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 № 3**

з дисципліни « Програмування мовою Java »

*назва дисципліни*

на тему: «РОЗРОБКА ЗАСТОСУНКІВ МОВОЮ ПРОГРАМУВАННЯ JAVA. АРИФМЕТИЧНІ 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](#_Toc146832749)

[**ЗМІСТ ЗВІТУ** 3](#_Toc146832750)

[**ВИКОНАННЯ РОБОТИ** 4](#_Toc146832751)

[**Загальні відомості** 4](#_Toc146832752)

[**ВИСНОВОК** 15](#_Toc146832753)

[**Контрольні питання** 16](#_Toc146832754)

Мета роботи**:** отримання практичних навичок роботи з арифметичними операторами мови програмування Java.

# **ПОСТАНОВКА ЗАВДАННЯ**

Навчитися працювати з арифметичними операторами.

# **ЗМІСТ ЗВІТУ**

1. Постановка завдання.
2. Встановити завдання, які вирішуються в лабораторній роботі
3. Коротко описати особливості арифметичних операторів.
4. Коротко описати особливості укороченої форми арифметичних операторів.
5. Коротко описати інкремент і декремент.
6. Виконати завдання 3.1-3.6
7. Висновки по роботі.

**ПОРЯДОК ВИКОНАННЯ РОБОТИ**

1. Прочитати теоретичний матеріал
2. Вивчити арифметичні оператори
3. Вивчити скорочену форму арифметичних операторів
4. Вивчити інкремент і декремент
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

Особливості арифметичних операторів

Арифметичні оператори використовуються для обчислень так само як в алгебрі (див. Таблицю зі зведенням арифметичних операторів нижче). Допустимі операнди повинні мати числові типи. Наприклад, використовувати ці оператори для роботи з логічними типами не можна, а для роботи з типом char можна, оскільки в Java тип char - це підмножина типу int.

|  |  |  |  |
| --- | --- | --- | --- |
| Оператор | Результат | Оператор | Результат |
| + | Додавання | += | Складання з привласненням |
| – | Віднімання (також унарний мінус) | – = | Віднімання з привласненням |
| \* | Множення | \*= | Множення з привласненням |
| / | Ділення | /= | Ділити з привласненням |
| % | Ділення по модулю | %= | Ділення по модулю з привласненням |
| ++ | інкремент | – – | Декремент |

Особливості укороченої форми арифметичних операторів.

Укорочена форма арифметичних операторів - це спеціальний спосіб запису арифметичних операцій в багатьох програмувальних мовах програмування, який дозволяє коротше і зрозуміліше виразити операції над змінними. Основні особливості укороченої форми арифметичних операторів включають:

Короткий синтаксис: Укорочена форма дозволяє виразити операцію, яка змінює значення змінної, в більш короткому вигляді, зменшуючи кількість коду, який потрібно писати.

Модифікація змінної: Укорочена форма операторів дозволяє виконати арифметичну операцію і одночасно змінити значення змінної, без потреби виразно вказувати змінну двічі (наприклад, x = x + 1 стає x++ в укороченій формі).

Прирост і декремент: Укорочена форма дозволяє легко збільшувати (інкремент) або зменшувати (декремент) значення змінних на 1 одиницю. Наприклад, x++ збільшує значення x на 1.

Підтримка різних арифметичних операцій: Укорочена форма може бути використана з різними арифметичними операціями, такими як додавання (+), віднімання (-), множення (\*), ділення (/) і т.д.

Підвищена читабельність: Укорочена форма може робити код більш зрозумілим, оскільки менше символів і операцій потрібно для виразу тієї ж логіки.

Можливість оптимізації: Деякі компілятори можуть оптимізувати укорочені форми арифметичних операцій для покращення швидкодії програми.

Загалом, укорочена форма арифметичних операторів дозволяє писати більш зрозумілий і компактний код для виразів, які використовують арифметичні операції.

