

Введення в мову програмування JAVA.

Категорії програм Java

- **Додаток** (*application*) — аналог звичайної прикладної програми
- **Аплет** (*applet*) — спеціалізована програма з обмеженими можливостями, що працює у вікні WWW-документа під управлінням браузера.
- **Мидлет** (*midlet*) — спеціалізована програма з обмеженими можливостями, що працює на мобільному пристрої.
- **Сервлет** (*servlet*) — спеціалізована програма з обмеженими можливостями, що працює в WWW на стороні сервера.
- **Серверний додаток** (*Enterprise application*) — призначений для багаторазового виконання на стороні сервера.
- **Бібліотека** (*Java Class Library* — бібліотека класів, або *NetBeans Module* - модуль платформи NetBeans) - призначена для багаторазового використання програмами Java

Способи установки Java-машини на клієнтський комп'ютер:

- JRE з поставки **Java Development Kit** (JDK) - комплект розробки програмного забезпечення на мові Java. Як синонім може вживатися термін SDK - Software Development Kit (комплект розробки програмного забезпечення);

<https://www.oracle.com/java/>

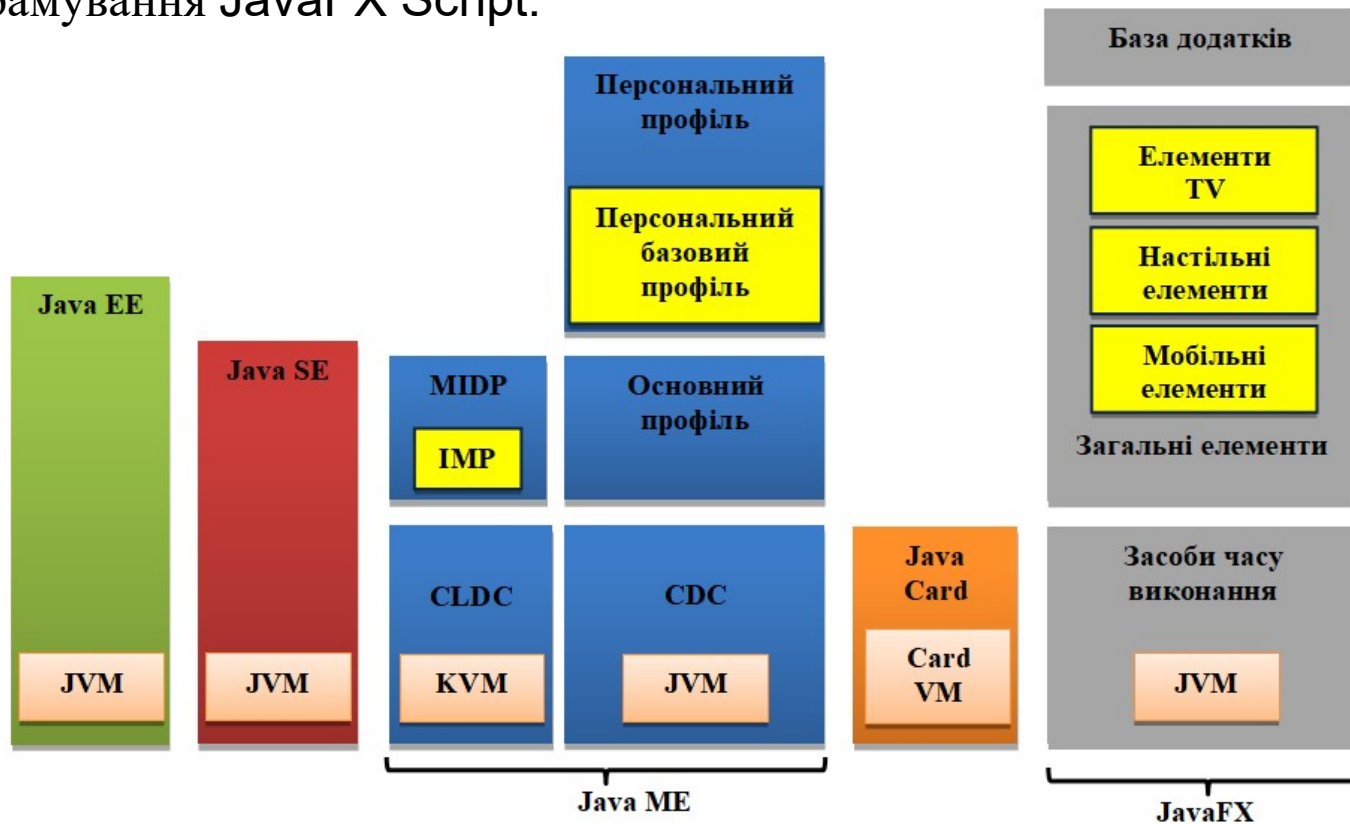
- Спеціалізований варіант JRE в складі інтернет-браузера, який має назву **Java plugin**.

