



Java Starter

Знайомство з мовою Java

Java Starter

Після уроку обов'язково



Повторіть цей урок у відеоформаті на [ITVDN.com](http://itvdn.com)

Доступ можна отримати через керівництво вашого навчального центру



Перевірте, як Ви засвоїли цей матеріал на [TestProvider.com](http://testprovider.com)

Знайомство з мовою Java

Мова програмування JAVA

JAVA Programming Language

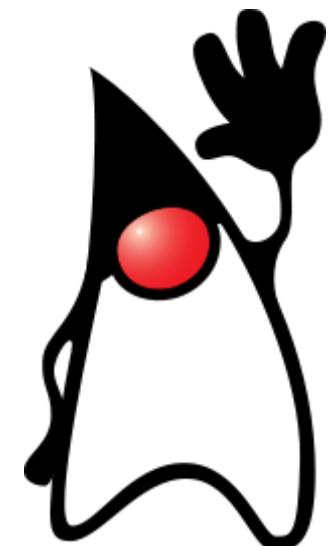


JAVA (*вимовляється Джава*) – об'єктоорієнтована мова програмування, яка розроблена компанією Sun Microsystems (надалі придбаною компанією Oracle).

Зазвичай застосунки Java транслуються у спеціальний байт-код, тому вони можуть працювати на будь-якій віртуальній Java-машині незалежно від комп'ютерної архітектури. Дата офіційного випуску – 23 травня 1995 року.

JAVA належить до сімейства мов із С-подібним синтаксисом, її синтаксис найбільш близький до C++ і C#.

Назва «Джава» походить від сорту кави Java, яка вирощується на однійменному острові Java. В українській мові прийнята транскрипція назви острова «Ява», але в англійській мові загальноприйнята транскрипція «Джава».



Джеймс Гослінг

James Gosling



Джеймс Гослінг (англ. *James Gosling*; 19 травня 1955 року, Калгарі, Канада) – автор об'єктоорієнтованої та кросплатформної мови програмування Java. Крім цього, він є творцем віконної системи News, Gosling Emacs, а також був одним з розробників Star Seven.

У 1977 році закінчив університет Калгарі зі ступенем бакалавра інформатики, а вже в 1983 році здобув ступінь доктора в університеті Карнегі-Меллон.

З 1984 року працював у Sun Microsystems.

З 28 березня 2011 року Джеймс Гослінг почав працювати в [Google](#).

Наприкінці серпня 2011 року в стрічках новин пройшло повідомлення, що Гослінг залишив інтернет-гіганта та перейшов працювати в стартап, невелику фірму Liquid Robotics, яка займається розробкою робототехніки для дослідження океану.

