



LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG với JAVA

Chương II

Lập trình Java căn bản

Biên soạn: Lý Quỳnh Trân

Nội dung

- Bộ ký tự, từ khóa, chú thích
- Biến & hằng
- Kiểu dữ liệu
 - Nguyên thủy
 - Tham chiếu
- Chuyển kiểu & ép kiểu
- Các cấu trúc lệnh
 - Gán
 - Rẽ nhánh
 - Lặp
- Hàm main()





Cấu trúc chương trình Java

- Tên lớp Class
- Phương thức main
- Câu lệnh
- Dấu kết thúc lệnh ;
- Từ khóa
- Comments giải thích
- Khối

```
6 package helloworld;
7
8 /**
9  *
10  * @author Quynh Tran Ly
11  */
12 public class HelloWorld {
13
14     public static void main(String[] args) {
15         // Prints "Hello, World" to the terminal window.
16         System.out.println("Hello, World!");
17     }
18
19 }
```



Class Name

- Mỗi chương trình Java phải chứa ít nhất là một lớp
- Tên lớp phải viết hoa, ví dụ HelloJava.

```
// This program prints Welcome to Java!  
public class HelloJava {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```

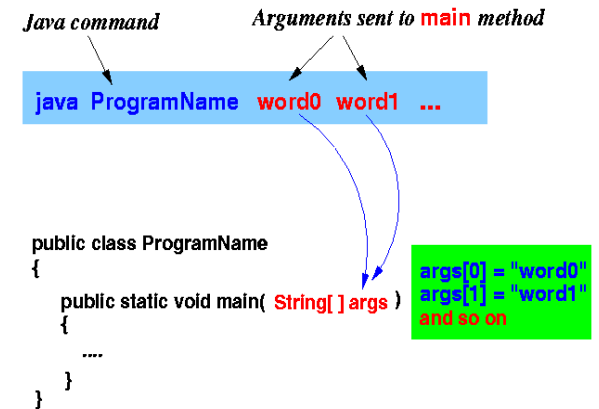
Hàm main()

- Tất cả các ứng dụng Java thi hành được, **phải có hàm main() – nơi chương trình bắt đầu thi hành.**
- `public static void main(String[] args)`
 - `public`: từ khóa chỉ định phạm vi truy xuất
 - `Static`: từ khóa chỉ ra rằng mọi thể hiện của ứng dụng này đều dùng chung hàm `main()`
 - `void` không trả về giá trị nào
 - `String[] args` :
 - đối số của hàm `main`
 - thường được truyền từ cửa sổ lệnh khi chạy ct
 - `args[0]` là đối số thứ nhất..

```
Test.java ✖
1 package com.journaldev.java.examples1;
2
3 public class Test {
4
5     public static void main(String args[]) {
6
7         System.out.println("Hello World");
8
9     }
10 }
```

Main : biến args

- **args[0]** là phần tử đầu tiên **first element**
Kiểu dữ liệu **a[0]** là **String**
- **args[1]** là phần tử thứ hai **second element**
Kiểu dữ liệu **a[1]** là **String**
- **args.length()** kích thước của array.



```
public class Argument1
{
    public static void main(String[] args)
    {
        for ( int i = 0; i < args.length; i++ )
            System.out.println( args[i] );
    }
}
```

Câu lệnh

- Câu lệnh thể hiện hành động hoặc một chuỗi hành động
- `System.out.println("Welcome to Java!")` hiển thị "Welcome to Java!".

```
// This program prints Welcome to Java!  
public class Welcome {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```



Dấu kết thúc câu lệnh

Dấu kết thúc câu lệnh (;).

```
// This program prints Welcome to Java!  
public class Welcome {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```


Từ khóa

- Từ khóa là từ được cung cấp bởi ngôn ngữ Java, được viết theo cú pháp tạo thành câu lệnh

```
// This program prints Welcome to Java!  
public class Welcome {  
    public static void main(String[] args) {  
        System.out.println("Welcome to Java!");  
    }  
}
```



Ký tự đặc biệt

Character	Name	Description
{ }	Opening and closing braces	Denotes a block to enclose statements.
()	Opening and closing parentheses	Used with methods.
[]	Opening and closing brackets	Denotes an array.
//	Double slashes	Precedes a comment line.
" "	Opening and closing quotation marks	Enclosing a string (i.e., sequence of characters).
;	Semicolon	Marks the end of a statement.

Bộ ký tự - từ khóa – chú thích

- Ký tự chữ, số, ký hiệu..
 - Phân biệt chữ thường & chữ hoa
- Khoảng 50 từ khóa
 - Phân biệt từ khóa & tên chuẩn
- Quy ước đặt tên
 - Tên lớp bắt đầu bằng 1 ký tự chữ in hoa
 - Tên biến, hàm bắt đầu bằng ký tự chữ thường
 - Tên gọi nhớ, nối các từ bằng chữ đầu in hoa. Ví dụ: `int numberOfStudent`
`class PhanSo`
- Lời chú thích
 - Trên 1 dòng sau ký hiệu `//` chú thích
 - Nhiều dòng: bao giữa `/*` chú thích `*/`

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

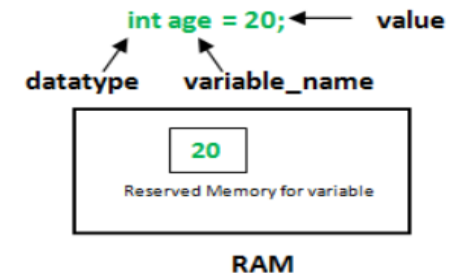
```
JavaExample.java
1 package com.beginnersbook;
2 public class JavaExample{
3     public static void main(String args[]){
4         String str = "122.202";
5
6         /* Convert String to double using
7          * parseDouble(String) method of Double
8          * wrapper class
9          */
10        double dnum = Double.parseDouble(str);
11        |
12        //displaying the value of variable dnum
13        System.out.println(dnum);
14    }
15 }
```

Problems @ Javadoc Declaration Console Pri

<terminated> JavaExample [Java Application] /Library/Java/JavaVirtual
122.202

Biến & hằng

- **Biến** là tên của một vùng bộ nhớ được dùng để lưu dữ liệu trong khi chương trình chạy
- **Cú pháp khai báo biến**
 - **TenkieuDL** tenbien1, tenbien2....Tenbienn;
Vd: `int age=20; char xepLoai='E';`
 - Kết thúc bằng dấu chấm phẩy ;
- Chúng ta có thể truy nhập, gán hay thay đổi giá trị của các biến, khi biến được gán một giá trị mới, giá trị cũ sẽ bị ghi đè lên.
- Java yêu cầu mỗi biến trước khi dùng phải được khai báo và kết thúc dòng khai báo bằng dấu chấm phẩy.
- **Hằng** là biến được khai báo sau từ khóa **final**
 - Phải khởi gán giá trị & không được phép thay đổi
 - Vd: `final int a=10;`
- Vị trí khai báo quyết định phạm vi hiệu lực của biến & hằng



```
15 import java.util.Scanner;
16 public class HinhTron {
17
18     public static void main(String[] args) {
19         // TODO code application logic here
20         int banKinh;
21         final double soPI = 3.14;
22         double area;
23         Scanner scanner = new Scanner(System.in);
24         System.out.print("Nhập vào bán kính hình tròn: ");
25         banKinh = scanner.nextInt();
26         area = soPI * banKinh * banKinh;
27         System.out.printf("Diện tích hình tròn = %.3f", area);
28         System.out.println();
29     }
30 }
```



Phạm vi của biến

- **Biến cục bộ (local variable)** phải được khởi gán trước khi truy xuất và có thể trùng tên biến của lớp, che dấu biến của lớp. **Biến cục bộ được khai báo tại hàm nào thì có hiệu lực ở bên trong hàm đó và các biến cục bộ bên trong khối & hàm không được trùng tên.**
 - Biến cục bộ được tạo bên trong các phương thức, constructor, block và sẽ **bị phá hủy** khi kết thúc các phương thức, constructor và block.
- **Biến toàn cục (global variable)** trong java
 - Khai **báo trong một lớp(class), bên ngoài các phương thức, constructor** và các block.
 - Biến toàn cục có thể được sử dụng bởi các phương thức, constructor, block, ... Nhưng nó phải được sử dụng thông qua một đối tượng cụ thể.
- **Biến static trong java**
 - Biến static được khai báo trong một class với từ khóa "static", phía bên ngoài các phương thức, constructor và block.
 - Biến static được tạo khi chương trình bắt đầu chạy và chỉ bị phá hủy khi chương trình dừng.
 - Biến static được truy cập thông qua tên của class chứa nó, với cú pháp: **TenClass.tenBien.**

Phạm vi của biến

```
12 public class StudentDetails2 {
13
14     public void StudentAge()
15     { //local variable age
16         int age = 0;
17         age = age + 5;
18     }
19
20     public static void main(String args[])
21     {
22         //using local variable age outside it's scope
23         System.out.println("Student age is : " + age);
24     }
25
26 }
```

run:

```
Exception in thread "main" java.lang.RuntimeException: Uncompilable source code - Erroneous tree
    at phamvibien.StudentDetails2.main(StudentDetails2.java:23)
```

C:\Users\Quynh Tran Ly\AppData\Local\NetBeans\Cache\8.2\executor-snippets\run.xml:53: Java returned: 1

BUILD FAILED (total time: 1 second)

