1. **Flow control**
   1. **If.. else**

* Biểu thức điều kiện có giá trị là true hoặc false
* Thực hiện từng trường hợp 1
* Tối đa là 3 if else. Nếu 4 if else thì dùng switch case
* Cấu trúc như sau:
* **Base**

if (Condition 1){

// Task 1

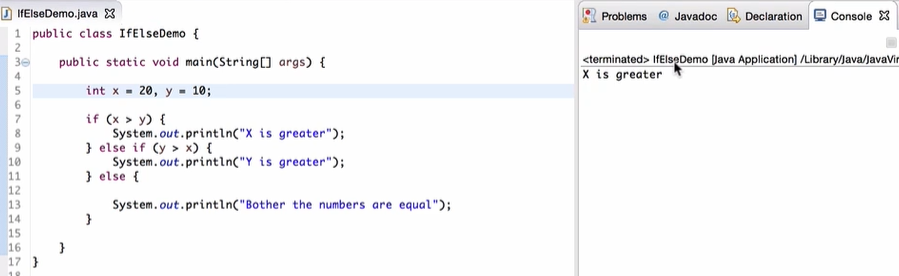
} else if (Condition 2){

// Task 2

} else{

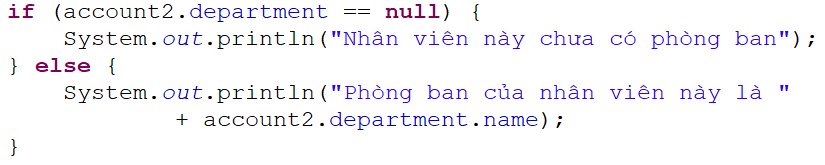
// Task 3

}



* **Ternary**

(Condition) ? (condition = true) : (condition = false)



* 1. **Switch…Case**
* Thực hiện tất cả case cùng lúc
* Nhanh hơn, tốn hiệu năng nhiều hơn
* Cấu trúc như sau:

switch (variable) {

case value1:

// task 1;

**break;**

case value2:

// task 2;

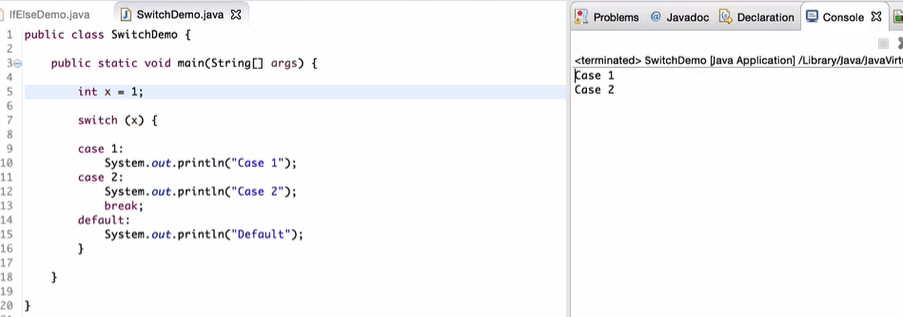
**break;**

default:

// default task;

**break;**

}



* 1. **For**
* Tối thiểu không chạy lần nào
* Cấu trúc như sau
* **Base**

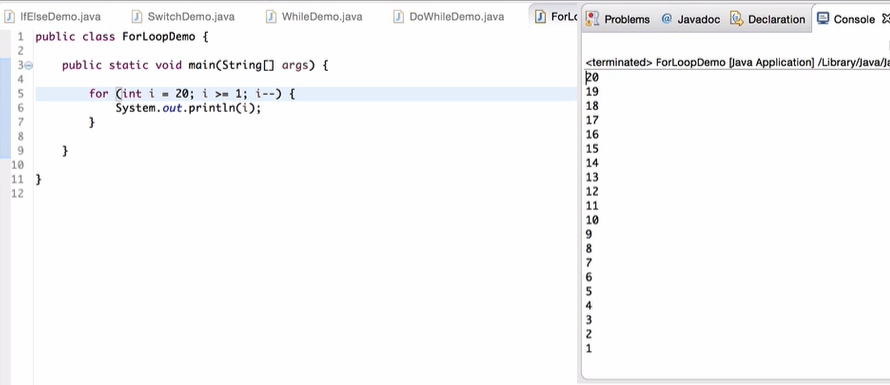
for( init value ; Condition ; step){

// statement

}

Ex:

|  |  |  |
| --- | --- | --- |
| for ( int i=1; i<= 100; i++){  // statement  } | for ( int i=1; i<= 100; ){  // statement  i++;  } | int i=1;  for ( ; i<= 100; ){  // statement  i++;  } |

