

“HỌC ĐỂ LÀM VIỆC”

Java for Base

Contents

1

Method và Scope Access

2

Class và Object

3

Sử dụng từ khóa this, static

Java for Base



Method và Scope Access

■ Methods

- Là một đơn vị chương trình độc lập dùng để thực hiện một phần việc nào đó như: Nhập số liệu, in kết quả hay thực hiện một số phép tính toán.

- **Cú pháp**

```
[Thuộc tính] [Bổ từ truy xuất] Tên phương thức( [Tham số] )  
{  
    // Nội dung xử lý  
}
```

■ Methods

- Là một đơn vị chương trình độc lập dùng để thực hiện một phần việc nào đó như: Nhập số liệu, in kết quả hay thực hiện một số phép tính toán.

Ví dụ:

```
public float TinhTong(float a, float b)
{
    float tongso=0;
    tongso = a + b;
    return tongso;
}
```

■ Methods

■ Phương thức chia thành 2 loại chính:

- Phương thức không có giá trị trả về thì sử dụng từ khóa **void**
 - Phương thức không có giá trị trả về và không có tham số truyền vào.
 - Phương thức không có giá trị trả về và có tham số truyền vào.
- Phương thức có giá trị trả về sử dụng từ khóa **return** trong nội dung hàm để trả về kiểu dữ liệu tương ứng
 - Phương thức có giá trị trả về và không có tham số truyền vào.
 - Phương thức có giá trị trả về và có tham số truyền vào

- **Scope Access**

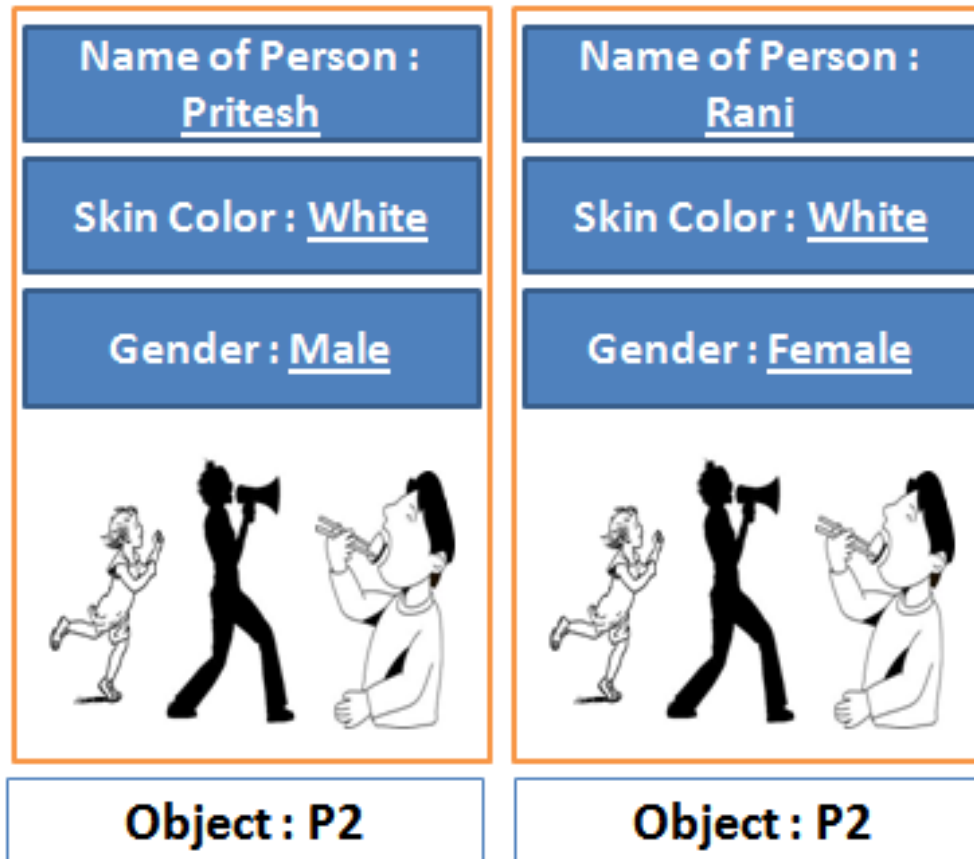


■ Scope Access

Properties	Access modifies
Public	+ Không hạn chế + Sử dụng bất kỳ phương thức của lớp và đối tượng khác
Private	+ Truy cập với các phương thức của chính lớp đó
Protected	+ Được truy cập bởi các phương thức của lớp đó. + Và những lớp đối tượng kế thừa từ nó mới sử dụng được.