```
12 public class StudentDetails3 {
13     static int age = 0;
14     public static void StudentAge()
15     { //local variable age
16         age = age + 5;
17     }
18
19     public static void main(String args[])
20     {
21         //using local variable age outside it's scope
22         StudentAge();
23         System.out.println("Student age is : " + age);
24     }
25
26 }
```

run:

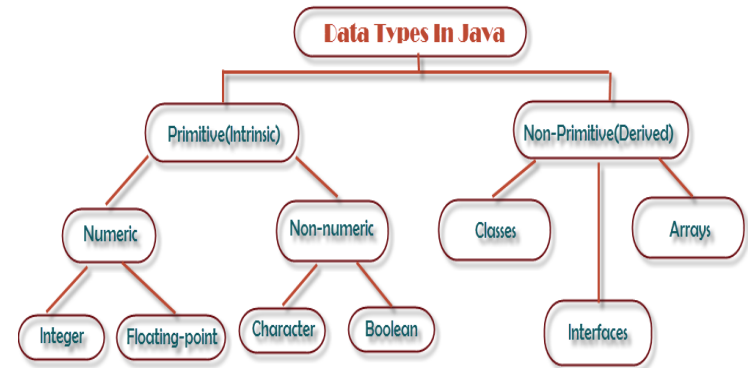
Student age is : 5

BUILD SUCCESSFUL (total time: 1 second)

.. .

Kiểu dữ liệu

- Hai loại kiểu dữ liệu :
 - Nguyên thủy (Primitive types)
 - Tham chiếu (Reference types)
- Kiểu nguyên thủy
 - Kiểu logic : boolean
 - Kiểu số
 - Nguyên: byte, short, int long, char
 - Thực: float, double.
 - Kiểu ký tự: char
- Kiểu tham chiếu: class, interface và array



Data types

```
int a = 77;  
Person person = new Person();
```

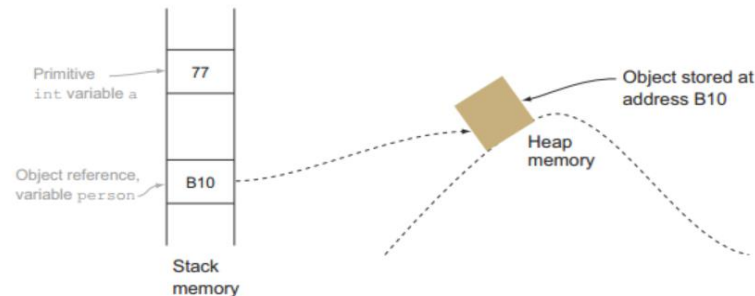
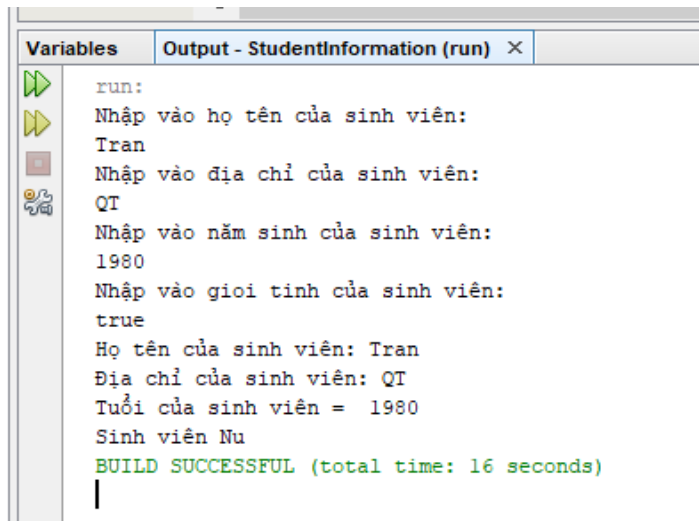


Figure 2.13 Primitive variables store the actual values, whereas object reference variables store the addresses of the objects they refer to.

Data Type	Size	Range of values that can be stored	Default value
byte	1 byte	-128 to 127	0
short	2 bytes	-32768 to 32767	0
int	4 bytes	-2,147,483,648 to 2,147,483,647	0
long	8 bytes	-9,223,372,036,854,775,808 to 9223372036854750000	0
float	4 bytes	3.4e-038 to 3.4e+038	0.0f
double	8 bytes	1.7e-308 to 1.7e+038	0.0d
boolean	1 bit	true or false	false
char	2 bytes		\u0000

Kiểu logic

- boolean
- true, false
- 1 byte / bit
- Ví dụ
 - boolean gender = false







```
run:
Nhập vào họ tên của sinh viên:
Tran
Nhập vào địa chỉ của sinh viên:
QT
Nhập vào năm sinh của sinh viên:
1980
Nhập vào giới tính của sinh viên:
true
Họ tên của sinh viên: Tran
Địa chỉ của sinh viên: QT
Tuổi của sinh viên = 1980
Sinh viên Nu
BUILD SUCCESSFUL (total time: 16 seconds)
```

```
8 import java.util.Scanner;
9
10 /**
11  *
12  * @author Quynh Tran Ly
13  */
14 public class StudentInformation {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21         String name;
22         int year;
23         String address;
24         boolean gender;
25         Scanner scanner = new Scanner(System.in);
26         System.out.println("Nhập vào họ tên của sinh viên: ");
27         name = scanner.nextLine();
28         System.out.println("Nhập vào địa chỉ của sinh viên: ");
29         address = scanner.nextLine();
30         System.out.println("Nhập vào năm sinh của sinh viên: ");
31         year = scanner.nextInt();
32         System.out.println("Nhập vào giới tính của sinh viên: ");
33         gender = scanner.nextBoolean();
34
35         System.out.println("Họ tên của sinh viên: " + name);
36         System.out.println("Địa chỉ của sinh viên: " + address);
37         System.out.println("Tuổi của sinh viên = " + year);
38         if (gender==true) {
39             System.out.println("Sinh viên Nu ");
40         }else
41         {
42             System.out.println("Sinh viên Nam ");
43         }
44     }
45 }
46
47 }
48
```

Kiểu logic

```
8 import java.util.Scanner;
9
10 /**
11  *
12  * @author Quynh Tran Ly
13  */
14 public class StudentInformation {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21         String name;
22         int year;
23         String address;
24         boolean gender;
25         Scanner scanner = new Scanner(System.in);
26         System.out.println("Nhập vào họ tên của sinh viên: ");
27         name = scanner.nextLine();
28         System.out.println("Nhập vào địa chỉ của sinh viên: ");
29         address = scanner.nextLine();
30         System.out.println("Nhập vào năm sinh của sinh viên: ");
31         year = scanner.nextInt();
32         System.out.println("Nhập vào giới tính của sinh viên: ");
33         gender = scanner.nextBoolean();
34
35         System.out.println("Họ tên của sinh viên: " + name);
36         System.out.println("Địa chỉ của sinh viên: " + address);
37         System.out.println("Tuổi của sinh viên = " + year);
38         if (gender==true) {
39             System.out.println("Sinh viên Nu ");
40         }else
41         {
42             System.out.println("Sinh viên Nam ");
43         }
44     }
45 }
46
47 }
48
```

Variables	Output - StudentInformation (run) ×
	run:
	Nhập vào họ tên của sinh viên:
	Tran
	Nhập vào địa chỉ của sinh viên:
	QT
	Nhập vào năm sinh của sinh viên:
	1980
	Nhập vào giới tính của sinh viên:
	true
	Họ tên của sinh viên: Tran
	Địa chỉ của sinh viên: QT
	Tuổi của sinh viên = 1980
	Sinh viên Nu
	BUILD SUCCESSFUL (total time: 16 seconds)

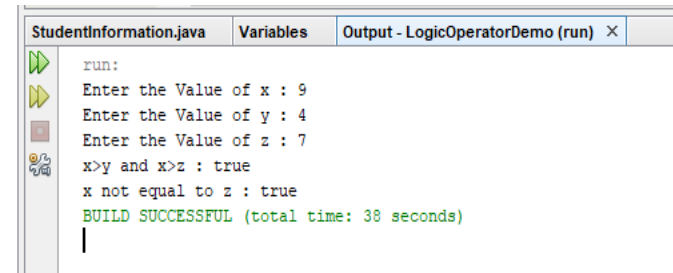
Toán tử Logic

- So sánh 2 hoặc nhiều biểu thức với nhau
- boolean
- true, false
- Các phép toán logic thực hiện từ trái sang phải cho đến khi đủ xác định giá trị của biểu thức logic
 - if (x!=0 && y/x>=3)... Việc tính toán dừng lại nếu x=0

Operator	Description	true Example	false Example
<	Less than	3 < 8	8 < 3
>	Greater than	4 > 2	2 > 4
==	Equal to	7 == 7	3 == 9
<=	Less than or equal to	5 <= 5	8 <= 6
>=	Greater than or equal to	7 >= 3	1 >= 2
!=	Not equal to	5 != 6	3 != 3

Ví dụ về toán tử Logic

```
8 import java.util.Scanner;
9
10 /**
11  *
12  * @author Quynh Tran Ly
13  */
14 public class LogicOperatorDemo {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21         int x, y, z;
22         Scanner scan = new Scanner(System.in);
23         System.out.print("Enter the Value of x : ");
24         x = scan.nextInt();
25         System.out.print("Enter the Value of y : ");
26         y = scan.nextInt();
27         System.out.print("Enter the Value of z : ");
28         z = scan.nextInt();
29         System.out.println("x>y and x>z : " + ((x > y) && (x > z)));
30         System.out.println("x not equal to z : " + (x != z));
31     }
32 }
33
34 }
35
```



The screenshot shows an IDE window titled "StudentInformation.java" with tabs for "Variables" and "Output - LogicOperatorDemo (run)". The output window displays the following text:

