

# Bài 1

**Ngôn ngữ lập trình Java**

Module: ADVANCED PROGRAMMING WITH JAVA

## Mụctiêu

* Trình bày *đượ*c n*ộ*i dung, yêu c*ầ*u, l*ị*ch trình và k*ế*t qu*ả* c*ủ*a môn h*ọ*cBC-JAVA
* S*ử*d*ụ*ng *đượ*c cú pháp Java *để* thao tác v*ớ*i bi*ế*n
* S*ử*d*ụ*ng *đượ*c cú pháp Java *để* thao tác v*ớ*i c*ấ*u trúc *đ*i*ề*u ki*ệ*n
* Sử dụng được cấu trúc if-else
* Sử dụng được cấu trúc switch-case

### ModuleADVANCED PROGRAMMING WITH JAVA

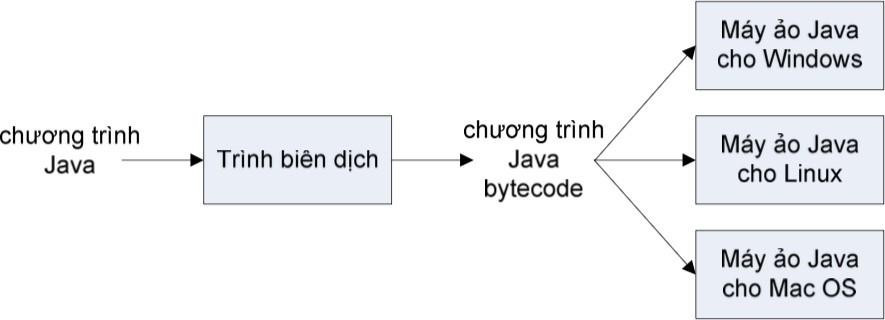
* Mục đích: Giúp học viên làm chủ các kiến thức lập trình cơ bản và tư duy giải quyết vấn đề. Hoàn thành khoá học, học viên có đủ kiến thức và kỹ năng nền tảng về lập trình để bước sang giai đoạn học lập trìnhchuyên sâu.
* Thời gian: 25bài
* Đánh giá:
  + Thi thực hành và lý thuyết cuối module, điểmđạt: 75%
  + Bảng đánh giá kỹ năng theo chuẩn đầura
* Yêu cầu:
  + Phần mềmInte**l** J



# Thảoluận

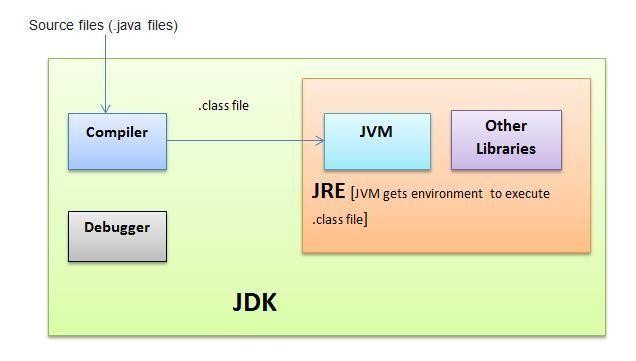
Ngôn ng*ữ* l*ậ*p trình Java Intel iJ IDEA

* Là m*ộ*t ngôn ng*ữ* l*ậ*p trình h*ướ*ng *đố*i t*ượ*ng
* Có kh*ả* n*ă*ng th*ự*c thi *ở* nhi*ề*ulo*ạ*i thi*ế*t b*ị*
* *Đượ*c s*ử* d*ụ*ng r*ộ*ngrãi
* Java có tính *độ*c l*ậ*p n*ề*nt*ả*ng (platform independent)
* M*ộ*t ch*ươ*ng trình Java có th*ể* ch*ạ*y trên các n*ề*nt*ả*ng khác nhau mà không ph*ả*i biên d*ị*ch l*ạ*i



