



Dữ liệu & cấu trúc điều khiển trong Java

Môn học: Kỹ thuật lập trình [Tuần 2]

GV: ThS. Trần Đức Hiếu

Cấu trúc chương trình

Cấu trúc chương trình đơn giản

- Bao gồm 2 phần
 - Khai báo thư viện, module:
import <tên thư viện>;
package <tên module, gói thư viện>;
 - Class & hàm main
public class <tenClass>{
 public static void main(String[] args)
 {
 *//--- Mã nguồn dưới dạng các lệnh cần thực thi*

 }
}

Cấu trúc chương trình

Khai báo class với từ khóa public

Khai báo: package, import, ...

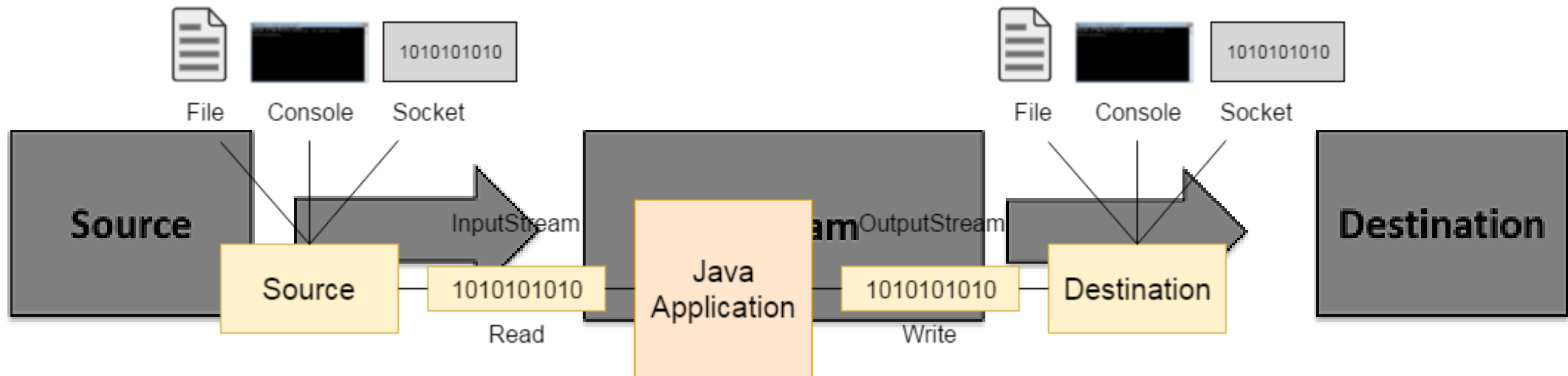
```
1 import java.util.Scanner;
2
3 public class HelloWorld {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         System.out.println("Hello World");
8         System.out.println("Hello World");
9     }
10
11 }
12
```

public class phải có hàm main, được khai báo public static void

Lệnh được thực hiện dựa vào Class, Object & Method. Trong trường hợp trên, ta gọi đối tượng out của lớp System, và thông qua phương thức println để xuất chuỗi ra màn hình

“Luồng dữ liệu” - Stream

- Stream là 1 khái niệm được mô tả như một dãy dữ liệu tuần tự, liên tiếp được truyền từ “bên ngoài vào” (Input) cho chương trình, hoặc được truyền từ chương trình “ra bên ngoài” (Output). “Bên ngoài” ở đây chính là các thiết bị được kết nối với máy tính, VD: *Bàn phím, Máy in, Màn hình, Ổ đĩa, ... hay 1 máy tính khác đang kết nối với máy tính hiện hành để truyền dữ liệu.*
- Stream được chia làm 2 khái niệm cụ thể
 - InputStream**: Mô tả luồng dữ liệu được đọc từ 1 nguồn nào đó vào trong chương trình
 - OutputStream**: dữ liệu được ghi, hay chuyển đến cho 1 thiết bị nào đó



