

CÚ PHÁP CƠ BẢN JAVA

Giảng viên : Nguyễn Hoàng Anh

Email: nhanh@fit.hcmus.edu.vn

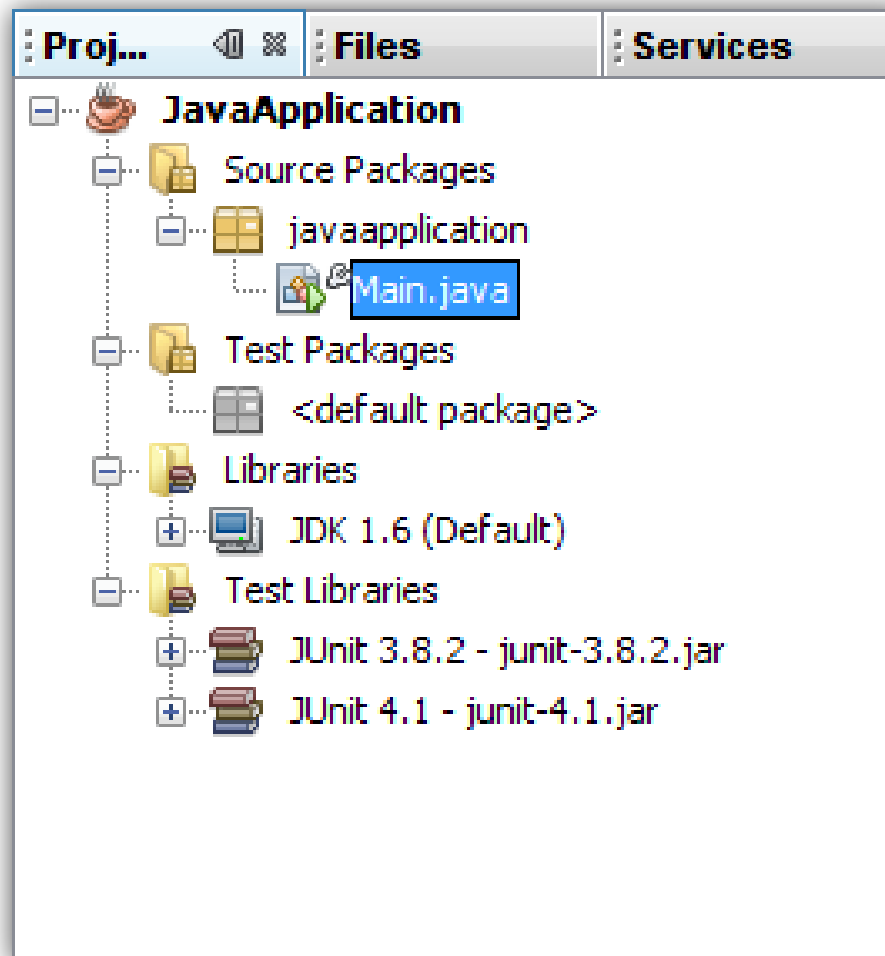


Nội dung



- Cấu trúc chương trình
- Ghi chú
- Kiểu dữ liệu cơ bản
- Nhập xuất console
- Cấu trúc rẽ nhánh
- Cấu trúc lặp
- Lệnh break và continue
- Math, String

Cấu trúc chương trình



Cấu trúc chương trình

```
package javaapplication;
```

```
import java.io.*;
```

```
import java.util.ArrayList;
```

```
/**
```

```
 *
```

```
 * @author NHA nh
```

```
 */
```

```
public class Main {
```

```
    /**
```

```
     * @param args the command line arguments
```

```
     */
```

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

```
        // TODO code application logic here
```

```
    }
```

```
}
```

Ghi chú



- Cách 1:

// Ghi chú 1 dòng

- Cách 2:

*/**

** Ghi chú nhiều dòng*

**/*

Ghi chú



- Cách 3:

```
/* *
```

```
 * Ghi chú Javadoc
```

```
*/
```

- Quy tắc vàng



Khi ghi chú nên sử dụng ghi chú Javadoc

Kiểu dữ liệu cơ sở

Kiểu dữ liệu	Kích thước	Giá trị mặc định
boolean	1 bit	false
byte	8 bit	0
short	16 bit	0
char	16 bit	'\u0000'
int	32 bit	0
long	64 bit	0l
float	32-bit IEEE 754	0f
double	64-bit IEEE 754	0

String	Chuỗi các ký tự	null
--------	-----------------	------

Hằng số



- Cách khai báo:

```
final KieuDuLieu TENHANGSO = GiaTri;
```

- Ví dụ:

```
final double PI = 3.14;
```

- Quy tắc vàng:

 **TÊN HẰNG SỐ** nên **VIẾT HOA**

Toán tử



Toán tử	Mức độ ưu tiên
postfix	expr++ expr--
unary	++expr --expr +expr -expr ~ !
multiplicative	* / %
additive	+ -
shift	<< >> >>>
relational	< > <= >= instanceof
equality	== !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	
logical AND	&&
logical OR	
ternary	? :
assignment	= += -= *= /= %= &= ^= = <<= >>= >>>=