Можливості мови JAVA

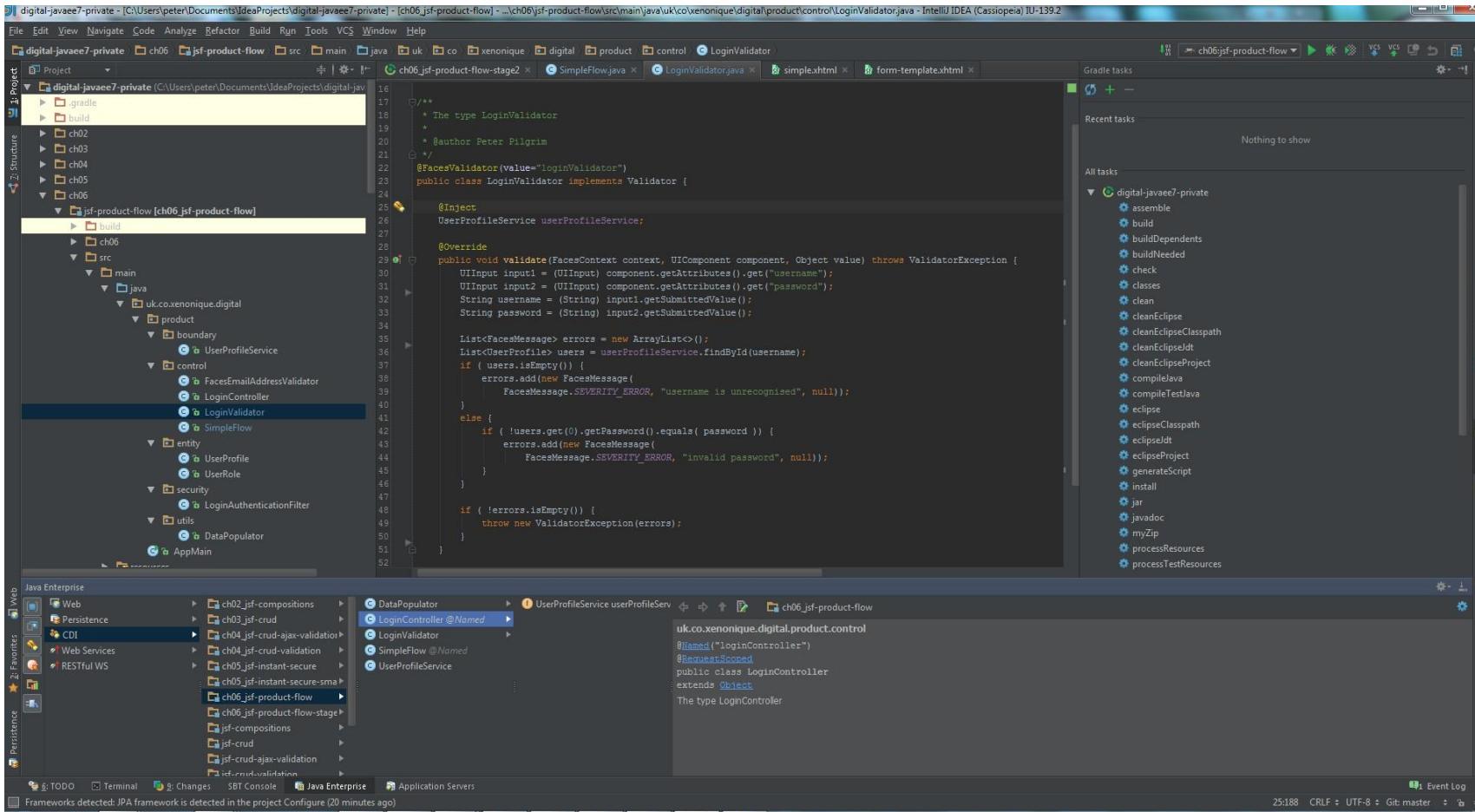
Possibilities of JAVA Language

Використовуючи мову Java, можна створювати такі програми:

- настільні застосунки;
- web-застосунки;
- web-сервіси;
- програми для мобільних пристройів;
- ігри;
- рішення для бізнесу;
- програми для мікроконтролерів та аплети для смарткарт;
- інтерактивний контент для Blu-ray.