Standard Streams

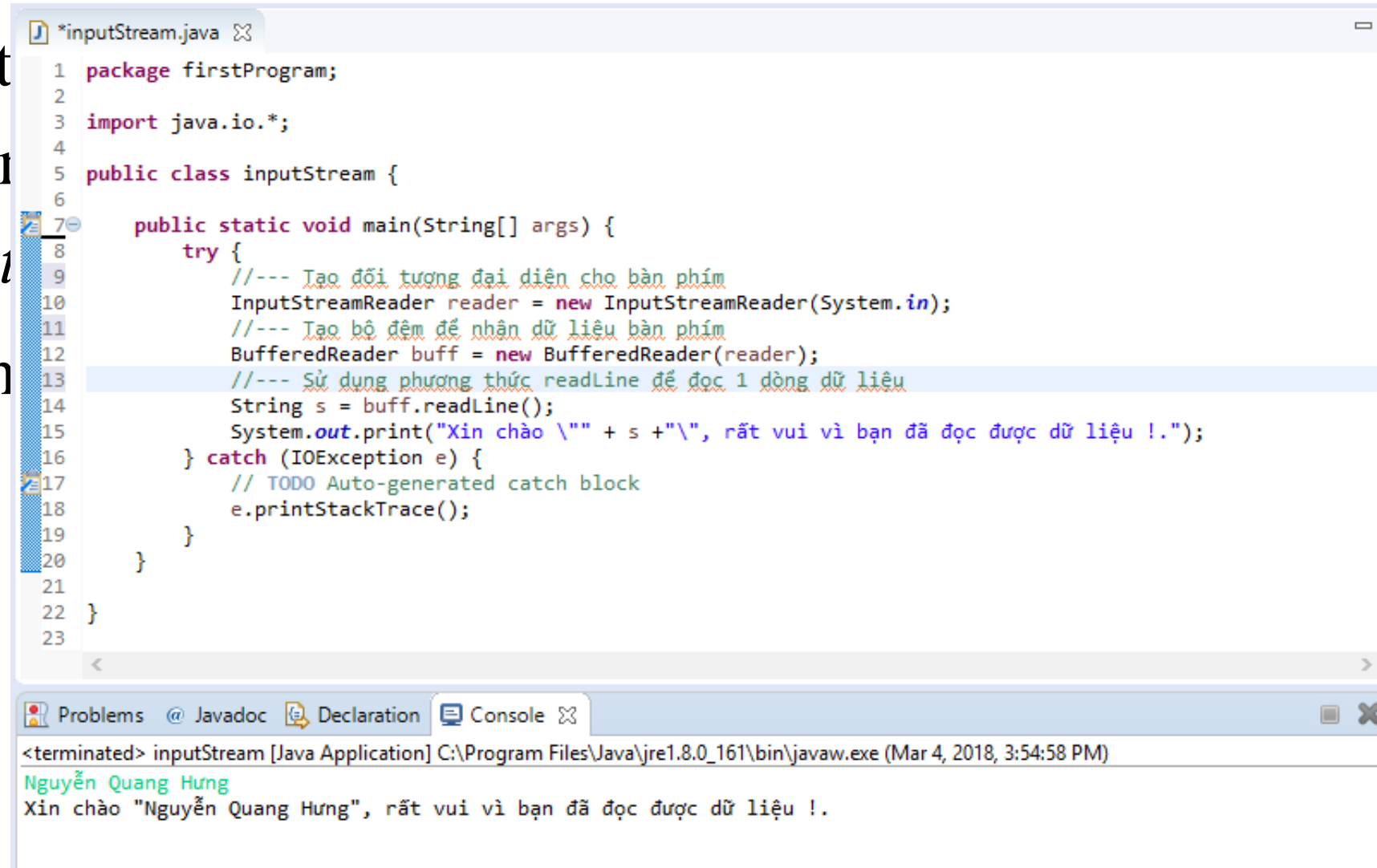
- **Standard Input** – Được sử dụng để nhập dữ liệu cho chương trình và mặc nhiên, bàn phím được xem là “thiết bị nhập chuẩn”. Java cung cấp đối tượng **System.in** đại diện cho *InputStream*.
- **Standard Output** – Được sử dụng để xuất dữ liệu của chương trình, mặc nhiên, màn hình máy tính được xác định là thiết bị xuất chuẩn và được xác định bởi đối tượng **System.out**.
- **Standard Error** – Mô tả cho luồng thông báo lỗi của Java khi chương trình thi hành, đối tượng **System.err** sẽ nhận các thông báo lỗi của chương trình.