Class và Object

■ Class và Object



■ Class và Object

1

Lớp là tập hợp các đối tượng có cùng thuộc tính

2

Các phương thức, thuộc tính, biến

3

Đối tượng là tập hợp các thuộc tính mô tả đối tượng đó.

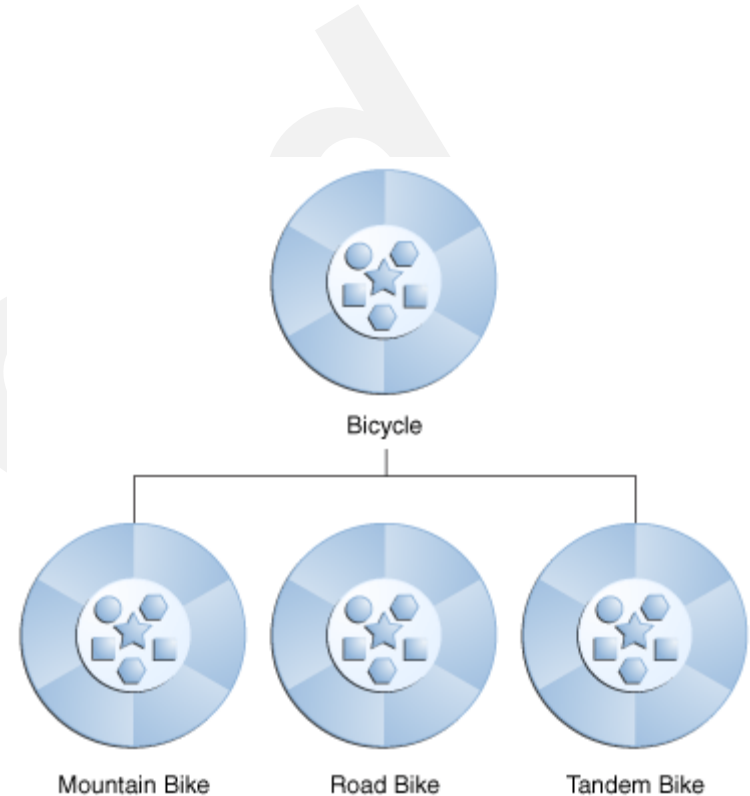
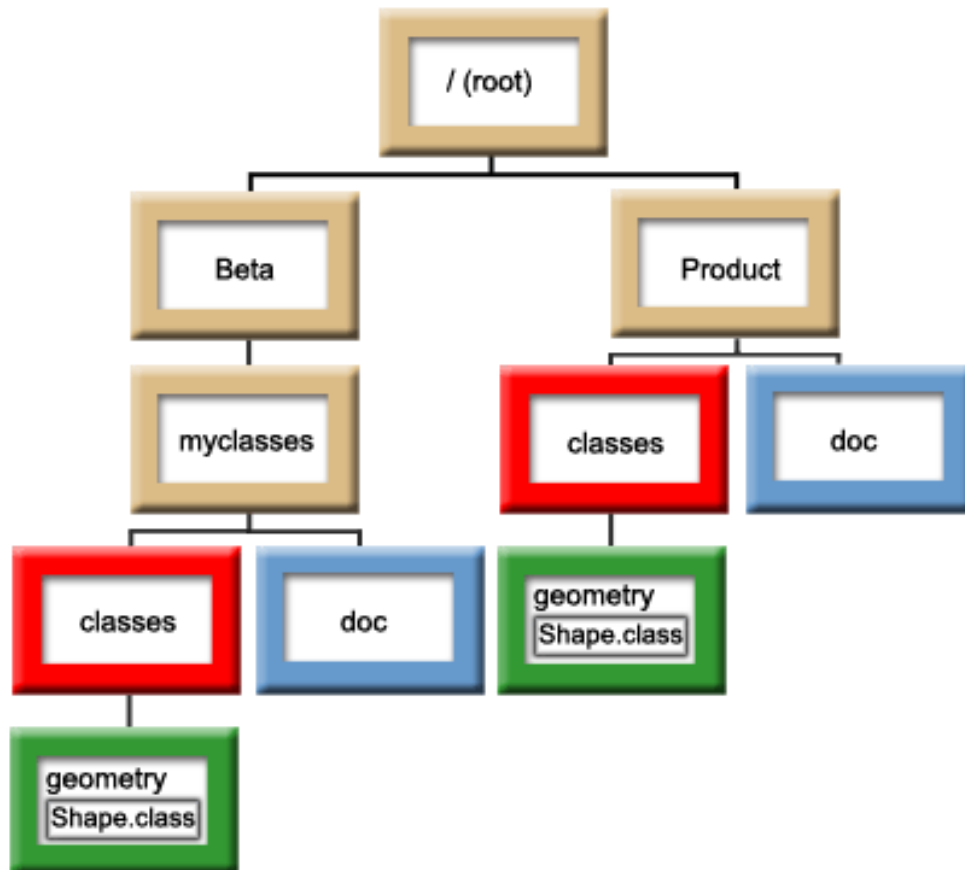
■ Class

