

2.11. Các cấu trúc điều khiển

2.11.1. Lệnh rẽ nhánh if

Cú pháp: if (biểu thức điều kiện) {

```
// Các lệnh sẽ được thực hiện nếu giá trị của biểu thức điều kiện là true  
}
```

Hoặc:

if (biểu thức điều kiện) {

```
// Các lệnh sẽ được thực hiện nếu giá trị của biểu thức điều kiện là true  
} else {
```

```
// Các lệnh sẽ được thực hiện nếu giá trị của biểu thức điều kiện là false  
}
```

Thí dụ: Lưu chương trình sau vào tập tin IfDemo.java:

```
import java.io.*;  
  
public class IfDemo {  
    public static void main(String args[]) {  
        System.out.print("Vui long nhap mot ky tu:");  
  
        try {  
            int ch = System.in.read(); if (ch == 'A') {  
                System.out.print("Ban rat may man !");  
            } else {  
                System.out.print("Ban khong gap may !");  
            } } catch(IOException ie) { System.out.print("Error:"+ie); } } }
```

2.11.2. Lệnh switch

Cú pháp:

```
switch ( variable ) {  
    case value1 : {  
        // Các tác vụ sẽ được thực thi nếu giá trị của variable là value1  
        break; }  
    case value2 : {  
        // Các tác vụ sẽ được thực thi nếu giá trị của variable là value2  
        break; }
```

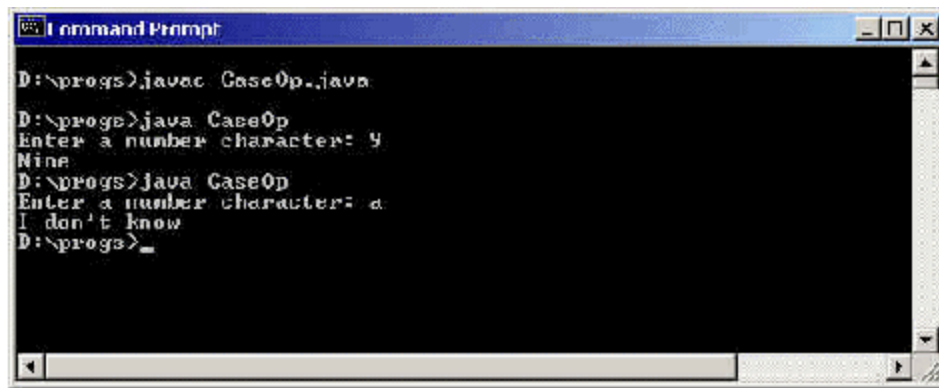
```
...
default:
    // Tác vụ sẽ được thực thi nếu giá trị của variable không là các giá trị
    trên }
```

Thí dụ: Lưu chương trình sau vào tập tin CaseOp.java:

```
import java.io.*;

public class CaseOp {
    public static void main(String args[]) {
        System.out.print("Enter a number character: ");
        try {
            int ch = System.in.read();
            switch(ch) {
                case '0': { System.out.print("Zero"); break; }
                case '1': { System.out.print("One"); break; }
                case '2': { System.out.print("Two"); break; }
                case '3': { System.out.print("Three"); break; }
                case '4': { System.out.print("Four"); break; }
                case '5': { System.out.print("Five"); break; }
                case '6': { System.out.print("Six"); break; }
                case '7': { System.out.print("Seven"); break; }
                case '8': { System.out.print("Eight"); break; }
                case '9': { System.out.print("Nine"); break; }
                default: { System.out.print("I don't know"); break; }
            }
        } catch (IOException ie) { System.out.print("Error "+ie); } } }
```

Biên dịch và thực thi được kết quả sau:



```
Command Prompt
D:\progs>javac CaseOp.java
D:\progs>java CaseOp
Enter a number character: 9
Nine
D:\progs>java CaseOp
Enter a number character: a
I don't know
D:\progs>
```

2.11.3. Lệnh while

Cú pháp:

```
while (biểu thức điều kiện) {
    Các lệnh sẽ được thực hiện lặp lại nếu giá trị của biểu thức điều kiện đúng }
```