Làm sao để nhập dữ liệu từ bàn phím ?

- Thêm vào gói `java.io.*` là thư viện phục vụ cho việc xuất nhập từ các luồng dữ liệu, serialization và tập tin trong Java
- Sử dụng một đối tượng đại diện cho “*luồng nhập*” để đọc dữ liệu từ bàn phím
InputStreamReader reader = `new` **InputStreamReader**(System.in);
- Quá trình đọc dữ liệu cần đến “*bộ đệm* – Buffer”, do đó chúng ta phải khai báo như sau
BufferedReader buff = `new` **BufferedReader**(reader);
- Bây giờ, để nhận dữ liệu, bạn chỉ cần sử dụng phương thức `read()` hoặc `readline()` của đối tượng buff đã khai báo ở trên
- `String s = buff.readline();`

Ví dụ

- Hãy viết chương trình nhập tên và họ của bạn. Chương trình sẽ in ra màn hình lời chào. Xin chào “*Họ và tên*”
- Lưu ý: “Họ và tên



```
*inputStream.java
1 package firstProgram;
2
3 import java.io.*;
4
5 public class inputStream {
6
7     public static void main(String[] args) {
8         try {
9             //--- Tạo đối tượng đại diện cho bàn phím
10            InputStreamReader reader = new InputStreamReader(System.in);
11            //--- Tạo bộ đệm để nhận dữ liệu bàn phím
12            BufferedReader buff = new BufferedReader(reader);
13            //--- Sử dụng phương thức readLine để đọc 1 dòng dữ liệu
14            String s = buff.readLine();
15            System.out.print("Xin chào \"" + s + "\", rất vui vì bạn đã đọc được dữ liệu !.");
16        } catch (IOException e) {
17            // TODO Auto-generated catch block
18            e.printStackTrace();
19        }
20    }
21
22 }
23
```

Problems @ Javadoc Declaration Console

<terminated> inputStream [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (Mar 4, 2018, 3:54:58 PM)

Nguyễn Quang Hưng
Xin chào "Nguyễn Quang Hưng", rất vui vì bạn đã đọc được dữ liệu !.

Kiểu dữ liệu & phép toán

Kiểu dữ liệu

- Java cung cấp 8 kiểu dữ liệu nguyên thủy “primitive data”, hay còn gọi là built-in data type. Bao gồm:
 - 4 kiểu dùng cho chứa số nguyên: byte, short, int, long
 - 2 kiểu chứa số thực, có dấu chấm động: float, double
 - 1 kiểu chứa dữ liệu logic : boolean
 - 1 kiểu chứa ký tự: char

Primitive data types

Type	Size (bits)	Range
byte	8	-128 to 127
short	16	-32,768 to 32,767
int	32	-2,147,483,648 to 2,147,483,647
long	64	-9,223,372,036,854,775,808L to 9,223,372,036,854,775,807L
float	32	+/- 3.4E+38F (6-7 significant digits)
double	64	+/- 1.8E+308 (15 significant digits)
char	16	ISO Unicode character set
boolean	1	true or false

Các phép toán trên dữ liệu

Phép toán số học

Operator	Description	Example
+ (Addition)	Adds values on either side of the operator.	$A + B$ will give 30
- (Subtraction)	Subtracts right-hand operand from left-hand operand.	$A - B$ will give -10
* (Multiplication)	Multiplies values on either side of the operator.	$A * B$ will give 200
/ (Division)	Divides left-hand operand by right-hand operand.	B / A will give 2
% (Modulus)	Divides left-hand operand by right-hand operand and returns remainder.	$B \% A$ will give 0
++ (Increment)	Increases the value of operand by 1.	$B++$ gives 21
-- (Decrement)	Decreases the value of operand by 1.	$B--$ gives 19

Các phép toán trên dữ liệu

Các phép toán quan hệ, dùng khi so sánh trong các biểu thức logic

Operator	Description	Example
<code>==</code> (equal to)	Checks if the values of two operands are equal or not, if yes then condition becomes true.	<code>(A == B)</code> is not true.
<code>!=</code> (not equal to)	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	<code>(A != B)</code> is true.
<code>></code> (greater than)	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	<code>(A > B)</code> is not true.
<code><</code> (less than)	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	<code>(A < B)</code> is true.
<code>>=</code> (greater than or equal to)	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	<code>(A >= B)</code> is not true.
<code><=</code> (less than or equal to)	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	<code>(A <= B)</code> is true.