Настільні застосунки

Desktop application

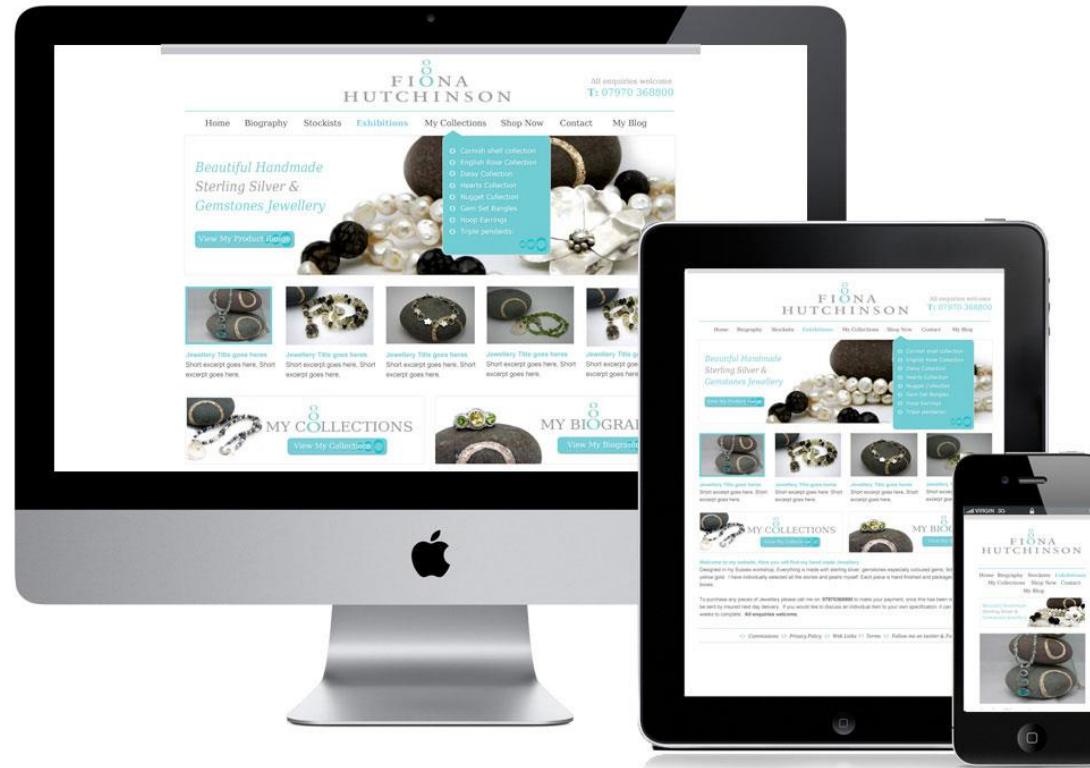


The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The project is named "digital-javee7-private". The "ch06_jsf-product-flow" module is selected. Inside "src/main/java", there is a package "uk.co.xenonique.digital.product.control". A file "LoginValidator.java" is open, showing Java code for a Validator implementation.
- Code:** The code defines a class `LoginValidator` that implements `Validator`. It uses `@Inject` to inject `UserProfileService`. The `validate` method checks if the username exists and if the password is correct.
- Java Enterprise Tools:** The bottom-left panel shows the Java Enterprise structure, with "CDI" selected. It lists various beans: `DataPopulator`, `UserProfileService`, `LoginController @Named`, `LoginValidator`, `SimpleFlow @Named`, and `UserProfileService`.
- Toolbars and Status Bar:** The top has standard IDE toolbars. The status bar at the bottom right shows "25188 CRLF - UTF-8 Git: master".