Інкремент і декремент

Укорочена форма арифметичних операторів - це спеціальний спосіб запису арифметичних операцій в багатьох програмувальних мовах програмування, який дозволяє коротше і зрозуміліше виразити операції над змінними. Основні особливості укороченої форми арифметичних операторів включають:

Короткий синтаксис: Укорочена форма дозволяє виразити операцію, яка змінює значення змінної, в більш короткому вигляді, зменшуючи кількість коду, який потрібно писати.

Модифікація змінної: Укорочена форма операторів дозволяє виконати арифметичну операцію і одночасно змінити значення змінної, без потреби виразно вказувати змінну двічі (наприклад, x = x + 1 стає x++ в укороченій формі).

Прирост і декремент: Укорочена форма дозволяє легко збільшувати (інкремент) або зменшувати (декремент) значення змінних на 1 одиницю. Наприклад, x++ збільшує значення x на 1.

Підтримка різних арифметичних операцій: Укорочена форма може бути використана з різними арифметичними операціями, такими як додавання (+), віднімання (-), множення (\*), ділення (/) і т.д.

Підвищена читабельність: Укорочена форма може робити код більш зрозумілим, оскільки менше символів і операцій потрібно для виразу тієї ж логіки.

Можливість оптимізації: Деякі компілятори можуть оптимізувати укорочені форми арифметичних операцій для покращення швидкодії програми.

Загалом, укорочена форма арифметичних операторів дозволяє писати більш зрозумілий і компактний код для виразів, які використовують арифметичні операції.

Завдання 3.1**.** Розробити, відлагодити та запустити програму для обчислень (рис. 1 в “Методичних вказівках до виконання лабораторної роботи 3”).