Các phép toán trên dữ liệu

Các phép toán logic

Operator	Description	Example
&& (logical and)	Called Logical AND operator. If both the operands are non-zero, then the condition becomes true.	(A && B) is false
(logical or)	Called Logical OR Operator. If any of the two operands are non-zero, then the condition becomes true.	(A B) is true
! (logical not)	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) is true

Các phép toán trên dữ liệu

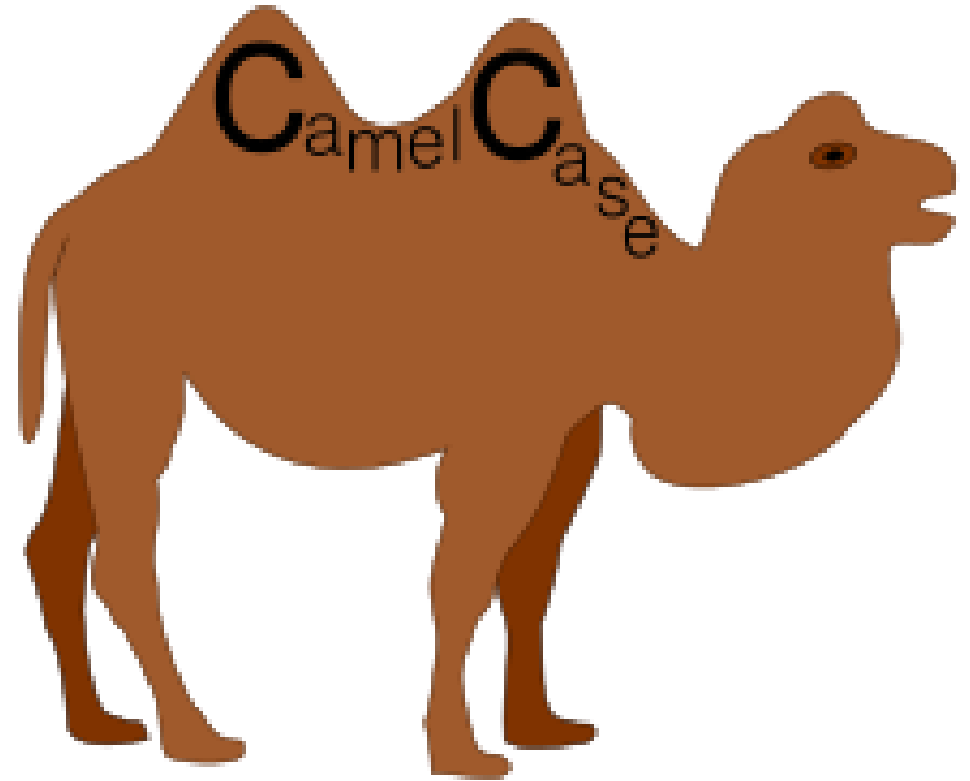
■ Phép gán

Operator	Description	Example
=	Simple assignment operator. Assigns values from right side operands to left side operand.	$C = A + B$ will assign value of $A + B$ into C
+=	Add AND assignment operator. It adds right operand to the left operand and assign the result to left operand.	$C += A$ is equivalent to $C = C + A$
-=	Subtract AND assignment operator. It subtracts right operand from the left operand and assign the result to left operand.	$C -= A$ is equivalent to $C = C - A$
*=	Multiply AND assignment operator. It multiplies right operand with the left operand and assign the result to left operand.	$C *= A$ is equivalent to $C = C * A$
/=	Divide AND assignment operator. It divides left operand with the right operand and assign the result to left operand.	$C /= A$ is equivalent to $C = C / A$
%=	Modulus AND assignment operator. It takes modulus using two operands and assign the result to left operand.	$C \% = A$ is equivalent to $C = C \% A$

Quy ước đặt tên

Camel case

- Tuân theo nguyên tắc về đặt tên cho Identifier trong lập trình
- Bắt đầu các từ phải là chữ viết hoa, các ký tự còn lại phải là chữ thường, duy nhất từ đầu tiên phải viết chữ thường toàn bộ. VD iPhone, eBay, fedEx, chieuDai, tyGia, ...



