Fpt University

FPT POLYTECHNIC

cdio™

LẬP TRÌNH JAVA 1

BÀI 2: KIỂU, TOÁN TỬ, LỆNH IF, SWITCH

PHẦN 1

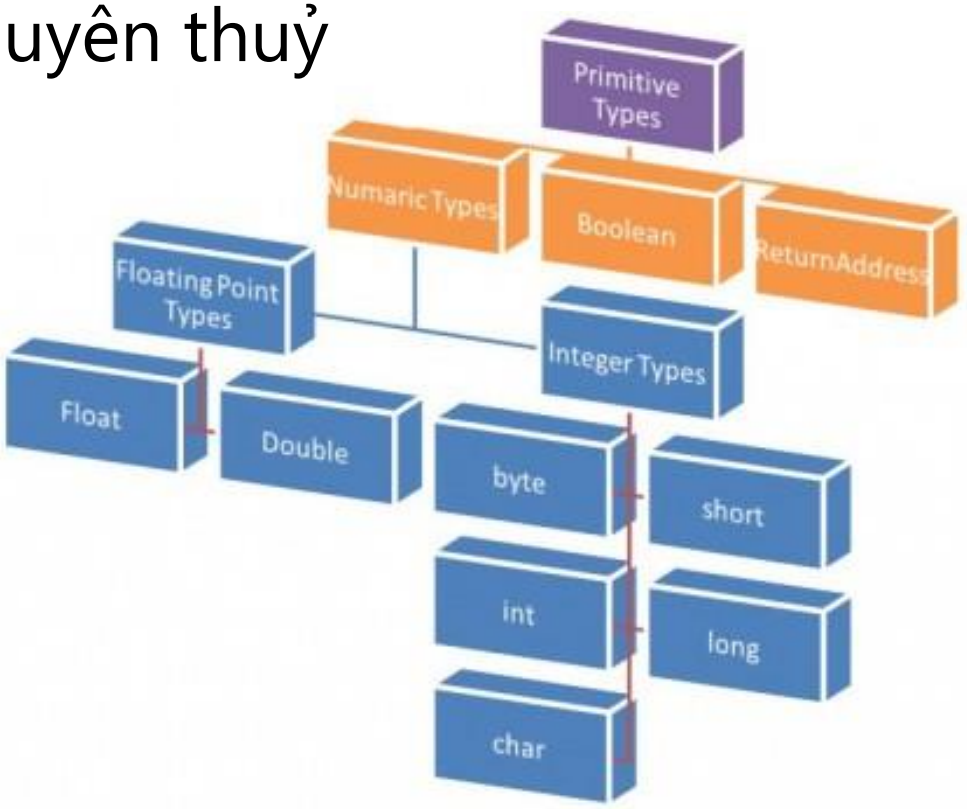
www.poly.edu.vn

Mục tiêu

- Kết thúc bài học này bạn có khả năng
  - Hiểu rõ và sử dụng kiểu nguyên thủy, lớp bao
  - Chuyển đổi chuỗi sang kiểu nguyên thủy
  - Sử dụng lệnh try...catch để bắt lỗi chuyển kiểu
  - Hiểu và sử dụng toán tử, xây dựng biểu thức
  - Sử dụng lệnh if
  - Sử dụng lệnh switch case
  - Biết cách tổ chức một chương trình

KIỂU DỮ LIỆU NGUYÊN THỦY

- Kiểu dữ liệu nguyên thủy là kiểu được giữ lại từ ngôn ngữ C (ngôn ngữ gốc của Java)
- Có 8 kiểu dữ liệu nguyên thuỷ
- Ví dụ
  - int a = 8;
  - double b;



KIỂU NGUYÊN THỦY

Kiểu	Mặc định	Bit	Khả năng lưu trữ	
			Giá trị nhỏ nhất	Giá trị lớn nhất
byte	0	8	-2^7	$+2^7-1$
short	0	16	-2^{15}	$+2^{15}-1$
int	0	32	-2^{31}	$+2^{31}-1$
long	0L	64	-2^{63}	$+2^{63}-1$
float	0.0F	32	$-3.40292347x10^{38}$	$+3.40292347x10^{38}$
double	0.0	64	$-1.79769313486231570x10^{308}$	$+1.79769313486231570x10^{308}$
char	'\u0000'	16	'\u0000'	'\uFFFF'
boolean	false	1	false	true

