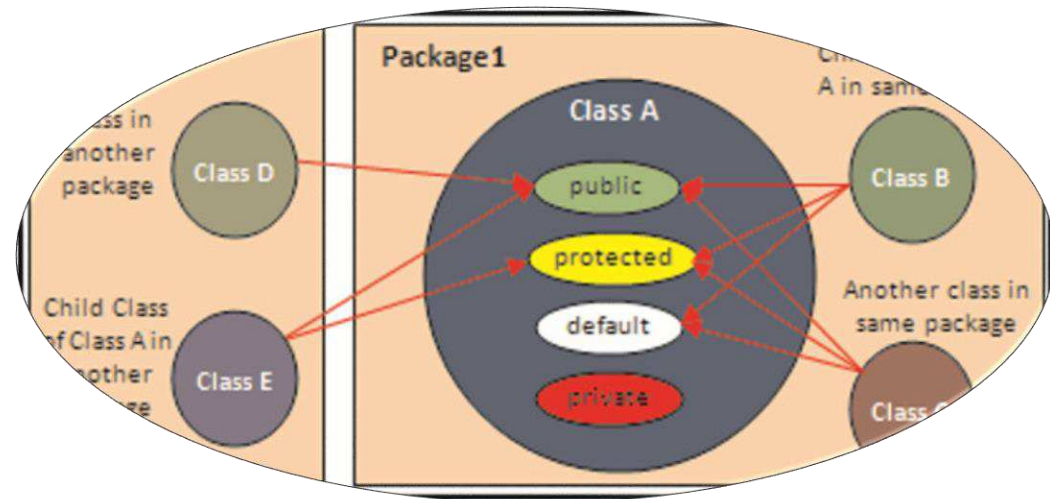
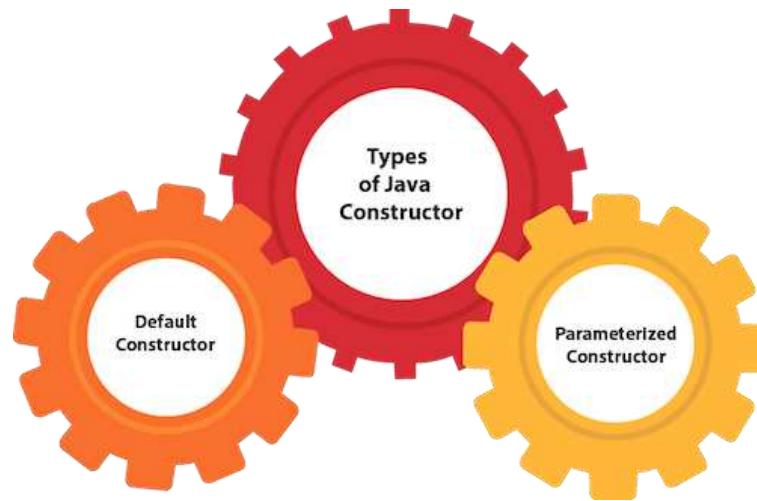


Session 06:

Constructors, Access Modifiers and Overloading Methods



1

Phương thức khởi tạo (Constructors)

2

Phạm vi truy cập (Access Modifiers)

3

Phương thức nạp chồng (Overloading Methods)

Phương thức khởi tạo (1)

Constructors

- **Constructor** là một phương thức đặc biệt được sử dụng để **khởi tạo thuộc tính** cho đối tượng
- Đặc điểm
 - Có tên trùng tên lớp
 - Không có kiểu trả về
 - Được gọi khi tạo đối tượng

```
Customer c1 = new Customer("1", "Kỳ", "HCM");  
Customer c2 = new Customer("2", "Minh", "HN");
```

```
public class Customer {  
    // Thuộc tính  
    int customerID;  
    String customerName;  
    String customerAddress;  
  
    // Phương thức khởi tạo (Constructor)  
    public Customer(int id, String name, String address) {  
        this.customerID = id;  
        this.customerName = name;  
        this.customerAddress = address;  
    }  
}
```

