

JAVA**CẤU TRÚC LẬP TRÌNH****1. Hướng dẫn làm bài tập**

- ✓ Tất cả các dữ liệu nhập trên các **TextField** đều là dạng chuỗi. Do vậy nếu sau khi lấy được dữ liệu từ các đối tượng controls này cần thực hiện tính toán thì cần chuyển về kiểu dữ liệu tương ứng phù hợp với mục đích công việc của mình như kiểu số nguyên (int), kiểu số thập phân (float, double),...
- ✓ Để lấy thông tin từ một đối tượng **TextField** người ta sử dụng phương thức có tên là **getText()** trong Java.

Ví dụ: Cho một TextField có tên là txtHoTen để lấy thông tin người dùng nhập vào trên giao diện sử dụng như sau:

```
String strHoTen = txtHoTen.getText();
```

- ✓ Để gán hoặc hiển thị thông tin lên giao diện cho một đối tượng controls như Label, TextField,... người lập trình có thể sử dụng phương thức **setText()** trong Java.

Ví dụ: Hiển thị kết quả sau khi thực hiện tính toán lên một TextField có tên txtKetQua trong Java như sau:

```
int a=5; int b=4;
```

```
int tong = TongSo(a, b);
```

```
txtKetQua.setText(tong);
```

- ✓ Để thiết lập tiêu đề cho một frame trong Java người ta thiết lập thông tin trong sự kiện có tên là formWindowActivated với cú pháp như sau:

```
this.setText("Nội dung tiêu đề frame");
```

- ✓ Để đóng một cửa sổ hiển thị trong Java người ta có thể sử dụng phương thức sau:

this.dispose(); //Giải phóng tài nguyên

this.setVisible(false); //Đóng frame đang mở

- ✓ Sử dụng cấu trúc **If...Else, Switch...Case** để làm các bài tập được giao

2. Cấu trúc vòng lặp

Trong lập trình Java để thực hiện một số các công việc giống nhau, lặp đi lặp lại nhiều lần người ta có thể sử dụng cấu trúc vòng lặp while, do...while, for tùy theo mục đích sử dụng của mình.

✓ While

Thực hiện công việc trong vòng lặp nếu biểu thức điều kiện thỏa mãn

Cú pháp:

while(Bieu_thuc_dieu_kien)

{

 //Thực hiện các công việc trong vòng lặp while

}

Ví dụ: In thông tin ra màn hình nếu biến i còn nhỏ hơn 5

```
int i=0;
```

```
while(i<5)
```

```
{
```

```
    System.out.println("Số thứ tự là: " + i);
```

```
    i++;
```

```
}
```

✓ **do...while**

Thực hiện công việc ít nhất một lần mà không quan tâm biểu thức thỏa mãn hay không => **Làm trước sau đó mới hỏi.**

Cú pháp:**do**

{

//Thực hiện công việc trong vòng lặp

}

while(Bieu_thuc_dieu_kien)**Ví dụ:**

//Khai báo biến

int i=0;

//Sử dụng do...while trong Java

do

{

System.out.println("Số thứ tự là: " + i);

}

while(i<0);

✓ **for**

Thực hiện công việc trong vòng lặp theo các thông tin thỏa mãn trong vòng lặp bao gồm: khai báo biến, giá trị khởi tạo; biểu thức điều kiện; bước lặp

Sử dụng để thực hiện, thao tác với một tập dữ liệu nhỏ trong tập dữ liệu lớn hoặc tất cả cả dữ liệu.

Cú pháp:**for**(Khai_bao_bien, gan_gia_tri; **Bieu_thuc_dieu_kien**; Buoc_lap)

```
{  
    //Thực hiện công việc trong vòng lặp  
}
```

Ví dụ:

```
//Sử dụng vòng lặp for  
for(i=2; i<5; i+=2)  
{  
    System.out.println("Số thứ tự là: " + i);  
}
```

Sử dụng để thực hiện, thao tác với tất cả dữ liệu trong mảng, danh sách,...

for(Kieu_du_lieu Ten_bien : Tap_du_lieu)

```
{  
    //Thực hiện công việc trong vòng lặp  
}
```

Ví dụ: Cho một mảng gồm 8 phần tử như ở dưới sau đó thực hiện in ra màn hình các phần tử đó.

//Sử dụng vòng for để duyệt tất cả các phần tử trong mảng hoặc danh sách

```
int arr[] = new int[]{10,40,20,50,60,70,90,80};  
System.out.println("Số phần tử là: ");  
for(int a : arr)  
{  
    System.out.println(a);  
}
```