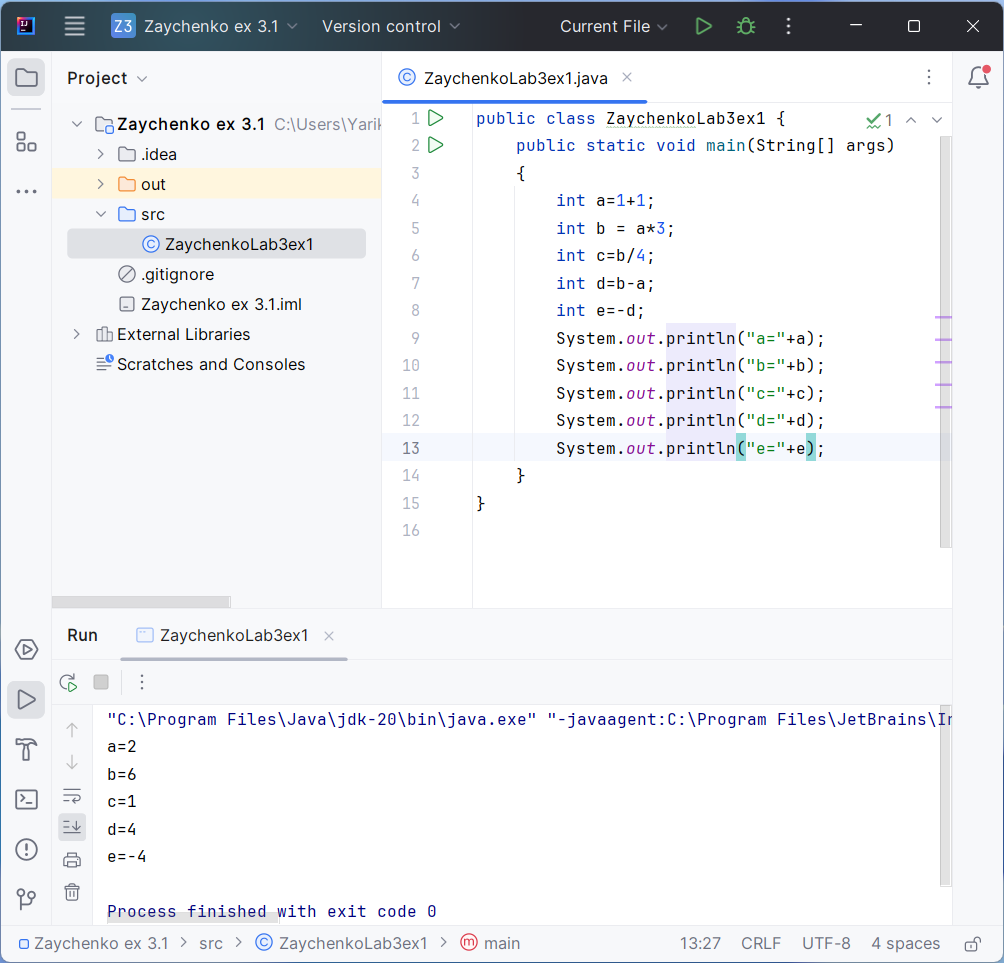


Рисунок 1 – результат виконання Завдання 1-3

Завдання 3.2

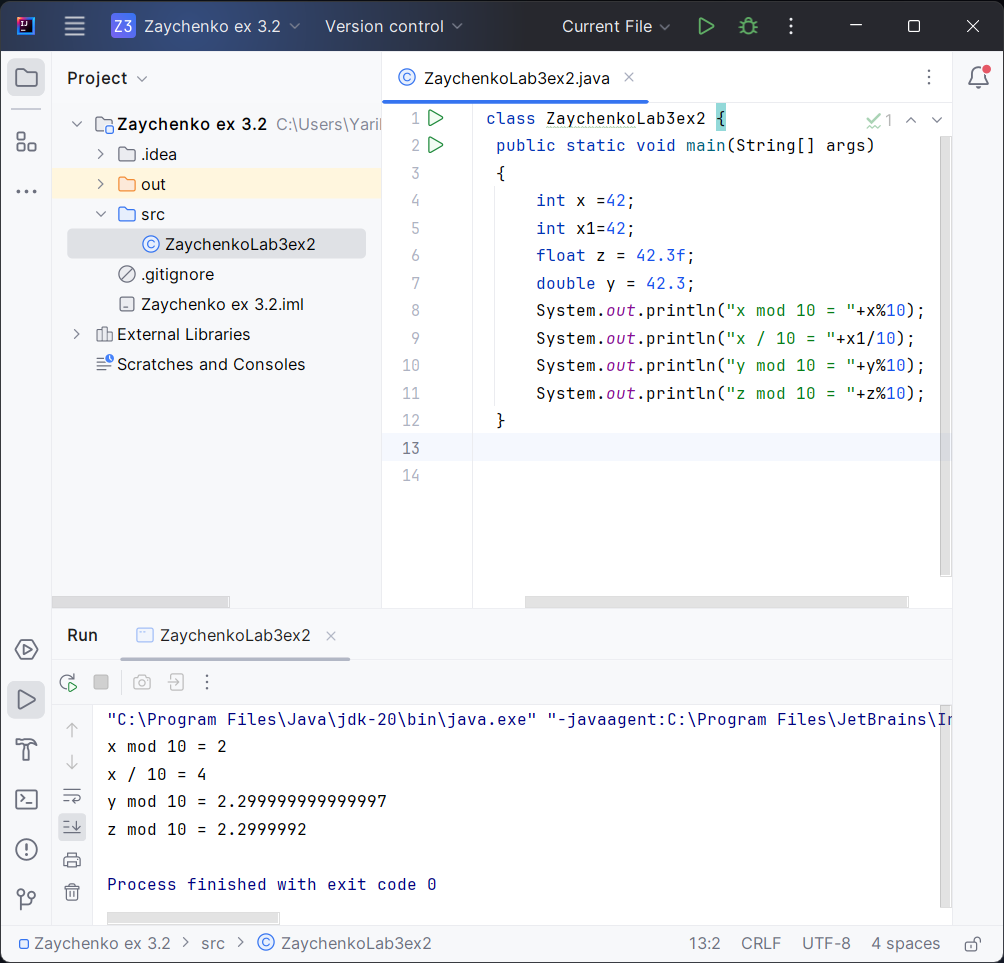


Рисунок 2 – Особливості використання арифметичних операторів, при роботі з дійсними типами

