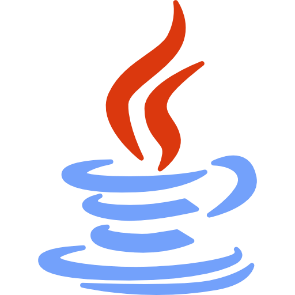
**Lesson 1. Get Started**

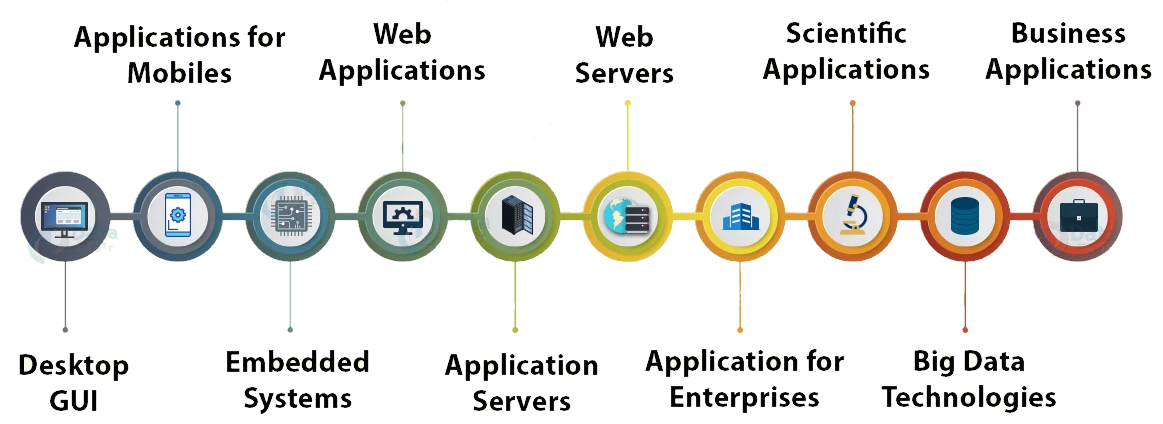
Java – мова програмування, створена в 1995 році, яка і досі входить у Топ 3 найпопулярніших мов програмування. Java належить компанії Oracle, і понад 3 мільярди пристроїв працюють на Java.

Пропонуємо з цього заняття почати занурення у мистецтво Java разом із Robot School. Ми проведемо вас звивистими стежками цієї прекрасної мови і обов’язково допоможемо, якщо буде важко!

**Why to learn Java?**

Питання одне а відповідей багато:

* Java можна використовувати для створення майже для будь-чого (мобільні застосунки, веб-сайти, програми для ПК та ігри);
* Java працює на різних платформах (Windows, Mac, Linux, Raspberry Pi тощо);
* Це одна з найпопулярніших мов програмування у світі;
* Java легко освоїти та використовувати;
* Java відкрита і безкоштовна мова;
* Java – безпечна, швидка та потужна;
* Java має величезну підтримку спільноти (десятки мільйонів розробників по всьому світу).

