

## &lt; Lập trình hướng đối tượng Java: biểu diễn đối tượng bằng sơ đồ lớp



tuanlq7

Đơn Giản

 100 Điểm

Giới hạn kí tự: 3000

Tiếng Việt



89



## Bài tập

Bạn hãy tạo ra lớp `Student` trong file `Student.java` với thông tin giống như sau:

Student
-name: String -age: int
+Student() +getInformation(): void +display(): void

Giải thích: hình trên là mô tả của lớp `Student` với các thuộc tính và phương thức giống như sau:

- `name` là thuộc tính kiểu `String` với phạm vi truy cập `private`.
- `age` là thuộc tính kiểu `int` với phạm vi truy cập `private`.
- `Student()` là phương thức khởi tạo không tham số có phạm vi truy cập là `public`.
- `getInformation()` và `display()` là hai phương thức có phạm vi truy cập `public`.

Trong thực tế, bạn không nên viết code của hai class trong cùng 1 file giống như các bài tập ở phần trước ( `class Student` và `class Entry` được viết trong cùng 1 file). Với mỗi một class bạn nên tạo ra 1 file riêng cho nó. Ví dụ: file `Student.java` sẽ chứa code của lớp `Student`, file `Entry.java` sẽ chứa code của lớp `Entry`, ...

Quay lại phần bài tập, cửa sổ code trong bài này chính là file `Student.java`, trong file này bạn cần tạo ra lớp `Student` với đầy đủ các thuộc tính và phương thức giống như đã mô tả (bạn chỉ cần khai báo các thuộc tính và phương thức, không cần viết code bên trong phương thức).

Lưu ý: khi nộp bài lên hệ thống, bạn không cần khai báo package.

## Lý thuyết

Hình minh họa về lớp `Student` bên trên chính là một **sơ đồ lớp**. Ở các bài trước bạn đã được làm quen với sơ đồ này nhưng ở bài này bạn sẽ được hiểu sâu hơn về nó.

### Sơ đồ lớp

mô tả của các lớp, mối quan hệ giữa các đối tượng trong hệ thống (mối quan hệ giữa các đối tượng bạn sẽ được học trong các bài tiếp theo).

Một ký hiệu lớp sẽ có 3 phần là Class name (tên lớp), attributes (danh sách các thuộc tính) và operations (các thủ tục, để đơn giản thì bạn hãy coi thủ tục chính là phương thức). Một lớp trong biểu đồ lớp sẽ được mô tả giống như sau:

Tên lớp
-Thuộc tính 1 -Thuộc tính 2 -...
+Phương thức 1() +Phương thức 2() +...()

Trong sơ đồ lớp, bạn có thể mô tả phạm vi truy cập của các thuộc tính, phương thức bằng cách đặt các ký tự `-`, `~`, `#`, `+` trước các thuộc tính và phương thức.

- Ký tự `'-'` sẽ tương ứng với phạm vi truy cập `private`.
- Ký tự `'~'` sẽ tương ứng với phạm vi truy cập `default`.
- Ký tự `'#'` sẽ tương ứng với phạm vi truy cập `protected`.
- Ký tự `'+'` sẽ tương ứng với phạm vi truy cập `public`.

Ví dụ lớp `Student` giống như sau:

```
class Student {  
    private int id;  
    String name;  
    protected int age;  
    public double gpa;  
  
    public Student() {  
  
    }  
  
    void getInformation() {  
  
    }  
  
    public void display() {  
  
    }  
}
```

Sẽ được biểu mô tả bằng sơ đồ lớp giống như sau (bạn hãy để ý các ký tự `-`, `~`, `#`, `+`):

Student
-id: int ~name: String #age: int +gpa: double
+Student() ~getInformation(): void +display(): void

Lưu ý: khi một phương thức không được định nghĩa kiểu trả về và tên phương thức trùng với tên lớp thì đây chính là phương thức khởi tạo. Như trong sơ đồ lớp trên thì `+Student()` chính là phương thức khởi tạo `public`.

Để chắc chắn rằng bạn đã hiểu về sơ đồ lớp, hãy xem ví dụ tiếp theo về lớp `Product` (lớp dùng để mô tả các đối tượng sản phẩm):

```
class Product {  
    public int id;  
    String name;  
    protected double price;  
    private String description;  
  
    Product() {  
  
    }  
  
    public void getInformation() {  
  
    }  
  
    public void display() {  
  
    }  
}
```

Lớp này sẽ được mô tả bằng sơ đồ lớp giống như sau:



~name: String #price: double -description: String
~Product() +getInformation(): void +display(): void

Đọc tới đây bạn đã hiểu được cách mô tả các lớp bằng ký hiệu lớp, hãy quay lại phần bài tập và làm thử.

## Hướng dẫn

Code mẫu:

```
public class Student {  
    private String name;  
    private int age;  
  
    public Student() {  
  
    }  
  
    public void getInformation() {  
  
    }  
  
    public void display() {  
  
    }  
}
```



Java



Chạy thử



Nộp bài(9)

Student.java



```
1 public class Student {  
2     private String name;  
3     private int age;  
4 }
```