<http://www.java.com/>

За застосовуваними засобами розробки і умовами виконання додатків Java розрізняють платформи:

- **Мікровидання** — Java Platform, Micro Edition (**Java ME**) для програмування «тонких апаратних клієнтів». Для розробки додатків застосовується не JDK, а спеціальний набір інструментів (Sun Java Wireless Toolkit for CLDC, Sun Java Toolkit for CDC, NetBeans 6.5, і т.д.).
- **Стандартне видання** — Java Platform, Standard Edition (**Java SE**) для програмування «товстих клієнтів». Основою засобів розробки служить JDK.
- **Корпоративне видання** — Java Platform, Enterprise Edition (**Java EE**) для написання серверного програмного забезпечення. Є надбудовою над платформою Java SE. У SDK даної платформи входить потужний програмний сервер Sun Java System Application Server (сервер додатків Java), а також велика кількість інших виконуючих середовищ, заснованих на технологіях Java.

- **Java Card** – платформа, яка використовується для створення прикладних програм, що працюють на смарт-картах та інших пристроях з дуже обмеженим обсягом пам'яті і можливостями обробки;
- **JavaFX** – платформа створення прикладних програм з графічним інтерфейсом для настільних комп'ютерів, мобільних і телевізійних пристроїв за допомогою декларативної мови програмування JavaFX Script.



Інтегровані середовища розробки для Java

- NetBeans** Безкоштовне середовище з відкритим вихідним кодом.
- JDeveloper** Безкоштовне середовище, орієнтоване на роботу з базами даних Oracle на основі технологій Java EE, але може використовуватися як універсальне середовище розробки для Java.
- Eclipse** Безкоштовне середовище з відкритим вихідним кодом. Є основою (платформою) для багатьох комерційних середовищ розробки
- JBuilder 2008** Є три редакції - безкоштовна (*JBuilder 2008 Turbo*) і дві комерційні (*JBuilder 2008 Professional* і *JBuilder 2008 Enterprise*).
- IntelliJ IDEA** Комерційне середовище. Доступне в двох версіях: безкоштовна *Community Edition* і комерційна *Ultimate Edition*. Дуже популярне в західних країнах.

<https://www.oracle.com/ua/java/>

oracle.com/ua/java/

Oracle Україна >

Продукти Ресурси Підтримка

Переглянути облікові записи

Download Java

Oracle Java

Java is the #1 programming language and development platform. It reduces costs, shortens development timeframes, drives innovation, and improves application services. With millions of developers running more than 45 billion Java Virtual Machines worldwide, Java continues to be the development platform of choice for enterprises and developers.

Protect your Java investment
Manage your Java SE installations, updates, and upgrades across your enterprise more cost effectively with an Oracle Java SE Subscription.

Get support

Live webcast: Application Development Innovation with Oracle Java 15
August 20, 2020, 11:00 a.m. PT—Learn about many of the Java ecosystem improvements and join the live discussion.

Register for the webcast

Continuous innovation from the leaders of Java

<https://netbeans.org/>

The screenshot shows the Apache NetBeans website homepage. At the top, there is a navigation bar with the NetBeans logo and links for Apache NetBeans, Plugins, Docs & Support, and Community. A search bar is located on the right side of the navigation bar. Below the navigation bar, the main content area features a large blue banner with the text "Apache NetBeans Fits the Pieces Together". To the right of this text is a "NEW!" starburst graphic and the text "Apache NetBeans IDE". Below the banner, there are two orange buttons labeled "Learn More" and "Download".

Below the banner, there is a "Featured News" section with the text "Apache NetBeans (incubating) 10 Released" and a "See All News" link. Below this, there are three featured news items:

- Best Support for Latest Java Technologies**: Accompanied by an image of the NetBeans IDE interface showing Java 8 and JavaFX logos. A "More" link is below the image.
- Fast & Smart Code Editing**: Accompanied by an image of the NetBeans IDE interface showing a code editor with a tooltip that says "Overridable method call in constructor". A "More" link is below the image.
- Join In: Moving to Apache!**: Accompanied by the Apache Software Foundation logo, which includes a feather icon and the text "THE APACHE SOFTWARE FOUNDATION". A "More" link is below the logo.