Thí dụ: Lưu chương trình sau vào tập tin WhileDemo.java:

```
import java.io.*;

public class WhileDemo {

    public static void main(String args[]) {
        int num = '9'; while (num > '0') {
            System.out.print((char)num + " "); num--; } } }
```

2.11.4. Lệnh do - while

Cú pháp:

```
do {
    // Lặp lại các tác vụ ở đây cho đến khi điều kiện condition có giá trị là false
} while (biểu thức điều kiện)
```

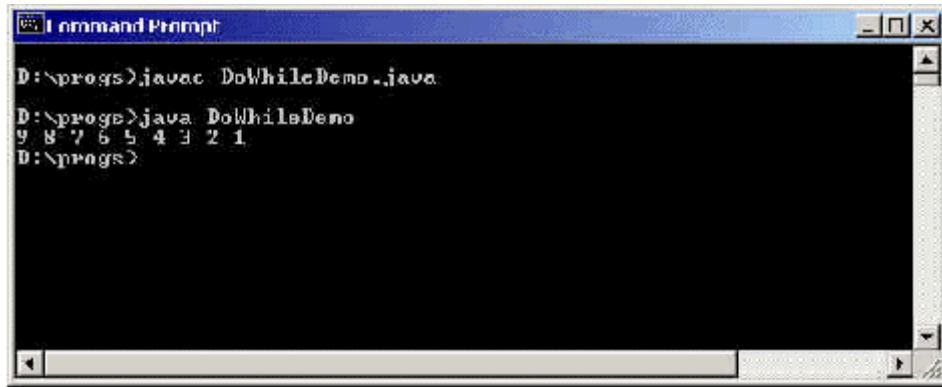
Thí dụ: Lưu chương trình sau vào tập tin DoWhileDemo.java:

```
import java.io.*;

public class DoWhileDemo {

    public static void main(String args[]) {
        int num = '9'; do {
            System.out.print((char)num + " "); num--; }
        while (num > '0'); } }
```

Biên dịch và thực thi được kết quả sau:



```
Command Prompt
D:\progs>javac DoWhileDemo.java
D:\progs>java DoWhileDemo
9 8 7 6 5 4 3 2 1
D:\progs>
```

2.11.5. Lệnh for

Cú pháp:

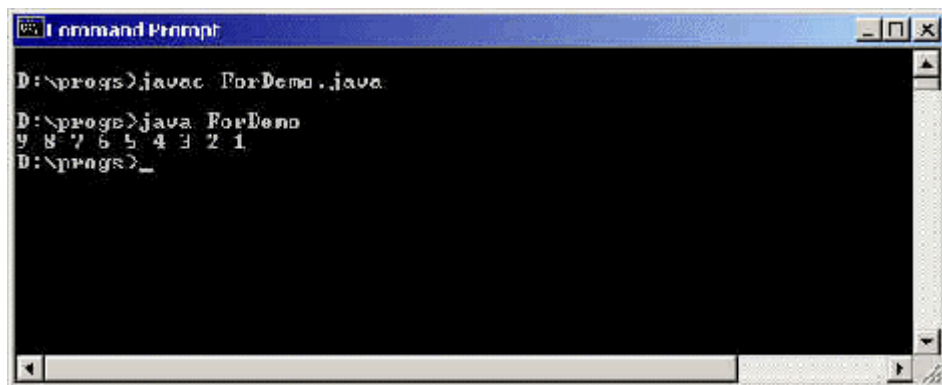
```
for (biểu thức 1; biểu thức điều kiện; biểu thức 2){
    // Các lệnh cần lặp lại
}
```

Thí dụ: Lưu chương trình sau vào tập tin ForDemo.java:

```
import java.io.*;

public class ForDemo {
    public static void main(String args[]) {
        for(int num = '9'; num>'0'; num --) {
            System.out.print((char)num + " "); } } }
```

Biên dịch và thực thi được kết quả như sau:



```
Command Prompt
D:\progs>javac ForDemo.java
D:\progs>java ForDemo
9 8 7 6 5 4 3 2 1
D:\progs>
```

