Основи програмування

**Лабораторна робота №1**

**Тема:** Перше знайомство с Java

**Варіант:** Група ІА-01, варіант 4

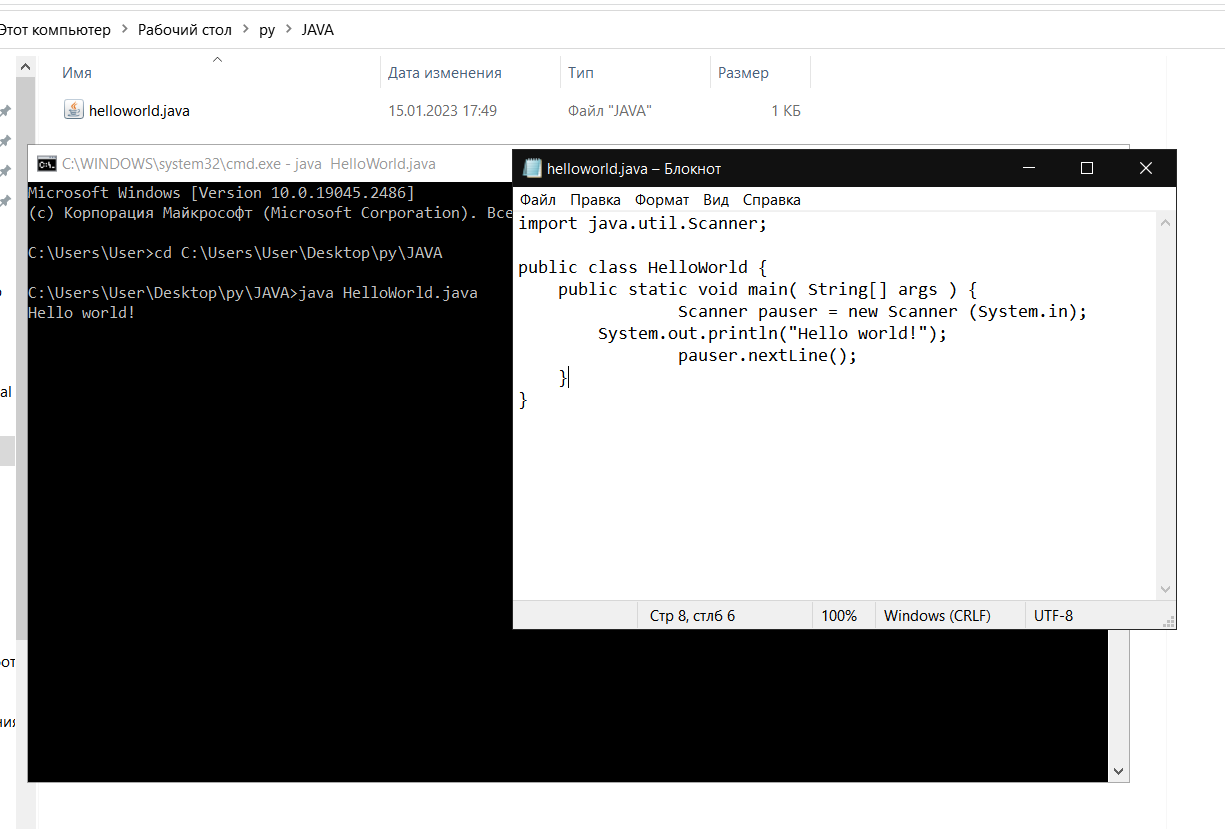
**Хід роботи:**

1. Встановити JDK

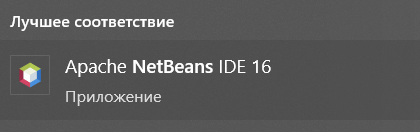
- не переплутайте з JRE

- не забудьте прописати в системі змінні PATH та JAVA\_HOME

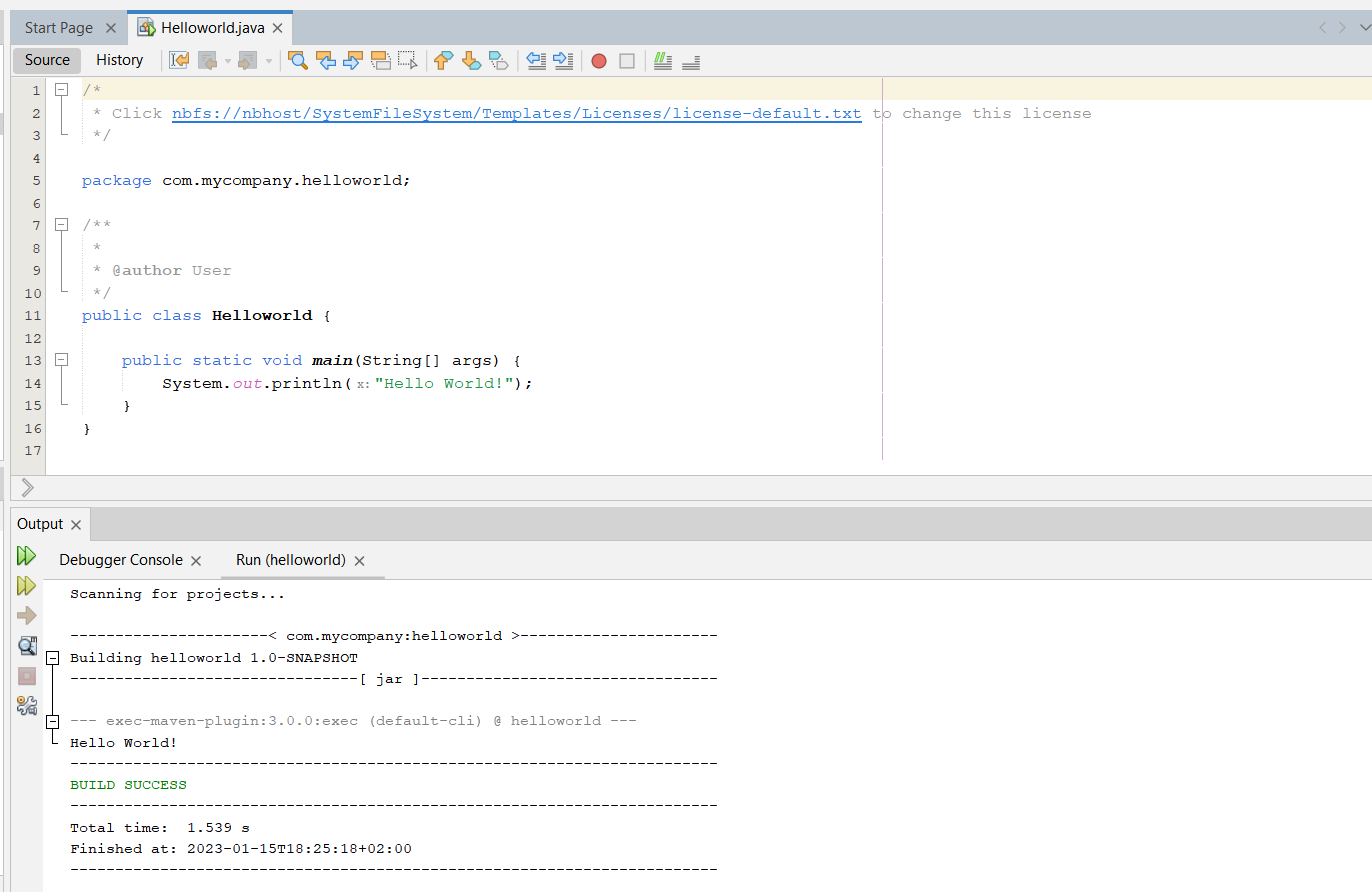
2. У консолі (без використання IDE) створити та запустити застосунок HelloWorld



3. Встановити IDE (NetBeans, Eclipse, JDeveloper або IntelliJ IDEA‎)



4. В IDE створити та запустити застосунок HelloWorld



**Контрольні питання:**

1. Компіляція та інтерпретація – це два різні методи виконання коду. Компіляція перетворює вихідний код на машинний код, який потім виконується безпосередньо процесором комп'ютера. Інтерпретація, з іншого боку, зчитує вихідний код рядково і перетворює його на машинний код на льоту, виконуючи код у міру його надходження. Перевага компіляції в тому, що вона призводить до швидшого виконання, оскільки код вже перетворено на машинний код. Перевага інтерпретації полягає в тому, що вона може бути більш гнучкою та простішою у налагодженні, оскільки вихідний код завжди доступний.

2. Вихідний код — це код, написаний розробником мовою програмування високого рівня, як-от Java. Байт-код – це низькорівневий код, який створюється під час компіляції вихідного коду. Це набір інструкцій, які виконуються віртуальною машиною Java (JVM). Машинний код — це код, який безпосередньо виконується процесором комп'ютера. Компіляція JIT (Just-In-Time) - це метод, що використовується JVM для підвищення продуктивності коду Java шляхом компіляції байт-коду в машинний код під час виконання, а не заздалегідь.