Phương thức khởi tạo (2)

Constructors

- Trong một lớp có thể định nghĩa nhiều constructor với danh sách tham số khác nhau
- Mỗi constructor cung cấp 1 cách tạo đối tượng
- Khi không khai báo constructor thì Java tự động cung cấp constructor mặc định (không tham số)

```
public class Customer {  
  
    // Phương thức khởi tạo (Constructor)  
    public Customer(String name, String address) {  
        this.customerName = name;  
        this.customerAddress = address;  
    }  
  
    public Customer(int id, String name, String address) {  
        this.customerID = id;  
        this.customerName = name;  
        this.customerAddress = address;  
    }  
}
```

Phương thức khởi tạo (3)

this keyword

- **this** được sử dụng để đại diện cho đối tượng hiện tại
- **this** được sử dụng trong lớp để tham chiếu tới các thành viên của lớp (properties, methods)

```
public class Customer {  
    // Thuộc tính  
    int customerID;  
    String name;  
    String customerAddress;  
  
    // Phương thức khởi tạo (Constructor)  
    public Customer(String name, String address) {  
        this.name = name;  
        customerAddress = address;  
    }  
}
```

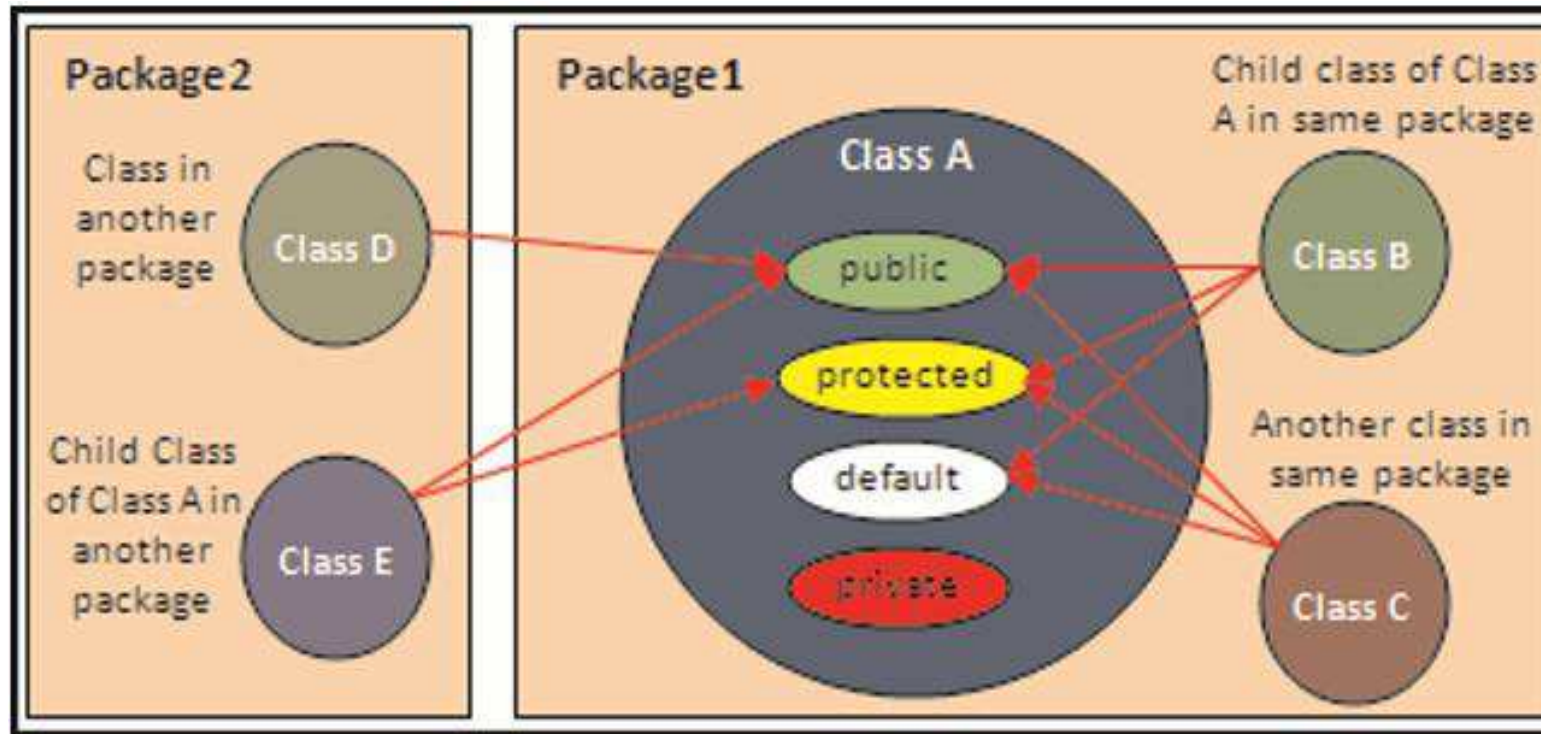
2

Phạm vi truy cập (Access Modifiers)

Phạm vi truy cập (1)

Access Modifiers

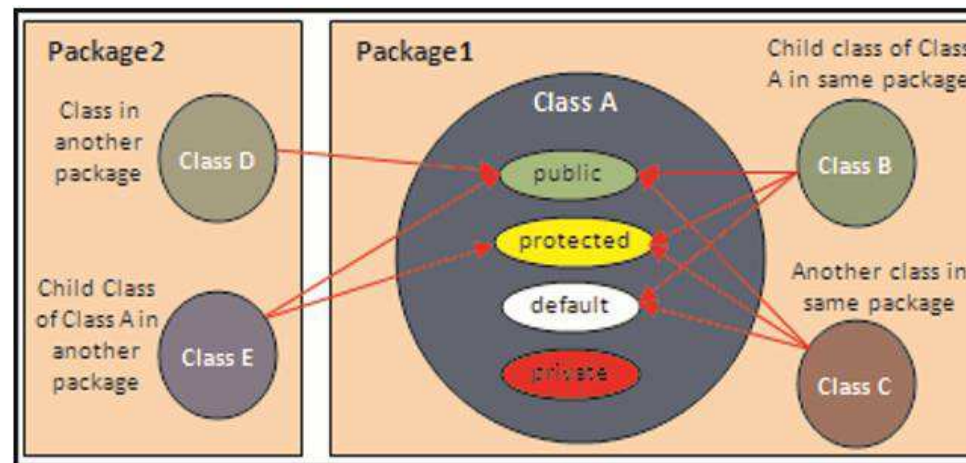
- Phạm vi truy cập được sử dụng để khai báo khả năng truy cập đến các thành viên của lớp từ lớp khác



Phạm vi truy cập (2)

Access Modifiers

- **private:** Chỉ được phép truy cập tại lớp khai báo
- **protected:** Được phép truy cập tại lớp khai báo và lớp con (lớp được kế thừa)
- **default:** Được truy cập ở những lớp cùng gói (package)
- **public:** Được truy cập ở tất cả các lớp

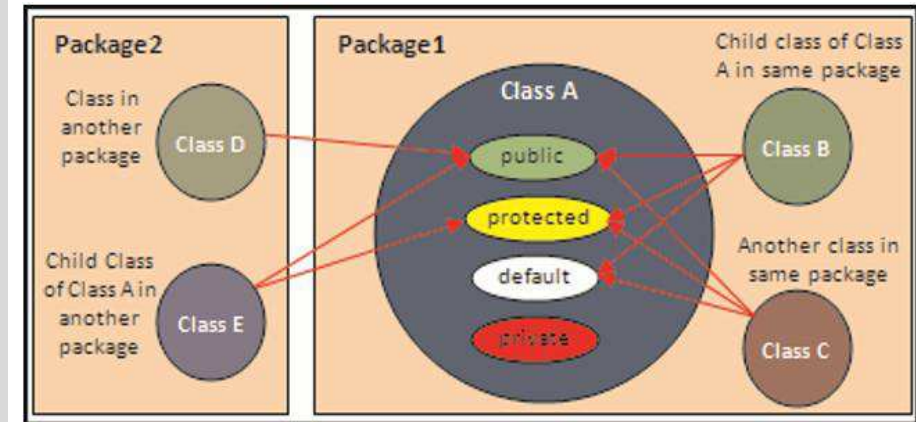


Phạm vi truy cập (3)