****

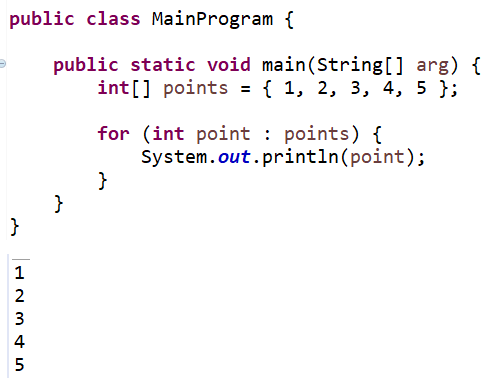
* **For each**

for(datatype : array ){

// statement

}

VD:



* 1. **While**
* Trong khi mệnh đề điều kiện còn đúng thì còn thực hiện công việc
* Lặp ít nhất 0 lần
* Cấu trúc như sau

while (Condition){

// statement

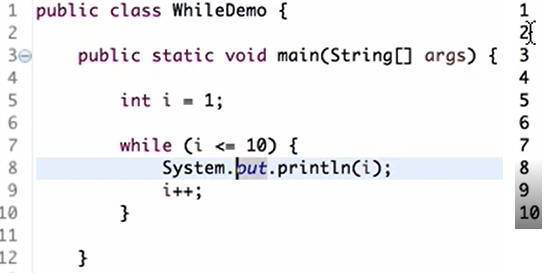
}

// tương tự như for

Giống for( ; biểuThứcGiớiHạn; ){

//statement

}

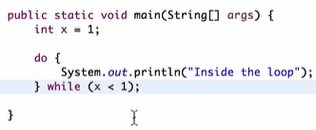


* 1. **Do while**
* Thực hiện trước rồi kiểm tra điều kiện
* Lặp ít nhất 1 lần
* Cấu trúc như sau

do {

// statement

} while (mệnh đề điều kiện);

****

* 1. **Continue**

**Continue** sẽ tiếp tục vòng lượt tiếp theo của vòng for (không chạy các câu lệnh ở dưới)

for( int i=0; i<=100; i++){

if( i%2 !=0){

continue;

}else{

break;

}

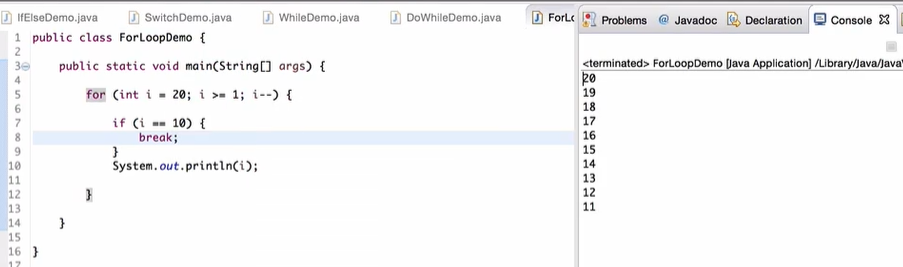
tong = tong+I;