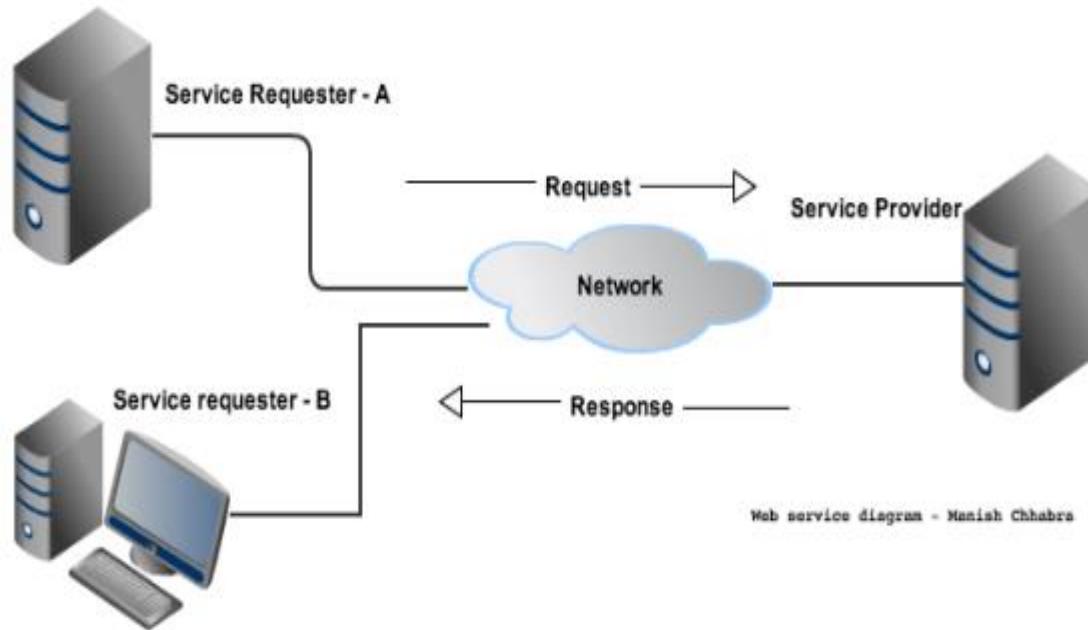
Web-застосунки

Web-application



Web-сервіси

Web-services



Програми для мобільних пристрій

Mobile applications



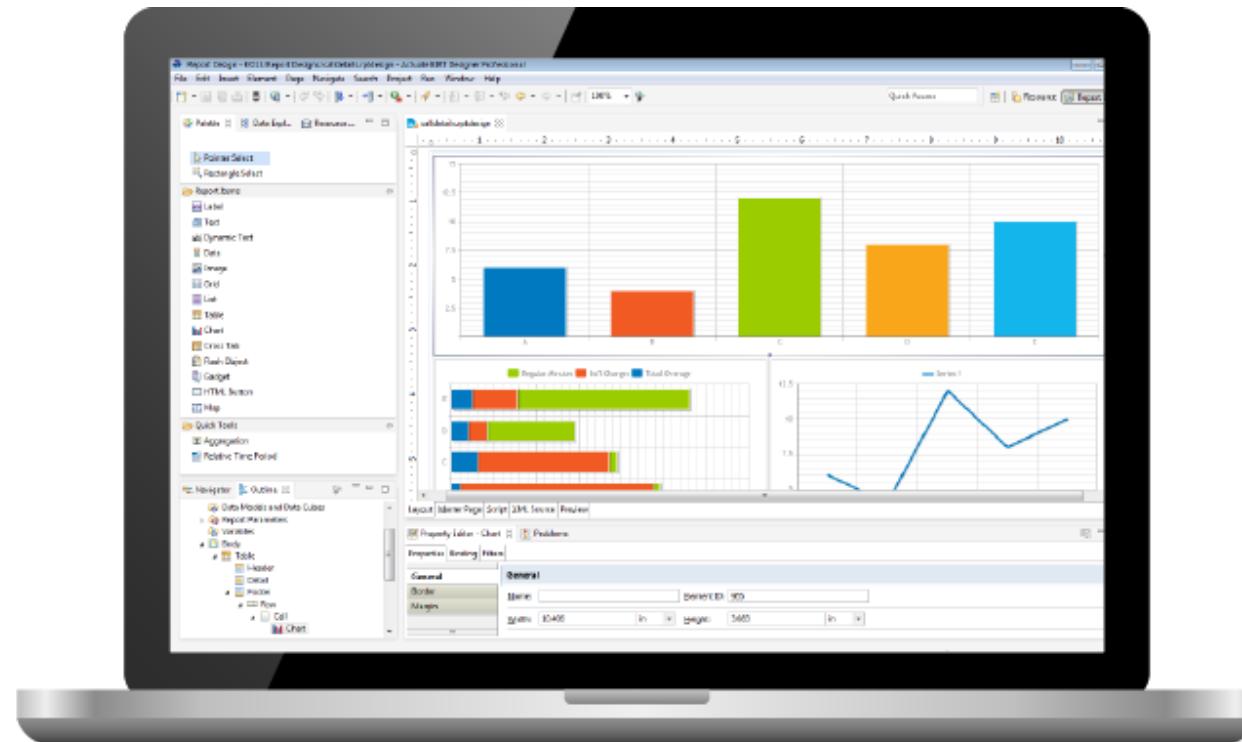
Ігри

Games



Рішення для бізнесу

Business applications



Аплети для смарткарт

Applets for smart cards



Интерактивный контент для Blu-ray

Interactive content for Blu-ray



Переваги мови JAVA

Advantages of JAVA Language

JAVA – це об'єктоорієнтована, проста та водночас потужна мова програмування, яка дає змогу розробникам створювати багатофункціональні програми.

