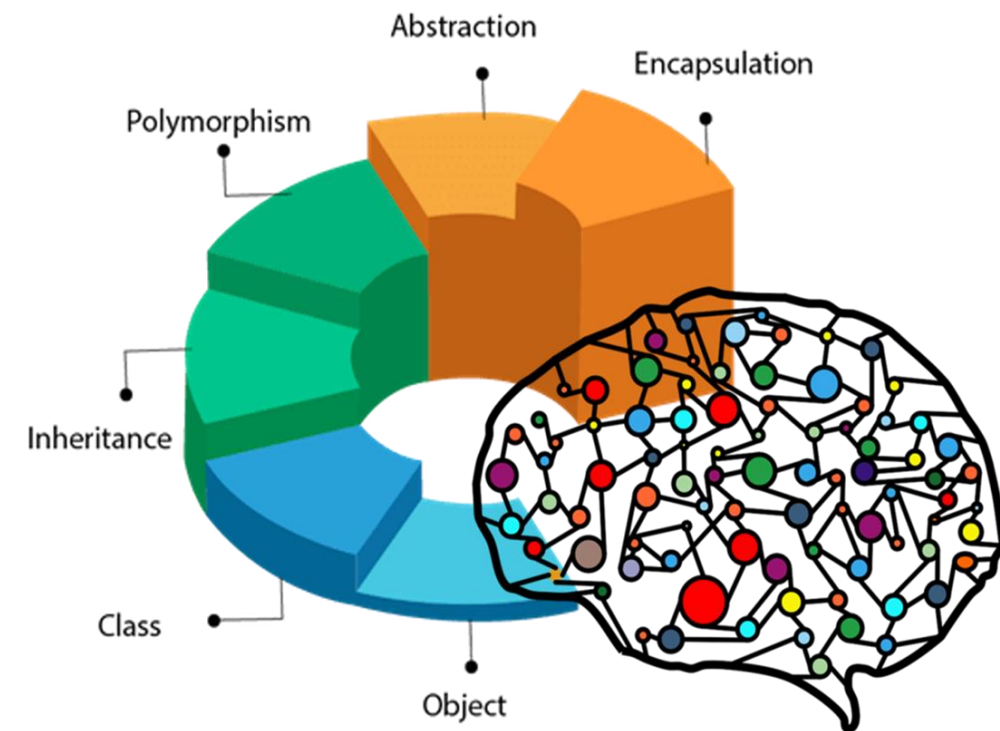


KỸ THUẬT LẬP TRÌNH

ĐẶNG VĂN NGHĨA
0975079414
nghiadv@donga.edu.vn

1. Biểu thức
2. Toán tử (Operators)
3. Chuyển đổi kiểu
4. Câu lệnh gán và biểu thức gán
5. Biểu thức điều kiện
6. Thứ tự ưu tiên của các phép toán



1. BIỂU THỨC

- ❖ **Biểu thức** là sự kết hợp giữa các phép toán và các toán hạng để diễn đạt một công thức toán học cụ thể. Khi viết biểu thức nên dùng các dấu ngoặc tròn thể hiện đúng trình tự tính toán trong biểu thức. Mỗi biểu thức sẽ có một giá trị. Như vậy hằng, biến, phân tử mảng và hàm cũng được xem là biểu thức.
- ❖ **Biểu thức** được phân loại theo kiểu giá trị: nguyên và thực. Trong các mệnh đề logic, biểu thức được phân thành đúng (giá trị khác không) và sai (giá trị bằng 0).
- ❖ **Biểu thức** thường được dùng trong:
 - Vế phải của câu lệnh gán
 - Làm tham số thực sự của hàm (như hàm printf)
 - Làm chỉ số
 - Trong các toán tử if, switch, for, while, do while.

1. BIỂU THỨC

❖ **Ví dụ:** tính diện tích tam giác

- $p = (a+b+c)/2$; //nửa chu vi

- $s = \text{Math.sqrt}(p*(p-a)*(p-b)*(p-c));$

❖ Như vậy có hai khái niệm chính tạo nên biểu thức là toán hạng và phép toán. Toán hạng gồm: hằng, biến, phần tử mảng và hàm.