- Lớp là tập hợp các đối tượng có cùng thuộc tính giống nhau.

- **Cú pháp:**

```
[Thuộc tính][Bổ từ truy xuất] class <Tên lớp>[extends Lớp cơ sở]
{
    // Biến
    // Thuộc tính
    // Phương thức
}
```

■ Class



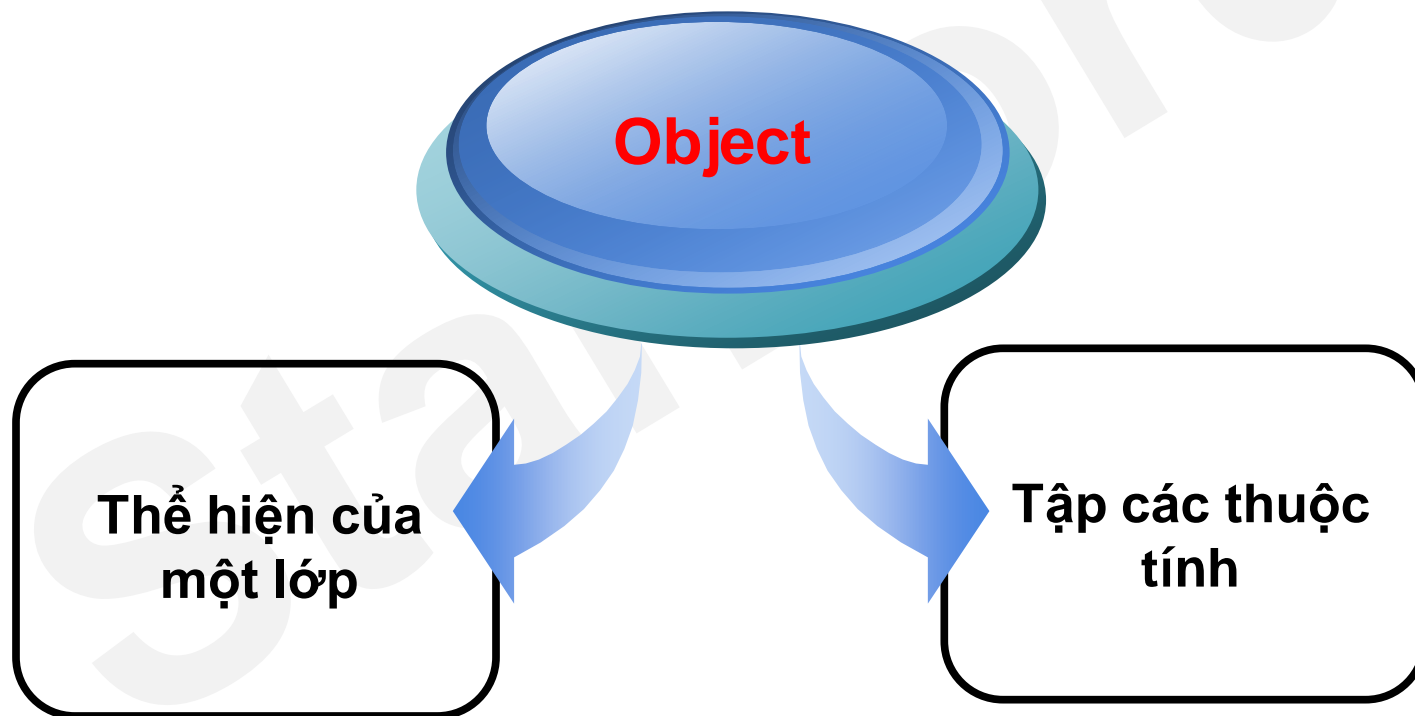
■ Class

- Ví dụ:

```
class Circle {  
    /** The radius of this circle */  
    double radius = 1.0; ← Data field  
  
    /** Construct a circle object */  
    Circle() {  
    }  
  
    /** Construct a circle object */  
    Circle(double newRadius) {  
        radius = newRadius;  
    } ← Constructors  
  
    /** Return the area of this circle */  
    double getArea() { ← Method  
        return radius * radius * Math.PI;  
    }  
}
```

■ Object

- Là một thể hiện cụ thể của lớp bao gồm tập hợp các thuộc tính mô tả đối tượng đó.



■ Object

- Khai báo một đối tượng
 - [Thuộc tính] [Bổ tử truy xuất] **[Tên lớp]** [Đặt tên đối tượng]
- Khởi tạo một đối tượng mới
 - [Đặt tên đối tượng] = **new** [Tên lớp]()
 - Sử dụng từ khóa new để khởi tạo một đối tượng mới.

Ví dụ:

```
TinhToan Obj1 = new TinhToan();
```

```
SinhVien Obj2 = new SinhVien();
```

```
Person Obj3 = new Person();
```

■ Object

■ Ví dụ:

```
/**
 *
 * @author DANG QUANG
 */
public class SinhVien
{
    private String _MaSV;
    private String _HoTen;
    private String _GioiTinh;
    private Date _NgaySinh;
    private String _QueQuan;

    public SinhVien()
    {
        _GioiTinh="Nam";
    }

    public void SetMaSV(String masv)
    {
        _MaSV = masv;
    }
}
```