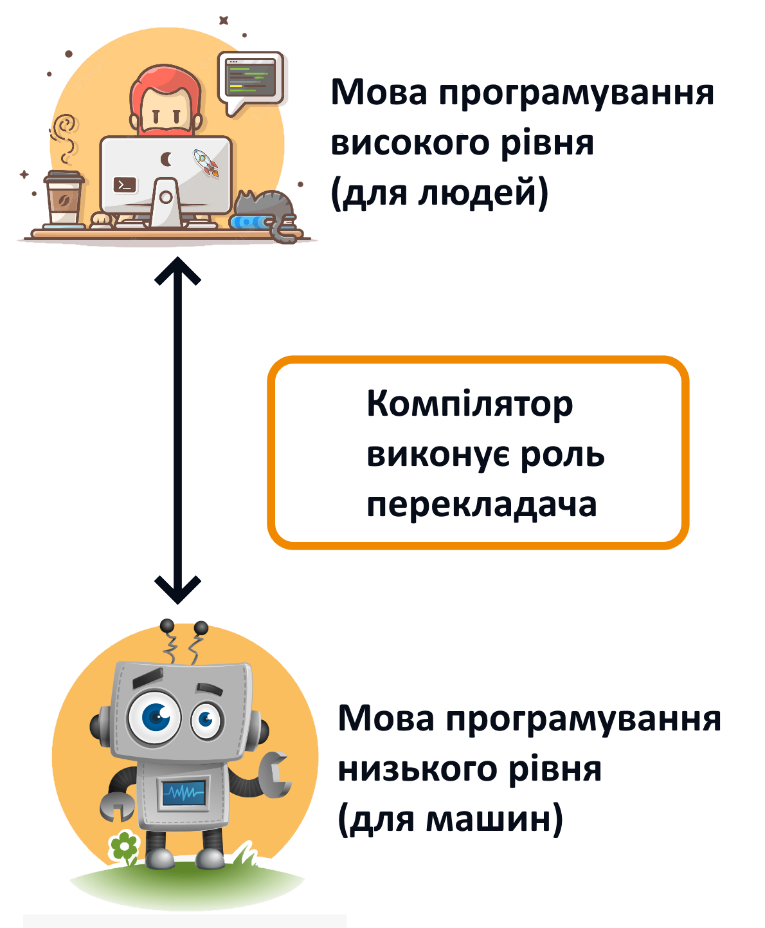


Крім того **Java** – це об’єктно-орієнтована мова, яка надає чітку структуру програмам і дозволяє повторно використовувати код, знижуючи час на його розробку.

Оскільки Java близька до C++ і C#, це дозволяє програмістам легко переходити з багатьох інших мов на Java або навпаки.

**How talk to machine?**

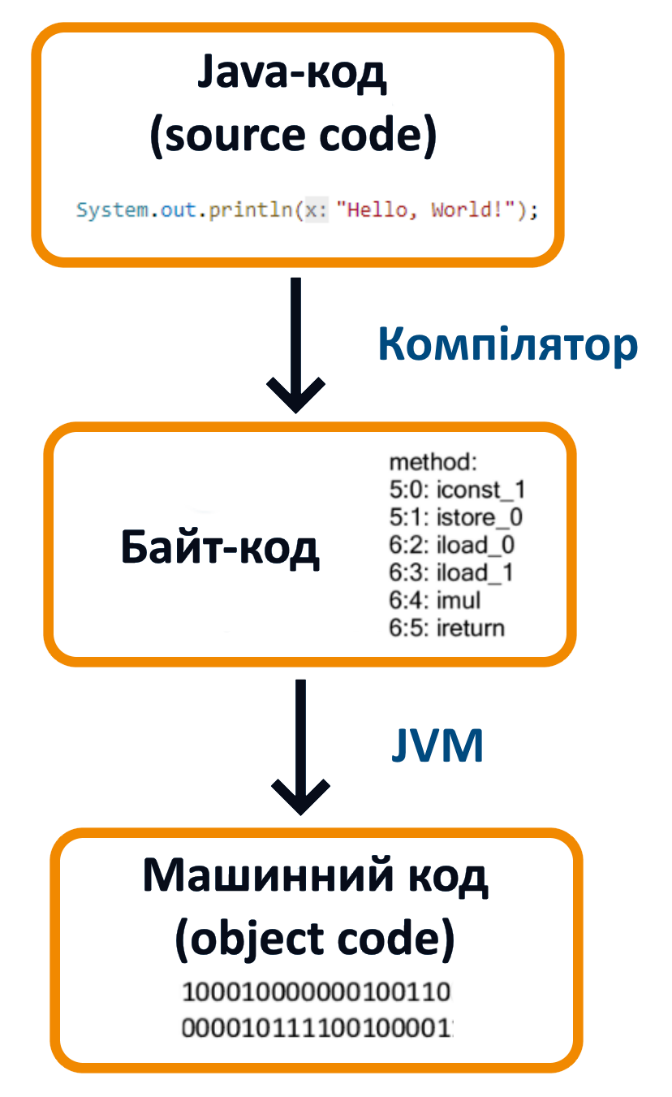
Люди можуть спілкуватись із комп’ютерами за допомогою мов програмування. Так, саме для цього вони і потрібні. Якщо ви хочете, змусити ваш комп’ютер робити що-небудь, ви повинні сказати йому про це, а також пояснити як саме ви хочете щоб це було зроблено, що означає – написати програму.

Мов програмування є багато, і ми можемо розділити їх на дві великі категорії:

**Мови програмування високого рівня** – ті, на яких люди пишуть програми, у тому числі Java та багато інших. Машини напряму не розуміють цих мов.

**Мови програмування низького рівня** – ті, які може читати машина і які дуже складно читати людям (наприклад мова асемблера, чи машинний двійковий код).

Для того, щоб було машина могла зрозуміти код написаний програмістом мовою високого рівня, використовуються спеціальні програми, які називають компіляторами.

**Компілятори** – це програми, які перетворюють код програми високого рівня, у машинний код, або іншу мову програмування низького рівня.