Giá trị mặc định là giá trị sẽ được gán cho biến khi khai báo không khởi đầu giá trị cho biến

GIÁ TRỊ HẰNG (LITERAL)

- Giá trị hằng là dữ liệu có kiểu là một trong các kiểu nguyên thuỷ

Kiểu int

```
int i = 3;
```

Kiểu long

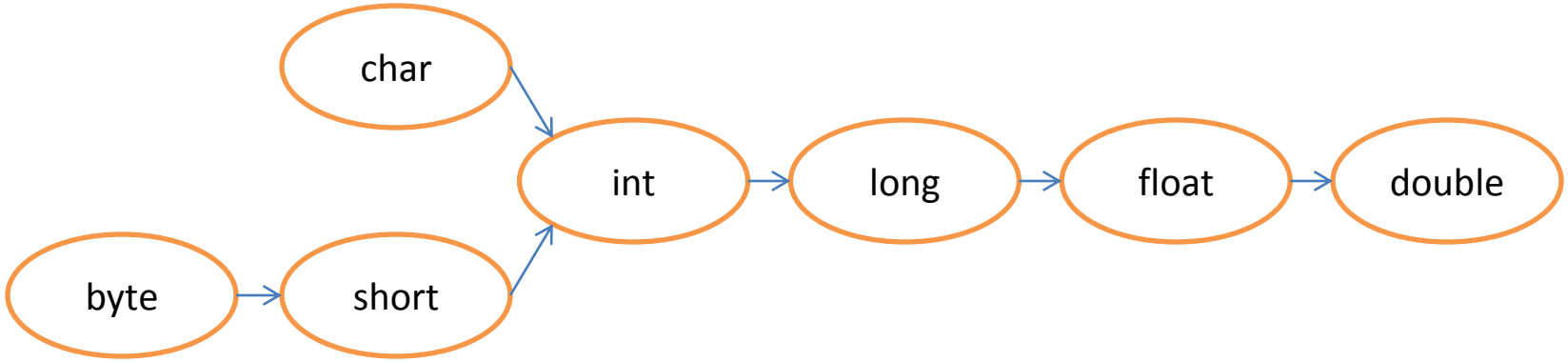
```
long l = 12L;
```

Kiểu float

```
float = 10.19F;
```

QUI LUẬT ÉP KIỂU

- Đối với kiểu nguyên thủy, ép kiểu tự động xảy ra theo chiều mũi tên



- Ví dụ

```
int a = 5;
double b = 9.4;
b = a;          //ép kiểu tự động
a = (int)b; //ép kiểu tường minh phần thập phân sẽ bị bỏ
```

CHUYỂN 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)

SỬ DỤNG TRY...CATCH ĐỂ KIỂM LỖI

- Xét trường hợp

  int a = **scanner.nextInt()**;

  hoặc

  int a = **Integer.parseInt(s)**;

- Điều gì sẽ xảy ra khi người dùng **nhập không phải số** hoặc chuỗi **s không phải là chuỗi chứa số**

- Hãy sử dụng lệnh **try...catch** để kiểm soát các lỗi trên

```
try{
        int a = scanner.nextInt();
        System.out.println("Bạn đã nhập đúng");
}
catch (Exception ex){
        System.out.println("Vui lòng nhập số !");
}
```

LỚP BAO KIỂU NGUYÊN THỦY (WRAPPER)

- Tương ứng với mỗi kiểu nguyên thủy Java định nghĩa một lớp bao để bao giá trị của kiểu nguyên thủy tương ứng gọi là lớp bao kiểu nguyên thủy
- Rất nhiều hàm trong Java chỉ làm việc với đối tượng mà không làm việc với kiểu nguyên thủy

Nguyên Thủy		Lớp bao
byte	⇔	**B**yte
short	⇔	**S**hort
int	⇔	**Integer**
long	⇔	**L**ong
float	⇔	**F**loat
double	⇔	**D**ouble
char	⇔	**Character**
boolean	⇔	**B**oolean

BAO (BOXING)/MỞ BAO(UNBOXING)