* M*ộ*t ch*ươ*ng trình vi*ế*t b*ằ*ng ngôn ng*ữ* b*ậ*c cao c*ầ*n *đượ*c d*ị*ch sang ngôn ng*ữ* máy tr*ướ*c khi có th*ể đượ*c ch*ạ*y trên máy tính
* Mã ngu*ồ*n Java không *đượ*c d*ị*ch tr*ự*c ti*ế*p ra ngôn ng*ữ* máy mà *đượ*c biên d*ị*ch (*compile*)ra byte code
* byte code là ngôn ng*ữ đượ*c th*ự*c thi trên m*ộ*t máy tính *ả*o g*ọ*i là JVM (Java Virtual Machine)
* Khi m*ộ*t ch*ươ*ng trình Java *đượ*c th*ự*c thi thì JVM s*ẽ* thông d*ị*ch (*interpret*) ra ngôn ng*ữ* máy th*ự*c s*ự*



**JDKvsJRE**

JRE giúp th*ự*c thi các ch*ươ*ng trình Java trongJVM

JDK giúp phát tri*ể*n và biên d*ị*ch mã Java thành ch*ươ*ng trình Java

JRE c*ũ*ng *đượ*c kèm theo trong JDK

## InteliJIDEA

* T*ả*i Inte**l** iJIDEAt*ạ*i: https://[www.jetbrains.com/idea/](http://www.jetbrains.com/idea/)



### Demo:Tạoứngdụng Java- 1

* B*ướ*c 1:T*ạ*oproject m*ớ*i trên Inte**l** iJ IDEA*đặ*t tên he**l** oworld
* B*ướ*c 2:T*ạ*ol*ớ*p He**l** oWorld v*ớ*i n*ộ*i dung:

**public class** HelloWorld {

**public static void** main(String[] args) {

System.***out***.println(**"Hello world"**);

}

}

**Có thể sử dụng các cách gõ tắt sau:**

* psvm + TAB: viết nhanh hàm main()
* sout + TAB: viết nhanh hàm System.out.println()

### Demo:Tạoứngdụng Java- 2

* B*ướ*c 3:Ch*ạ*y *ứ*ng d*ụ*ng: Ch*ọ*n m*ụ*c “Run‘HelloWorld.main()’.



**Có thể sử dụng phím tắt để chạy ứngdụng.**

# Demo

T*ạ*o *ứ*ng d*ụ*ngJava *đầ*u tiên

# Thảoluận

Bi*ế*n vàh*ằ*ng Ki*ể*u d*ữ* li*ệ*u Toán t*ử*

* Java yêu c*ầ*u ph*ả*i khai báo bi*ế*n tr*ướ*c khi s*ử* d*ụ*ng
* Cú pháp:

*datatypevariableName;*

Trong *đ*ó:

- datatypelà ki*ể*u d*ữ* li*ệ*u c*ủ*a bi*ế*n

- variableNamelà *đị*nh danh (tên) c*ủ*a bi*ế*n

* Có th*ể* khai báo nhi*ề*u bi*ế*n cùng ki*ể*u giá tr*ị* trong m*ộ*t câu l*ệ*nh:

*datatypevariable1,variable2,...,variablen;*

* Ví d*ụ*:

**int** i, j, k; //Khai báo các biến i, j, k là kiểu số nguyên

* Có th*ể* gán giá tr*ị* cho bi*ế*n ngay t*ạ*i th*ờ*i *đ*i*ể*m khai báo
* Ví d*ụ*:

**int** count = **1**;

* Ví d*ụ* trên t*ươ*ng *đươ*ng v*ớ*i:

**int** count; count = **1**;

* Có th*ể* gán giá tr*ị* cho nhi*ề*u bi*ế*n t*ạ*i th*ờ*i *đ*i*ể*m khai báo
* Ví d*ụ*:

**int** i = **1**, j = **2**;

* H*ằ*ng là m*ộ*t tên g*ọ*i *đạ*i di*ệ*n cho m*ộ*t giá tr*ị* c*ố đị*nh
* Giá tr*ị* c*ủ*a h*ằ*ng không th*ể* thay *đổ*i
* Giá tr*ị* c*ủ*a h*ằ*ng c*ầ*n ph*ả*i *đượ*c gán t*ạ*i th*ờ*i *đ*i*ể*m khai báo
* Ví d*ụ*, s*ử*d*ụ*ng h*ằ*ng PI thay cho giá tr*ị* 3.14159:

**double** area = radius \* radius \* **3.14159**;

*Đượ*c thay b*ằ*ng:

**final double** PI = **3.14159**; **double** area = radius \* radius \* PI;

* Cú pháp khai báoh*ằ*ng:

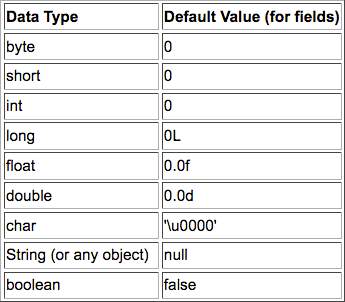
**final** datatype CONSTANTNAME = value;

Trong *đ*ó:

* + *final* là t*ừ* khoá b*ắ*t bu*ộ*c *để* khai báo h*ằ*ng
  + *datatype*là ki*ể*u d*ữ* li*ệ*u c*ủ*a h*ằ*ng
  + *CONSTANTNAME* là tên c*ủ*a h*ằ*ng
  + *value* là giá tr*ị* c*ủ*a h*ằ*ng

|  |  |
| --- | --- |
| **Kiểu dữ liệu** | **Mô tả** |
| **byte** | Kiểu số nguyên có kích thước 1 byte. Các giá trị nằm trong khoảng -128 đến 127. |
| **short** | Kiểu số nguyên có kích thước 2 byte. Các giá trị nằm trong khoảng -32768 đến  32767. |
| **int** | Kiểu số nguyên có kích thước 4 byte.Các giá trị nằm trong khoảng -231 đến 231 -1. |
| **long** | Kiểu số nguyên có kích thước 8 byte. Các giá trị nằm trong khoảng -263 đến 263 - 1. |
| **float** | Kiểu số thực có kích thước 4 byte. |
| **double** | Kiểu số thực có kích thước 8 byte. |
| **boolean** | Bao gồm 2 giá trị là true vàfalse. |
| **char** | Kiểu ký tự Unicode có kích thước 2 byte. Có giá trị nhỏ nhất là '\u0000' (tương đương với 0) và giá trị lớn nhất là '\uffff' (tương đương với 65535) |

* Khi khai báo m*ộ*t bi*ế*n c*ủ*a *đố*i t*ượ*ng (không ph*ả*i bi*ế*n *đị*a ph*ươ*ng) mà không gán giá tr*ị* cho nó thì nó s*ẽ* có giá tr*ị* m*ặ*c *đị*nh



|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Mô tả** | **Ví dụ** |
| + | Phép cộng | int result = 1 + 2; //result = 3 |
| - | Phép trừ | int result = 1 - 2; //result = -1 |
| \* | Phép nhân | int result = 2 \* 3; //result = 6 |
| / | Phép chia | int result = 3 / 2; //result = 1 |
| % | Phép chia lấy số dư | int result = 5 % 3; //result = 2 |

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Mô tả** | **Ví dụ** |
| + | Toán tử cộng. | *int result = +1; //result = 1* |
| - | Toán tử trừ | *int result = -1; //result = -1* |
| ++ | Toán tử tăng 1 giá trị | *int result = 1;*  *int result++; //result = 2* |
| -- | Toán tử giảm 1 giá trị | *int result = 1;*  *int result--; //result = 0* |
| ! | Toán tử phủ định | *int value = true;*  *result = !value;//result – false* |