3. Віртуальна машина – це програмна емуляція фізичного комп'ютера, яка може виконувати код, написаний для цієї фізичної машини. Віртуальна машина Java — це конкретний приклад віртуальної машини, яка призначена для виконання байт-коду Java.

4. Кросплатформенність відноситься до здатності програмної програми або системи працювати в декількох різних операційних системах або платформах без модифікації. Програми Java зазвичай вважаються кросплатформовими, оскільки вони можуть працювати в будь-якій операційній системі, в якій встановлена JVM. Сама JVM також є кросплатформною, оскільки може бути реалізована у різних операційних системах.

5.

a. Клас: визначає план об'єкта.

b. public: вказує, що клас доступний всім інших класів.

c. static: вказує на те, що метод є методом класу, який можна викликати без створення екземпляра класу.

d. void: вказує, що метод не повертає жодного значення.

e. String: це клас у бібліотеці Java, що представляє послідовність символів.

f. System: це клас у бібліотеці Java, який забезпечує доступ до системної інформації та ресурсів.

g. out: це статична змінна класу System, що становить стандартний потік виведення.

h. println: це метод класу PrintStream, який друкує рядок тексту у стандартний потік виведення та переміщує курсор на наступний рядок.

6. Якщо змінити регістр літер в одному з цих слів, наприклад, написати "КЛАС" замість "клас", це викликає помилку компіляції, оскільки Java чутлива до регістру і ключове слово "клас" не збігається з "КЛАС"

7. В одній системі можна встановити кілька різних версій Java одночасно, але для цього потрібно додаткове налаштування та керування, щоб переконатися, що для кожної програми використовується правильна версія. Це може бути корисно, якщо для різних програм потрібні різні версії Java або якщо ви розробляєте та тестуєте код на кількох версіях. Однак, як правило, простіше використовувати одну версію Java та уникнути можливих конфліктів.