- Boxing là việc tạo đối tượng từ lớp bao để bọc giá trị nguyên thủy.
  - Có 3 cách để bao giá trị nguyên thủy sau
    - Integer a = Integer.**valueOf**(5) **// bao tường minh**
    - Integer a = **new Integer(5) // bao tường minh**
    - Integer a = **5 // bao ngầm định**
- Unboxing là việc mở lấy giá trị nguyên thủy từ đối tượng của lớp bao
  - Có 2 cách mở bao để lấy giá trị nguyên thủy sau
    - int b = a.**intValue**() **// mở bao tường minh**
    - int b = **a**; **// mở bao ngầm định**

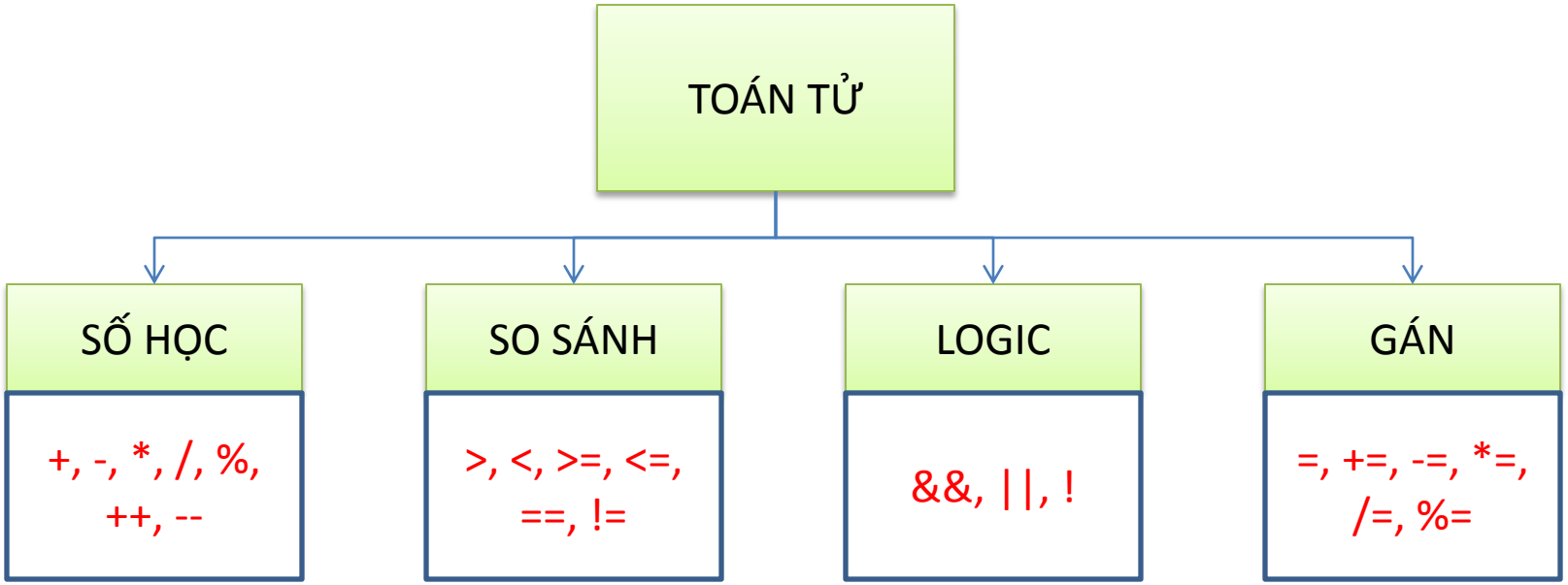
BOXING/UNBOXING

Boxing	Unboxing	Ví dụ

Byte.valueOf(byte)	<<Byte>>.byteValue()	long a = 5L;
		Long b = Long.valueOf(a);
		long c = b.longValue();

Short.valueOf(short)	<<Short>>.shortValue()	
Integer.valueOf(int)	<<Integer>>.intValue()	
Long.valueOf(long)	<<Long>>.longValue()	
Float.valueOf(float)	<<Float>>.floatValue()	
Double.valueOf(double)	<<Double>>.doubleValue()	
Boolean.valueOf(boolean)	<<Boolean>>.booleanValue()	

TOÁN TỬ & BIỂU THỨC



Biểu thức là sự kết hợp giữa toán tử và toán hạng. Kết quả của biểu thức là một giá trị.