* Toán t*ử* t*ă*ng (++)và gi*ả*m (--)s*ẽ* cho k*ế*t qu*ả* khác nhau, tu*ỳ* thu*ộ*c vào v*ị* trí c*ủ*a nó so v*ớ*i toán h*ạ*ng
* N*ế*u *đặ*t tr*ướ*c (prefix) toán h*ạ*ng thì giá tr*ị* s*ẽ đượ*c t*ă*ng ho*ặ*c gi*ả*m tr*ướ*c khi bi*ể*u th*ứ*c *đượ*c *đ*ánh giá
* N*ế*u *đặ*t sau (postfix) toán h*ạ*ng thì giá tr*ị* s*ẽ đượ*c t*ă*ng ho*ặ*c gi*ả*m sau khi bi*ể*u th*ứ*c *đượ*c *đ*ánh giá
* Ví d*ụ*:

**int** i = 3;

**int** j = ++i;*// i = 4 và j = 4;*

**int** i = 3;

**int** j = i++;*// i = 4 và j = 3;*

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Mô tả** | **Ví dụ**  **int** a = 5;  **int** b = 6; |
| == | So sánh bằng | boolean result = a == b; //result = false |
| != | So sánh khác | boolean result = a != b; //result = true |
| > | Lớn hơn | boolean result = a > b; //result = false |
| >= | Lớn hơn hoặc bằng | boolean result = a >= b; //result = false |
| < | Nhỏ hơn | boolean result = a < b; //result = true |
| <= | Nhỏ hơn hoặc bằng | boolean result = a <= b; //result = true |

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Mô tả** | **Ví dụ**  **int** a = true;  **int** b = false; |
| && | AND (và): Trả về đúng nếu cả 2 vế đều đúng | boolean result = a && b; //result = false |
| || | Trả về đúng nếu ít nhất  một vế đúng | boolean result = a || b; //result = true |
| ! | Phủ định | boolean result = !a; //result = false |

# Demo

Bi*ế*n, toán t*ử* và ki*ể*u d*ữ* li*ệ*u

# Thảoluận

L*ệ*nh if L*ệ*nh if-else

L*ệ*nh if l*ồ*ng nhau L*ệ*nh if b*ậ*c thang L*ệ*nh switch-case

* Cú pháp:

if (condition) {

// one or more statements;

}

Trong *đ*ó:

* + condition: là bi*ể*u th*ứ*c tr*ả*v*ề* giá tr*ị* ki*ể*u boolean
  + statements: Các câu l*ệ*nh s*ẽđượ*c th*ự*c thi n*ế*u *đ*i*ề*u ki*ệ*n tr*ả*v*ề* **true**

|  |  |  |
| --- | --- | --- |
| * Cú pháp: |  |  |
| if (condition) | { |
| // one or | more | statements; |
| }  else {  // one or | more | statements; |
| } |  |  |

Trong *đ*ó:

* + condition: *đ*i*ề*u ki*ệ*n *để đ*ánh giá. N*ế*u condition tr*ả* v*ề* **true** thì kh*ố*i l*ệ*nh bên trong **if**

*đượ*c th*ự*c thi.N*ế*u conditiontr*ả*v*ề* **false** thì kh*ố*i l*ệ*nh trong**else** *đượ*c th*ự*c thi.

* M*ộ*t câu l*ệ*nh if có th*ể đượ*c *đặ*t trong câu l*ệ*nh if khác:

if(condition1) {

if(condition2)

true-block statement(s); else

false-block statement(s);

}

}

else {

}

false-block statement(s);