Завдання 3.3

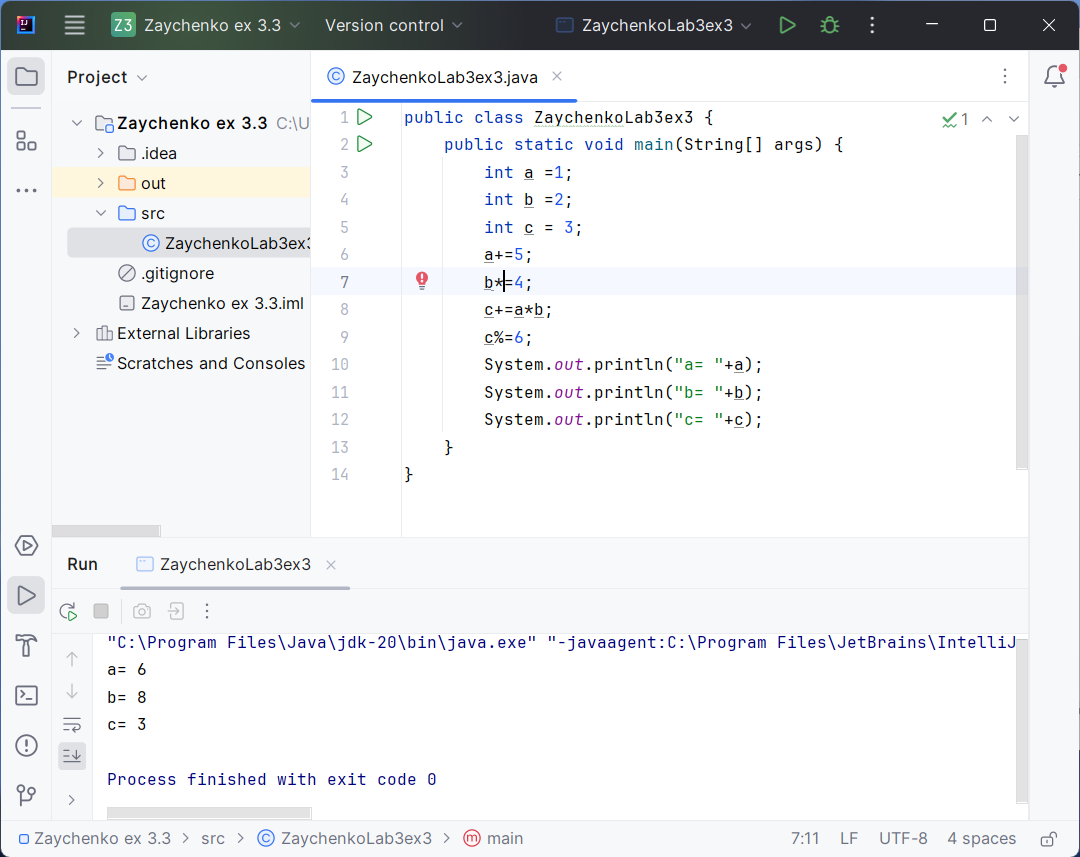


Рисунок 3 – використання укороченої форми арифметичних операторів

Завдання 3.4