At the bottom of the page, there is an "About Us" link on the left and social media icons for Facebook, YouTube, Google+, and Twitter on the right.

Вміст папки JDK

- **bin** — містяться файли інструментів розробки;
 - **demo** — файли прикладів з вихідними кодами;
 - **include** — заголовки файлів C для доступу до ряду бібліотек Java та отладчику віртуальної Java-машини на платформи-залежному рівні - на основі інтерфейсів JNI (Java Native Interface) і JVMDI (Java Virtual Machine Debugging Interface) відповідно;
 - **jre** — файли, необхідні для роботи з віртуальною Java-машиною;
 - **lib** — ряд бібліотек і супровідних файлів для роботи інструментів з папки **bin**;
 - **sample** — приклади з вихідними кодами.
- src.zip** — файли з вихідними кодами стандартних класів Java.

Оформлення програми на мові Java

Програма на мові Java записується в текстовому вигляді і складається з введених елементів. Програма розбивається на рядки.

Введені елементи можуть бути наступних типів: пробільні символи, коментарі та елементи програми (tokens).

Елементами програми є: ключові слова, ідентифікатори, константи (літерали), роздільники і операції.

Символи верхнього і нижнього регістра в ключових словах й ідентифікаторах вважаються різними символами.

Алфавіт мови Java

Латинські літери ASCII:

ABCD...XYZ — заголовні;

abcd...xyz — рядкові.

Додаткові літерні символи ASCII:

_ — знак підкреслення;

\$ — знак долара.

Національні літери на прикладі українського алфавіту:

АБВГ...БЮЯ — заголовні (прописні);

абвг...ьюя — рядкові.

Десяткові цифри:

0 1 2 3 4 5 6 7 8 9

Роздільники:

() { } [] ; , .

Вісімкові і шістнадцяткові цифри

Шістнадцяткове значення задається за допомогою символів "0x" або "0X", за яким йде значення числа

0 1 2 3 4 5 6 7 8 9 A B C D E F

$$0xF = F_{<16>} = 15;$$

$$0x10 = 10_{<16>} = 16;$$

$$0x100 = 100_{<16>} = 16 \cdot 16 = 256;$$

$$0x1000 = 1000_{<16>} = (16)^3 = 4096;$$

$$0x1F = 1F_{<16>} = 1 \cdot 16 + 15 = 31;$$

$$0x21 = 21_{<16>} = 2 \cdot 16 + 1 = 33;$$

$$0xFF = FF_{<16>} = 15 \cdot 16 + 15 = 255;$$

Число в **вісімковому записі** має починатися з нуля, за яким слідує одна або кілька вісімкових цифр, наприклад **077777**.

Ключові слова

<code>abstract</code>	<code>assert</code>	<code>boolean</code>	<code>break</code>	<code>byte</code>
<code>case</code>	<code>catch</code>	<code>char</code>	<code>class</code>	<code>const</code>
<code>continue</code>	<code>default</code>	<code>do</code>	<code>double</code>	<code>else</code>
<code>enum</code>	<code>extends</code>	<code>final</code>	<code>finally</code>	<code>float</code>
<code>for</code>	<code>goto</code>	<code>if</code>	<code>implements</code>	<code>import</code>
<code>instanceof</code>	<code>int</code>	<code>interface</code>	<code>long</code>	<code>native</code>
<code>new</code>	<code>package</code>	<code>private</code>	<code>protected</code>	<code>public</code>
<code>return</code>	<code>short</code>	<code>static</code>	<code>strictfp</code>	<code>super</code>
<code>switch</code>	<code>synchronized</code>	<code>this</code>	<code>throw</code>	<code>throws</code>
<code>transient</code>	<code>try</code>	<code>void</code>	<code>volatile</code>	<code>while</code>

Літерали **false, null, true**

Ключові слова **const** і **goto** не використовуються в Java. Вони введені для того, щоб згенерувати повідомлення про помилку компіляції при появі цих ключових слів C++ в програмі.

Якщо в тексті програми потрібні символи, які звичайним чином в текст ввести не можна, то вдаються до керуючої послідовності.

Символ	Значення
<code>\b</code>	Повернення на один символ назад
<code>\f</code>	Переведення на нову сторінку
<code>\n</code>	Переведення на новий рядок
<code>\r</code>	Повернення до початку рядка
<code>\t</code>	Горизонтальна табуляція
<code>\'</code>	Апостроф
<code>\"</code>	Подвійні лапки
<code>\\</code>	Зворотна коса риска
<code>\u</code>	Початок кодування символу Unicode

Керуючі послідовності – символи Unicode

`\u0030` — `\u0039` — цифри ISO-LATIN від 0 до 9;

`\u0024` — знак долара \$;

`\u0041` — `\u005a` — літери від *A* до *Z*;

`\u0061` — `\u007a` — літери від *a* до *z*.