}



* 1. **Break**

được sử dụng để thoát ra khỏi switch, loop hoặc block



* 1. **Return**

1. **Operator**
   1. **i++, ++i, --i, i--**

i++: tăng i lên 1 đơn vị **sau**

****

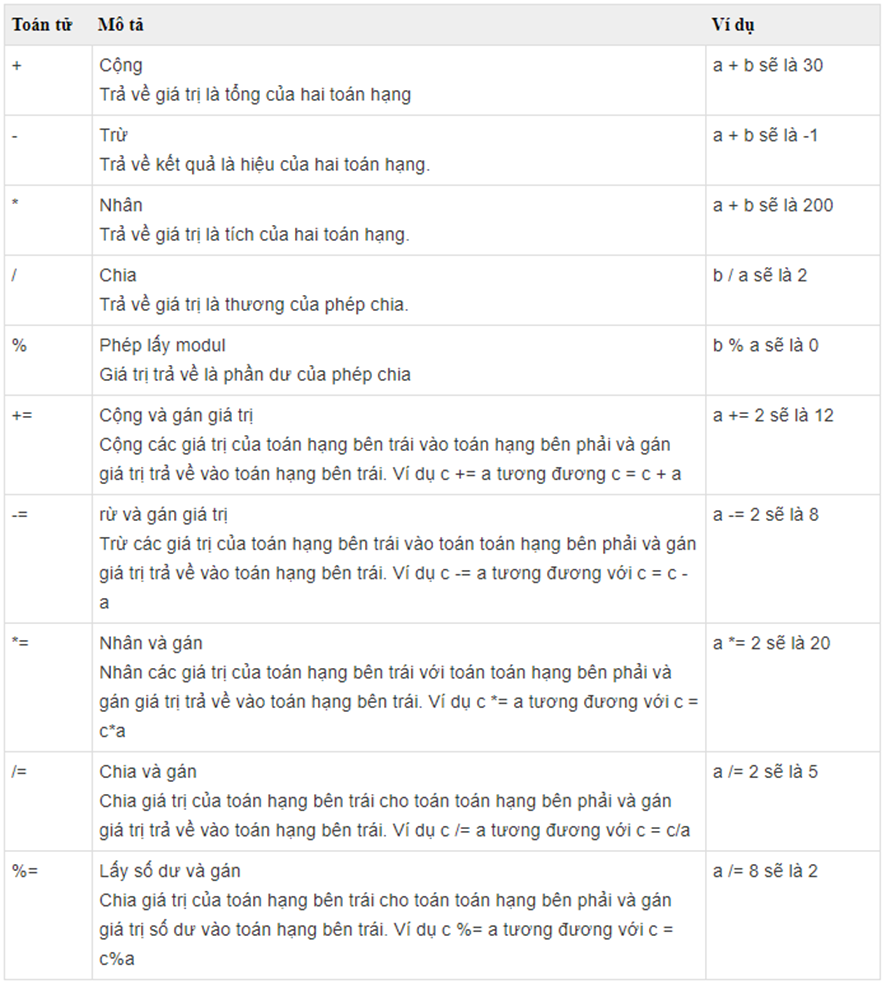
++i: tăng i lên 1 đơn vị **trước**

****

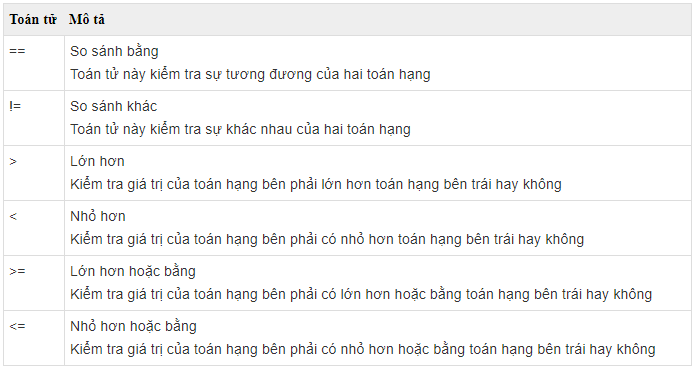
i--: trừ I đi 1 đơn vị **sau**

--i: trừ i đi 1 đơn vị **trước**

