МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

факультет програмної інженерії та бізнесу

кафедра інженерії програмного забезпечення

**Лабораторна робота № 5**

з дисципліни « Програмування мовою Java »

*назва дисципліни*

на тему: «РОЗРОБКА ЗАСТОСУНКІВ МОВОЮ ПРОГРАМУВАННЯ JAVA. УМОВНІ ОПЕРАТОРИ»

Виконав: студент 2 курсу групи № 622п

освітньої програми

121 інженерія програмного забезпечення

(шифр і назва ОП)

Зайченко Ярослав Ігорович

(прізвище й ініціали студента)

Прийняв: доцент

Симбірський Генадій Дмитрович

(посада, науковий ступінь, прізвище й ініціали)

Кількість балів:

Харків – 2023

ЗМІСТ

[**ПОСТАНОВКА ЗАВДАННЯ 3**](#_Toc148216875)

[**ЗМІСТ ЗВІТУ 3**](#_Toc148216876)

[**ПОРЯДОК ВИКОНАННЯ РОБОТИ 3**](#_Toc148216877)

[**ВИКОНАННЯ РОБОТИ 4**](#_Toc148216878)

[**ЗАГАЛЬНІ ВІДОМОСТІ 4**](#_Toc148216879)

[**ВИСНОВОК 11**](#_Toc148216880)

[**КОНТРОЛЬНІ ПИТАННЯ 12**](#_Toc148216881)

Мета роботи: отримання практичних навичок роботи з операторами мови програмування Java: умовні оператори.

ПОСТАНОВКА ЗАВДАННЯ

Навчитися працювати з умовними операторами if, if-else і switch.

# **ЗМІСТ ЗВІТУ**

1. Постановка мети.
2. Встановити завдання, які вирішуються в лабораторній роботі
3. Коротко описати роботу оператора if-else.
4. Коротко описати роботу оператора switch-case.
5. Записати код вирішення завдань.
6. Висновки по роботі

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Прочитати теоретичний матеріал
2. Вивчити роботу оператора if-else
3. Вивчити роботу оператора switch-case
4. Вирішити завдання з використанням вивченого матеріалу
5. Зробити висновки по роботі.

# **ВИКОНАННЯ РОБОТИ**

## **Загальні відомості**

Мова програмування – Java

Операційна система – Windows 11

Процесор – Apple Silicon M1 Pro with 4 CPU cores

Тип компілятора – IntelliJ IDEA 2023

Робота оператора if-else.

У кожній операції порівняння беруть участь два операнда. А виглядає вона як звичайне нерівність в математиці. Наприклад: a> b.

Операція порівняння повертає:

* значення true ( "істина"), якщо висловлювання з оператором правдиве (умова виконується),
* false ( «брехня») - якщо висловлювання з оператором помилкове (умова не виконується).

Наприклад, ми хочемо порівняти в Java значення змінних a і b. Для цього використовуємо оператор >, який повертає логічне значення (true або false).

A > b одне true, коли значення змінної a більше значення змінної b (операнд зліва від > більше правого), а інакше - false.

|  |  |  |  |
| --- | --- | --- | --- |
| **Оператор порівняння** | **Позначення в**  **Java** | **Приклад операції** | **Результат операції** |
| менше | < | a < b | true, якщо операнд зліва менше правого, інакше false |
| більше | > | a > b | true, якщо операнд зліва більше правого, інакше false |
| Менше або дорівнює | <= | a <= b | true, якщо операнд зліва менше правого або вони рівні, інакше false |
| Більше чи рівно | >= | a >= b | true, якщо операнд зліва більше правого або дорівнює йому, інакше false |
| дорівнює | == | a == b | true, якщо операнд зліва дорівнює операнду справа |
| не дорівнює | != | a != b | true, якщо операнд зліва не дорівнює операнду справа |

Оператор розгалуження дозволяє вибрати і виконати певний блок коду в залежності від спрацювання деякого умови.

Є два оператора розгалуження (умовні конструкції): if-else і switch. оператор if

Його синтаксис виглядає так:

if (умова) {

// код

} else {

// код

}

В круглих дужках після if вказують умовне вираз (вислів). Якщо воно істинне (тобто результат його обчислення дорівнює true), то виконується код в фігурних дужках після умови, інакше виконується код по гілці else (якщо вона є).