Quy ước đặt tên

Package name – tên gói đặt tên bằng các kí tự chữ thường

Class name – tên lớp đặt tên theo qui tắc UpperCamelCase

Interface name – tên Interface nên là một tính từ và bắt đầu bằng kí tự chữ hoa. VD: Serializable, Clonable, Iterable,...

Method name – tên phương thức nên là một động từ và tuân theo qui tắc lowerCamelCase

Variable name – tên biến đặt theo qui tắc lowerCamelCase

Constant Variable – tên hằng số viết hoa toàn bộ và phân cách nhau bằng dấu gạch dưới (underscore)

Ví dụ minh họa

- 1/- Viết chương trình cho phép nhập vào số lượng, đơn giá để tính trị giá của một mặt hàng
- 2/- Viết chương trình cho phép nhập chiều dài, chiều rộng để tính diện tích, chu vi hình chữ nhật
- 3/- Viết chương trình cho phép nhập vào bán kính, để tính chu vi, diện tích của một hình tròn

Tính diện tích

- Sử dụng **InputStream** để nhận dữ liệu
- Sau khi nhập, dữ liệu được xử lý bằng **Class** với phương thức **parse**
- Sử dụng từ khóa **final** để khai báo hằng số
- Nhớ sử dụng kỹ thuật **try-catch** để xử lý ngoại lệ

hinhTron.java

```
1 package firstProgram;
2
3 import java.io.*;
4
5 public class hinhTron {
6
7     public static void main(String[] args) {
8         try {
9             //--- Tạo đối tượng đại diện cho bàn phím
10            InputStreamReader reader = new InputStreamReader(System.in);
11            //--- Tạo bộ đệm để nhận dữ liệu bàn phím
12            BufferedReader buff = new BufferedReader(reader);
13            System.out.print("Hãy nhập bán kính: ");
14            //--- Sử dụng phương thức readLine để nhận dữ liệu từ bàn phím và s
15            String s = buff.readLine();
16            //--- Dựa vào Wrapper class để chuyển dữ liệu từ String thành int
17            int r = Integer.parseInt(s);
18            //--- Khai báo hằng số pi kiểu double thông qua từ khóa final
19            final double pi = 3.14;
20            //--- Tính diện tích, chu vi
21            double dt = (double)pi*r*r;
22            double cv = (double)pi*2*r;
23            System.out.println("Diện tích: " + dt);
24            System.out.println("Chu vi: " + cv);
25        } catch (IOException e) {
26            // TODO Auto-generated catch block
27            e.printStackTrace();
28        }
29    }
30 }
```

Problems @ Javadoc Declaration Console

<terminated> hinhTron [Java Application] C:\Program Files\Java\jre1.8.0_161\bin\javaw.exe (Mar 4, 2018, 4:30:52 PM)