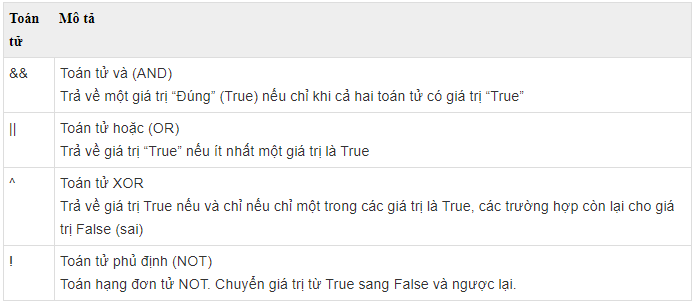
* 1. **Arithmetic**

****

* 1. **Relational**

****

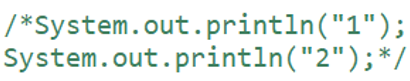
* 1. **Logic**

****

1. **Comment**
   1. **1 line**

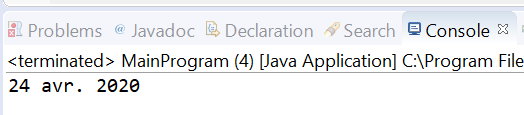
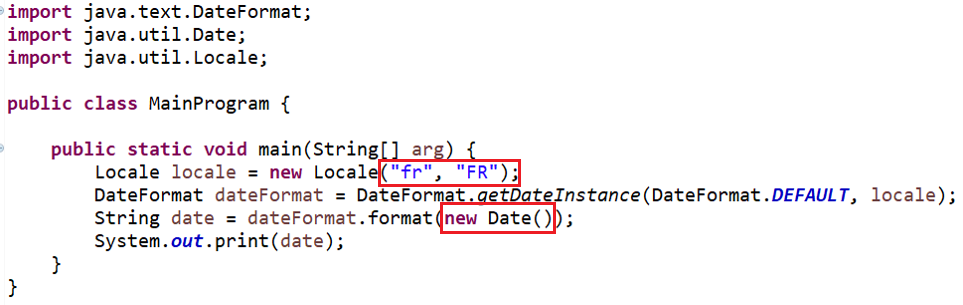
****

* 1. **Multiple line**

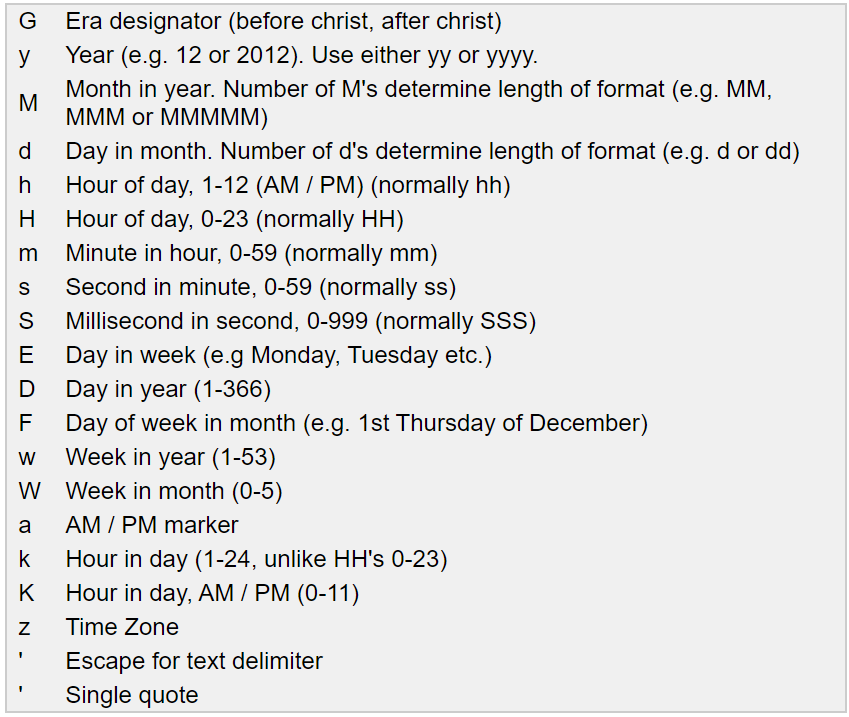
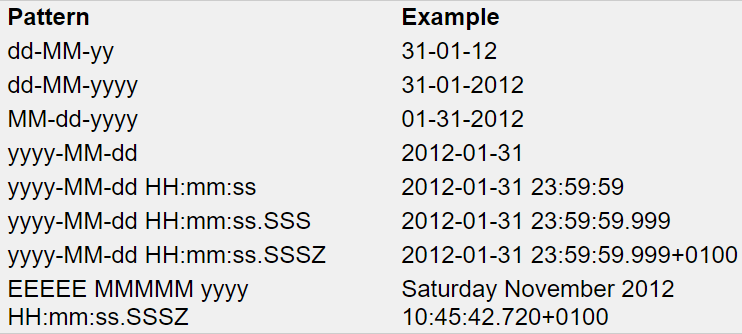