Спеціальні символи

+	Оператор складання
-	Оператор вирахування
*	Оператор множення
/	Оператор ділення
%	Оператор залишку від цілочисельного ділення
=	Оператор присвоювання
~	Оператор побітового доповнення (побітове НЕ)
^	Оператор XOR (виключаюче АБО)
&	Оператор AND (І)
	Оператор OR (АБО)
!	Оператор NOT (НЕ)
>	Більше
<	Менше

Спеціальні символи

- ? Знак питання - частина тернарного (що складається з трьох частин) умовного оператора "?:"
- : Двокрапка - частина умовного оператора "?:". Також використовується для завдання мітки - ставиться після імені мітки
- { Ліва фігурна дужка - роздільник, що відкриває блок коду
- } Права фігурна дужка - роздільник, що закриває блок коду
- , Кома - роздільник в списку параметрів оператора; роздільник в складеному операторі
- . Крапка - десятковий роздільник в числовому літерному вираженні; роздільник в складеному імені для доступу до елементу пакета, класу, об'єкта, інтерфейсу
- (Ліва кругла дужка - роздільник, що відкриває список параметрів в операторах і згрупованих частин у виразах
-) Права кругла дужка - роздільник, що закриває список параметрів в операторах і згрупованих частин у виразах

Спеціальні символи

- [Ліва квадратна дужка - роздільник, що відкриває індекс масиву
 -] Права квадратна дужка - роздільник, що закриває індекс масиву
 - ; Крапка з комою - роздільник, що позначає закінчення оператора
 - ' Апостроф (одинарні лапки) - відкриття і закриття символу
 - “ Подвійні лапки - відкриття і закриття рядка символів
 - \ Зворотня коса риска (backslash) - використовується для завдання керуючих послідовностей символів
 - @ Комерційне а (“ет”) - знак початку анотації
- знак пробілу
- знак табуляції
- Символи # □ «» № § не мають спеціального призначення

Складені спеціальні символи

++	Оператор інкременту (збільшення на 1) еквівалентно $x = x + 1$
--	Оператор декременту (зменшення на 1) еквівалентно $x = x - 1$
&&	Оператор AND (логічне І)
	Оператор OR (логічне АБО)
<<	Оператор лівого побітового зсуву
>>>	Оператор без знакового правого побітового зсуву
>>	Оператор правого побітового зсуву зі збереженням знака
==	Дорівнює
!=	Не дорівнює
+=	$y += x$ еквівалентно $y = y + x$
-=	$y -= x$ еквівалентно $y = y - x$
*=	$y *= x$ еквівалентно $y = y * x$
/=	$y /= x$ еквівалентно $y = y / x$

Складені спеціальні символи

`%=` $Y \% = x$ еквівалентно $y = y \% x$

`|=` $Y | = x$ еквівалентно $y = y | x$

`^=` $Y \wedge = x$ еквівалентно $y = y \wedge x$

`>>=` $y >> = x$ еквівалентно $y = y >> x$

`>>>=` $y >>> = x$ еквівалентно $y = y >>> x$

`<<=` $y << = x$ еквівалентно $y = y << x$

`/*` Початок багаторядкового коментаря

`/**` Початок багаторядкового коментаря, призначеного для автоматичного створення документації по класу

`*/` Кінець багаторядкового коментаря (відкривається як `/*` або `/**`)

`//` Однорядковий коментар

Ідентифікатори

Ідентифікаторами називають імена, що привласнюються різним елементам програми. Ідентифікатори в мові Java будуються з символів стандарту Unicode.

Згідно зі специфікацією Java, **ідентифікатори** повинні складатися з літер і цифр (без пробілів), причому першим символом не повинна бути цифра. Ідентифікатори не повинні збігатися з ключовими словами, булевими константами **true** і **false**, а також константою **null**.

Літери можуть бути будь-яким літерним символом Unicode (в тому числі і літерою кирилиці - тільки для версії Multilanguage). Однак слід зазначити, що буква К латинського алфавіту і буква К кирилиці вважаються різними літерами. Ідентифікатор, таким чином, може мати довільну довжину.

Крім того, для сумісності з ранніми версіями Java літерами вважаються символ підкреслення "_" і символ долара "\$".