```
run:
Enter the Value of x : 9
Enter the Value of y : 4
Enter the Value of z : 7
x>y and x>z : true
x not equal to z : true
BUILD SUCCESSFUL (total time: 38 seconds)
```



Kiểu số nguyên

Type	Minimum Value	Maximum Value	Size in Bytes
byte	-128	127	1
short	-32,768	32,767	2
int	-2,147,483,648	2,147,483,647	4
long	-9,223,372,036,854,775,808	9,223,372,036,854,775,807	8



Kiểu số thực

- **Phạm vi :**
 - **float –32 bit - có thể chính xác đến 6 hoặc 7 chữ số**
 - **double – 64 bit - chính xác đến 14 hoặc 15 chữ số**
- **Các hằng giá trị sau đây để biểu diễn các giá trị đặc biệt (định nghĩa trong lớp Float and Double)**
 - **Float.NaN**
 - **Float.NEGATIVE_INFINITY**
 - **Float.POSITIVE_INFINITY**
 - **Double.NaN**
 - **Double.NEGATIVE_INFINITY**
 - **Double.POSITIVE_INFINITY**

Phép toán trên kiểu số

- **Tất cả các dữ liệu kiểu số sử dụng các toán tử sau:**

- Phép toán số học $+$, $-$, $*$, $/$
 - byte $a=3$, $b=2$, c ;
 - $c=a*b$; /* error phép nhân $a*b$ tự động chuyển thành int, vượt phạm vi của c */
 - $15/2$ bằng 7 – $15.0/2$ bằng 7.5
- Phép chia lấy số dư $\%$
 - $5\%3 = 2$
 - $3.14\%2.2=0.94$
- Phép tăng $++$ và giảm $--$, trước & sau
 - int $a=5$, $b=7$, c ;
 $c=a++ +b$; // $c=12$ $a=6$
 $c= ++a +b$; // $c=13$ $a=6$
- Kết hợp gán và số học $+=$, $-=$, $*=$, $/=$, $\% =$
 - $x+=3$; tương đương với $x=x+3$

- **Các phép toán sau đây chỉ áp dụng cho kiểu nguyên**

- Phép dịch bit trái, phải $<<$, $>>$, and $>>>$
- Phép thao tác trên bit (bitwise) $\&$ -and, $|$ -or, \wedge -xor, \sim -not



Kiểu ký tự

- Là kiểu nguyên không dấu
 - **char**
 - Các ký tự Java là Unicode 16-bit
 - Từ 0 đến $2^{16} - 1$
 - Biểu diễn giá trị char giữa cặp dấu ngoặc đơn: 'a'
- Một số ký tự đặc biệt
 - \t phím tab
 - \r về đầu dòng
 - \n xuống hàng
 - Dấu \ được đặt trước các ký tự đặc biệt khi muốn in ra bản thân các ký tự đó \" , \'

Ví dụ Kiểu dữ liệu nguyên thủy

```
12 class DataTypeSample
13 {
14     public static void main(String args[])
15     {
16         // declaring character
17         char a = 'G';
18
19         // Integer data type is generally
20         // used for numeric values
21         int i=89;
22
23         // use byte and short if memory is a constraint
24         byte b = 4;
25
26         // this will give error as number is
27         // larger than byte range
28         // byte b1 = 7888888955;
29
30         short s = 56;
31
32         // this will give error as number is
33         // larger than short range
34         // short s1 = 87878787878;
35
36
37         // by default fraction value is double in java
38         double d = 4.355453532;
39
40         // for float use 'f' as suffix
41         float f = 4.7333434f;
42
43         System.out.println("char: " + a);
44         System.out.println("integer: " + i);
45         System.out.println("byte: " + b);
46         System.out.println("short: " + s);
47         System.out.println("float: " + f);
48         System.out.println("double: " + d);
49     }
50 }
```

```
run:
char: G
integer: 89
byte: 4
short: 56
float: 4.7333436
double: 4.355453532
BUILD SUCCESSFUL (total time: 0 seconds)
|
```

Kiểu tham chiếu

- reference types
- Gồm class, interface, mảng.
- Giá trị của 1 biến kiểu tham chiếu là địa chỉ (reference) của 1 đối tượng (object)
 - 1 đối tượng(object) là 1 thể hiện của 1 lớp hoặc 1 mảng.
 - 1 giá trị tham chiếu thực chất là con trỏ chỉ đến 1 biến đối tượng,
- Hằng null dùng để chỉ đến 1 giá trị tham chiếu rỗng.
- Biến tham chiếu thường phải được khởi tạo bằng từ khóa new
 - String hoTen= new String("John Smith")
 - int[] a = new int[10];
- Toán tử instanceof để kiểm tra 1 biến tham chiếu thuộc lớp
 - a instanceof Integer // trả về true nếu a là biến thể hiện của Integer

```
13 class Tutor
14 {
15     int idNumber;
16     String fullName;
17     String course;
18     Tutor(int idNumber, String fullName, String course)
19     {
20         this.idNumber = idNumber;
21         this.fullName = fullName;
22         this.course = course;
23     }
24 }
25 public class ReferenceTypes
26 {
27     public static void main(String[] args)
28     {
29         Tutor tutor1 = new Tutor(1, "Nguyen Le", "Java" );
30         Tutor tutor2 = new Tutor(1, "Nguyen Le", "Java" );
31         System.out.println("Địa chỉ của biến tutor 1: "+ tutor1);
32         System.out.println("Tên của tutor 1: "+ tutor1.fullName);
33         tutor2 = null;
34     }
35 }
```

String

- Kiểu tham chiếu để biểu diễn hằng chuỗi
 - Lớp String
 - Biểu diễn giá trị chuỗi giữa 2 dấu “abc”
 - Toán tử + để nối 2 chuỗi
 - Tất cả các kiểu dữ liệu được tự động biến thành chuỗi khi tham gia phép + với 1 chuỗi
 - Một hằng chuỗi được xem là 1 biến của lớp String
“abc”.length() trả về 3
- Các hàm thường dùng
 - length()
 - equals()
 - substring()
 - charAt()
 - indexOf()..
- String Pool
- StringBuffer, StringTokenizer

```
13 public class StringMethodsDemo {
14     public static void main(String[] args) {
15         String targetString = "Java is fun to learn";
16         String s1 = "JAVA";
17         String s2 = "Java";
18         String s3 = " Hello Java ";
19
20         System.out.println("Char at index 2(third position): " + targetString.charAt(2));
21         System.out.println("After Concat: " + targetString.concat("-Enjoy-"));
22         System.out.println("Checking equals ignoring case: " + s2.equalsIgnoreCase(s1));
23         System.out.println("Checking equals with case: " + s2.equals(s1));
24         System.out.println("Checking Length: " + targetString.length());
25         System.out.println("Replace function: " + targetString.replace("fun", "easy"));
26         System.out.println("SubString of targetString: " + targetString.substring(8));
27         System.out.println("SubString of targetString: " + targetString.substring(8, 12));
28         System.out.println("Converting to lower case: " + targetString.toLowerCase());
29         System.out.println("Converting to upper case: " + targetString.toUpperCase());
30         System.out.println("Trimming string: " + s3.trim());
31         System.out.println("searching s1 in targetString: " + targetString.contains(s1));
32         System.out.println("searching s2 in targetString: " + targetString.contains(s2));
33
34         char [] charArray = s2.toCharArray();
35         System.out.println("Size of char array: " + charArray.length);
36         System.out.println("Printing last element of array: " + charArray[3]);
37     }
38 }
```

```
run:
Char at index 2(third position): v
After Concat: Java is fun to learn-Enjoy-
Checking equals ignoring case: true
Checking equals with case: false
Checking Length: 20
Replace function: Java is easy to learn
SubString of targetString: fun to learn
SubString of targetString: fun
Converting to lower case: java is fun to learn
Converting to upper case: JAVA IS FUN TO LEARN
Trimming string: Hello Java
searching s1 in targetString: false
searching s2 in targetString: true
Size of char array: 4
Printing last element of array: a
BUILD SUCCESSFUL (total time: 0 seconds)
```

Toán tử điều kiện

- Cú pháp:
 - Expr1? Expr2:Expr3
 - Vd: int a= (x>y)?10:5
- Biểu thức 1 expr1 phải có giá trị kiểu logic.
- Biểu thức 2 và 3 có thể nhận các giá trị kiểu số, logic hoặc tham chiếu nhưng phải cùng kiểu
- Biểu thức toán hạng 2 & 3 có thể là một lời gọi hàm trả về cùng kiểu – không phải void

```
12 //Program to Find greatest of three numbers using Conditional Operator
13 import java.util.Scanner; //program uses Scanner class
14 public class ConditionalOperator
15 {
16     public static void main(String[] args)
17     {
18         int a,b,c,result;
19         //create Scanner object to obtain input from keyboard
20         Scanner input=new Scanner(System.in);
21         System.out.print("Enter the Three Number : "); //prompt for input
22         a=input.nextInt(); //Read First number
23         b=input.nextInt(); //Read Second number
24         c=input.nextInt(); //Read third number
25         result = (a>b)? ((a>c)?a:c) : ((b>c)?b:c);
26         System.out.println( result + " is Greatest");
27     }
28 }
29 }
```

```
run:
Enter the Three Number : 8
9
15
15 is Greatest
BUILD SUCCESSFUL (total time: 4 seconds)
```

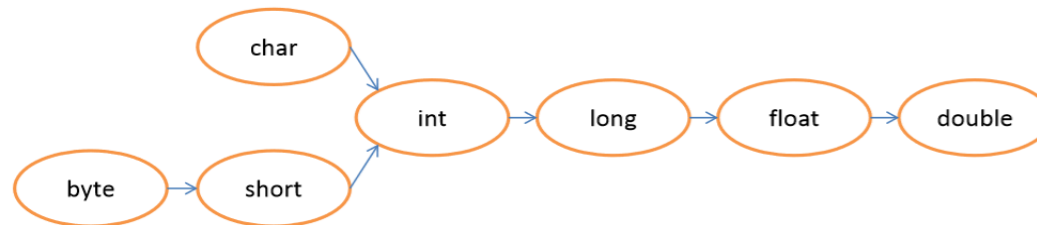
Thứ tự ưu tiên các toán tử

Operators	Associativity
[] . () (method call)	left to right
! ~ ++ -- + (unary) - (unary) () (cast) new	right to left
* / %	left to right
+ -	left to right
<< >> >>>	left to right
< <= > >= instanceof	left to right
== !=	left to right
&	left to right
^	left to right
	left to right
&&	left to right
	left to right
?:	left to right

Chuyển kiểu – ép kiểu

Chuyển kiểu là quá trình biến đổi kiểu dữ liệu của 1 giá trị/biến/biểu thức, tuân theo các quy luật sau:

- Không được chuyển từ kiểu tham chiếu thành kiểu nguyên thủy & ngược lại.
- Các giá trị bất kỳ có thể chuyển thành kiểu String.
- Kiểu mảng có thể chuyển thành kiểu Object hoặc String.
- Kiểu số được tự động chuyển sang kiểu số có phạm vi lớn hơn khi 2 toán hạng khác kiểu.



■ Cú pháp:

- Đặt (tenkiểu) trước giá trị/biểu thức cần chuyển kiểu
- Ví dụ:

