Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Науково-навчальний  інститут телекомунікаційних систем

Кредитний модуль

 «Основи програмування Java»

Лабораторна робота №1

Підготовка робочого середовища для програмування на Java

Виконав студент групи ТЗ-22

Єгоров Олексій Андрійович

Перевірив: Руренко Олександр

Григорович

17 вересня 2024 р.

Київ 2023

**Виконання роботи:**

1. Попередньо встановлено JDK.
2. Створив текстовий файл з розширенням .java, написав код програми Hello World, скомпілював та запустив програму.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

1. Створив новий файл lab1.java в IDE Eclipse, написав код, скомпілював та запустив програму.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Контрольні запитання:

1. Чим компіляція відрізняється від інтерпретації? Які переваги кожної з них?

Інтерпретація це коли код виконується одразу, рядок за рядком, а компіляція це коли код перетворюється в машинний код, а потім запускається. Плюси інтерпретації – гнучкість, легше тестувати код, плюси компіляції – швидкість виконання, можливість оптимізації компілятором.

1. Що таке початковий код, байткод, машинний код? Яка між ними різниця? Що таке JIT-компіляція?

Початковий код — це код, написаний мовою високого рівня

Байткод — це проміжний код, що генерується після компіляції

Машинний код — це код, що виконується безпосередньо процесором комп'ютера, специфічний для архітектури процесора.

JIT-компіляція (Just-In-Time) — це техніка, коли байткод компілюється в машинний код під час виконання програми, щоб забезпечити швидше виконання програм.

1. Що таке віртуальна машина?

Це програмне забезпечення, яке імітує апаратне забезпечення комп'ютера

1. Що таке кросплатформність? Чи є кросплатформними застосунки на мові Java? Чи є кросплатформною віртуальна машина Java?

Кросплатформність — це здатність програмного забезпечення працювати на різних операційних системах

Застосунки на Java є кросплатформними завдяки тому, що вони компілюються в байткод, який виконується на JVM.

JVM також є кросплатформною, оскільки вона доступна для різних операційних систем (Windows, macOS, Linux тощо), що дозволяє одному і тому ж байткоду виконуватись на будь-якій системі з встановленою JVM.

1. Дати пояснення кожному зі слів у початковому коді HelloWorld:

* **class** — ключове слово для оголошення класу. Клас є основним будівельним блоком у Java, який містить дані та методи.
* **public** — модифікатор доступу, що вказує на те, що клас або метод є доступним з будь-якої точки програми.
* **static** — означає, що метод або змінна належить класу, а не конкретному об'єкту цього класу.
* **void** — вказує, що метод не повертає значення.
* **String** — клас у Java, який представляє текстові рядки.
* **System** — клас, що містить корисні методи і змінні, які стосуються системних ресурсів.
* **out** — це статичний член класу System, який відповідає за виведення даних.
* **println** — метод, який друкує переданий йому рядок та переводить курсор на новий рядок.

1. Що буде, якщо в одному з цих слів змінити регістр літер, наприклад, замість «class» написати «CLASS»?

Компілятор видасть помилку. Java чутлива до реєстру.

1. Чи можна в одній системі встановити одночасно кілька різних версій Java? Якщо «ні» - пояснити чому. Якщо «так» - пояснити як, і взагалі для чого комусь таке робити?

Так, можна. Шляхом налаштування змінних середовища, або за допомогою jvm менеджерів. Це потрібно Це може бути корисно для розробників, які працюють з різними проектами, що потребують різних версій JVM.