2. TOÁN TỬ (OPERATOR)

❖ **Toán tử** là biểu tượng (ký tự) đặc biệt thực hiện các hoạt động trên toán hạng (hằng, biến, phần tử của mảng, hàm và giá trị).

- Toán tử số học
- Toán tử gán
- Toán tử quan hệ
- Toán tử logic
- Toán tử Ternary
- Toán tử bit

2. TOÁN TỬ (OPERATOR)

❖ **Toán tử số học:** được sử dụng để tính toán. Cho 2 số nguyên a và b, với $a = 8$, $b = 5$.

Ta có:

Toán tử	Ý nghĩa	Ví dụ	Kết quả	Toán tử	Ý nghĩa	Ví dụ	Kết quả
+	Cộng hai toán hạng hoặc nối chuỗi	$a + b$	13	+	Biểu diễn số dương	$+a$	8
-	Trừ hai toán hạng	$a - b$	3	-	Biểu diễn số âm	$-a$	-8
*	Nhân hai toán hạng	$a * b$	40	++	Tăng 1 giá trị	$++a$ $a++$	9 9
/	Chia hai toán hạng	a / b	1	--	Giảm 1 giá trị	$--a$ $a--$	7 7
%	Lấy phần dư của phép chia hai toán hạng	$a \% b$	3				
+=	Cộng và gán giá trị	$a += 2$	10				
-=	Trừ và gán giá trị	$a -= 2$	6				
*=	Nhân và gán giá trị	$a *= 2$	16				
/=	Chia và gán giá trị	$a /= 2$	4				
%=	Lấy phần dư của phép chia và gán giá trị	$a \% = 2$	0				

2. TOÁN TỬ (OPERATOR)

❖ **Toán tử gán:** được sử dụng để gán giá trị cho một biến, có thể gán nhiều giá trị cho nhiều biến cùng lúc. Cho 2 số nguyên a và b, với $a = 8$, $b = 5$. Ta có:

Toán tử	Ý nghĩa	Ví dụ	Kết quả
=	Toán tử gán	$a = b$ $a = b = 7$	$a = b = 5$ $a = b = 7$

2. TOÁN TỬ (OPERATOR)

❖ **Toán tử quan hệ:** nếu mệnh đề đúng trả về giá trị true, ngược lại trả về false. Cho 2 số nguyên a và b, với $a = 8$, $b = 5$. Ta có:

Toán tử	Ý nghĩa	Ví dụ	Kết quả
<code>==</code>	so sánh bằng	<code>a == b</code>	False
<code>!=</code>	so sánh không bằng	<code>a != b</code>	True
<code>></code>	so sánh lớn hơn	<code>a > b</code>	True
<code><</code>	so sánh nhỏ hơn	<code>a < b</code>	False
<code>>=</code>	so sánh lớn hơn hoặc bằng	<code>a >= b</code>	True
<code><=</code>	so sánh nhỏ hơn hoặc bằng	<code>a <= b</code>	False

2. TOÁN TỬ (OPERATOR)

❖ **Toán tử logic:** Cho 2 số nguyên a và b, với $a = 8$, $b = 5$. Ta có:

Toán tử	Ý nghĩa	Ví dụ	Kết quả
&& (AND)	Trả về giá trị true nếu tất cả điều kiện đều đúng (true)	$(a > 3 \ \&\& \ b < 10)$	True
 (OR)	Trả về giá trị true nếu có ít nhất một điều kiện đúng (true)	$(a \geq 9 \ \ b \geq 5)$	True
!	Phủ định giá trị logic	$!(a > 3 \ \&\& \ b < 10)$	False