```
int a = (int)3.14; //ép giá trị thực thành nguyên
float b= 4;        //tự động chuyển kiểu
long c=b;          // không hợp lệ, phải ép kiểu
Vector d = (Vector)x; // x được ép kiểu Vector
b= a+c;
```

Chuỗi sang kiểu nguyên thủy

Xét biểu thức 1

```
String a = "3";  
String b = "4";  
String c = a + b;
```

=> c là **"34"**

Xét biểu thức 2

```
int a = Integer.parseInt("3");  
int b = Integer.parseInt("4");  
int c = a + b;
```

=> c là **7**

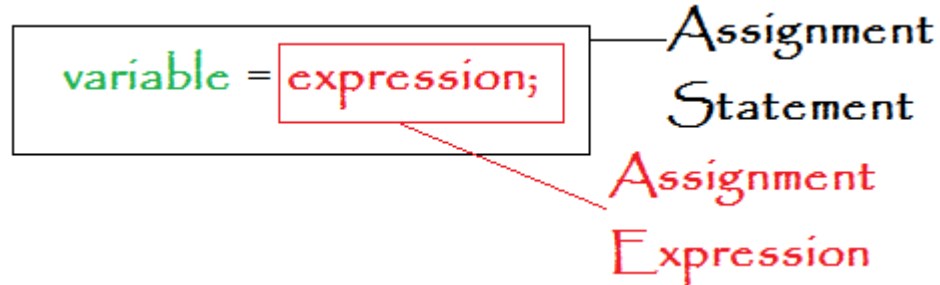
Chuỗi => Nguyên thủy
byte Byte. parseByte (String)
short Short. parseShort (String)
int Integer. parseInt (String)
long Long. parseLong (String)
float Float. parseFloat (String)
double Double. parseDouble (String)
boolean Boolean. parseBoolean (String)

Ví dụ về chuyển kiểu

```
12 public class TypeCast {
13
14     public static void main(String[] args) {
15         // TODO code application logic here
16         int myInt = 123;
17         double myDouble = 123.0;
18         String myStr = "123";
19         String txtStr = "Hello, World!";
20
21         /* It doesn't give actual value because result will be integer and it doesn't include decimal numbers */
22         System.out.println("myInt/5 gives :- " + (myInt / 5));
23
24         // Its modulus gives remainder as result
25         System.out.println("myInt%5 gives :- " + (myInt % 5));
26
27         /* It gives actual answer because double is decimal type and result is double */
28         System.out.println("myDouble/5 gives :- " + (myDouble / 5));
29
30         // Another type-cast by forcing double math
31         System.out.println("myInt/5.0 gives :- " + (myInt / 5.0));
32
33         // Attempt to convert string to int but it gives error
34         //System.out.println("myStr to int :- " + (int)myStr);
35
36         /* java has a Integer class which has a method parseInt it to convert string to integer */
37         System.out.println("Converting string to int :- " + Integer.parseInt(myStr));
38
39         // converting a text string to int gives runtime error
40         //System.out.println("Converting text string to int :- " + Integer.parseInt(txtStr));
41     }
42 }
43 }
44 }
```

StudentInformation.java	Variables	Output - TypeCast (run) ×
run:		
myInt/5 gives :- 24		
myInt%5 gives :- 3		
myDouble/5 gives :- 24.6		
myInt/5.0 gives :- 24.6		
Converting string to int :- 123		
BUILD SUCCESSFUL (total time: 0 seconds)		

Lệnh gán



- Dấu = được dùng cho lệnh gán
 - Tính toán giá trị của biểu thức ở vế phải, gán cho biến ở vế trái
 - Phải cùng kiểu hoặc tự động chuyển kiểu hợp lệ
 - Trả về giá trị được gán
- Ví dụ
 - `double x,y,z;`
`x=y=z=3.14; // gán đồng thời x,y,z giá trị 3.14`
 - `String xepLoai='c'; // không hợp lệ`
`String namSinh="" + 1985; // hợp lệ`

Ví dụ Lệnh gán

```
12 public class Java1LenhGan {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         int a = 20;
20         int b = 5;
21         int c = 10;
22         int result1=(a + b * c);
23         int result2=((a + b) * c);
24         int result3=(a / b - c);
25         int result4=(a / (b - c));
26         System.out.println("a + b * c = " + result1);
27         System.out.println("(a + b) * c = " + result2);
28         System.out.println("a / b - c = " + result3);
29         System.out.println("a / (b - c) = " + result4);
30
31     }
32
33 }
```

```
run:
a + b * c      = 70
(a + b) * c    = 250
a / b - c      = -6
a / (b - c)    = -4
BUILD SUCCESSFUL (total time: 0 seconds)
```

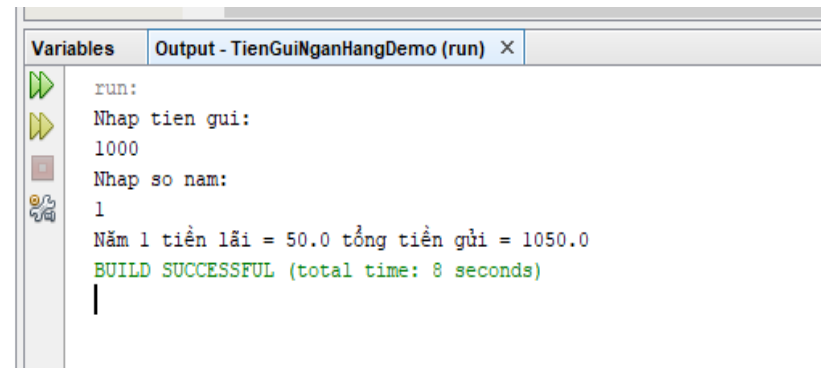


Bài tập về lệnh gán

- **Viết chương trình tính số tiền trong ngân hàng sau soNam với soNam là giá trị nhập từ bàn phím**
- Giả sử hiện tại bạn đang gửi ngân hàng là \$10,000. Hãy viết chương trình số tiền trong ngân hàng của bạn sau 10 năm với lãi suất là 5%/năm.
- Input: double soNam, soTien, tienLai
- Body:
 - `tienLai=(soTien*5/100)*soNam;`
 - `soTien=soTien+tienLai;`
- Output: double soTien;

Bài tập về lệnh gán

```
8 import java.util.Scanner;
9
10 /**
11  *
12  * @author Quynh Tran Ly
13  */
14 public class TienGuiNganHangDemo {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21         int soNam = 0;
22         double tienGui, tienLai;
23         Scanner scanner = new Scanner(System.in);
24         System.out.println("Nhap tien gui: ");
25         tienGui = scanner.nextDouble();
26         System.out.println("Nhap so nam: ");
27         soNam = scanner.nextInt();
28
29         tienLai = tienGui * 5 / 100; // tiền lãi
30         tienGui += tienLai;
31         System.out.println("Năm " + soNam + " tiền lãi = " + tienLai
32             + " tổng tiền gửi = " + tienGui);
33     }
34 }
35
```



Variables	Output - TienGuiNganHangDemo (run) X
run:	
	Nhap tien gui:
	1000
	Nhap so nam:
	1
	Năm 1 tiền lãi = 50.0 tổng tiền gửi = 1050.0
	BUILD SUCCESSFUL (total time: 8 seconds)

Cấu trúc rẽ nhánh if()/else

■ Cú pháp

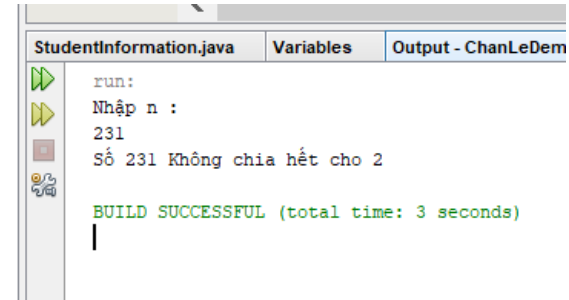
```
if (<<điều kiện>>)  
{  
    << công việc 1 >>  
}  
else  
{  
    << công việc 2 >>  
}
```

```
double diem = 4;  
if (diem < 5) {  
    System.out.println("Rớt");  
}  
else {  
    System.out.println("Đậu");  
}
```

- Nếu điều kiện có giá trị true thì công việc 1 được thực hiện, ngược lại công việc 2 được thực hiện
- Công việc 1, công việc 2 có thể là 1 khối bao giữa cặp toán tử { }
- Có thể có nhiều if()/else lồng nhau
- Có thể không có mệnh đề else

Ví dụ về if()/else

```
8 import java.util.Scanner;
9
10 /**
11  *
12  * @author Quynh Tran Ly
13  */
14 public class ChanLeDemo {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21
22         Scanner input = new Scanner(System.in);
23         System.out.println("Nhập n : ");
24         int n = input.nextInt();
25         if(n % 2 == 0)
26             System.out.println("Số " + n + " Chia hết cho 2");
27         else
28             System.out.println("Số " + n + " Không chia hết cho 2");
29
30         System.out.println();
31     }
32 }
33 }
34 }
```



Ví dụ về if()/else