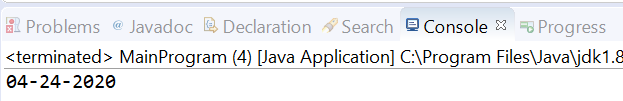
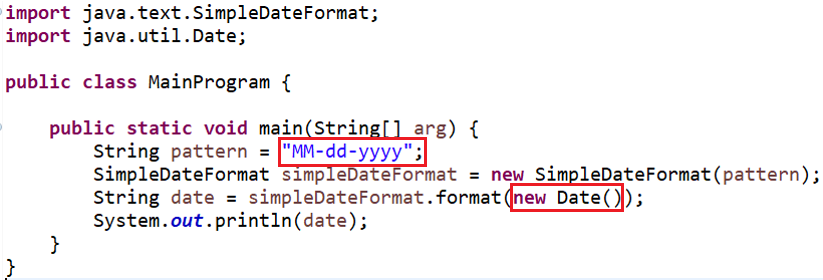
1. **DateFormat**
   1. **Country**

In ra date được định dạng theo country của từng nước



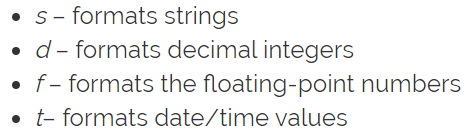
* 1. **Pattern**

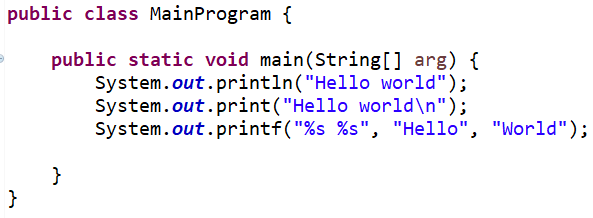
** **



* 1. **Print value**
* **System.out.println()**
* **System.out.print()**
* **System.out.printf()**

VD1:

