

Trang 05

1. Cho biết kết quả chương trình sau:

```
byte value = 9;
byte mask = 6;

if(value == 9 & ++mask == 6){
    System.out.println("Led turn ON");
} else {
    System.out.println("Led turn OFF");
}
System.out.println(mask);
```

Chương trình ra kết quả

```
Led turn OFF
7
```

Giải thích

Trong điều kiện của if có toán tử tiền tố ++ nên thứ tự ưu tiên như sau:

```
mask = mask + 1
```

→ `value == 9 & mask == 6` trả về TRUE/FALSE

Vì `7 != 6` nên biểu thức `FALSE`, chạy vào phần code khối của “OFF” và giá trị của `mask` là 7

```
byte value = 9;
byte mask = 6;

if(value == 9 || mask++ == 6){
    System.out.println("Led turn ON");
} else {
    System.out.println("Led turn OFF");
}
System.out.println(mask);
```

Chương trình ra kết quả

```
Led turn ON
7
```

Giải thích:

Trong điều kiện của if có toán tử hậu tố ++ nên thứ tự ưu tiên như sau:

```
value == 9 & mask == 6 trả về TRUE/FALSE
```

→ `mask = mask + 1`

Vì `6 = 6` nên biểu thức `TRUE`, chạy vào phần code khối của “ON” và giá trị của `mask` là 7

Trang 08

- Cho biết kết quả chương trình

```
char keyPess = 'A';
switch (keyPess)
{
default:
System.out.println("Program is
still a live");
case 'A':
case 'a':
System.out.println("Exit
program");
break;
}
```

Chương trình ra kết quả

Exit program

Giải thích

Khi chạy lệnh `switch`, `case` sẽ được ưu tiên chạy trước

`case 'A'` thoả mãn giá trị của `keyPess` nên sẽ tiếp tục chạy lần lượt tuần tự cho đến khi gặp `break;` hoặc kết thúc block,

trường hợp này in ra `Exit program`

```
char keyPess = 'B';
switch (keyPess)
{
default:
System.out.println("Program is
still a live");
case 'A':
case 'a':
System.out.println("Exit
program");
break;
}
```

Chương trình ra kết quả

Program is still a live
Exit program

Giải thích

Khi chạy lệnh `switch`, `case` sẽ được ưu tiên chạy trước

do không có giá trị case nào thoả mãn `keyPess` nên sẽ nhận vào giá trị `default` và tuần tự đến khi gặp `break;` hoặc kết thúc block,

trường hợp này sẽ in `Program is still a live`, sau đó in `Exit program`

Trang 10

- Cho biết kết quả chương trình

```
static boolean b;  
public static void main(String [] args)  
{  
    short hand = 42;  
    if ( hand < 50 && !b ) /* Line 7 */  
        hand++;  
    if ( hand > 50 );      /* Line 9 */  
    else if ( hand > 40 )  
    {  
        hand += 7;  
        hand++;  
    }  
    else  
        --hand;  
    System.out.println(hand);  
}
```

Chương trình ra kết quả

51

Giải thích

Biến `b` không có giá trị khởi tạo ban đầu nên mặc định là `FALSE`

Khi đến dòng 7, do `hand = 42 < 50` và `!b = FALSE` nên thoả điều kiện và `hand` tăng lên 1 đơn vị; lúc này `hand = 43`

Khi đến dòng 9, do `hand = 43 < 50` nên sẽ chạy lệnh `else if`, do `hand = 43 > 40` thoả điều kiện nên sẽ chạy vào block này; lúc này `hand` tăng 7 đơn vị và tăng 1 đơn vị, `hand = 43 + 7 + 1 = 51`, `hand = 51`

Lúc này kết quả in ra như trên

Trang 19 (1)

- Kết quả hiển thị ra màn hình

```
n= 6;  
while(n-- > 0)  
{  
    System.out.println(n);  
}
```

Chương trình ra kết quả

```
5  
4  
3  
2  
1  
0
```

Giải thích

Khi chạy lệnh `while` với điều kiện `(n-- > 0)`, do có hậu tố toán tử `--`, nên thứ tự ưu tiên thực hiện là:

`(n > 0)` trả về `TRUE/FALSE`

→ `n= n - 1`

→ nếu `TRUE` thì thực hiện trong khối

nên giá trị `n` in ra lần lượt 5 đến 0

Lần	<code>n-- > 0</code>	<code>n--</code>	<code>n</code>
1 (<code>n = 6</code>)	TRUE	5	5
2 (<code>n = 5</code>)	TRUE	4	4
3 (<code>n = 4</code>)	TRUE	3	3
4 (<code>n = 3</code>)	TRUE	2	2
5 (<code>n = 2</code>)	TRUE	1	1
6 (<code>n = 1</code>)	TRUE	0	0
7 (<code>n = 0</code>)	FALSE	-1	break
Thoát vòng lặp while			

```
n= 6;  
do  
{  
    System.out.println(n);  
}while(--n > 0);
```

