їх суми, різниці та добутку.
13. Для введених двох чисел визначити найбільшу величину серед їх суми, добутку та частки.
14. Дано три числа. Знайти суму двох менших з них.
15. Дано три числа. Знайти різницю між найбільшим і найменшим з них.
16. Задано натуральне число з діапазону 1 – 9999. Вивести його опис у вигляді: «одноцифрове число», «двоцифрове число» і т. д.

Приклад виконання роботи

Завдання: 16. Задано натуральне число з діапазону 1 – 9999. Вивести його опис у вигляді: «одноцифрове число», «двоцифрове число» і т. д.

Можна побудувати декілька алгоритмів розв’язання поставленої задачі. Зупинимось на варіанті, в якому спочатку перевіримо чи введене число відповідає умові задачі (лежить в діапазоні від 1 до 9999). Якщо введене число дійсно належить вказаному діапазону, то подальшу перевірку можна здійснювати повною формою розгалуження за схемою: числа менші за 10 – одноцифрові, інші, але менші за 100, – двоцифрові, інші, але менші за 1000 – трицифрові, всі решта – чотирицифрові.

Запускаємо Eclipse. Створюємо новий проект з назвою Lab2, в ньому – одноіменний клас з головним методом main, де записуємо наступний код.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class Lab2 {
    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        System.out.println("Введіть натуральне число від 1 до 9999");
        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(reader.readLine());
        if (n > 0 && n < 10000) {
            if (n < 10) System.out.println("Одноцифрове число");
            else {
                if (n < 100) System.out.println("Двоцифрове число");
                else {
                    if (n < 1000) System.out.println("Трицифрове число");
                    else System.out.println("Чотирицифрове число");
                }
            }
        }
        else System.out.println("Введене число лежить за межами вказаного діапазону");
    }
}
```

Лабораторна робота № 4

Програмування алгоритмів з багатьма вітками розгалуження на прикладі розв'язання алгебраїчної нерівності. Використання статичних методів класу.

Мета роботи: вдосконалення практичних навичок програмування алгоритмів розгалуженої структури, знайомство з деякими елементами об'єктно-орієнтованого програмування.

Короткі теоретичні відомості.

Структура класів, властивості і методи

Пакети, область видимості

Основні домовленості з іменування об'єктів

Завдання

1. Скласти програму для розв'язування алгебраїчної нерівності (системи нерівностей) згідно варіанту. Алгоритм знаходження розв'язку нерівності оформити у вигляді окремого методу, який для заданих значень числових коефіцієнтів a , b та c визначає розв'язок і повертає текстовий результат у вигляді проміжків на числовій осі. В головному методі класу організувати виклик методу розв'язування нерівності з різними значеннями коефіцієнтів a , b і c так, щоб продемонструвати роботу методу за усіма можливими вітками розгалуження.

Варіанти завдань

№ вар.	Завдання	№ вар.	Завдання
1	$\frac{a}{x^2 + bx + c} \leq 0$	9	$\frac{a}{x^2 + bx + c} > 0$
2	$ax^2 - bx > c$	10	$ax^2 - bx \leq c$
3	$\frac{x^2}{ax + b} < c$	11	$\frac{x^2}{ax + b} \geq c$
4	$\frac{x - a}{x^2 + bx + c} \geq 0$	12	$\frac{x - a}{x^2 + bx + c} < 0$
5	$\frac{x^2 - ax + b}{x + c} \leq 0$	13	$\frac{x^2 - ax + b}{x + c} > 0$
6	$(x - a)(x^2 + bx + c) < 0$	14	$(x + a)(x^2 - bx + c) > 0$
7	$\begin{cases} x - a < 0, \\ x^2 + bx + c > 0 \end{cases}$	15	$\begin{cases} x + a > 0, \\ x^2 + bx + c \leq 0 \end{cases}$
8	$\begin{cases} x - a \geq 0, \\ x^2 + bx + c < 0 \end{cases}$	16	$\begin{cases} x - a \geq 0, \\ x^2 + bx + c > 0 \end{cases}$

Приклад виконання (варіант № 16)

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

Результатом виконання роботи має бути програма, яка отримавши від користувача конкретні числові значення коефіцієнтів a , b , c , видавала б розв'язок відносно x нерівності у вигляді проміжків на числовій осі. Головною запорукою правильної роботи 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 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 має розв'язком усю множину дійсних чисел \mathbb{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 $x_2 = \frac{-b + \sqrt{D}}{2}$ (очевидно, що $x_1 < x_2$)

квадратного тричлена в другій нерівності. 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 проведений аналіз, код методу для розв'язування нерівності можна оформити наступним чином:

```
public static String solve(double a, double b, double c){
    String result;
    double d = b*b-4*c;
    if (d<0) result = "["+a+";+inf)";
    else{
        if (d==0){
            double x0 = -b/2;
            if (a < x0) result = "x ∈ ["+a+";" + x0 + ")U("+x0+";+inf)";
            else {
                if(a == x0) result = "x ∈ ("+a+";+inf)";
                else result = "x ∈ ["+a+";+inf)";
            }
        }
        else{
            double x1 = (-b-Math.sqrt(d))/2;
            double x2 = (-b+Math.sqrt(d))/2;
            if (a < x1) result = "x ∈ ["+a+";" + x1 + ")U("+x2+";+inf)";
            else{
                if (a <= x2) result = "x ∈ ("+x2+";+inf)";
                else result = "x ∈ ["+a+";+inf)";
            }
        }
    }
    return result;
}
```

Створюємо новий клас з головним методом main, в тілі класу записуємо код наведеного вище методу solve. В тілі головного методу main слід реалізувати виклики методу solve з різними наборами числових значень коефіцієнтів a, b та c, кожен з яких відповідає одному з варіантів системи, розглянутих в аналізі. Він може виглядати так:

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    System.out.println("a=-10, b=-1, c=-6\n" + solve(-10, -1, -6));
    System.out.println("a=-2, b=-1, c=-6\n" + solve(-2, -1, -6));
    System.out.println("a=-1, b=-1, c=-6\n" + solve(-1, -1, -6));
    System.out.println("a=3, b=-1, c=-6\n" + solve(2, -1, -6));
    System.out.println("a=10, b=-1, c=-6\n" + solve(10, -1, -6));
    System.out.println("a=-10, b=2, c=1\n" + solve(-10, 2, 1));
    System.out.println("a=-1, b=2, c=1\n" + solve(-1, 2, 1));
    System.out.println("a=10, b=2, c=1\n" + solve(10, 2, 1));
    System.out.println("a=10, b=2, c=10\n" + solve(10, 2, 10));
}
```