На рисунку 1 представлені різні варіанти синтаксису оператора if-else. Один рядок не входить в структуру оператора. Знайдіть його.

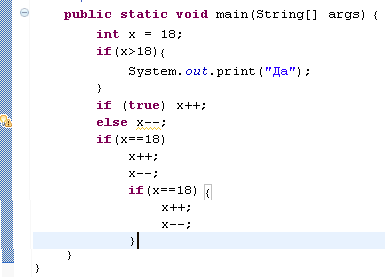


Рисунок 1 – Приклади використання оператора if

**Відповідь:** х--; не входить в структуру оператора if.

Визначте, до якого оператора if відноситься оператор else на поданому фрагменті коду:

int x = 18;

if (x>18)

if(x>21)x++;

else x--;

**Відповідь:** оператор else відносить до другого оператору if(x>21).

У минулій лабораторній роботі був вивчений тернарний оператор. Використання конструкції if-else дозволяє проводити аналогічні дії тернарного оператору

if (x!=0) a = a \* 2;

else a = a +1;

a = (x!=0)?a \* 2 : a+1;

Важливо пам'ятати, що спрацьовує тільки один варіант конструкції, навіть якщо за підсумками роботи умова починає задовольняти іншим частинам конструкції if-else (рис.3)

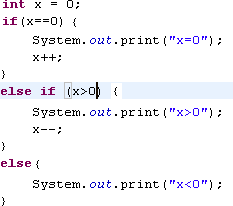


Рисунок 2 – одиночне спрацьовування

Усередині дужок може виконуватися безліч умов. Головне, щоб результатом було true або false.

int x = 0, y = 0;

if (x==0 && y ==0)

Вкладені конструкції

Припустимо, ми хочемо перевірити якусь умову. Якщо умова не виповнилася - перевірити іншу умову.

вкладений if

if (умова1) {

// дії, якщо умова1 виконано;

} else {

if (умова2) {

// дії, якщо умова2 виконано;

}

<...>

else {

// дії, якщо всі попередні умови не виконалися;

}

конструкція else if

if (умова1) {

// дії, якщо умова1 виконано;

}

else if (умова2) {

// дії, якщо умова2 виконано;

}

else if (умова3) {

// дії, якщо умова3 виконано;

}

else {

// дії, якщо умова останнього if теж не виповнилося;

}

Робота оператора switch-case.

Оператор множинного вибору

Він застосовується, коли потрібно виконати один з декількох блоків коду в залежності від значення аргументу.

Вигода від switch тим очевидніше, чим більше перевірок і варіантів дій нам потрібно.

int dayOfWeekNum = 5; switch (dayOfWeekNum) {

case 1:

System.out.println("Понеділок"); break;

case 2:

System.out.println("Вівторок"); break;

case 3:

System.out.println("Середа"); break;

case 4:

System.out.println("Четвер"); break;

case 5:

System.out.println("П’ятниця"); break;

case 6:

System.out.println("Субота"); break;

case 7:

System.out.println("Неділя"); break;

}

Приклад використання аналогічних рішень наведено на рисунку 4.

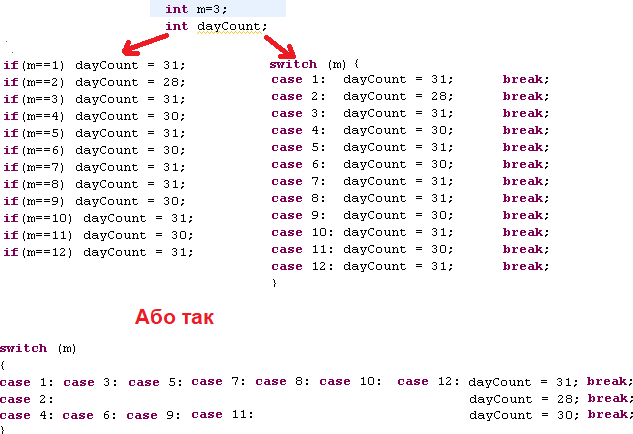


Рисунок 3 – приклади використання switch-case

Оператор break необхідний інакше будуть виконуватися всі дії конструкції case (рис.5).