Hãy nhập bán kính: 6
Diện tích: 113.03999999999999
Chu vi: 37.68

Các cấu trúc điều khiển

Cấu trúc rẽ nhánh

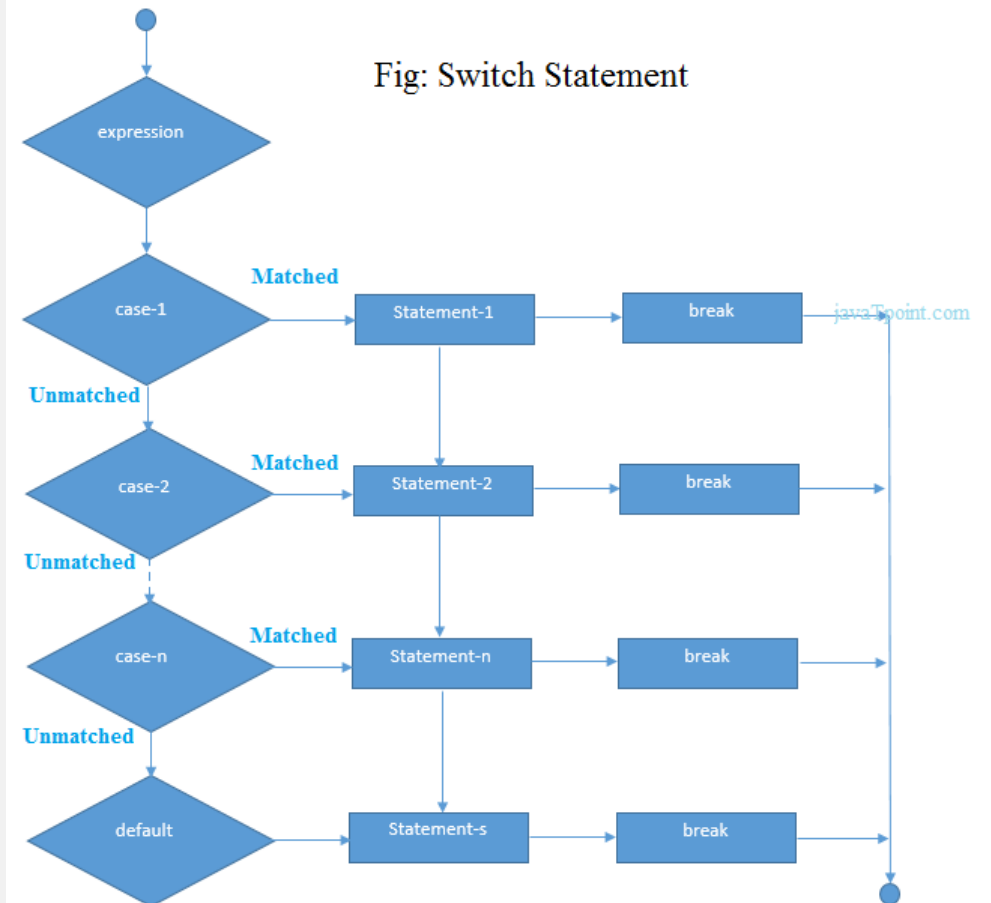
if () { ... }

General form	Example
if (boolean) statement;	if (x == y) a = 20; if (x == z) { b = 10; c = 20; }
if (boolean) statement1; else statement2;	if (x == y) { a = 10; b = 20; } else x = y;
if (boolean1) statement1; ... else if (booleanN) statementN; else statement;	if (x > 60) y = 20; else if (x < 30) { z += y; y = 25; } else y = 40;

Cấu trúc rẽ nhánh

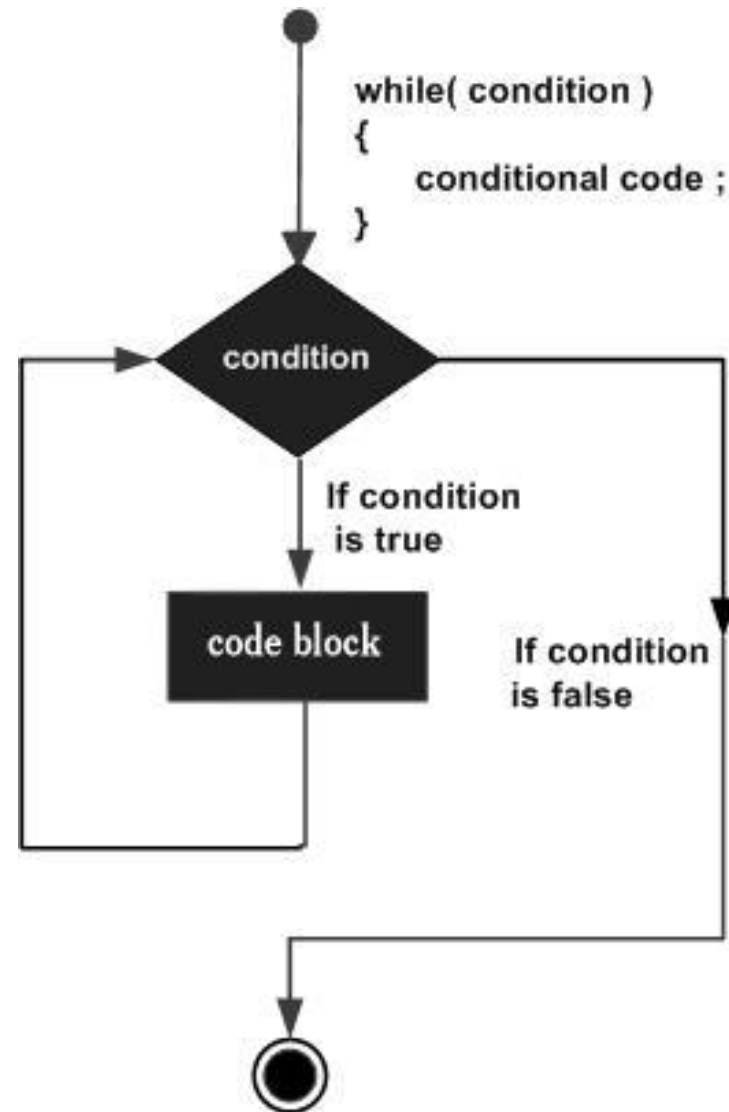
switch () case

```
switch(expression) {  
    case value :  
        // Statements  
        break; // optional  
  
    case value :  
        // Statements  
        break; // optional  
  
    // You can have any number of case statements.  
    default : // Optional  
        // Statements  
}
```