Giá trị của các biểu thức sau?

```
int x = 11 % 4;
boolean a = 9 < 2 && true || 4 > 3;
```

TOÁN TỬ SỐ HỌC

- Toán tử số học là các phép toán thao tác trên các số nguyên và số thực

+	Tính tổng của 2 số
-	Tính hiệu của 2 số
*	Tính tích của 2 số
/	Tích thương của 2 số
%	Thực hiện chia có dư của 2 số
++	Tăng giá trị của biến lên 1 đơn vị
--	Giảm giá trị của biến xuống 1 đơn vị

TOÁN TỬ SO SÁNH

- Toán tử so sánh là các phép toán so sánh hai toán hạng

==	So sánh bằng
>	So sánh lớn hơn
>=	So sánh lớn hơn hoặc bằng
<	So sánh nhỏ hơn
<=	So sánh nhỏ hơn hoặc bằng
!=	So sánh khác

TOÁN TỬ LOGIC

- Toán tử logic là các phép toán thao tác trên các toán hạng logic

&&	Trả về giá trị true khi tất cả biểu thức tham gia biểu thức có giá trị true
||	Trả về giá trị true khi có 1 biểu thức tham gia biểu thức có giá trị là true
!	Lấy giá trị phủ định của biểu thức

TOÁN TỬ ĐIỀU KIỆN

- Toán tử điều kiện là toán tử 3 ngôi duy nhất trong ngôn ngữ Java
- Cú pháp:

  <điều kiện> ? <giá trị đúng> : <giá trị sai>
- Diễn giải:
  - Nếu biểu thức **<điều kiện>** có giá trị là true thì kết quả của biểu thức là **<giá trị đúng>**, ngược lại là **<giá trị sai>**
- Ví dụ: tìm số lớn nhất của 2 số a và b

```
int a = 1, b = 9;
int max = a > b ? a : b;
```

Tìm số lớn nhất trong 3 số a, b và c?

LỆNH IF

- Cú pháp

```
if(<<điều kiện>>)
{
    << Công việc >>
}
```

- Diễn giải:
  - Nếu **điều kiện** có giá trị true thì **công việc** được thực hiện

- Ví dụ:

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

- Diễn giải:
  - Đoạn mã trên không xuất gì ra màn hình cả vì biểu thức điều kiện **diem >= 5** có giá trị false


DEMO
Nhập số từ bàn phím.
Nếu số dương thì tính và xuất căn bậc 2
của số đó ra màn hình

LAB 2 BUỔI 1

- Lab 2 – bài 1
- Lab 2 – bài 2

Fpt University

FPT POLYTECHNIC

cdio™

LẬP TRÌNH JAVA 1

BÀI 2: KIỂU, TOÁN TỬ, LỆNH IF, SWITCH

PHẦN 2

www.poly.edu.vn

LỆNH IF...ELSE

- Cú pháp

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

- Diễn giải
  - Nếu **điều kiện** có giá trị true thì **công việc 1** được thực hiện, ngược lại **công việc 2** được thực hiện

LỆNH IF...ELSE

- Ví dụ

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

- Diễn giải:
  - Đoạn mã trên xuất chữ "Rớt" ra màn hình vì điều kiện **diem < 5** có giá trị là **true**.


DEMO
Nhập số từ bàn phím.
Nếu số dương thì tính và xuất căn bậc 2
của số đó ra màn hình, ngược lại thì
thông báo lỗi

NHIỀU LỆNH IF

- Cú pháp

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

- Diễn giải
  - Chương trình sẽ kiểm tra từ **điều kiện 1 đến N** nếu gặp **điều kiện i** đầu tiên có giá trị true thì sẽ thực hiện **công việc i**, ngược lại sẽ thực hiện **công việc N+1**

NHIỀU LỆNH IF

- Ví dụ

```
double delta = Math.pow(b, 2) – 4 * a * c;
if(delta < 0) {
    System.out.println("Vô nghiệm");
}
else if(delta == 0) {
    System.out.println("Nghiệm kép");
}
else {
    System.out.println("2 nghiệm");
}
```

- Diễn giải
  - Đoạn mã trên biện luận và giải phương trình bậc 2


DEMO
Tính thuế thu nhập mô tả slide sau

TÍNH THUẾ THU NHẬP

- Viết chương trình tính thuế thu nhập. Giả sử thu nhập gồm lương và thưởng
- Thuế thu nhập được tính như sau
  - Dưới 9 triệu: không đóng thuế
  - Từ 9 đến 15 triệu: thuế 10%
  - Từ 15 đến 30 triệu: 15%
  - Trên 30 triệu: 20%