HỌC ĐỂ LÀM VIỆC



■ Thuộc tính của đối tượng

- Là thông tin mô tả các đối tượng thuộc một lớp nào đó
- Cho phép người dùng khai báo để đọc (**read**) hoặc ghi (**write**) thông tin để mô tả đối tượng cụ thể.

Cú pháp khai báo:

```
private|public [Kieu_Du_Lieu] [Ten_bien];
```

Đọc thông tin:

```
public [Kieu_Du_Lieu] getTen_Ham { return [Kieu_Du_Lieu_Tra_ve];}
```

Ghi thông tin:

```
public [void] setTen_Ham([Kieu_Du_Lieu] value)  
{
```

```
    [Ten_bien] = value;}
```

■ Thuộc tính của đối tượng

■ Ví dụ:

```
private String _MaSV;
```

Đọc thông tin:

```
public String getMaSV  
{  
    return _MaSV;  
}
```

Ghi thông tin:

```
public [void] setMaSV(String masv)  
{  
    _MaSV = masv;  
}
```

Sử dụng từ khóa `this`, `static`

■ Từ khóa **this**

- Từ khóa **this** đại diện cho 1 class hay 1 object mà mình đang làm việc trên nó.

Ví dụ:

```
private int Nam = 0;  
public void SetYear(int Nam)  
{  
    this.Nam = Nam;  
}
```

■ Từ khóa static

■ Biến tĩnh:

- Là biến có từ khóa static trước kiểu dữ liệu của biến.
- Biến tĩnh chỉ được khởi tạo một lần trong bộ nhớ cho tất cả các thể hiện của đối tượng đó.

Ví dụ: `static string KeyName = "";`

■ Từ khóa static

■ Phương thức tĩnh:

- Là phương thức có từ khóa static trước kiểu dữ liệu trả về của phương thức đó khi khai báo phương thức.
- Trong thân của phương thức chỉ được phép truy xuất những thành viên tĩnh (static member).

Ví dụ:

```
public static TinhTong(int a, int b){  
    //Nội dung xử lý  
};
```


Exercises

“HỌC ĐỂ LÀM VIỆC”

Thank You !