2. TOÁN TỬ (OPERATOR)

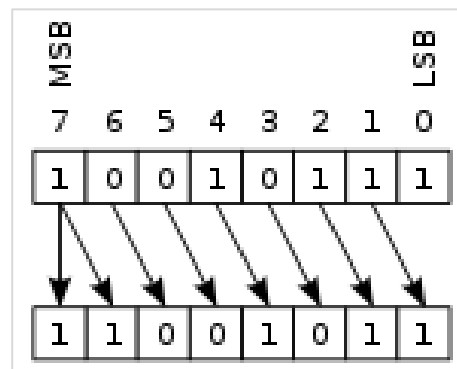
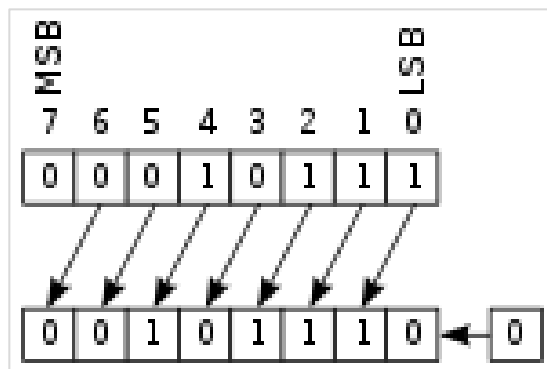
❖ **Toán tử Ternary:** dùng trong biểu thức điều kiện. Cho 2 số nguyên a và b, với $a = 8$, $b = 5$. Ta có:

Toán tử	Ý nghĩa	Ví dụ	Kết quả
?: (Ternary)	Toán tử 3 ngôi. Cú pháp: btđk ? gt1 : gt2 Nếu btđk đúng trả về gt1, ngược lại btđk trả về gt2	$(a > b) ? \text{"True"} : \text{"False"}$	True
		$(a < b) ? \text{"True"} : \text{"False"}$	False

2. TOÁN TỬ (OPERATOR)

❖ Toán tử bit

A	B	A & B (AND)	A B (OR)	A ^ B (XOR)	~A (NOT)
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0



2. TOÁN TỬ (OPERATOR)

❖ **Toán tử bit:** Cho 2 số nguyên a và b, với $a = 8$, $b = 5$. Ta có:

Toán tử	Ý nghĩa	Ví dụ	Kết quả
&	AND	$a \& b$	0
	OR	$a b$	13
~	NOT	$\sim a$	-9
^	XOR	$a \wedge b$	13
«	Dịch bit sang trái	$a \ll 1$	16
»	Dịch bit sang phải	$a \gg 1$	4

Toán tử	Ý nghĩa	Ví dụ	Kết quả
&=	AND và gán giá trị	$a \&= b$	$a = 0$
=	OR và gán giá trị	$a = b$	$a = 13$
~=	NOT và gán giá trị	$a \sim= a$	$a = -9$
^=	XOR và gán giá trị	$a \wedge= b$	$a = 4$
«=	Dịch bit sang trái và gán giá trị	$a \ll= 1$	$a = 16$
»=	Dịch bit sang phải và gán giá trị	$a \gg= 1$	4

2. TOÁN TỬ (OPERATOR)

❖ **Ví dụ 1:** Hãy cho biết kết quả thực hiện chương trình cho bên dưới và giải thích

```
public class Example {  
    public static void main(String[] args) {  
        int a = 3;  
        System.out.println(a++);  
        System.out.println(++a);  
        System.out.println(a+++1);  
        System.out.println(++a+1);  
    }  
}
```

2. TOÁN TỬ (OPERATOR)

❖ **Ví dụ 2:** Hãy cho biết kết quả thực hiện chương trình cho bên dưới và giải thích

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        int a = 0, b, c;  
        a++;  
        b = a;  
        c = a++;  
        System.out.println(a);  
        System.out.println(b);  
        System.out.println(c);  
    }  
}
```

2. TOÁN TỬ (OPERATOR)

❖ **Ví dụ 3:** Hãy cho biết kết quả thực hiện chương trình cho bên dưới và giải thích

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        int a = 0, b, c;  
        ++a;  
        b = a;  
        c = ++a;  
        System.out.println(a);  
        System.out.println(b);  
        System.out.println(c);  
    }  
}
```

2. TOÁN TỬ (OPERATOR)

❖ **Ví dụ 4:** Hãy cho biết kết quả thực hiện chương trình cho bên dưới và giải thích