LỆNH SWITCH

- Cú pháp

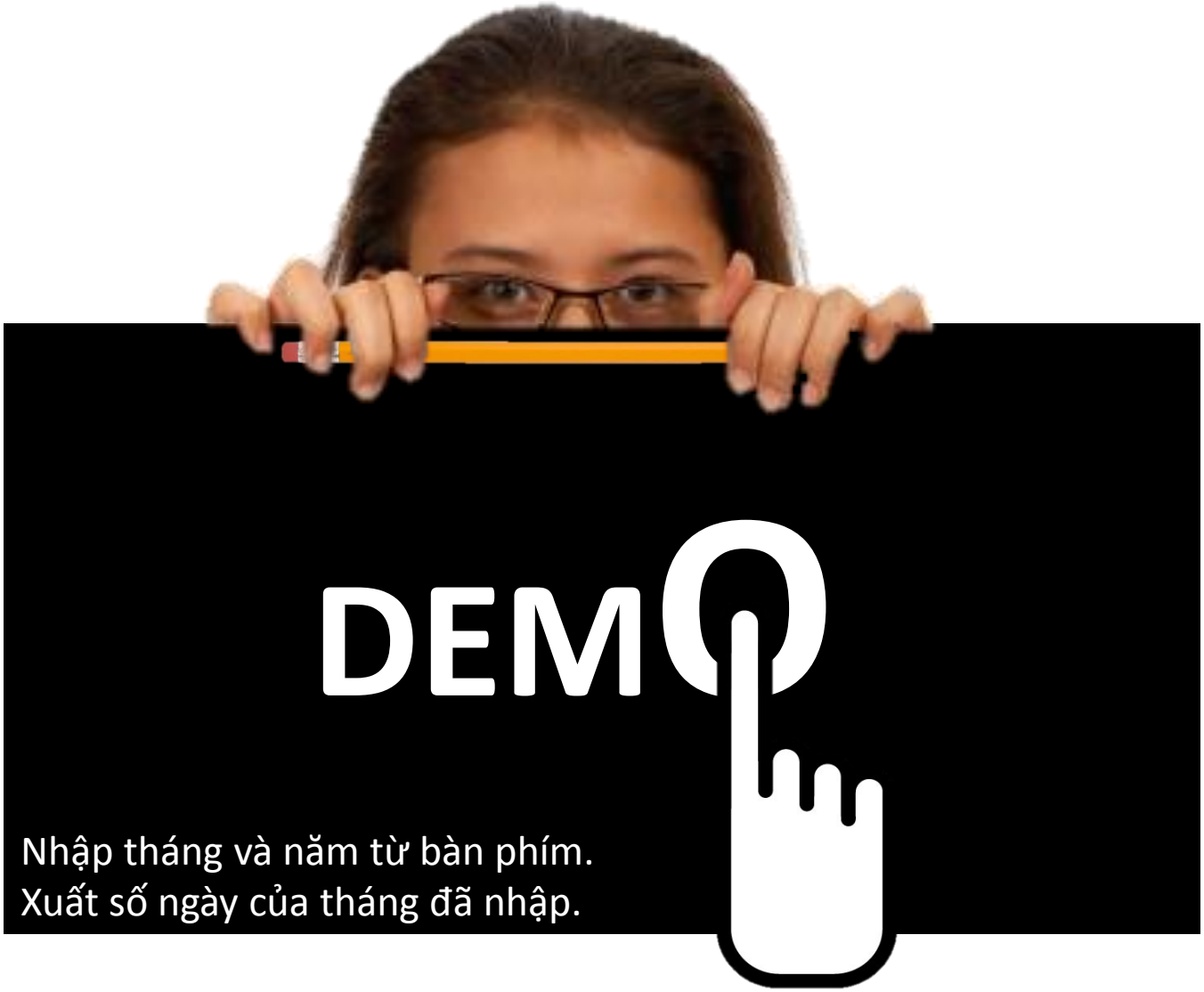
```
switch (<<biểu thức>>)
{
        case <<giá trị 1>>:
                // Công việc 1
                break;
        case <<giá trị 2>>:
                // Công việc 2
                break;
        ...
        default:
                // Công việc N+1
                break;
}
```

