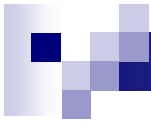




Lớp và đối tượng trong Java



Nội dung

- Định nghĩa lớp
- Thuộc tính
- Phương thức
- Kiểm soát truy cập
- Phương thức khởi tạo
- Thao tác với đối tượng



Tài liệu tham khảo

- *Thinking in Java*, chapter 1, 2
- *Java how to program*, chapter 8



Định nghĩa lớp

Lớp được định nghĩa bởi

```
class class_name {  
  
    ...  
}
```

Ví dụ:

```
class MyDate {  
  
}
```



Đối tượng

- Đối tượng được thao tác thông qua *tham chiếu*
 - Tham chiếu đóng vai trò gần giống như một con trỏ
- Đối tượng phải được tạo ra một cách tường minh bằng toán tử **new**

```
MyDate d;
```

```
d = new MyDate();
```

```
MyDate myBirthday = d;
```



Thuộc tính, phương thức và kiểm soát truy cập

```
class MyDate {  
    private int year, mon, day;  
    public int getYear() {  
        return year;  
    }  
    public boolean setYear(int y) {  
        ...  
    }  
    ...  
}
```



```
MyDate d = new MyDate();  
...  
d.year = 2005;    // compile error  
d.setYear(2005);  
System.out.println("Year=" + d.getYear());
```



Phương thức trùng tên (overload)

- Có thể định nghĩa các phương thức trùng tên, tuy nhiên phải phân biệt bởi danh sách tham số

```
class MyDate {  
    ...  
    public boolean setMonth(int m) { ...}  
    public boolean setMonth(String s) { ...}  
}
```

```
d.setMonth(9);  
d.setMonth("September");
```




Phương thức khởi tạo (constructor)

- Dữ liệu nên được khởi tạo trước khi sử dụng
 - lỗi khởi tạo là một trong các lỗi phổ biến
- Phương thức khởi tạo
 - là phương thức đặc biệt được gọi tự động sau khi tạo ra đối tượng
 - nhằm mục đích chính là khởi tạo cho các thuộc tính của đối tượng



Phương thức khởi tạo

- Có tên trùng với tên lớp
- Không nhận giá trị trả lại
- Mỗi khi đối tượng được tạo ra bởi toán tử `new`, hệ thống sẽ tự động gọi phương thức khởi tạo.
 - nếu không khai báo, hệ thống sẽ gọi constructor mặc định là một phương thức rỗng



Ví dụ: Constructor rỗng

```
class SayMsg {  
}
```

...

```
SayMsg msg = new SayMsg( ) ;
```



Ví dụ: Constructor mặc định

```
class SayMsg {  
    SayMsg( ) {  
        System.out.println( "Hello" );  
    }  
}  
  
...  
SayMsg msg = new SayMsg( );
```



Ví dụ:

```
class SayMsg {  
    SayMsg() {  
        System.out.println( "Hello" );  
    }  
    SayMsg(String s) {  
        System.out.println(s);  
    }  
}  
  
...  
SayMsg msg1 = new SayMsg( );  
SayMsg msg2 = new SayMsg( "Java" );
```



Ví dụ:

```
class SayMsg {  
    SayMsg(String s) {  
        System.out.println(s);  
    }  
}  
  
...  
  
SayMsg msg1 = new SayMsg();           // error  
SayMsg msg2 = new SayMsg(" ");
```



Copy constructor

- Khởi tạo đối tượng bằng một đối tượng khác

```
public class MyDate {  
    private int year, month, day;  
    public MyDate() {...}  
    public MyDate(MyDate d) {  
        year = d.year;  
        month = d.month;  
        day = d.day;  
    }  
  
    ...  
}
```

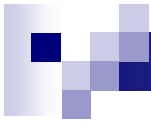


```
MyDate d = new MyDate( );  
d.setYear(2005);  
d.setMonth(9);  
d.setDay(12);  
MyDate openDay = new MyDate(d);  
MyDate dd = d;  
dd = new MyDate( );
```




Kiểm soát truy cập

```
public class MyDate {  
    private int year, month, day;  
    public MyDate() {...}  
    public MyDate(MyDate d) {  
        year = d.year; // year = d.getYear();  
        month = d.month;  
        day = d.day;  
    }  
    ...  
}
```



Bài tập

- Nghiên cứu, so sánh về tạo lớp đối tượng trong ngôn ngữ C++
- Tìm hiểu về phương thức hủy destructor trong C++



Tự thực hành

- Hoàn thiện lớp MyDate
 - các constructor
 - các phương thức get và set cần thiết
 - thêm phương thức dayOfWeek()
- Kiểm nghiệm tính đóng gói/che dấu thông tin