Access Modifiers

```
public class Customer {  
    // Thuộc tính  
    private int customerID;  
    public String name;  
    String customerAddress;  
    protected String email;  
  
    // Phương thức khởi tạo (Constructor)  
    public Customer(String name, String address) {  
        this.name = name;  
        customerAddress = address;  
    }  
}
```

Khai báo thuộc tính



3

Phương thức nạp chồng (Overloading Methods)

Phương thức nạp chồng (1)

Overloading Methods

- Xét trường hợp

```
public class Calculator {  
    public float sum(float n1, float n2) {  
        return n1 + n2;  
    }  
  
    public int sum(int n1, int n2) {  
        return n1 + n2;  
    }  
}
```

- Với lớp trên

```
Calculator c = new Calculator();  
int result1 = c.sum(10, 20);
```

```
float result2 = c.sum(10.2, 20.8);  
System.out.println(result1 + "," + result2);
```

Phương thức nạp chồng (2)

Overloading Methods

- Trong một lớp có thể có nhiều phương thức **trùng tên** nhưng **khác nhau về chữ ký** (kiểu dữ liệu và số lượng các tham số)

```
public class Calculator {  
    public float sum(float n1, float n2) {  
        return n1 + n2;  
    }  
}
```

```
    public int sum(int n1, int n2) {  
        return n1 + n2;  
    }  
}
```

```
    public int sum(int n1, int n2, int n3) {  
        return n1 + n2 + n3;  
    }  
}
```

1. Phương thức khởi tạo

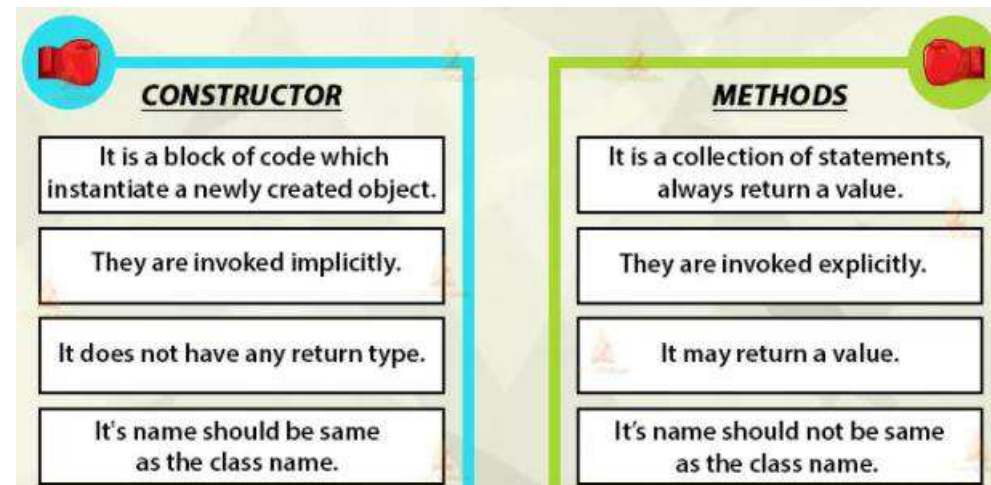
- Trùng tên với tên lớp
- Không có kiểu trả về
- Khởi tạo giá trị cho thuộc tính
- Được gọi khi tạo đối tượng

2. Phạm vi truy cập

- private
- Protected
- default
- public

3. Phương thức nạp chồng

- Nhiều phương thức trùng tên
- Khác nhau về chữ ký
 - Số lượng tham số
 - Kiểu dữ liệu





Thankyou!