JAVA – платформно-незалежна мова. Це означає, що програма працює на будь-якій платформі, де є віртуальна машина Java. І тому є теза: «Write once, run anywhere».

JAVA відрізняється розширеними можливостями обробки виняткових ситуацій, автоматичним керуванням пам'яті.

JAVA поєднує найкращі ідеї сучасних мов програмування: C++, Visual Basic тощо.

Що таке JRE, JDK, JVM?

JRE, JDK, JVM

Java Runtime Environment (JRE) – для роботи – мінімальна реалізація віртуальної машини, яка необхідна для виконання Java-застосунків, без компілятора й інших засобів розробки. Складається з віртуальної машини – Java Virtual Machine та бібліотеки Java-класів.

Java Development Kit (JDK) – для програмування – безоплатно розповсюджуваний комплект розробника застосунків на мові Java, який вміщує компілятор Java (javac), стандартні бібліотеки класів Java, приклади, документацію, різні утиліти та виконавчу систему Java (JRE).

Java Virtual Machine (Java VM, JVM) – віртуальна машина Java – основна частина виконувальної системи Java, так званої Java Runtime Environment (JRE). Віртуальна машина Java інтерпретує байт-код Java, який попередньо створений із вихідного тексту Java-програми компілятором Java (javac). JVM також може використовуватися для виконання програм, написаних іншими мовами програмування.

Для чого призначений JRE, JDK, JVM ?

JRE, JDK, JVM

- *JRE* необхідний для запуску Java-програм (містить у собі *JVM*).
- *JDK* необхідний для розробки (це компілятор, налагоджувач тощо).
- *JDK* та *JRE* містять *JVM*, яка потрібна для запуску програм на Java.
- *JVM* є серцем мови програмування Java та забезпечує незалежність від платформи.

Середовище розробки Java

Integrated development environment

- Eclipse
- IntelliJ IDEA
- NetBeans

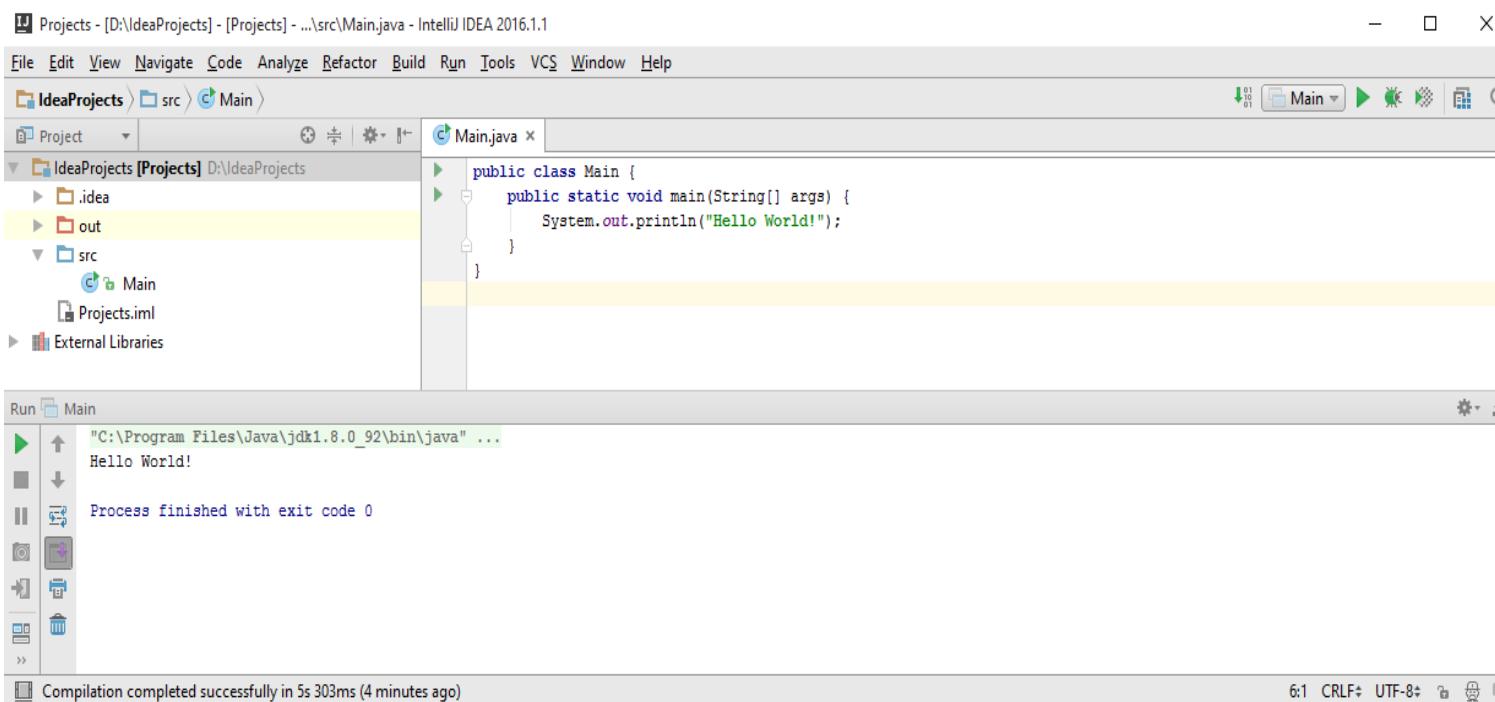
- Підсвічування синтаксису
- Автодоповнення, гіперпосилання
- Рефакторинг
- Інтерактивний налагоджувач