**JDK, JRE, JVM**

Тепер розглянемо як спілкування з машиною відбувається мовою Java. У випадку із Java ми не одразу перетворюємо написаний Java-код у машинний код, а спочатку у **байт-код** – код низького рівня, що створюється компілятором.

Байт-код є проміжним етапом, для його читання і відтворення використовується ще одна програма – **інтерпретатор**. Інтерпретатор перетворює байт-код у машинний, вже у процесі його виконання.

Виконання байт-коду Java відбувається за допомогою програми, що має назву JVM (Java Virtual Machine). **JVM** є частиною **JRE** (Java Runtime Environment), яке входить до **JDK** (Java Development Kit).

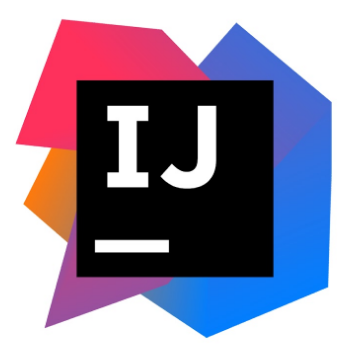
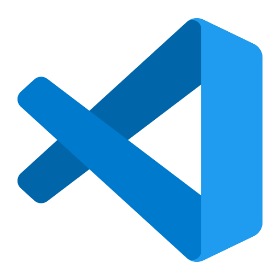
Що означають всі ці абревіатури ми не будемо розбирати детально, а те, що нам необхідно про них пам’ятати можна знайти в ілюстрації на наступній сторінці.

Для того, що почати писати код на Java нам необхідна одна остання річ, а саме **IDE** (так, знову абревіатура – Integrated Development Environment).



**When do we start to write code?**

Якщо коротко, то це програма, яка дозволить нам зручно створювати (писати, редагувати, перевіряти, тестувати) код. Серед IDE у яких можна створювати програми на Java можемо виділити **VS Code**, **Eclipse**, та **IntelliJ IDEA**.

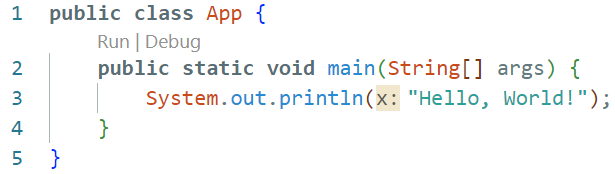


У майбутньому ви зможете протестувати кожне із середовищ і вибрати будь-яке, але протягом даного курсу ми будемо користуватись VS Code.

**Hello World!**

На даному етапі уроку ми вже маємо встановлений VS Code та JDK. Тепер можемо перейти до написання першої програми.

Традиційно першою програмою є простий код, що дозволяє вивести на екран текст із привітанням: *“Hello World!”*. Для написання коду нам необхідно спочатку створити новий проєкт-Java, і в головний файл (*ProjectName.Java*) помісити наступний код:



**Завдання**.Розгляньте наступний код. Схожий на код попередньої програми, чи не так?

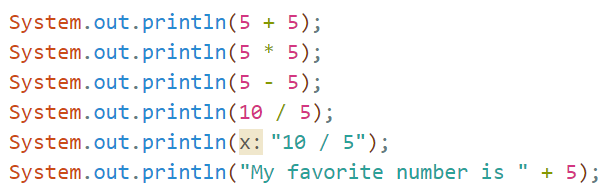
**Print and comment**

Зверніть увагу на сірий текст. Це – **коментарі**, їх компілятор не сприймає як код. Коментарі потрібні для того, аби додавати у код помітки, які полегшують його читання і розуміння.

Створіть подібний код зі своїм іменем, поекспериментуйте з коментарями та з’ясуйте, як утворюються **однорядкові** та **багаторядкові** коментарі, а також чим відрізняються функції print() та prtinln().

**Tasks & Examples**

**Завдання 1**. Запрограмуйте наступні рядки виведення. Проаналізуйте код і зробіть висновки, щодо роботи кожного рядка. Прокоментуйте ті з них, результати яких здивували вас.





**Завдання 2**. Виведіть у консоль наступний текст у трьох рядках:

**Завдання 3**. Виведіть у консоль результат обчислення наступного прикладу ((148+99) \*17-45) \* 3. Програма сама повинна обчислювати результат.

**Homework**

Завдання 1. Встановіть на власному комп’ютері все необхідне для роботи із Java (JDK, VS Code та необхідні розширення до нього). Протестуйте успішність встановлення найпростішою програмою «Hello, World!».