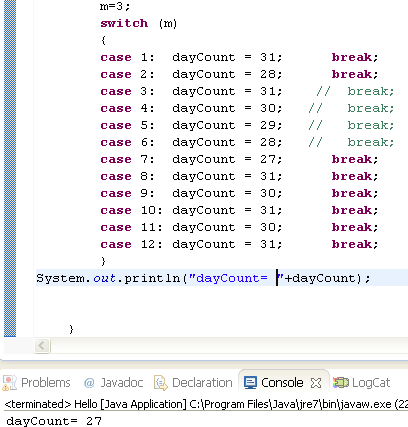


Рисунок 4 – Особливості спрацювання з оператором break

Вирішення завдань

Завдання 1. Дано 5 чисел (а, в, с, д, е). Знайти найбільше

Код створеної програми:

import java.util.Scanner;  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("Введіть значення a: ");  
 int a = scanner.nextInt();  
 System.out.print("Введіть значення b: ");  
 int b = scanner.nextInt();  
 System.out.print("Введіть значення c: ");  
 int c = scanner.nextInt();  
 System.out.print("Введіть значення d: ");  
 int d = scanner.nextInt();  
 System.out.print("Введіть значення e: ");  
 int e = scanner.nextInt();  
 int max = a; // Початково вважаємо, що 'a' – найбільше число  
 if (b > max) {  
 max = b;}  
 if (c > max) {  
 max = c;}  
 if (d > max) {  
 max = d;}  
 if (e > max) {  
 max = e;}  
 System.out.println("Найбільше число: " + max);}

}

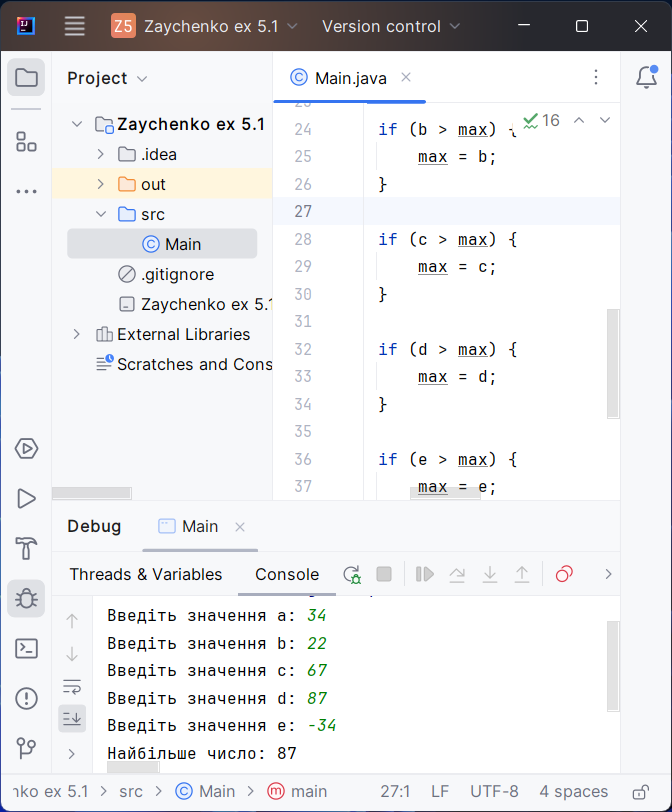


Рисунок 5 – результат роботи програми до завдання 1

Завдання 2. Дано 5 чисел (а, в, с, д, е). Знайти ДВА найбільших

Код створеної програми:

import java.util.Scanner;  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.in);  
 System.out.print("Введіть значення a: ");  
 int a = scanner.nextInt();  
 System.out.print("Введіть значення b: ");  
 int b = scanner.nextInt();  
 System.out.print("Введіть значення c: ");  
 int c = scanner.nextInt();  
 System.out.print("Введіть значення d: ");  
 int d = scanner.nextInt();  
 System.out.print("Введіть значення e: ");  
 int e = scanner.nextInt();  
 int firstMax = Math.max(a, Math.max(b, Math.max(c, Math.max(d, e))));  
 int secondMax = a; // Початково вважаємо, що 'a' - друге найбільше число  
 if (b > secondMax && b < firstMax) {  
 secondMax = b;}  
 if (c > secondMax && c < firstMax) {  
 secondMax = c;}  
 if (d > secondMax && d < firstMax) {  
 secondMax = d;}  
 if (e > secondMax && e < firstMax) {  
 secondMax = e;}  
 System.out.println("Найбільше число: " + firstMax);  
 System.out.println("Друге найбільше число: " + secondMax);}  
}