- Diễn giải
  - So sánh giá trị của biểu thức switch với giá trị của các 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 tiếp sau sẽ được thực hiện
  - default là tùy chọn

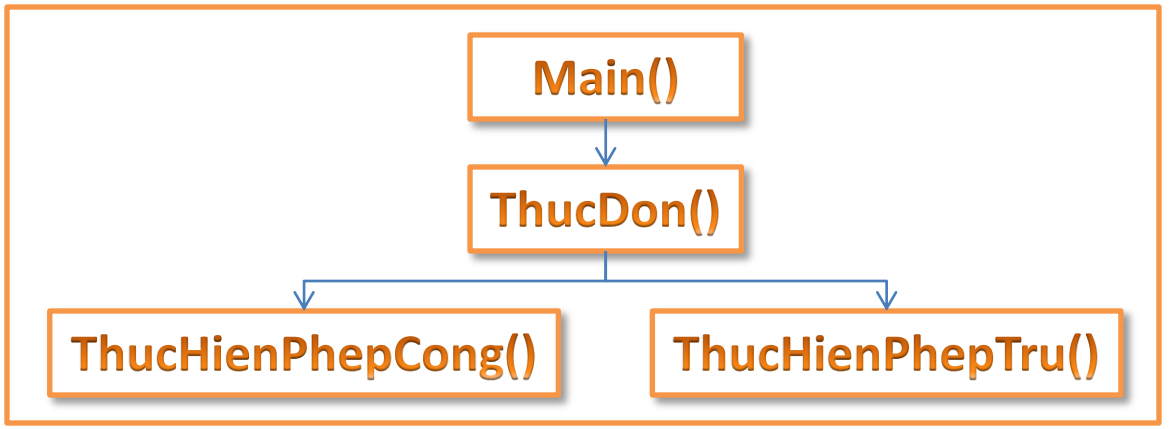
VÍ DỤ LỆNH SWITCH

```
double a = 5, b = 7, c = -1;
char op = '+';
switch(op){
    case '+':
        c = a + b;
        break;
    case '-':
        c = a - b;
        break;
    case 'x':
    case ':':
        System.out.println("Đang xây dựng");
        break;
    default:
        System.out.println("Vui lòng chọn +, -, x và :");
        break;
}
```

Không có **break**


DEMO
Nhập tháng và năm từ bàn phím.
Xuất số ngày của tháng đã nhập.

TỔ CHỨC CHƯƠNG TRÌNH




```
package com.fpoly;

import java.util.Scanner;

public class ChuongTrinh {
    public static void main(String[] args) {
        thucDon();
    }

    public static void thucDon() {

    public static void thucHienPhepCong() {

    public static void thucHienPhepTru() {
}
```

Hiển thị thực đơn chính của chương trình

THIẾT KẾ THỰC ĐƠN

```
System.out.println(">> MÁY TÍNH CÁ NHÂN <<");
System.out.println("+---------------------+");
System.out.println("| 1. Cộng             |");
System.out.println("| 2. Trừ              |");
System.out.println("| 3. Kết thúc         |");
System.out.println("+---------------------+");
System.out.println(" >> Chọn chức năng? ");

Scanner scanner = new Scanner(System.in);
int answer = scanner.nextInt();
if(answer == 1){
    thucHienPhepCong();
}
else if(answer == 2){
    thucHienPhepTru();
}
else if(answer == 3){
    System.exit(0);
}
```

Gọi phương thức thực hiện phép cộng

Gọi phương thức thực hiện phép trừ

Thoát ứng dụng


DEMO
Tổ chức chương trình trên
bằng cách đổi if…else sang switch…case

FPT POLYTECHNIC

TỔNG KẾT NỘI DUNG BÀI HỌC

- Kiểu nguyên thủy
- Qui luật ép kiểu nguyên thủy
- Lớp bao giá trị kiểu nguyên thủy
- Boxing/Unboxing
- Chuyển đổi kiểu dữ liệu
- Toán tử và biểu thức
- Lệnh if
- Lệnh switch case
- Tổ chức chương trình

LAB 2 BUỔI 2

- Lab 2 – bài 3
- Lab 2 – bài 4
- Lab 2 – bài 5 (giảng viên cho thêm)