```
9  * Giải phương trình bậc 2:  $ax^2 + bx + c = 0$ 
10 *
11 * @param a: hệ số bậc 2
12 * @param b: hệ số bậc 1
13 * @param c: số hạng tự do
14 */
15 import java.util.Scanner;
16 public class Java1PTBacHai {
17     private static Scanner scanner = new Scanner(System.in);
18     public static void giaiPTBac2(float a, float b, float c) {
19         // kiểm tra các hệ số
20         if (a == 0) {
21             if (b == 0) {
22                 System.out.println("Phương trình vô nghiệm!");
23             } else {
24                 System.out.println("Phương trình có một nghiệm: "
25                     + "x = " + (-c / b));
26             }
27             return;
28         }
29         // tính delta
30         float delta = b*b - 4*a*c;
31         float x1;
32         float x2;
33         // tính nghiệm
34         if (delta > 0) {
35             x1 = (float) ((-b + Math.sqrt(delta)) / (2*a));
36             x2 = (float) ((-b - Math.sqrt(delta)) / (2*a));
37             System.out.println("Phương trình có 2 nghiệm là: "
38                 + "x1 = " + x1 + " và x2 = " + x2);
39         } else if (delta == 0) {
40             x1 = (-b / (2 * a));
41             System.out.println("Phương trình có nghiệm kép: "
42                 + "x1 = x2 = " + x1);
43         } else {
44             System.out.println("Phương trình vô nghiệm!");
45         }
46     }
47 }
```

```
47 public static void main(String[] args) {
48     System.out.print("Nhập số a = ");
49     float a = scanner.nextFloat();
50     System.out.print("Nhập số b = ");
51     float b = scanner.nextFloat();
52     System.out.print("Nhập số c = ");
53     float c = scanner.nextFloat();
54     giaiPTBac2(a, b, c);
55 }
56
57
58 }
```

```
run:
Nhập hệ số bậc 2, a = -3
Nhập hệ số bậc 1, b = -4
Nhập hằng số tự do, c = 7
Phương trình có 2 nghiệm là: x1 = -2.3333333 và x2 = 1.0
BUILD SUCCESSFUL (total time: 8 seconds)
```

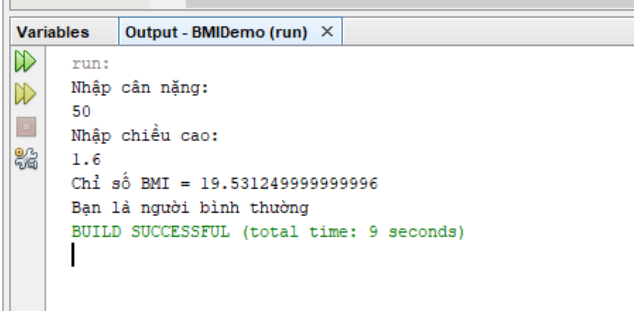


Bài tập về if()/else

- **Viết chương trình tính chỉ số BMI và dựa vào chỉ số đó để đưa ra kết luận**
- Chỉ số khối cơ thể (viết tắt là BMI - Body Mass Index) được dùng để đánh giá mức độ gầy hay béo của một người.
 - Chỉ số này có thể giúp xác định một người bị bệnh béo phì hay bị bệnh suy dinh dưỡng.
 - Cách tính chỉ số này như sau: $BMI = W / H^2$
 - trong đó W là khối lượng của một người (tính bằng kg) và H là chiều cao của người đó (tính bằng m).
- Dựa vào chỉ số BMI chúng ta có thể phân loại như sau:
 - $BMI < 18$: người gầy.
 - $BMI = 18 - 24.9$: người bình thường.
 - $BMI = 25 - 29.9$: người béo phì độ I.
 - $BMI = 30 - 34.9$: người béo phì độ II.
 - $BMI > 35$: người béo phì độ III.
- Hãy xây dựng chương trình tính chỉ số BMI và đưa ra kết luận.
- **Lưu ý:** kết quả chỉ số BMI phải được làm tròn đến 2 chữ số thập phân.

Bài tập về if()/else

```
8 import java.util.Scanner;
9
10 /**
11  *
12  * @author Quynh Tran Ly
13  */
14 public class BMIDemo {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21         double chieuCao, canNang, chiSoBMI;
22         String ketQua;
23
24         Scanner scanner = new Scanner(System.in);
25         System.out.println("Nhập cân nặng: ");
26         canNang = scanner.nextDouble();
27         System.out.println("Nhập chiều cao: ");
28         chieuCao = scanner.nextDouble();
29
30         // tính chỉ số BMI
31         chiSoBMI = canNang / (chieuCao * chieuCao);
32         System.out.println("Chỉ số BMI = " + chiSoBMI);
33
34         // phân loại dựa vào chỉ số BMI
35         if (chiSoBMI < 18) {
36             ketQua = "Bạn là người gầy!";
37         } else if (chiSoBMI <= 24.9) {
38             ketQua = "Bạn là người bình thường";
39         } else if (chiSoBMI <= 29.9) {
40             ketQua = "Bạn bị béo phì độ I";
41         } else if (chiSoBMI <= 34.9) {
42             ketQua = "Bạn bị béo phì độ II";
43         } else {
44             ketQua = "Bạn bị béo phì độ III";
45         }
46     }
47 }
```



The screenshot shows an IDE window titled "Output - BMIDemo (run)". The output text is as follows:

```
run:
Nhập cân nặng:
50
Nhập chiều cao:
1.6
Chỉ số BMI = 19.531249999999996
Bạn là người bình thường
BUILD SUCCESSFUL (total time: 9 seconds)
```



Bài tập về if()/else

- **Viết chương trình tính chỉ số BMI và dựa vào chỉ số đó để đưa ra kết luận**
- Chỉ số khối cơ thể (viết tắt là BMI - Body Mass Index) được dùng để đánh giá mức độ gầy hay béo của một người.
 - Chỉ số này có thể giúp xác định một người bị bệnh béo phì hay bị bệnh suy dinh dưỡng.
 - Cách tính chỉ số này như sau: $BMI = W / H^2$
 - trong đó W là khối lượng của một người (tính bằng kg) và H là chiều cao của người đó (tính bằng m).
- Dựa vào chỉ số BMI chúng ta có thể phân loại như sau:
 - $BMI < 18$: người gầy.
 - $BMI = 18 - 24.9$: người bình thường.
 - $BMI = 25 - 29.9$: người béo phì độ I.
 - $BMI = 30 - 34.9$: người béo phì độ II.
 - $BMI > 35$: người béo phì độ III.
- Hãy xây dựng chương trình tính chỉ số BMI và đưa ra kết luận.
- **Lưu ý:** kết quả chỉ số BMI phải được làm tròn đến 2 chữ số thập phân.



Bài tập về if()/else

- Viết chương trình nhập vào số điện sử dụng của tháng và tính tiền điện theo phương pháp lũy tiến
 - Nếu số điện sử dụng từ 0 đến 50 thì giá mỗi số điện là 1000
 - Nếu số điện sử dụng trên 50 thì giá mỗi số điện vượt hạn mức là 1200
- Viết chương trình nhập vào 3 số a,b,c. Kiểm tra 3 số nhập vào có phải là 3 cạnh của tam giác? Nếu đúng 3 cạnh tam giác, cho biết tam giác loại gì (BTLTJava, trang 13)

Cấu trúc switch()

- Switch: rẽ nhiều nhánh theo điều kiện nguyên
 - Cú pháp
 - Diễn giải:

```
switch (<<biểu thức>>)
{
    case <<giá trị 1>>:
        // Công việc 1
        break;
    case <<giá trị 2>>:
        // Công việc 2
        break;
    ...
    default:
        // Công việc N+1
        break;
}
```

- So sánh các giá trị của biểu thức switch với các giá trị của biểu thức case. Nếu bằng với giá trị của case nào thì sẽ thực hiện công việc của case đó, ngược lại sẽ thực hiện công việc của **default**
- Nếu công việc của case không chứa lệnh **break**; thì case sau tiếp tục sẽ được thực hiện.
- default là tùy chọn

Ví dụ về switch()

Switch: rẽ nhiều nhánh theo điều kiện nguyên

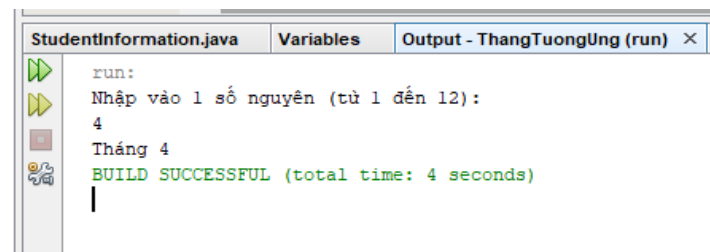
```
12 import java.util.Scanner;
13 public class SwitchExample
14 { public static void main(String[] args)
15   { Scanner input = new Scanner(System.in);
16     System.out.print("Enter your grade: ");
17     String userInput = input.next();
18     char grade = userInput.charAt(0);
19     switch (grade)
20     {
21         case 'A':    System.out.println("Excellent!"); break;
22         case 'B':    System.out.println("Great!");
23         case 'C':
24         case 'D':    System.out.println("Well done!"); break;
25         case 'F':    System.out.println("Sorry, you failed."); break;
26         default:     System.out.println("Error! Invalid grade.");break;
27     }
28 }
29 }
```

