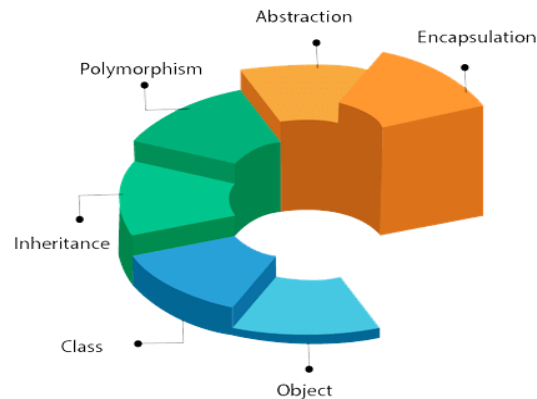


# Lập Trình Hướng Đối Tượng – CT176



TS. Phan Thượng Cang

[ptcang@cit.ctu.edu.vn](mailto:ptcang@cit.ctu.edu.vn)

Khoa CNTT&TT-Đại học Cần Thơ



## Phần 3

# Lập Trình Hướng Đối Tượng với Java

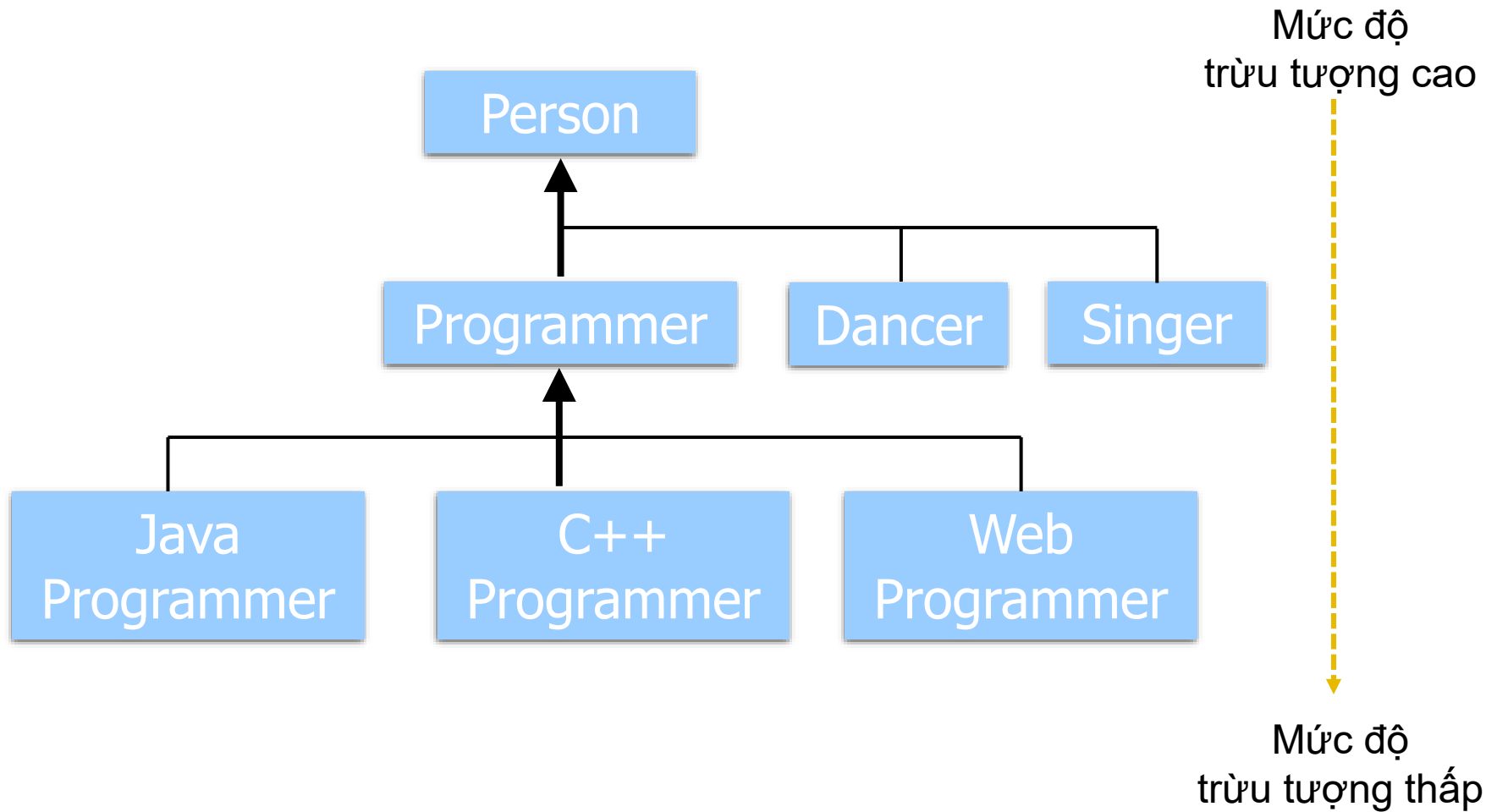
## Chương 4:

# Trừu Tượng Hóa

TS. Phan Thượng Cang

Khoa CNTT&TT-Đại học Cần Thơ

# Trừu Tượng



# Lớp trừu tượng

⌘ Khai báo:

```
<phạm vi> abstract class A {
```

```
...
```

```
}
```

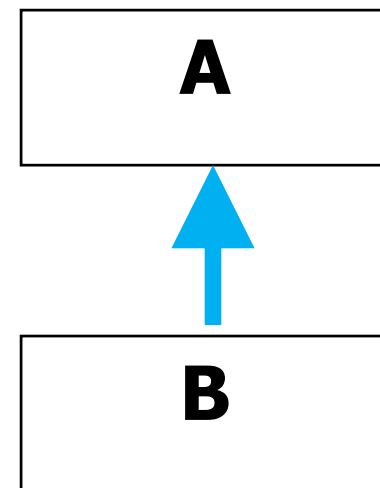
```
<phạm vi> class B extends A{
```

```
...
```

```
}
```

Lưu ý:

- + Lớp trừu tượng **không** được dùng để tạo đối tượng.
- + Nó được dùng để tạo các lớp con và các tham chiếu.
- + Nó là 1 lớp tổng quát đóng vai trò là 1 chuẩn hay giao diện.



# Lớp trừu tượng

⌘ Ví dụ:

```
public abstract class ConVat {  
    private String giong, mau;  
    private float cnang;  
    public abstract void keu();  
  
    public ConVat(){  
        giong= new String(); mau = new String(); cnang=0.0f;  
    }  
    public void nhap() { ... }  
    public void in() { ... }  
    .....  
}
```

# Phương thức trừu tượng

⌘ Ví dụ:

```
public class Bo extends ConVat {  
    public void keu() {  
        System.out.println(" bo ... bo .. ");  
    }  
  
    public Bo(){  
        super();  
    }  
  
    .....  
}
```

# Sử dụng Lớp trừu tượng

- Ví dụ:

```
public class SDConVat{
```

```
    public static void main(String args[]) {
```

```
        ConVat c; // không tạo = new ConVat()
```

```
        c = new Bo();
```

```
        c.nhap();
```

```
        c.in();
```

```
        c.keu(); // Bo.keu() -> lien ket dong
```

```
        c = new Heo();
```

```
        c.nhap();
```

```
        c.in();
```

```
        c.keu(); // Heo.keu() -> lien ket dong
```

```
    }
```

```
}
```

# Giao diện

⌘ Khai báo:

```
<phạm vi> interface I1 {
```

```
    // khai báo các phương thức chung
```

```
}
```

```
<phạm vi> class C implements I1, I2{
```

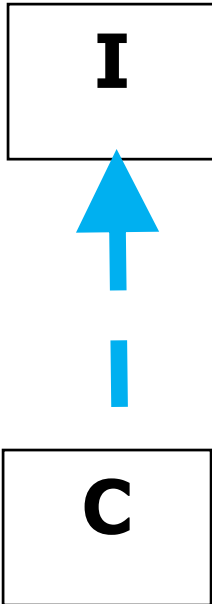
```
    // các thành phần của lớp
```

```
    // định nghĩa các phương thức giao diện
```

```
}
```

Lưu ý:

- + Giao diện chứa các khai báo phương thức chung, một cam kết gồm những khả năng gì mà không định nghĩa cụ thể.
- + Nó **không** được dùng để tạo đối tượng.
- + Nó được dùng để tạo các lớp và các tham chiếu.





# Giao diện

- Ví dụ:

```
public interface Moveable{
```

```
    public void moveUp();  
    public void moveDown();  
    public void moveLeft();  
    public void moveRight();
```

```
}
```

```
public class Diem implements Moveable {
```

```
    private int x, y ;
```

```
    public Diem(){ x = 0; y = 0; }  
    public Diem(int x1, int y1){ x = x1; y = y1; }  
    public void moveUp() {y++; }  
    public void moveDown() { y-- ; }  
    public void moveLeft() { x-- ; }  
    public void moveRight() { x++ ; }  
    public static void main(String args[]) {
```

```
        Moveable I1 = new Diem(); I1.moveUp();
```

```
        Diem B = new Diem(13, 23); I1 = B; I1.moveDown(); }
```

```
}
```

# Thuộc tính tĩnh

- Ví dụ:

```
public class Diem{
```

```
    private int x, y ;
```

```
    public static int count = 0;
```

```
    public Diem(){
```

```
        x = 0; y = 0;
```

```
        count++ ;
```

```
    }
```

```
    public Diem(int x1, int y1){
```

```
        x = x1; y = y1;
```

```
        count++ ;
```

```
    }
```

```
    public static void main(String args[]){
```

```
        Diem A = new Diem();
```

```
        Diem B = new Diem(13, 23);
```

```
        System.out.print("A.count = " + A.count + "Diem.count = " + Diem.count);
```

```
    }
```

```
}
```