Змінні і типи

Змінна - це іменована комірка пам'яті, з якою зіставлений **тип** і вміст якої може змінюватися.

Перед тим як задіяти будь-яку змінну, вона повинна бути задана в області програми, що передує тому місцю, де ця змінна використовується. При оголошенні змінної спочатку вказується її тип, а потім ідентифікатор.

- **int** — ціле число;
- **float** — дійсне число;
- **boolean** — логічне значення;
- **Object** — найпростіший об'єктний тип (клас) Java;

Є можливість задавати власні об'єктні типи (класи):

```
MyType1 a1,b1;
```

Оголошення змінних може супроводжуватися їх *ініціалізацією* - привласненням початкових значень.

```
int i1=5;  
int i2=-78;
```

або

```
int i1=5, i2=-78;
```

Примітивними типами (primitive types) називаються такі, для яких дані містяться в одній комірці пам'яті, яка не має вкладених комірок.

Посилальними типами називаються такі, для яких в комірці пам'яті містяться не самі дані, а тільки адреси цих даних, тобто посилання на дані. При присвоєнні в посилальну змінну заноситься нова адреса, а не самі дані. Але безпосереднього доступу до адреси, що зберігається в посилальних змінних, немає. Це зроблено для забезпечення безпеки роботи з даними.

Константи представляють значення числових типів і символьного типу, а також типу **String** (рядкові константи).

Значення **null** також є константою.

Роздільниками в Java є такі 9 символів:

- (- відкриваюча кругла дужка;
-) - закриваюча кругла дужка;
- { - відкриваюча фігурна дужка;
- } - закриваюча фігурна дужка;
- [- відкриваюча квадратна дужка;
-] - закриваюча квадратна дужка;
- ; - крапка з комою;
- ,
- .

Операції Java - це символи унарних і бінарних операцій (арифметичних, логічних, операцій відносини і присвоювання), символи складових операцій (типу "+="), а також символи "?" і ":" умовного оператора.

Послідовність елементів програми утворює **речення** або **оператор** мови Java. В одному рядку може розміщуватися декілька операторів. Оператор може займати кілька рядків, причому перенесення на новий рядок можна робити в тому місці оператора, де знаходиться символ пробілу або символ-роздільник (до або після цього символу). Ознакою закінчення оператора є символ ";".

Оператори Java, як і оператори мови C, можуть утворювати **блоки**. Оператори блоку відокремлюють символами "{" і "}". Символ ";" після завершення блоку (символу "}") ставити не потрібно.

Коментарі в мові Java можна оформляти одним із способів:

1. */* текст */* Весь текст між цими поєднаннями символів, буде проігноровано. Такий коментар може займати кілька рядків.
2. *// текст* Весь текст, наступний після символів "//" до кінця рядка, ігнорується.

В Java існує ще один тип коментарів, використовуваний для автоматичного документування програми - коментарі Javadoc.

В Java діють наступні угоди про регістрі букв в ідентифікаторах:

- імена примітивних типів слід писати в нижньому регістрі, наприклад: ***int, float, boolean*** і т. д.;
- імена посилальних типів слід починати з великої літери, а далі для імен, що складаються з одного слова, писати всі інші літери в нижньому регістрі (наприклад, ***Object, Float, Boolean, Collection, Runnable***). Але якщо ім'я складене, то нову частину імені починають з великої літери (наприклад, ***JButton, JTextField, JFormattedTextField, MyType*** і т. д.).
- для змінних і методів імена, що складаються з одного слова, слід писати в нижньому регістрі (наприклад, ***i, j, object1***). Якщо ім'я складене, нову частину імені починають з великої літери: ***myVariable, jButton2, jTextField2.getText()*** і т. д.;
- імена пакетів слід писати повністю в нижньому регістрі навіть в тому випадку, коли ім'я складене: ***mypackage, desktopapplication1*** і т. д.;

В Java діють наступні угоди про регістрі букв в ідентифікаторах:

- імена констант слід писати в верхньому регістрі, розділяючи складові частини, що входять в ім'я, символом підкреслення "_", наприклад: *Double.MIN_VALUE*, *Double.MAX_VALUE*, *JOptionPane.INFORMATION_MESSAGE*, *MY_CHARS_COUNT* и т. п.;
- символ підкреслення "_" рекомендується для поділу складових частин імені тільки в іменах констант і пакетів.

Лектор:

Старший викладач кафедри Електроніки и комп'ютерної техніки Сумського державного університету

Горячев О. Є.

В лекції використано матеріали авторів:

Шонін В.А.

Монахов В.В.