```
run:
Enter your grade: C
Well done!
BUILD SUCCESSFUL (total time: 2 seconds)
```

Ví dụ về switch()

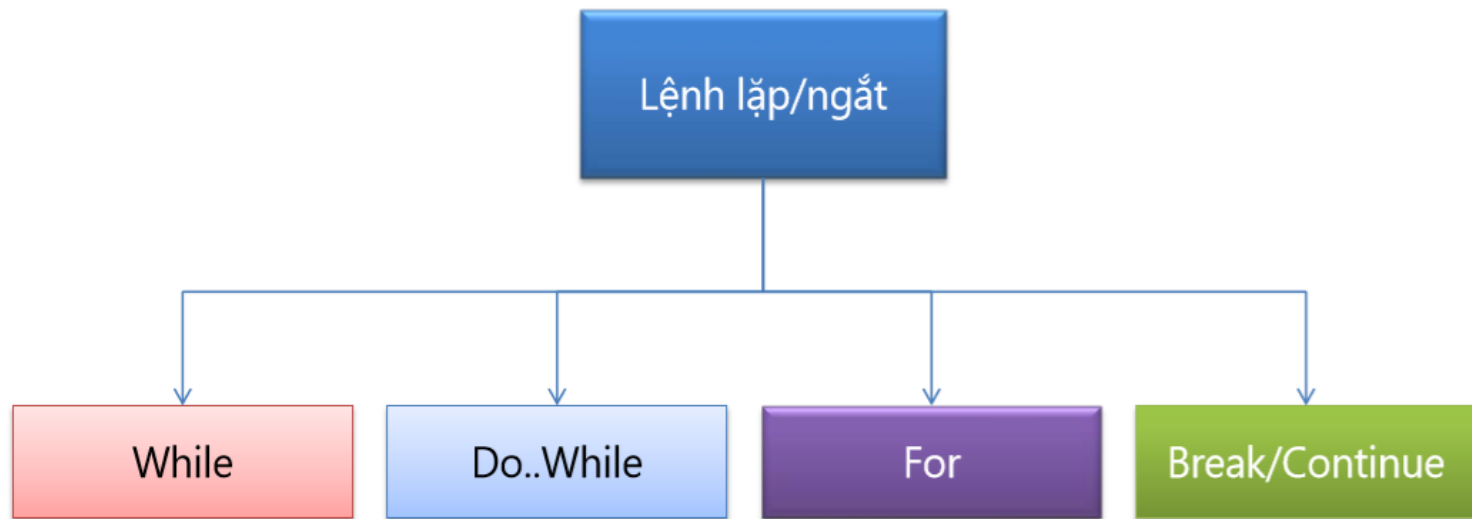
Switch: rẽ nhiều nhánh theo điều kiện nguyên

```
8 import java.util.Scanner;
9 public class ThangTuongUng {
10     public static void main(String[] args) {
11         int thang;
12         Scanner scanner = new Scanner(System.in);
13         System.out.println("Nhập vào 1 số nguyên (từ 1 đến 12): ");
14         thang = scanner.nextInt();
15         switch (thang) {
16             case 1:
17                 System.out.println("Tháng 1");
18                 break;
19             case 2:
20                 System.out.println("Tháng 2");
21                 break;
22             case 3:
23                 System.out.println("Tháng 3");
24                 break;
25             case 4:
26                 System.out.println("Tháng 4");
27                 break;
28             case 5:
29                 System.out.println("Tháng 5");
30                 break;
31             case 6:
32                 System.out.println("Tháng 6");
33                 break;
34             case 7:
35                 System.out.println("Tháng 7");
36                 break;
37             case 8:
38                 System.out.println("Tháng 8");
39                 break;
40             case 9:
41                 System.out.println("Tháng 9");
42                 break;
43             case 10:
44                 System.out.println("Tháng 10");
45                 break;
46             case 11:
47                 System.out.println("Tháng 11");
48                 break;
49             case 12:
50                 System.out.println("Tháng 12");
51                 break;
52             default:
53                 System.out.println("Số nhập vào phải nằm trong khoảng từ 1 đến 12.");
54         }
```



```
run:
Nhập vào 1 số nguyên (từ 1 đến 12):
4
Tháng 4
BUILD SUCCESSFUL (total time: 4 seconds)
```

Cấu trúc lặp



- Phần chương trình lặp đi lặp lại một lệnh hoặc một khối lệnh được gọi là một vòng lặp.
- Khi thiết kế một vòng lặp, ta cần xác định thân vòng lặp thực hiện hành động gì. Ngoài ra, ta còn cần một cơ chế để quyết định khi nào vòng lặp sẽ kết thúc

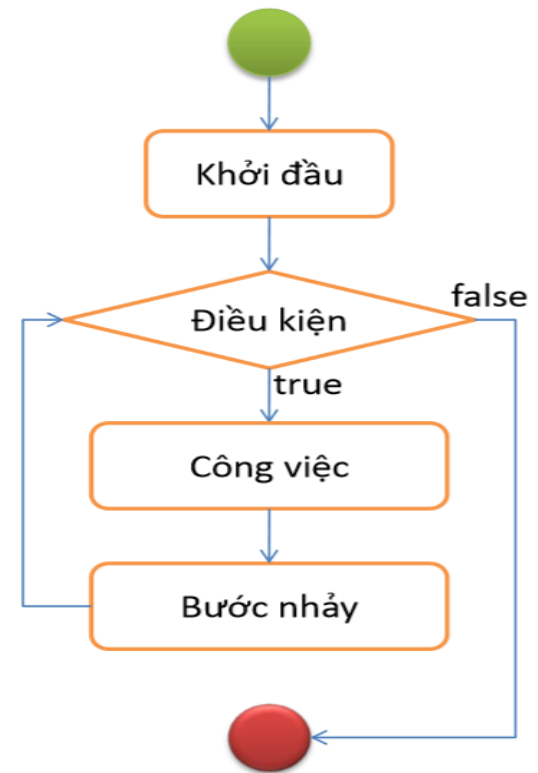
Cấu trúc lặp for

- Cú pháp

```
for (khởi đầu ; điều kiện; bước nhảy){  
    // công việc  
}
```






- Diễn giải:

- bước 1: Thực hiện khởi đầu:
 - khởi gán biến.
- bước 2: kiểm tra điều kiện
 - nếu true: Bước 3
 - Nếu false: Kết thúc
- bước 3: Thực hiện công việc,
- Bước 4: Thực hiện bước nhảy
- Bước 5: quay lại bước 2



Ví dụ về for

```
12 public class LenhForDemo {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         // Khai báo một biến 'step', mô tả bước của vòng lặp (Vòng lặp thứ mấy)
20         int value = 0;
21         for (int i = 0; i < 5; i++) {
22             System.out.println("Step =" + i + " value =" + value);
23             value = value + 5;
24         }
25     }
26 }
27
28 }
29
```

StudentInformation.java	Variables	Output - LenhForDemo (run)
		run:
		Step =0 value = 0
		Step =1 value = 5
		Step =2 value = 10
		Step =3 value = 15
		Step =4 value = 20
		BUILD SUCCESSFUL (total time: 0 seconds)

Exercises

- Write a program in C to display the n terms of harmonic series and their sum. The series is :
 $1 + 1/2 + 1/3 + 1/4 + 1/5 \dots 1/n$ terms
- Write a program in C to display the sum of the series [
 $1+x+x^2/2!+x^3/3!+....$].



Number of terms : 5



$$1/1 + 1/2 + 1/3 + 1/4 + 1/5$$

Sum of Series upto 5 terms : 2.283334

© w3resource.com



Exercises

- Write a program in C to Check Whether a Number can be perfect Numbers.

Perfect Number

Divisor of 28 : 1, 2, 4, 7, 14, 28

Sum of 1 + 2 + 4 + 7 + 14 = 28

Sum = Original Number

28 is Perfect number

Cấu trúc lặp for

- Viết chương trình tính tổng các số chẵn từ 2 đến N (Nguyên dương, nhập từ bàn phím) (BTLTJava, trang 14).
- Viết chương trình kiểm tra số nguyên dương N nhập vào từ bàn phím có phải là số nguyên tố không? (BTLTJava, trang 16).
- Viết chương trình tính $S = 7! + 10! + 12! + 14!$ (BTLTJava, trang 18)

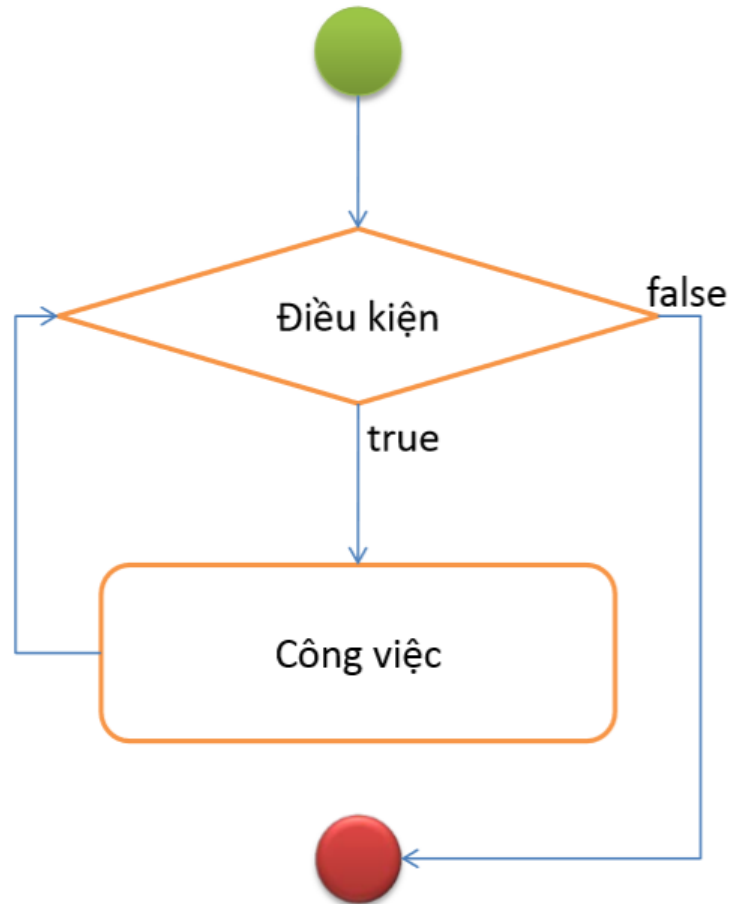
Cấu trúc lặp –while

- Cú pháp