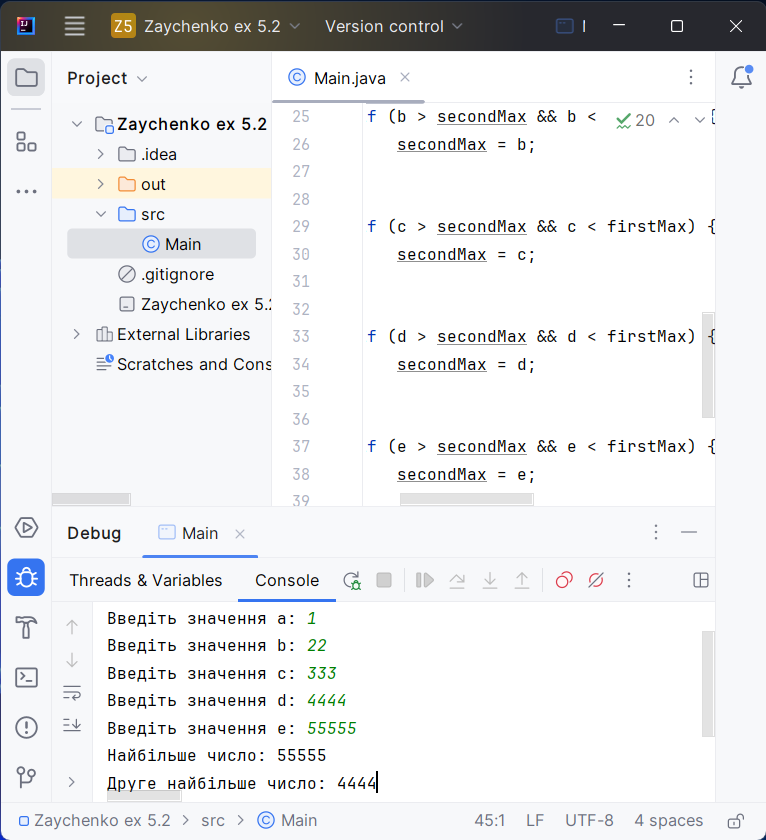


Рисунок 6 – результат роботи програми до завдання 2

# **ВИСНОВОК**

Отримано практичні навички роботи з операторами if, if-else і switch мови програмування Java.

# **Контрольні питання**

1. **Особливості роботи if**

* if - це умовний оператор, який використовується для виконання коду в залежності від заданої умови.
* Умова в if повинна бути булевим виразом, тобто виразом, який повертає true або false.
* Якщо умова в if є true, то код внутрішнього блоку виконується, інакше він пропускається.

1. **Особливості роботи else if**

* else if - це розширення конструкції if, яке дозволяє встановити додаткові умови для перевірки.
* else if використовується, коли вам потрібно перевірити кілька умов послідовно.
* Коли перша умова в if або попередньому else if є true, код цього else if виконується, і подальші else if та else блоки пропускаються.

1. **Особливості роботи else?**

* else - це опціональний блок, який виконується, якщо жодна з умов в if або else if не виконується.
* else не має умови, і він виконується, коли всі попередні умови в if і else if є false.

1. **Особливості роботи switch?**

* switch - це оператор вибору, який дозволяє вам вибирати один зі списку варіантів в залежності від значення виразу.
* Вираз у switch повинен бути цілим числом або рядком.
* Кожен case вказує на конкретний варіант, який порівнюється з виразом. Якщо вони співпадають, виконується код у case.
* break використовується для виходу з switch, інакше всі наступні case виконаються також.

1. **Особливості роботи case?**

* case - це мітка, яка вказує на конкретний варіант у switch.
* Код у case виконується, якщо вираз у switch відповідає значенню case.
* case може містити будь-який код, а також break, який використовується для виходу з switch після виконання коду case.