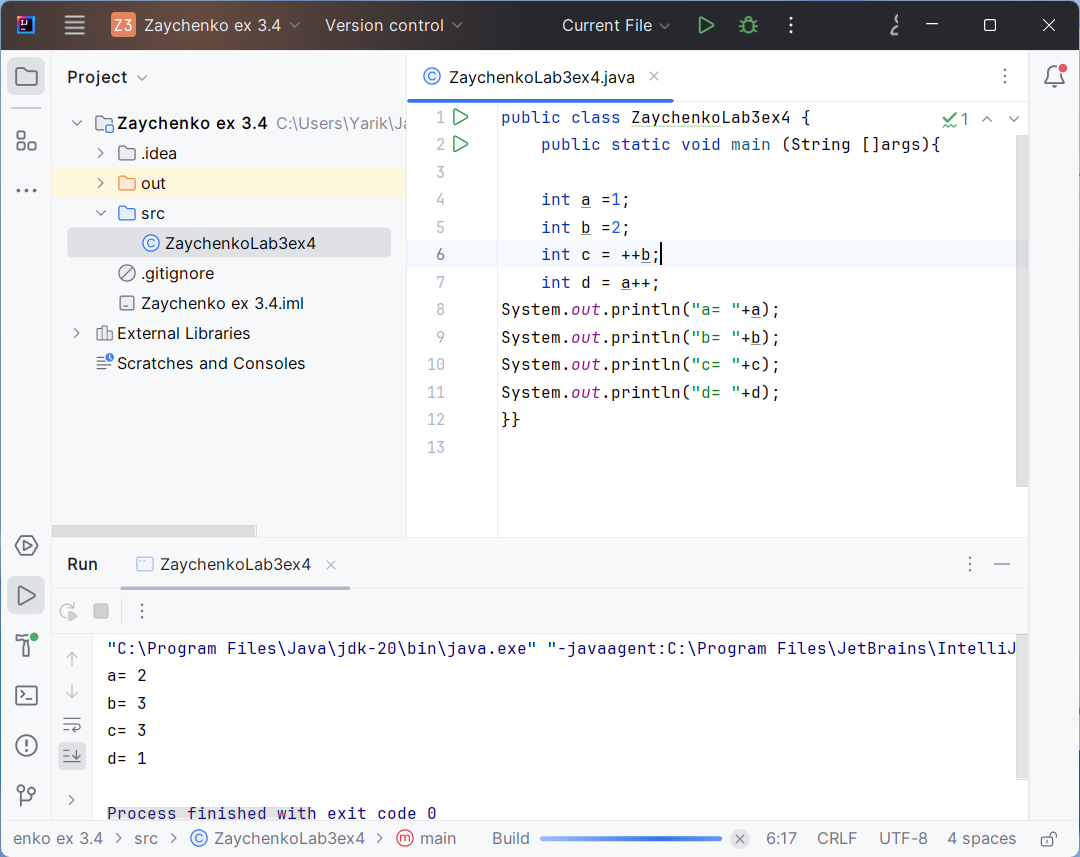


Рисунок 4 – особливість використання префіксних і постфіксних інкрементів

Завдання 3.5

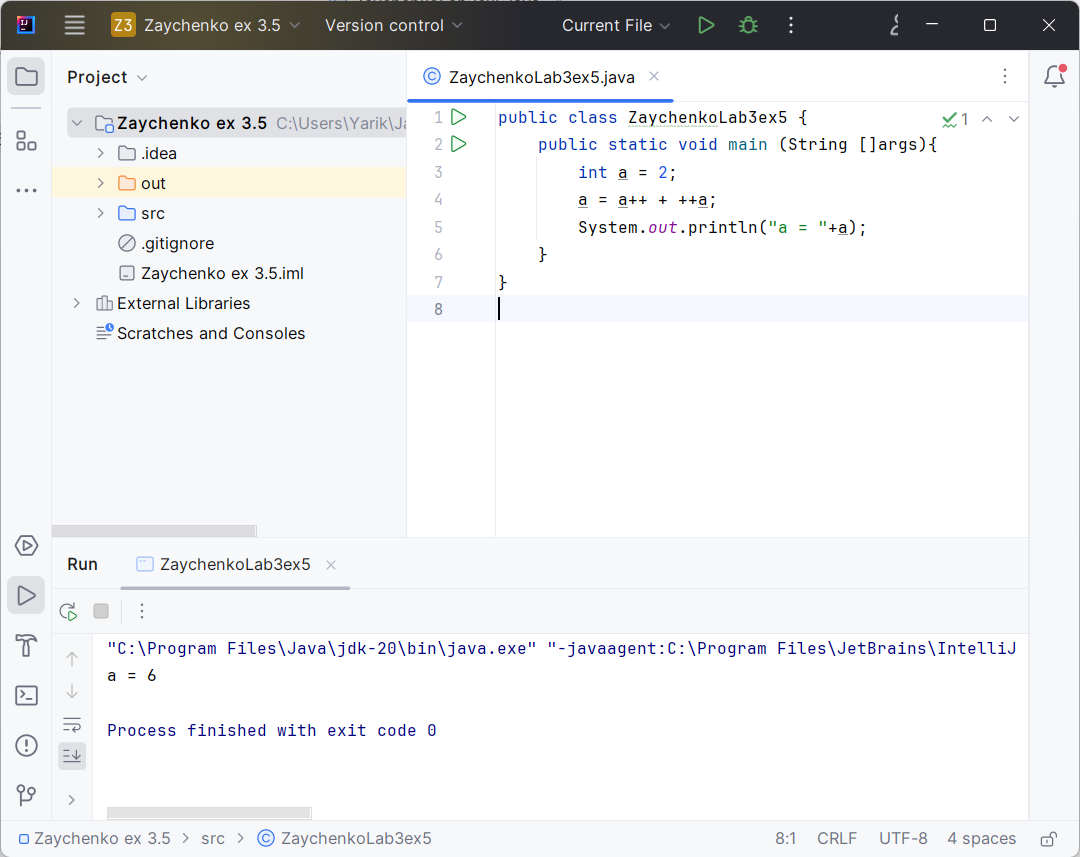