****

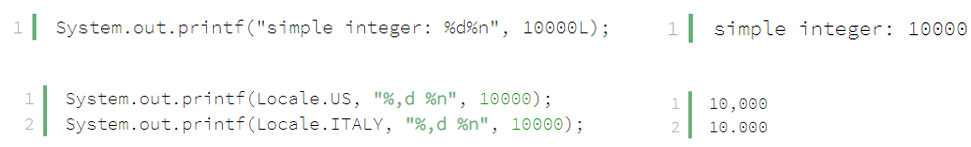


VD2: String

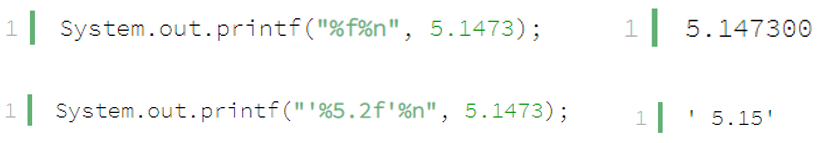
****

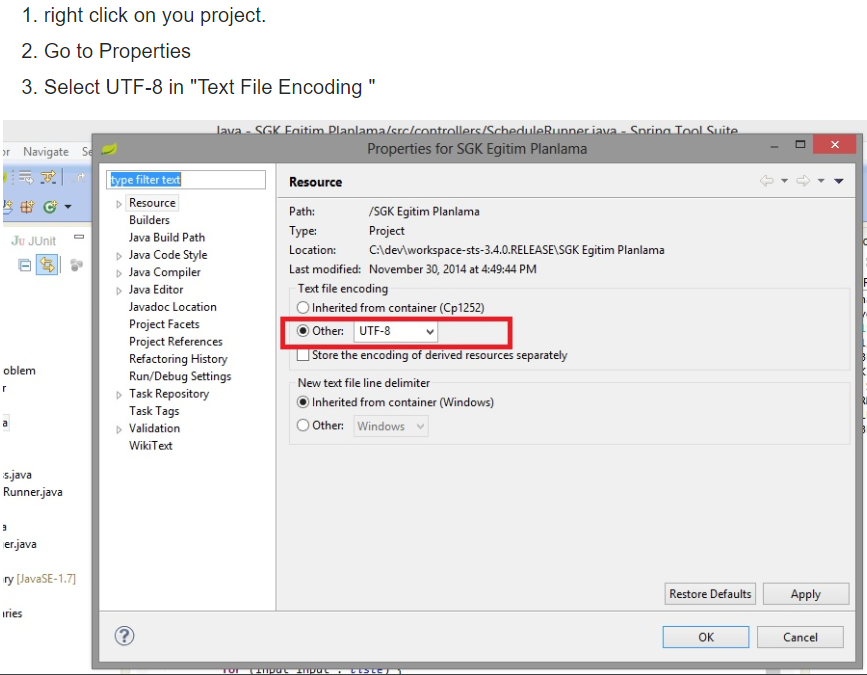


VD3: int



VD4: float





*utf-8*

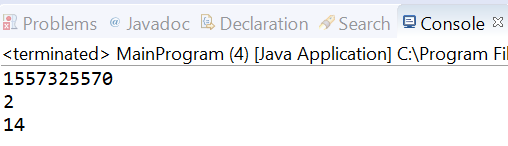
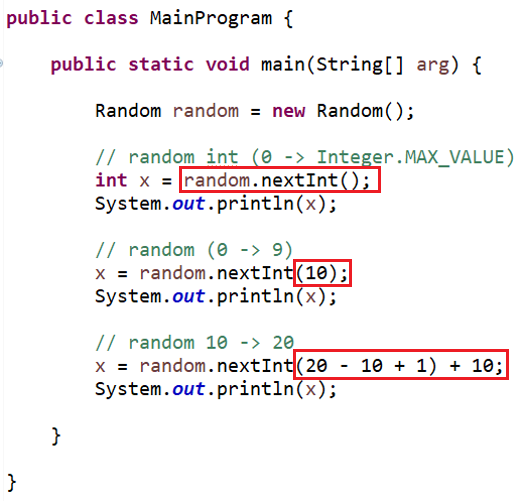
1. **Random number & Date**
   1. **Random int**

* **nextInt()**

Random số nguyên

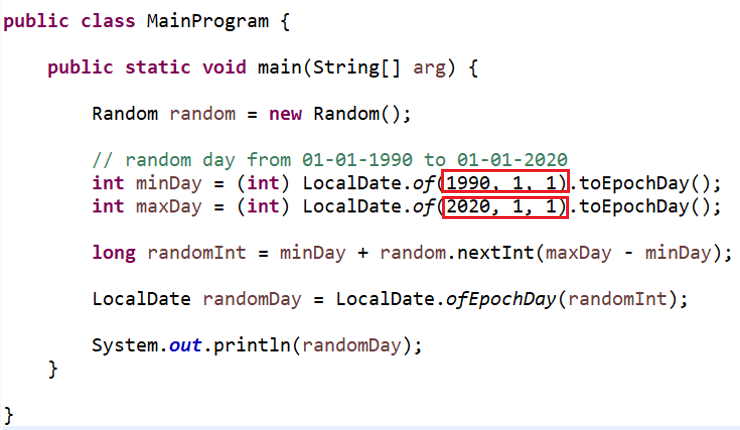
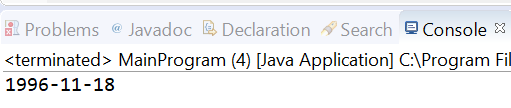
* **nextInt(n)**

Random số từ 0 🡪 n-1



*Random số từ 10 🡪 20*

* 1. **Random Date**

*Random 1 ngày từ 1-1-1990 🡪 1-1-2020*

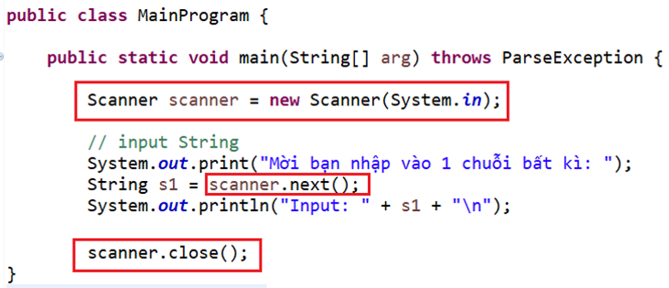
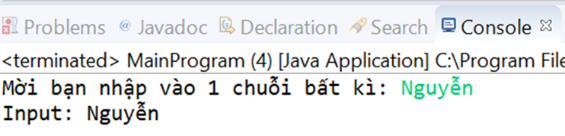
1. **Input from user**