* Có th*ể đặ*t các câu l*ệ*nh *đ*i*ề*u ki*ệ*n if-else liên ti*ế*p nhau

if(condition) {

// one or more statements

}

else if (condition) {

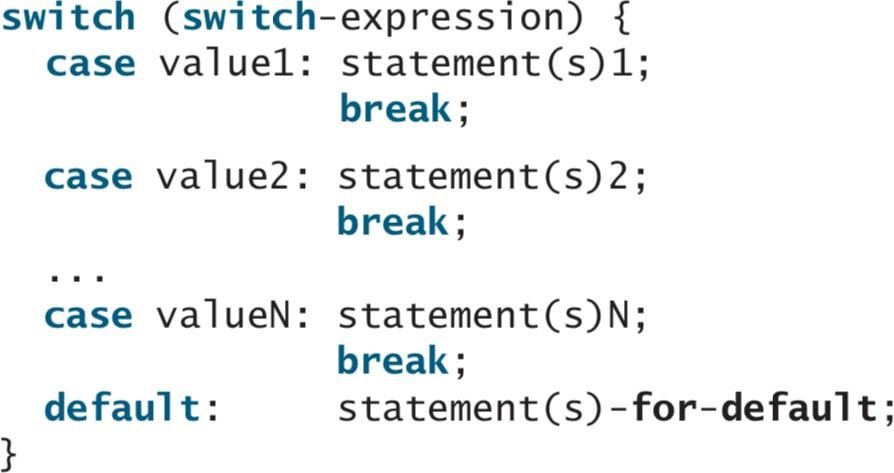
// one or more statements

}

else {

// one or more statements

}

* + Trong *đ*ó:
    - *switch-expression*: là bi*ể*u th*ứ*c tr*ả*v*ề*giá tr*ị* thu*ộ*c m*ộ*t trong các ki*ể*u: char, byte, short, int ho*ặ*c String
    - *value1,…valueN* có cùng ki*ể*u d*ữ* li*ệ*u so v*ớ*i

switch-expression

* + - *break* là t*ừ* khoá *để* d*ừ*ng th*ự*c thi cáccâu l*ệ*nh *ở*

phía sau.*break* là không b*ắ*t bu*ộ*c.

* + - *default*là t*ừ* khoá *để* quy *đị*nh kh*ố*i l*ệ*nh s*ẽđượ*c th*ự*c thi n*ế*u không có tr*ườ*ng h*ợ*p nào *ở* các *case* là *đ*úng. *default*là không b*ắ*t bu*ộ*c.

|  |  |
| --- | --- |
| **if** | **switch-case** |
| Có th*ể* s*ử* d*ụ*ng *để* so sánh l*ớ*n h*ơ*n, nh*ỏ* h*ơ*n… | Ch*ỉ* có th*ể* s*ử*d*ụ*ng *để* so sánh b*ằ*ng ho*ặ*c khác nhau |
| M*ỗ*i câu l*ệ*nh if có m*ộ*t bi*ể*u th*ứ*c *đ*i*ề*u ki*ệ*n, v*ớ*i giá tr*ị* tr*ả* v*ề* là **true** ho*ặ*c **false** | T*ấ*t c*ả* các tr*ườ*ng h*ợ*p (case) *đề*u so sánh v*ớ*i giá tr*ị* c*ủ*a bi*ể*u th*ứ*c *đ*i*ề*u ki*ệ*n duy nh*ấ*t |
| Bi*ể*u th*ứ*c *đ*i*ề*u ki*ệ*n c*ầ*n tr*ả* v*ề* giá tr*ị* ki*ể*u **boolean** | Bi*ể*u th*ứ*c *đ*i*ề*u ki*ệ*n c*ầ*n tr*ả* v*ề* giá tr*ị* là ki*ể*u  **byte**, **short**, **char**, **int**, ho*ặ*c **String** |
| Ch*ỉ* có m*ộ*t kh*ố*i l*ệ*nh *đượ*c th*ự*c thi n*ế*u *đ*i*ề*u ki*ệ*n*đ*úng | N*ế*u *đ*i*ề*u ki*ệ*n *đ*úng màkhông có câu l*ệ*nh break thì t*ấ*t c*ả* các kh*ố*i l*ệ*nh *ở* phía sau c*ũ*ng *đượ*c th*ự*cthi |



# Demo

if-else switch-case

## Tómtắt bài học

* Java h*ỗ* tr*ợ* nhi*ề*u ki*ể*u d*ữ* li*ệ*u khác nhau
* Các câu l*ệ*nh *đ*i*ề*u khi*ể*n giúp *đ*i*ề*u h*ướ*ng lu*ồ*ng th*ự*c thi c*ủ*a *ứ*ng d*ụ*ng



# Hướngdẫn

* H*ướ*ng d*ẫ*n làm bài th*ự*c hành và bài t*ậ*p
* Chu*ẩ*n b*ị* bài ti*ế*p: *Vòng lặp trong Java*