Рисунок 5 – особливість використання префіксних і постфіксних інкрементів

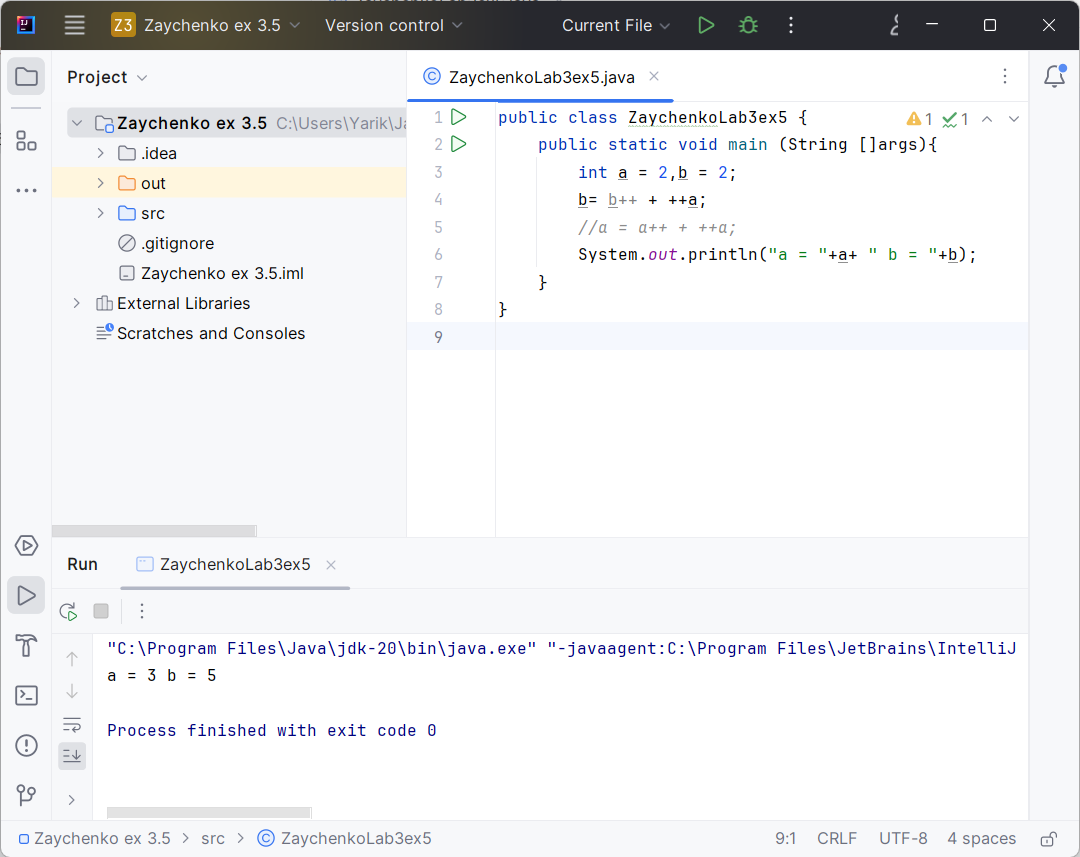