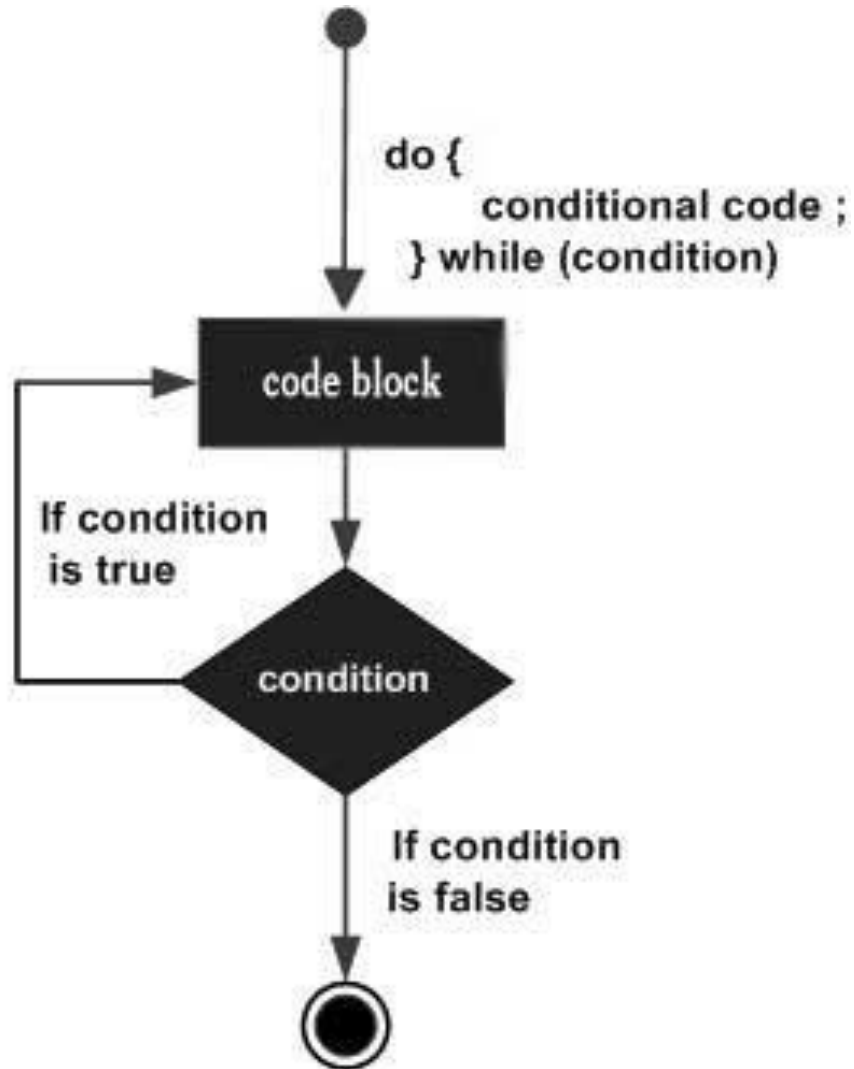
Cấu trúc lặp

while ()