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        int a = 0, b, c;  
        ++a;  
        b = ++a;  
        c = a++;  
        System.out.println(a);  
        System.out.println(b);  
        System.out.println(c);  
    }  
}
```


2. TOÁN TỬ (OPERATOR)

❖ **Ví dụ 5:** Hãy cho biết kết quả thực hiện chương trình cho bên dưới và giải thích

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        int a = 0, b, c;  
        ++a;  
        b = ++a;  
        c = b++;  
        System.out.println(a);  
        System.out.println(b);  
        System.out.println(c);  
    }  
}
```

2. TOÁN TỬ (OPERATOR)

❖ **Ví dụ 6:** Hãy cho biết kết quả thực hiện chương trình cho bên dưới và giải thích

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        int a = 0, b, c;  
        ++a;  
        b = a++;  
        c = b++;  
        c += c;  
        System.out.println(a);  
        System.out.println(b);  
        System.out.println(c);  
    }  
}
```

3. CHUYỂN ĐỔI KIỂU

❖ **Chuyển đổi kiểu (ép kiểu)** là gán giá trị của một biến có kiểu dữ liệu này đến một biến có kiểu dữ liệu khác.

❖ **Ví dụ 1:**

- `float c = 35.8f;`
- `int b = (int)c + 1;`

❖ **Có hai loại chuyển đổi kiểu dữ liệu:**

- **Nới rộng (widening):** Là quá trình làm tròn số từ kiểu dữ liệu có kích thước nhỏ sang kiểu dữ liệu có kích thước lớn hơn. Kiểu biến đổi này không làm mất thông tin.
- **Thu hẹp (narrowing):** Là quá trình làm tròn số từ kiểu dữ liệu có kích thước lớn sang kiểu dữ liệu có kích thước nhỏ hơn. Kiểu biến đổi này có thể làm mất thông tin.

3. CHUYỂN ĐỔI KIỂU

❖ **Nới rộng (widening):** byte → short → int → long → float → double

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        int in = 120;  
        long lo = in;    //không yêu cầu chỉ định ép kiểu  
        float fl = lo;   //không yêu cầu chỉ định ép kiểu  
        System.out.println("Giá trị Int: " + in);  
        System.out.println("Giá trị Long: " + lo);  
        System.out.println("Giá trị Float: " + fl);  
    }  
}
```

3. CHUYỂN ĐỔI KIỂU

❖ **Thu hẹp (narrowing):** double → float → long → int → short → byte

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        double dou = 120;  
        long lo = (long)dou; //yêu cầu chỉ định ép kiểu long  
        int in = (int)lo;    //yêu cầu chỉ định ép kiểu int  
        System.out.println("Giá trị Int: " + dou);  
        System.out.println("Giá trị Long: " + lo);  
        System.out.println("Giá trị Float: " + in);  
    }  
}
```

3. CHUYỂN ĐỔI KIỂU

❖ Chuyển đổi String thành int, long, double, float, short khi:

- thực hiện các phép toán trên chuỗi có chứa số;
- lấy dữ liệu từ textfield hoặc textarea;
- dữ liệu nhập vào như một chuỗi;
- sử dụng Integer.parseInt(), Long.parseLong(), Double.parseDouble(), Float.parseFloat(), Short.parseShort()

❖ Ví dụ 2: Chuyển đổi String thành int

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        String str = "200";  
        System.out.println(Integer.parseInt(str));  
    }  
}
```

3. CHUYỂN ĐỔI KIỂU

❖ Ví dụ 3: Chuyển đổi String thành long

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        String str = "200";  
        System.out.println(Long.parseLong(str));  
    }  
}
```

❖ Ví dụ 4: Chuyển đổi String thành double

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        String str = "200";  
        System.out.println(Double.parseDouble(str));  
    }  
}
```

❖ Ví dụ 5: Chuyển đổi String thành float

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        String str = "200";  
        System.out.println(Float.parseFloat(str));  
    }  
}
```

❖ Ví dụ 6: Chuyển đổi String thành short