2.11.6. Lệnh break

Vòng lặp của các lệnh while, do-while và for sẽ kết thúc khi lệnh break được thực hiện. Thí dụ: Lưu chương trình sau vào tập tin BreakDemo.java:

```
import java.io.*;

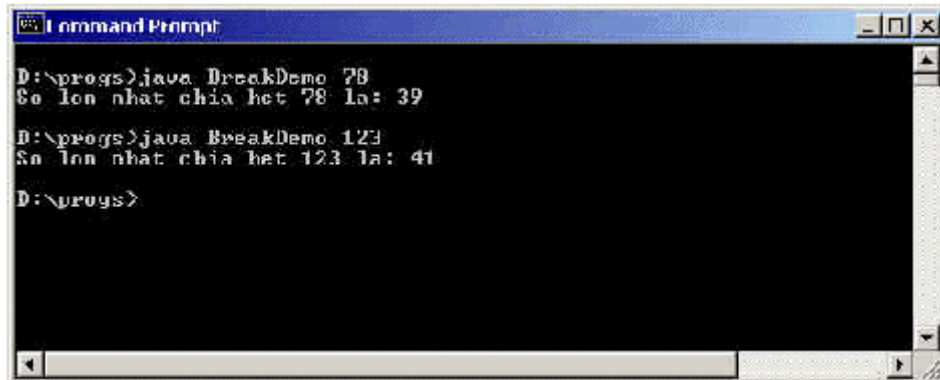
public class BreakDemo {
```

```

public static void main(String args[]){
    int num =Integer.valueOf(args[0]).intValue();
    int i= num /2;
    while(true){ if (num % i ==0) break; i--; }
    System.out.println("Số lớn nhất chia hết "+num+ " là: "+i); } }

```

Biên dịch và thực thi được kết quả sau:



```

D:\progs>java BreakDemo 78
Số lớn nhất chia hết 78 là: 39

D:\progs>java BreakDemo 123
Số lớn nhất chia hết 123 là: 41

D:\progs>

```

Chương trình trên đổi đổi số thứ nhất của nó (lưu trong args[0]) thành số (bằng lệnh Integer.valueOf(args[0]).intValue()) và tìm số lớn nhất chia hết số này.

2.11.7. Lệnh continue

Trong một lần lặp nào đó của các lệnh while, do-while và for, nếu gặp lệnh continue thì lần lặp sẽ kết thúc (bỏ qua các lệnh phía sau continue) để bắt đầu lần lặp tiếp theo.

Thí dụ: Lưu chương trình sau vào tập tin ContinueDemo.java:

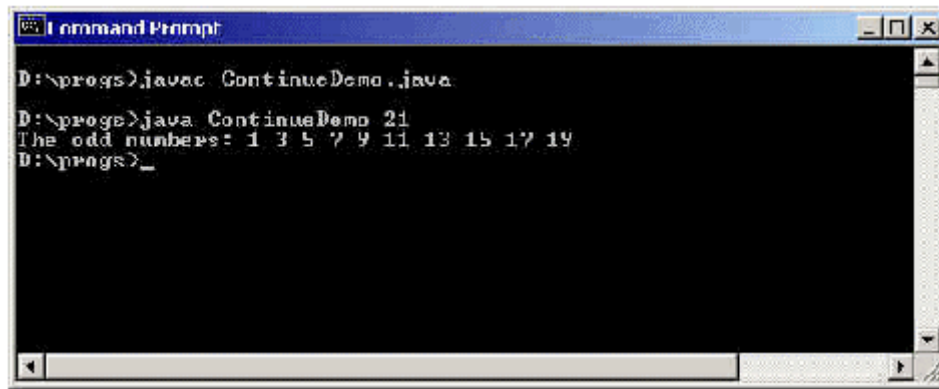
```

import java.io.*;

public class ContinueDemo{
    public static void main(String args[]){
        int num =Integer.valueOf(args[0]).intValue();
        System.out.print("The odd numbers: ");
        for (int i =0; i< num; i++ ){
            if (i % 2 ==0) continue; System.out.print(i+ " "); } } }

```

Biên dịch và thực thi được kết quả sau:



```
Command Prompt
D:\progs>javac ContinueDemo.java
D:\progs>java ContinueDemo 21
The odd numbers: 1 3 5 7 9 11 13 15 17 19
D:\progs>_
```

Chương trình này in ra tất cả các số lẻ nhỏ hơn số đưa vào từ đối số.