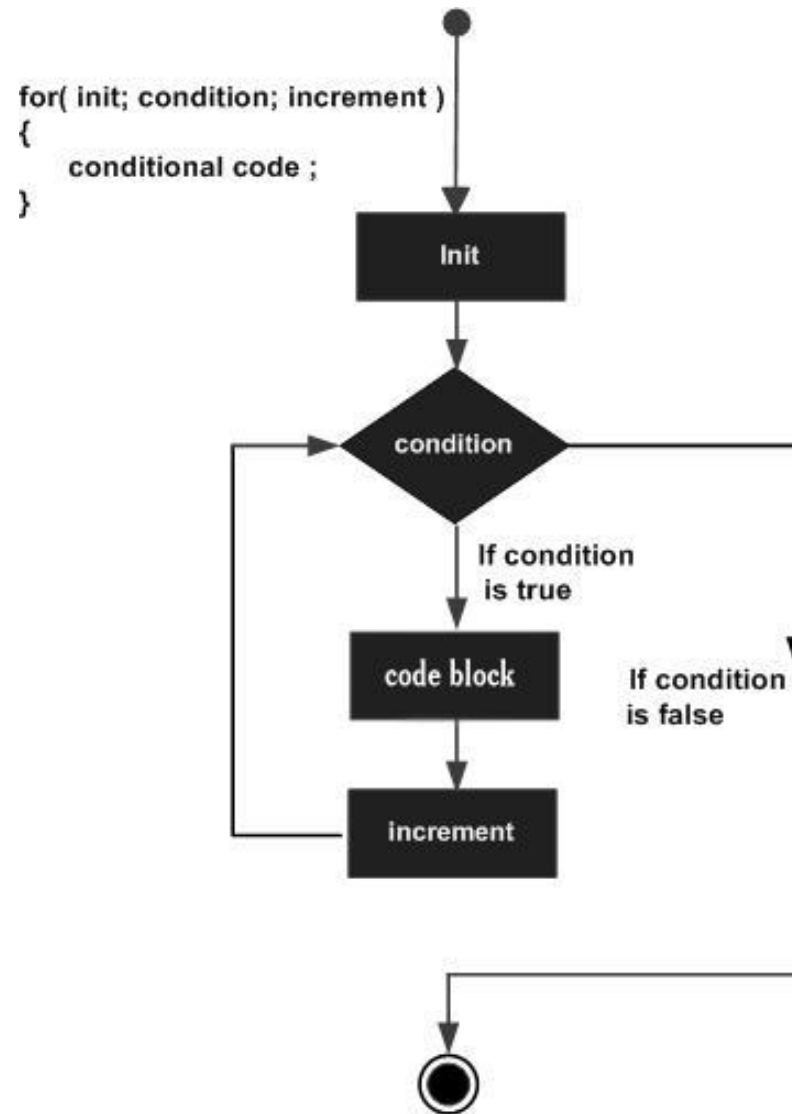
Cấu trúc lặp

do ... while ()



Cấu trúc lặp

for ()



Nhớ gì ?!!!

- Cấu trúc một chương trình đơn giản trong Java
- Khái niệm luồng xuất, nhập chuẩn
- Biết cách nhập dữ liệu từ bàn phím, tính toán & xuất ra màn hình
- Các kiểu dữ liệu nguyên thủy trong Java
- Các phép toán trên dữ liệu (số học, quan hệ, logic, gán)
- Quy ước đặt tên
- Khai báo hằng trong Java, phải dùng từ khóa **final**
- Các cấu trúc điều khiển

Tài liệu tham khảo

- Y. Daniel Lang, “**Introduction to Java Programming Comprehension Version**” 10th Edition.
- Jose M. Garrido, “**Object-Oriented Programming: From Problem Solving to Java**”
- Paul Deitel, Harvey Deitel, “**Java : How to program**”, 9th edition, 2012
- Oracle, “**The Java™ Tutorials**”,
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html>, 6:20PM,
18/01/2018
- Java tutorial, <https://howtodoinjava.com/java/basics/>, 15:00PM, 20/02/2020