Рисунок 6 – особливість використання префіксних і постфіксних інкрементів

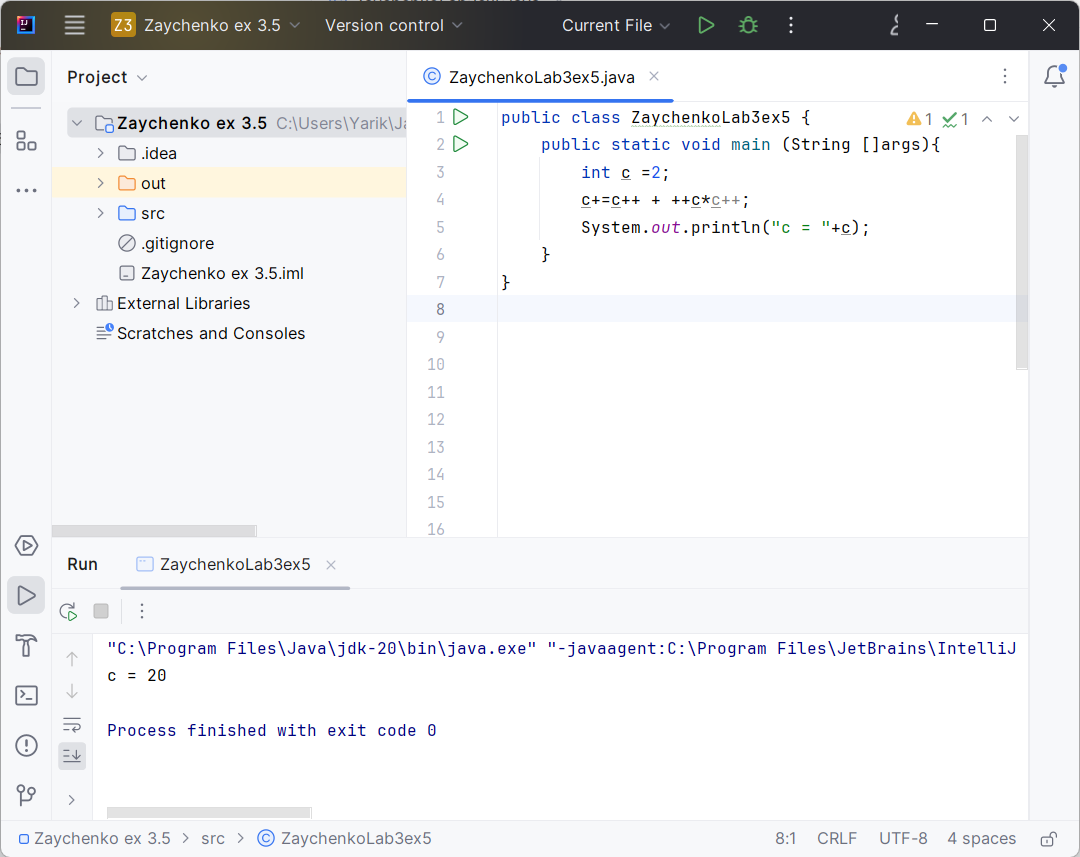
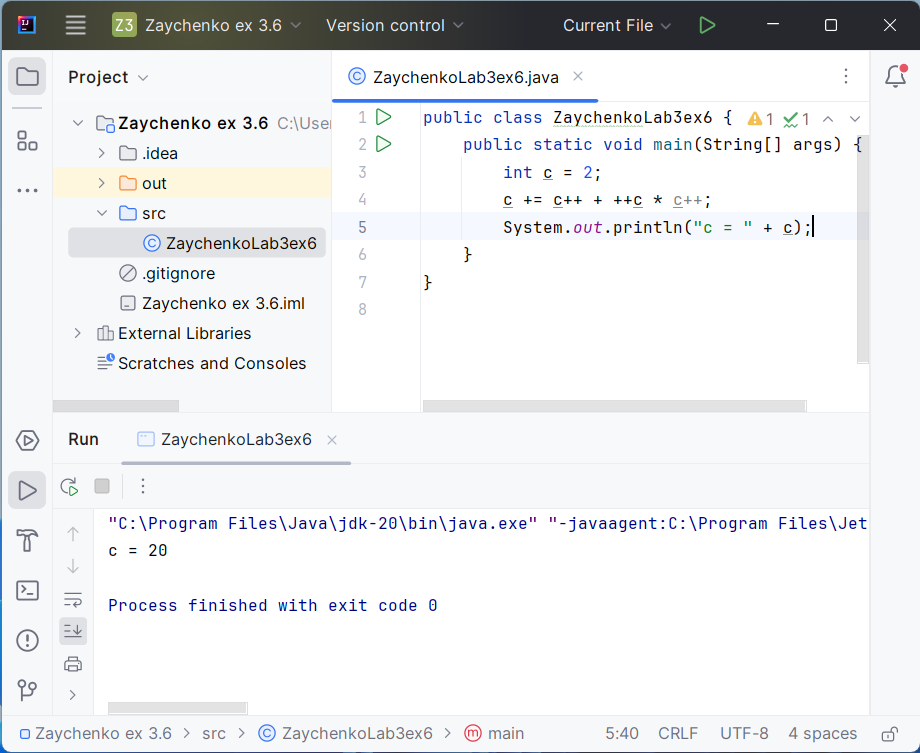