Nạp chồng toán tử



- Quy tắc vàng:

⚠ Java không hỗ trợ nạp chồng toán tử

Nhập từ console



Scanner	scan	= new Scanner (System.in);
boolean	bl	= scan.nextBoolean ();
byte	b	= scan.nextByte ();
short	s	= scan.nextShort ();
Char	c	= scan.nextChar ();
long	lg	= scan.nextLong ();
float	fl	= scan.nextFloat ();
double	db	= scan.nextDouble ();
String	str	= scan.next ();
String	str	= scan.nextLine ();

Xuất ra console



- `System.out.print(...);`
- `System.out.println(...);`
- `System.out.printf(...);`

Định dạng xuất



Định dạng	Kiểu dữ liệu
%b	boolean
%c	char
%d	int
%f	float, double
%s	String

Định dạng xuất



- `System.out.printf ("n = %d", 10);`
- `System.out.printf ("m = % f", 12.2.f);`
- `System.out.printf ("s = %.2f", 112.3);`
- `System.out.printf ("t = %.2f", 112.3);`
- `System.out.printf ("u = %d v = %f", 5, 2.3);`
- `System.out.printf ("str = %s", "nhanh");`

Định dạng xuất



- Định dạng:

`String.format` (“định dạng”, tham số 1, ..., tham số n);

- Ví dụ:

```
String str = String.format (“n = %d, m= %f”, n, m);  
System.out.println (str);
```

- Quy tắc vàng:

⚠ Nên sử dụng `String.format` kết hợp với `System.out.println` thay thế cho `System.out.printf`

Cấu trúc điều khiển if



- Cú pháp

```
if (biểu thức điều kiện) {  
    lệnh 1;  
    ....  
    lệnh n  
}
```

- Quy tắc vàng:



Dấu { nên để sau biểu thức điều kiện
Nếu sau if có một lệnh thì nên để lệnh ấy trong
{ }

Mỗi lệnh trên một dòng

Cấu trúc điều khiển if



```
System.out.println("n=");  
Scanner scan = new Scanner(System.in);  
int n = scan.nextInt();  
if (n > 0) {  
    System.out.println("n là số dương");  
}
```

.....

Cấu trúc if - else



- Cú pháp
if (**biểu thức điều kiện**) {
 lệnh 1;

 lệnh n
} **else** {
 lệnh 1;

 lệnh n
}

Cấu trúc if - else



- Quy tắc vàng:



Dấu { nên để sau if và sau else

Nếu sau if hoặc else dù chỉ có một lệnh thì nên để lệnh ấy trong { }

Mỗi lệnh trên một dòng

Cấu trúc if - else



```
System.out.println("n=");
Scanner scan = new Scanner();
int n = scan.nextInt();
if (n > 0) {
    System.out.println("n là số dương");
} else {
    System.out.println("n là số âm");
}
```

Cấu trúc điều khiển if - else if - else



```
if (biểu thức điều kiện) {  
    ...  
} else if (biểu thức điều kiện) {  
    ...  
}  
else if (biểu thức điều kiện) {  
    ...  
}  
else {  
    ....  
}
```

Cấu trúc điều khiển if - else if - else



- Quy tắc vàng:

⚠ Dấu { nên để sau if, sau else if và else

Nếu sau if, else if hoặc else dù chỉ có một lệnh thì nên để lệnh ấy trong { }

Mỗi lệnh trên một dòng

Cấu trúc điều khiển switch



```
switch (giá trị){  
    case giá trị 1:  
        .....  
        break;  
  
    .....  
    case giá trị N:  
        .....  
        break;  
    default:  
        ...  
}
```

- Giá trị : char, byte, short, int
- Giá trị 1, giá trị 2 có cùng kiểu dữ liệu với giá trị

Cấu trúc lặp while



while (điều kiện lặp) {

.....

}

Cấu trúc lặp do while



```
do{  
    .....  
}while (điều kiện lặp)
```

Cấu trúc lặp for



for (khởi tạo biến; điều kiện lặp; hành động sau mỗi lần lặp){

....

}

Câu lệnh break và continue



- Được sử dụng trong câu lệnh
 - for
 - do
 - While
- break:
 - Thoát khỏi vòng lặp
- continue :
 - Tiếp tục lặp

Math



- Math.abs
- Math.sqrt
- Math.min
- Math.max
- Math.exp
- Math.round
- Math.floor
- Math.log
- Math.log10
- Math.pow
- Math.random
- Math.sin
- Math.cos
- Math.tan
- Math.PI
- Math.E
- ...

Tham khảo



- Y.Daniel Liang Introduction to Java Programming Sixth Edition (2004)
- The Java Language Specification Third Edition (2005)

HỎI VÀ ĐÁP