```
while (<<điều kiện>>) {  
    // công việc  
}
```

- Diễn giải

- Thực hiện công việc trong khi biểu thức điều kiện có giá trị là true.



while() statements

How To Calculate Your Loan Interest



Interest

=



Principal

x



Rate

x



Time



For Example:

x

6%

x



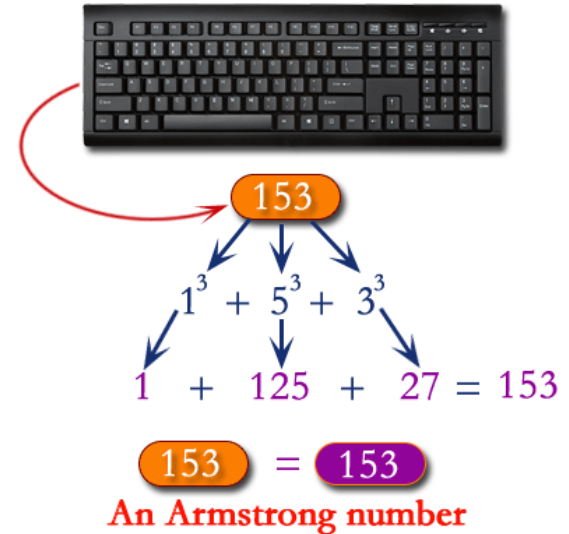
=



the balance

Exercises

- Write a program in Java to check Armstrong number of n digits.

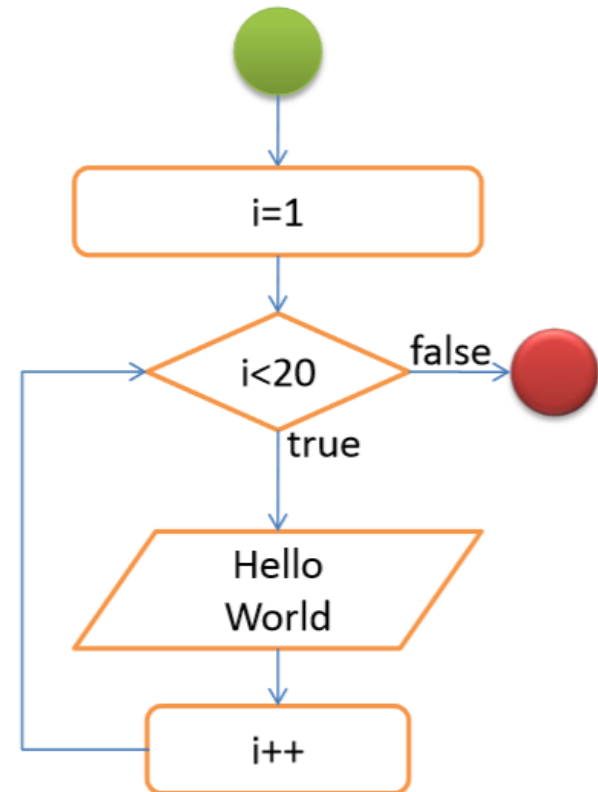


© w3resource.com

Ví dụ về while

```
14 import java.util.Scanner;
15 public class WhileDemo {
16     public static void main(String[] args) {
17         // TODO code application logic here
18         int i = 1;
19         Scanner input = new Scanner(System.in);
20         System.out.println("Moi nhap so lan: ");
21         int number = input.nextInt();
22         while (i <= number)
23         { System.out.println("Hello World");
24           i++;
25         }
26     }
27 }
```

```
Output - WhileDemo (run) x
run:
Moi nhap so lan:
5
Hello World
Hello World
Hello World
Hello World
Hello World
BUILD SUCCESSFUL (total time: 2 seconds)
```



Cấu trúc lặp – do.. while

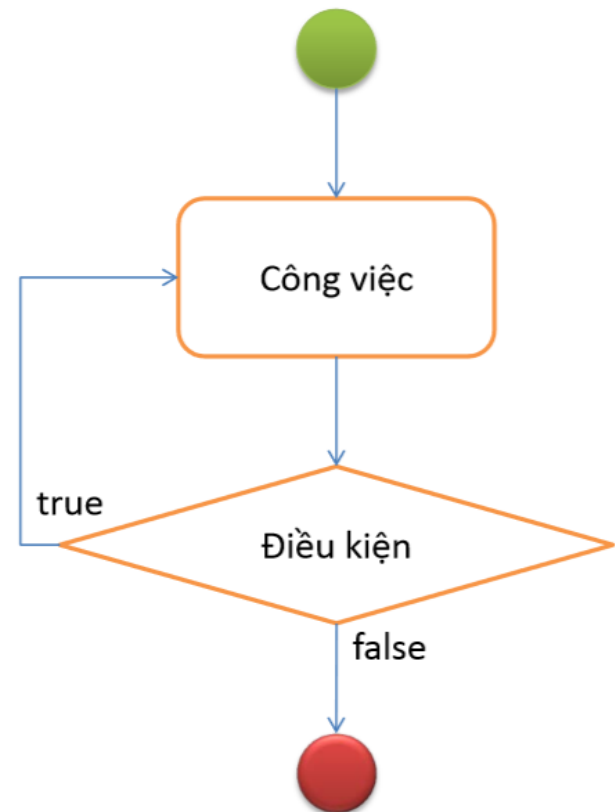
- Cú pháp

```
do {  
    // công việc  
}
```

```
while (<<điều kiện>>);
```

- **Diễn giải:**

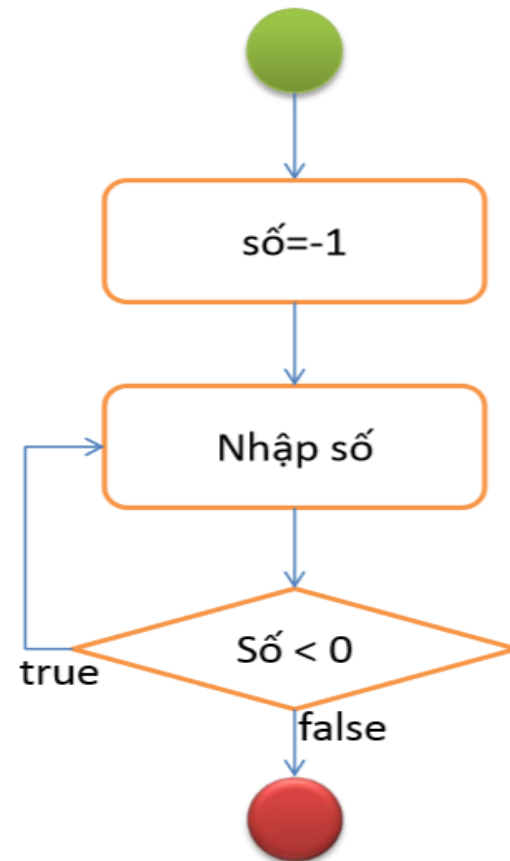
- Thực hiện công việc trong khi Điều kiện đúng. Công việc được thực hiện ít nhất một lần.



Ví dụ về do.. while

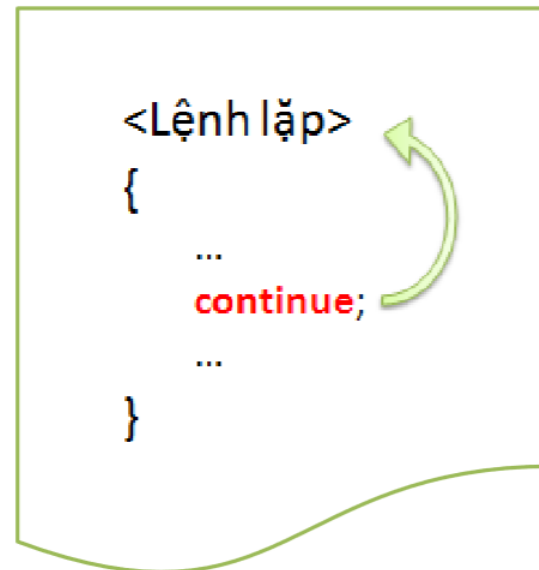
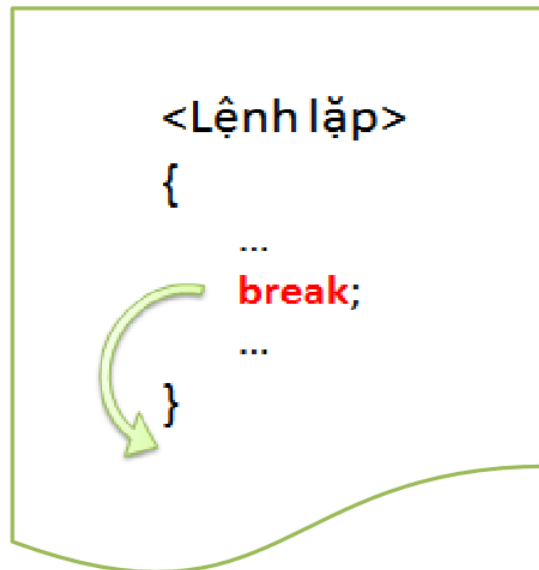
```
8 import java.util.Scanner;
9 public class DoWhileDemo {
10     public static void main(String[] args) {
11         // TODO code application logic here
12         Scanner input = new Scanner(System.in);
13         int so=-1;
14         do{
15             System.out.println("Moi nhap so: ");
16             so = input.nextInt();
17         }while(so<0);
18     }
19 }
20
```

```
run:
Moi nhap so:
-3|
Moi nhap so:
5
BUILD SUCCESSFUL (total time: 7 seconds)
```



Cấu trúc break và continue

- Cú pháp
 - `break;` dùng để ngắt
 - `Continue;` dùng để thực hiện lần lặp tiếp theo ngay lập tức



Ví dụ về break và continue

```
14 public class BreakDemo {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21         Scanner input=new Scanner(System.in);
22         int diem=0;
23         while(true)
24         {   System.out.print("Diem so ");
25             diem=input.nextInt();
26             if(diem>0 &&diem<=10)
27                 break;
28         }
29         System.out.print("Diem so phai tu 0 den 10");
30     }
31
32 }
```