```
public class Example {  
    public static void main(String[] args) {  
        //Các câu lệnh xử lý  
        String str = "200";  
        System.out.println(Short.parseShort(str));  
    }  
}
```


3. CHUYỂN ĐỔI KIỂU

❖ Chuyển đổi int, long, double, float, short thành String:

- khi hiển thị số trong textfield;
- sử dụng `String.valueOf()` hoặc `Integer.toString()`, `Long.toString()`, `Double.toString()`, `Float.toString()`, `Short.toString()`

❖ Ví dụ 7: Chuyển đổi int thành String

```
public class Example {  
    public static void main(String args[]) {  
        int num = 100;  
        //sử dụng String.valueOf()  
        System.out.println(String.valueOf(num));  
        //sử dụng Integer.toString()  
        System.out.println(Integer.toString(num));  
    }  
}
```

❖ Ví dụ 8: Chuyển đổi long thành String

```
public class Example {  
    public static void main(String args[]) {  
        long num = 9991991991L;  
        //sử dụng String.valueOf()  
        System.out.println(String.valueOf(num));  
        //sử dụng Long.toString()  
        System.out.println(Long.toString(num));  
    }  
}
```

❖ Ví dụ 9: Chuyển đổi float thành String

```
public class Example {  
    public static void main(String args[]) {  
        float num = 999.999F;  
        //sử dụng String.valueOf()  
        System.out.println(String.valueOf(num));  
        //sử dụng Float.toString()  
        System.out.println(Float.toString(num));  
    }  
}
```

❖ Ví dụ 10: Chuyển đổi double thành String

```
public class Example {  
    public static void main(String args[]) {  
        double num = 999.999D;  
        //sử dụng String.valueOf()  
        System.out.println(String.valueOf(num));  
        //sử dụng Double.toString()  
        System.out.println(Double.toString(num));  
    }  
}
```


❖ Ví dụ 11: Chuyển đổi short thành String

```
public class Example {  
    public static void main(String args[]) {  
        short num = 999;  
        //sử dụng String.valueOf()  
        System.out.println(String.valueOf(num));  
        //sử dụng Short.toString()  
        System.out.println(Short.toString(num));  
    }  
}
```

❖ Ví dụ 12: Chuyển đổi String thành Date

```
import java.text.SimpleDateFormat;
import java.util.Date;

public class Example {
    public static void main(String[] args) throws Exception {
        String str = "02/09/2022";
        Date date = new SimpleDateFormat("dd/MM/yyyy").parse(str);
        System.out.println(str + "\t" + date);
    }
}
```



Console

<terminated> Example [Java Application] C:\Program Files\Java\jre-7\bin\java.exe
02/09/2022 Fri Sep 02 00:00:00 ICT 2022

❖ Ví dụ 13: Chuyển đổi String thành Date

```
import java.text.SimpleDateFormat;
import java.util.Date;
public class Example {
    public static void main(String[] args) throws Exception {
        String str1 = "02/9/2022";
        String str2 = "02-09-2022";
        String str3 = "02/9/2022 13:30:50";
        String str4 = "02-09-2022 13:30:50";
        SimpleDateFormat fm1 = new SimpleDateFormat("dd/MM/yyyy");
        SimpleDateFormat fm2 = new SimpleDateFormat("dd-MM-yyyy");
        SimpleDateFormat fm3 = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");
        SimpleDateFormat fm4 = new SimpleDateFormat("dd-MM-yyyy HH:mm:ss");
        Date date1 = fm1.parse(str1);
        Date date2 = fm2.parse(str2);
        Date date3 = fm3.parse(str3);
        Date date4 = fm4.parse(str4);
        System.out.println(str1 + "\t" + date1);
        System.out.println(str2 + "\t" + date2);
        System.out.println(str3 + "\t" + date3);
        System.out.println(str4 + "\t" + date4);
    }
}
```