Огляд середовища розробки IntelliJ IDEA

Overview of IDE JetBrains IntelliJ IDEA

JetBrains IntelliJ IDEA – це програмне забезпечення, яке розроблене компанією JetBrains, щоб писати програми на JAVA та інших мовах.



Створення програми мовою JAVA

Java Program Development

За допомогою програм, які називають компіляторами, комп'ютери можуть перекладати вихідні коди програм з мови програмування на машинну мову.

Процес створення програми вміщує кілька етапів:

1. Написання програми мовою програмування JAVA (*наприклад, Hello.java*).
2. Перетворення кожного класу програми за допомогою компілятора на байт-код (*наприклад, Hello.class*).
3. У разі виявлення у програмі помилки компілятор повідомляє вам про це. Тоді необхідно виправити програму та повернутися до етапу 2.
4. Запуск програми (*часто через різні логічні помилки програма може виявитися непрацездатною. У цьому випадку необхідно переглянути та виправити її, а потім повторити етапи 1-4*).

Знайомство з методами

`System.out.print();` – Метод print() – друкує рядок.

The screenshot shows a Java code editor with the following code:

```
3 ► public class Test {  
4 ►   public static void main(String[] args) {  
5     System.out.print("Hello");  
6     System.out.print("World");  
7   }  
8 }
```

Below the code is a 'Run' window with the following details:

- Run configuration: `Test`
- Command: `C:\Users\Admin\.jdks\openjdk-17.0.2\bin\java`
- Output:
 - Process started
 - Output: `HelloWorld` (highlighted with a red arrow)
 - Process finished with exit code 0

`System.out.println();` – Метод println() – друкує рядок, а потім завершує рядок.
In розшифровується як line.

The screenshot shows a Java code editor with the following code:

```
3 ► public class Test {  
4 ►   public static void main(String[] args) {  
5     System.out.println("Hello");  
6     System.out.println("World");  
7   }  
8 }
```

Below the code is a 'Run' window with the following details:

- Run configuration: `Test`
- Command: `C:\Users\Admin\.jdks\openjdk-17.0.2\bin\java`
- Output:
 - Process started
 - Output: `Hello` (highlighted with a red arrow)
 - Output: `World` (highlighted with a red arrow)
 - Process finished with exit code 0

Q&A

Інформаційний відеосервіс для розробників програмного забезпечення