Sử dụng scanner để lấy input từ người dùng

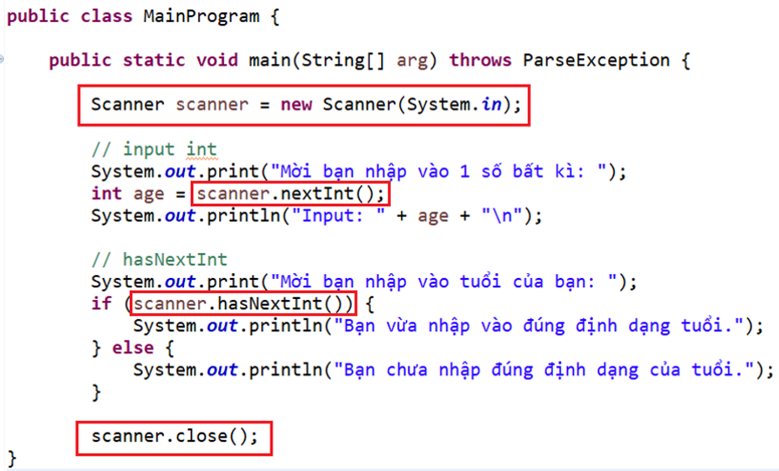
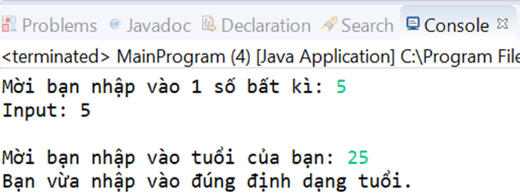
Method

* String nextLine()
* int nextInt()
* hasNextInt()

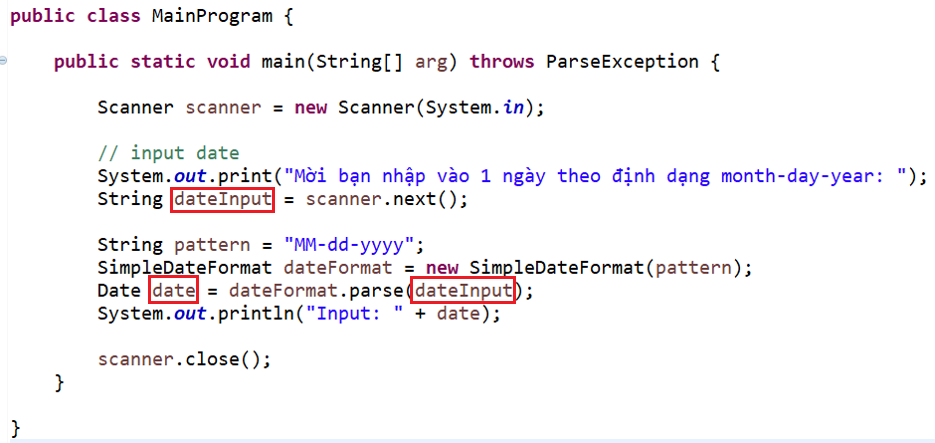
VD1: string

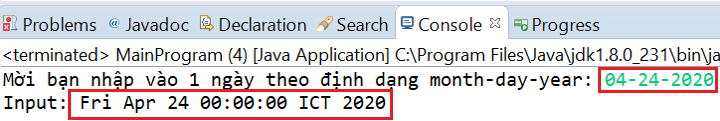
 

VD2: int

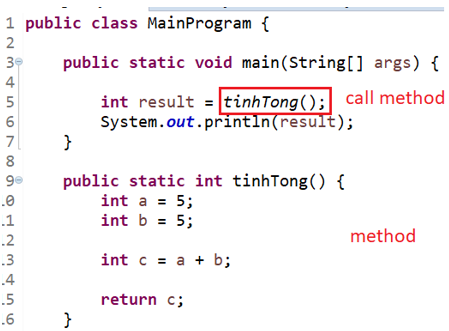
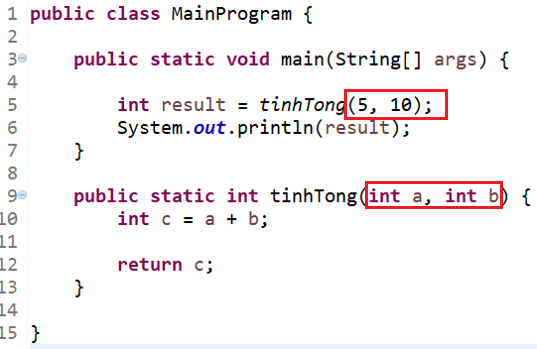
VD3: date





1. **Method**

Giống store ở trong SQL

**** ****

* **Reference**