Chương trình ra kết quả

```
6  
5  
4  
3  
2  
1
```

Giải thích

Khi chạy lệnh `do..while` với điều kiện `(--n > 0)`, do có tiền tố toán tử `--`, nên thứ tự ưu tiên thực hiện là:

thực hiện block lần đầu

→ `n= n - 1`

→ `(n > 0)` trả về `TRUE/FALSE`

→ nếu `TRUE` thì thực hiện trong khối

nên giá trị `n` in ra lần lượt 6 đến 1

Lần	<code>n</code>	<code>--n</code>	<code>--n > 0</code>
1 (<code>n = 6</code>)	6	5	TRUE
2 (<code>n = 5</code>)	5	4	TRUE
3 (<code>n = 4</code>)	4	3	TRUE
4 (<code>n = 3</code>)	3	2	TRUE
5 (<code>n = 2</code>)	2	1	TRUE
6 (<code>n = 1</code>)	1	0	FALSE
Thoát vòng lặp while			

Trang 19 (2)

- Kết quả hiển thị ra màn hình

```
byte abc = 112;  
do  
{  
    System.out.println(abc);  
}while(++abc > 111 && abc++ >  
112);
```

```
byte abc = 112;  
do  
{  
    System.out.println(abc);  
}while(++abc > 111 | abc++ >  
112);
```

2 chương trình ra kết quả giống nhau

```
112  
114  
116  
118  
120  
122  
124  
126  
-128
```

Giải thích

Khi chạy lệnh `do..while`, chương trình chạy lệnh trong khối trước, sau đó đến điều kiện `(++abc > 111 && abc++ > 112)` hoặc `(++abc > 111 | abc++ > 112)`, do có tiền tố lẫn hậu tố `++`, nên thứ tự ưu tiên của lệnh sẽ là:

`abc = abc + 1` (của tiền tố)

→ `(++abc > 111 && abc++ > 112)` hoặc `(++abc > 111 | abc++ > 112)` trả về giá trị `TRUE/FALSE`

→ `abc = abc + 1` (của hậu tố)

→ nếu điều kiện `TRUE` thì thực hiện tiếp lệnh trong khối

Giải thích vì sao -128 vẫn được in ra và dừng chương trình:

Sau khi in `abc = 126`, `abc = 127` do tiền tố, và thỏa điều kiện (`127 > 111` và `127 > 112` cũng như `127 > 111` hoặc `127 > 112`), sau đó `abc = 128` nhưng vượt quá phạm vi của byte `[-128..127]` nên `abc = -128` và in ra, sau đó không thỏa điều kiện nữa nên thoát vòng `do..while`

Trang 22

- Cho biết kết quả hiện thị ra màn hình

```
int term, sum;
for (term = 1, sum = 0; (term <= 5) | (sum++ <= 20); term++,
sum = sum + 3);
System.out.print(term);
System.out.println(" " + sum);
```

Chương trình ra kết quả

7 25

Giải thích

Vòng `for` có

- Giá trị khởi tạo ban đầu `term = 1, sum = 0`;
- Điều kiện `(term <= 5 hoặc sum++ <= 20)`;
- Giá trị cập nhật sau mỗi vòng là `term++` và `sum = sum + 3`;

Chú ý: Giá trị `sum` tăng lên 1 đơn vị sau khi xét điều kiện

Giá trị		term <= 5 sum++ <= 20
term	sum	
1	0	TRUE
2	0+1+3 = 4	TRUE
3	4 + 1 + 3 = 8	TRUE
4	8 + 1 + 3 = 12	TRUE
5	12 + 1 + 3 = 16	TRUE
6	16 + 1 + 3 = 20	TRUE
7	20 + 1 + 3 = 24	FALSE
break	24 + 1 = 25	break

Trang 26

- Tìm lỗi chương trình bên dưới

```
String s = "";
char c = ' ';
Scanner keyboard = new Scanner(System.in);
do
{
    System.out.println("Enter 'A' for option A " +
        "or 'B' for option B.");
    s = keyboard.next( );
    s.toLowerCase( );
    c = s.charAt(0);
}
while ((c != 'a') || (c != 'b'));
```

Lỗi (so với yêu cầu đề bài):

Dòng 9, biến `s` chưa cập nhật lại giá trị sau khi chuyển sang chữ thường

Dòng 12, sửa điều kiện so sánh là `&&` (và) thay vì `||` (hoặc),

vì nếu biến `c` khác `'a'` hoặc `'b'` thì điều kiện luôn `TRUE` và vòng lặp `while` chạy mãi, do đó sửa thành biến `c` khác cả `'a'` và `'b'` thì điều kiện `FALSE` và thoát khỏi vòng lặp

Chương trình sau khi sửa

```
String s = "";
char c = ' ';
Scanner keyboard = new Scanner(System.in);
do {
    System.out.println("Enter 'A' for option A " + "or 'B' for option
B.");
    s = keyboard.next();
    s = s.toLowerCase();
    c = s.charAt(0);
} while ((c != 'a') && (c != 'b'));
```