```
Console ✕
<terminated> Example [Java Application] C:\Program Files\Java\jdk-13\bin
02/9/2022      Fri Sep 02 00:00:00 ICT 2022
02-09-2022     Fri Sep 02 00:00:00 ICT 2022
02/9/2022 13:30:50  Fri Sep 02 13:30:50 ICT 2022
02-09-2022 13:30:50  Fri Sep 02 13:30:50 ICT 2022
```

4. CÂU LỆNH GÁN VÀ BIỂU THỨC GÁN

❖ Câu lệnh gán:

- $i = i + 2;$
- Trong đó vế trái có thể viết gọn lại: $i += 2;$
- $v += e;$
- Trong đó v là một biến hoặc phần tử mảng, e là biểu thức có thể áp dụng cho các phép toán 2 ngôi (+, -, *, /, %).
- **Ví dụ:**
 - ✓ $x = x*(y+3);$
 - ✓ Có thể viết lại: $x *= y+3;$

❖ Biểu thức gán:

- $v = e$
- Trong đó v là một biến (hay phần tử mảng), e là một biểu thức. Giá trị của biểu thức gán là giá trị của e , kiểu của nó là kiểu của v . Nếu đặt dấu ; vào sau biểu thức gán thì ta được toán tử gán: $v = e$;
- **Ví dụ:**
 - ✓ $a = b = 5$;
 - ✓ Có nghĩa là gán giá trị của biểu thức $b = 5$ cho biến a . Kết quả $b = 5$ và $a = 5$.
 - ✓ $a = b = c = d = 8$; //sẽ gán giá trị 8 cho a , b , c và d .
 - ✓ $z = (y = 2) * (x = 6)$; // y có giá trị 2, x có giá trị 6 và z có giá trị 12.

5. BIỂU THỨC ĐIỀU KIỆN

❖ Biểu thức điều kiện:

- Cú pháp: $\text{btđk} ? \text{gt1} : \text{gt2}$

❖ Toán tử logic: Cho 2 số nguyên a và b, với $a = 8$, $b = 5$. Ta có:

Toán tử	Ý nghĩa	Ví dụ	Kết quả
$? :$ (Ternary)	Toán tử 3 ngôi. Cú pháp: $\text{btđk} ? \text{gt1} : \text{gt2}$ Nếu btđk đúng trả về gt1 , ngược lại btđk trả về gt2	$(a > b) ? a : b$ Giá trị lớn nhất	8
		$(a < b) ? a : b$ Giá trị nhỏ nhất	5

6. THỨ TỰ ƯU TIÊN CỦA CÁC PHÉP TOÁN

❖ Thứ tự ưu tiên của các phép toán:

Toán tử	Thứ tự thực hiện
Các toán tử đơn	+, -, ++, --
Các toán tử số học và các toán tử dịch	*, /, +, -, <<, >>
Các toán tử quan hệ	>, <, >=, <=, ==, !=
Các toán tử logic và Bit	&&, , &, , ^
Các toán tử gán	=, *=, /=, +=, -=

❖ Thứ tự ưu tiên của các phép toán:

- Để thay đổi thứ tự ưu tiên một biểu thức có thể sử dụng dấu ngoặc đơn ()
- Phần trong ngoặc đơn thực hiện trước
- Nếu sử dụng nhiều ngoặc đơn lồng nhau thì toán tử nằm trong ngoặc đơn trong cùng sẽ thực thi trước, sau đó đến các ngoặc đơn phía ngoài
- Trong phạm vi một cặp ngoặc đơn thì quy tắc thứ tự ưu tiên vẫn giữ nguyên tác dụng

❖ **Ví dụ:**

- `int a = 20, b = 5, c = 10;`
- $a + b * c = ?$
- $(a + b) * c = ?$
- $a / b - c = ?$
- $a / (b - c) = ?$

❖ Câu hỏi:

- Phân biệt toán tử 1 ngôi, toán tử 2 ngôi và toán tử 3 ngôi?
- Phân biệt giữa toán hạng và toán tử?
- Phân biệt giữa câu lệnh gán và biểu thức gán?
- Cho biết thứ tự ưu tiên của các phép toán?
- Vì sao phải chuyển đổi kiểu (ép kiểu) dữ liệu?