Рисунок 7 – особливість використання префіксних і постфіксних інкрементів

Завдання 3.6

Рисунок 8 – особливість використання префіксних і постфіксних інкрементів

# **ВИСНОВОК**

У ході виконання цієї лабораторної роботи отримано практичні навички роботи з арифметичними операторами мови програмування Java.

# **Контрольні питання**

1. Що таке арифметичний оператор?

Це операція, яка виконує математичні дії над числами або змінними, такі як додавання (+), віднімання (-), множення (\*), ділення (/) і т. д. Вони використовуються для виконання різних обчислень в програмах.

1. Які особливості використання арифметичних операторів?

Включають використання операторів для виразів, обрання правильного оператора для певного обчислення, розрізнення між цілими числами і числами з плаваючою точкою, обробку порядку операцій (за допомогою дужок) і обробку можливих помилок (наприклад, ділення на нуль).

1. Які особливості використання арифметичних операторів укороченої форми?

Включають можливість виразити операцію і одночасно змінити значення змінної (наприклад, інкремент x++ збільшує x на 1 і повертає попереднє значення x). Вони дозволяють писати більш зрозумілий і компактний код для арифметичних операцій.

1. Які особливості використання арифметичних операторів інкремента і декремента?

Включають можливість збільшити (інкремент) або зменшити (декремент) значення змінної на 1 одиницю. Інкремент може бути префіксним (++variable) або постфіксним (variable++), і вони поводяться по-різному щодо повернення значення до або після зміни.