```
run:
Diem so 6
Diem so phai tu 0 den 10BUILD SUCCESSFUL (total time: 2 seconds)
|
```

Kiểu mảng và vòng lặp

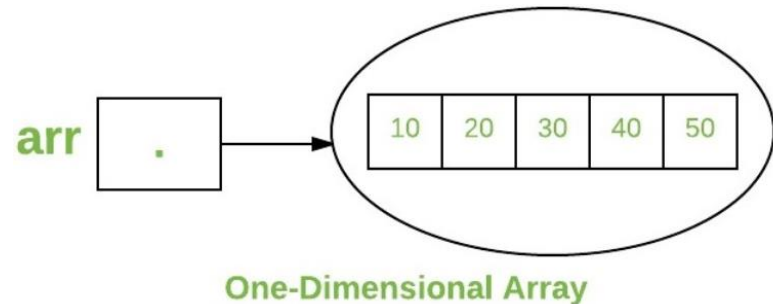
- Mảng là cấu trúc lưu trữ phần tử có cùng kiểu dữ liệu
- Để truy xuất các phần tử của mảng, ta cần biết tên mảng và chỉ số index
- Chỉ số được đánh từ 0

0	1	2	3	4	5	6	7	8	← Indices
5	7	9	1	45	1	9	9	2	← Elements

- Khai báo:
 - Cú pháp
 - `Tenkiieu[] tenMảng; hoặc Tenkiieu tenMảng[];`
 - `Tenkiieu[] tenMảng= new tenkiieu[sophantu];`
 - `Tenkiieu[] tenMảng={liệt kê phần tử mảng cách nhau bởi dấu phẩy};`
- Các thao tác trên mảng: khai báo, truy xuất phần tử, đọc ghi, duyệt mảng, sắp xếp mảng

Ví dụ Kiểu mảng

```
13 class GFGArray
14 {
15     public static void main (String[] args)
16     {
17         // declares an Array of integers.
18         int[] arr;
19
20         // allocating memory for 5 integers.
21         arr = new int[5];
22
23         // initialize the first elements of the array
24         arr[0] = 10;
25
26         // initialize the second elements of the array
27         arr[1] = 20;
28
29         //so on...
30         arr[2] = 30;
31         arr[3] = 40;
32         arr[4] = 50;
33
34         // accessing the elements of the specified array
35         for (int i = 0; i < arr.length; i++)
36             System.out.println("Element at index " + i +
37                               " : "+ arr[i]);
38     }
39 }
```



```
run:
Element at index 0 : 10
Element at index 1 : 20
Element at index 2 : 30
Element at index 3 : 40
Element at index 4 : 50
BUILD SUCCESSFUL (total time: 0 seconds)
```



Kiểu mảng

- **Ví dụ**
- Khai báo chưa khởi tạo
 - `int a[]; //khai báo biến a là mảng kiểu nguyên, chưa khởi tạo`
 - `int a[] = new int[10];`
- Khai báo có khởi tạo giá trị
 - `String b[]={“Java”,“Sun Mircosystem”,“Oak”,“1995”}`
 - `int c[]={1,2,3,4,5,6,7,8,9};`
- Thuộc tính length chỉ ra số phần tử mảng
`String a[]= {“a”,“b”,“c”,“def”} -> a.length bằng 4`
- Tham chiếu mảng
 - Theo chỉ số phần tử bắt đầu từ 0 đến n-1
 - Không được phép tham chiếu vượt chỉ số mảng
 - `a[0]=12;`

Ví dụ Kiểu mảng

```
15 class Student
16 {
17     public int roll_no;
18     public String name;
19     Student(int roll_no, String name)
20     {
21         this.roll_no = roll_no;
22         this.name = name;
23     }
24 }
25
26 // Elements of array are objects of a class Student.
27 public class GFGStudent
28 {
29     public static void main (String[] args)
30     {
31         // declares an Array of integers.
32         Student[] arr;
33
34         // allocating memory for 5 objects of type Student.
35         arr = new Student[5];
36
37         // initialize the first elements of the array
38         arr[0] = new Student(1, "aman");
39
40         // initialize the second elements of the array
41         arr[1] = new Student(2, "vaibhav");
42
43         // so on...
44         arr[2] = new Student(3, "shikar");
45         arr[3] = new Student(4, "dharmesh");
46         arr[4] = new Student(5, "mohit");
47
48         // accessing the elements of the specified array
49         for (int i = 0; i < arr.length; i++)
50             System.out.println("Element at " + i + " : " +
51                               arr[i].roll_no + " " + arr[i].name);
52     }
53 }
```

```
run:
Element at 0 : 1 aman
Element at 1 : 2 vaibhav
Element at 2 : 3 shikar
Element at 3 : 4 dharmesh
Element at 4 : 5 mohit
BUILD SUCCESSFUL (total time: 0 seconds)
```


Duyệt mảng

- Duyệt mảng
 - Sử dụng vòng lặp for để duyệt mảng

```
int[] a = {4, 3, 5, 9};  
for(int i=0; i<a.length; i++){  
    System.out.println(a[i]);  
}
```

← for(;;)

- Sao chép mảng
System.arraycopy(from, fromIndex, to, toIndex, count);
- Sắp xếp mảng
Arrays.sort(tenbienmang);
dùng sắp xếp các mảng kiểu nguyên thủy
- Mảng nhiều chiều thực chất là mảng của mảng

Duyệt mảng

- Sắp xếp mảng
`Arrays.sort(tenbienmang);`

```
int[] a = {9, 3, 8, 7, 3, 9, 4, 2};  
  
System.out.println("Mảng gốc: " + Arrays.toString(a));  
[9, 3, 8, 7, 3, 9, 4, 2]  
  
Arrays.sort(a);  
System.out.println("Sau sort: " + Arrays.toString(a));  
[2, 3, 3, 4, 7, 8, 9, 9]  
  
int i = Arrays.binarySearch(a, 8);  
System.out.println("Vị trí của 8 là " + i);  
Vị trí của 8 là 5  
  
Arrays.fill(a, 0);  
System.out.println("Sau fill: " + Arrays.toString(a));  
[0, 0, 0, 0, 0, 0, 0, 0]
```

Mảng gốc: [9, 3, 8, 7, 3, 9, 4, 2]
Sau sort: [2, 3, 3, 4, 7, 8, 9, 9]
Vị trí của 8 là 5
Sau fill: [0, 0, 0, 0, 0, 0, 0, 0]

Ví dụ Kiểu mảng

```
4 import java.util.Scanner;
5 public class Main {
6
7     private static final Scanner in = new Scanner(System.in);
8
9     //check user input a number integer
10    private static int checkInputInt() {
11        //loop until user input correct
12        while (true) {
13            try {
14                int result = Integer.parseInt(in.nextLine().trim());
15                return result;
16            } catch (NumberFormatException e) {
17                System.err.println("Please input number");
18                System.out.print("Enter again: ");
19            }
20        }
21    }
22    //allow user input number of array
23
24    private static int inputSizeOfArray() {
25        System.out.print("Enter number of array: ");
26        int n = checkInputInt();
27        return n;
28    }
29    //allow user input value of array
30
31    private static int[] inputValueOfArray(int n) {
32        int[] a = new int[n];
33        for (int i = 0; i < n; i++) {
34            System.out.print("Enter a[" + i + "]: ");
35            a[i] = checkInputInt();
36        }
37        return a;
38    }
39
40    //swap 2 number
41    private static void swap(int a, int b) {
42        int temp = a;
43        a = b;
44        b = temp;
45        System.out.println("hellllo");
46    }
47
```

```
48 //sort array by bubble sort
49 private static void sortArrayByBubbleSort(int[] a) {
50     System.out.print("Unsorted array: ");
51     for (int i = 0; i < a.length; i++) {
52         System.out.print(a[i] + " ");
53     }
54     for (int i = 0; i < a.length; i++) {
55         for (int j = 0; j < a.length - i - 1; j++) {
56             if (a[j] > a[j + 1]) {
57                 int temp = a[j];
58                 a[j] = a[j + 1];
59                 a[j + 1] = temp;
60             }
61         }
62     }
63     System.out.println();
64 }
65
66 //display array after sort
67 private static void print(int[] a) {
68     System.out.print("Sorted array: ");
69     for (int i = 0; i < a.length; i++) {
70         System.out.print(a[i] + " ");
71     }
72 }
73
74 public static void main(String[] args) {
75     int n = inputSizeOfArray();
76     int[] a = inputValueOfArray(n);
77     sortArrayByBubbleSort(a);
78     print(a);
79 }
80
```

Tổ chức chương trình Menu

- Ứng dụng Java thường được thiết kế theo cách sử dụng Menu để thuận tiện cho người sử dụng

run:

```
=====
|  MENU SELECTION DEMO  |
=====
| Options:              |
|    1. Option 1       |
|    2. Option 2       |
|    3. Exit           |
=====
```

Choose option

3

Exit selected

BUILD SUCCESSFUL (total time: 2 seconds)

```
8 import java.util.Scanner;
9 public class ConsoleMenuDemo {
10
11     static void function1()
12     {}
13     static void function2()
14     {}
15     public static void main(String[] args) {
16         // TODO code application logic here
17         Scanner input = new Scanner(System.in);
18         int choice;
19         while(true)
20         {
21             // Display menu graphics
22             System.out.println("=====");
23             System.out.println("|  MENU SELECTION DEMO  |");
24             System.out.println("=====");
25             System.out.println("| Options:              |");
26             System.out.println("|    1. Option 1       |");
27             System.out.println("|    2. Option 2       |");
28             System.out.println("|    3. Exit           |");
29             System.out.println("=====");
30             System.out.println("Choose option");
31             choice = input.nextInt();
32             // Switch construct
33             switch (choice) {
34                 case 1:
35                     System.out.println("Option 1 selected");
36                     function1();
37                     break;
38                 case 2:
39                     System.out.println("Option 2 selected");
40                     function2();
41                     break;
42                 case 3:
43                     System.out.println("Exit selected");
44                     return;
45                 default:
46                     System.out.println("Invalid selection");
47                     break; // This break is not really necessary
48             }
49         }
50     }
51 }
```



Tóm tắt

- Các cấu trúc lệnh
 - Gán
 - Rẽ nhánh
 - Lặp



Bài tập chương II

1. Chạy lại tất cả các bài trong Bài giảng (*)
2. Bài 3, 4, 5, 6, 7, 8, 9, 10 trang 11 đến 21 trong sách Bài tập (**)
3. Bài tập tự giải:
 - Bài 1 đến bài 11 trang 21 đến 22 trong (**)
- Sinh viên lưu folder dưới dạng **<class>-<name><Student ID>Chapter2** và nộp cho GV