



< Lập trình hướng đối tượng trong java

🕒 20 phút

❓ 1 câu hỏi

✍️ Không giới hạn lượt làm lại

🕒 2 phút

Bài tập

Bạn hãy cho biết kết quả khi chạy chương trình sau:

```
import java.util.Scanner;

public class Entry {
    public static void main(String[] args) {
        Student s = new Student();
        s.name = "Codelearn";
        s.display();
    }
}

class Student {
    String name;

    private void display() {
        System.out.println("Name: " + name);
    }
}
```

Lý thuyết

Nhìn vào đoạn code khai báo phương thức `display()` bạn sẽ thấy có từ khóa `private`. Với các bài trước bạn đã quen với việc khai báo một phương thức bằng từ khóa `public`, vậy sự khác nhau giữa `public` và `private` là gì? Bài này bạn sẽ giúp bạn hiểu rõ.

Phạm vi truy cập

Phạm vi truy cập là khái niệm để xác định xem một biến hay một phương thức có được truy xuất từ bên ngoài hay không. Trong Java có 4 loại phạm vi truy cập là `private`, `default`, `protected`, `public`. Xem bảng minh họa dưới đây để có cái nhìn tổng quan về phạm vi truy cập:

| Phạm vi truy cập | Truy cập bên trong class? | Truy cập bên trong package? | Truy cập bên ngoài package | Truy cập bên ngoài class và |
|------------------|---------------------------|-----------------------------|----------------------------|-----------------------------|
|------------------|---------------------------|-----------------------------|----------------------------|-----------------------------|



| | | | | |
|-----------|---|---|---|---|
| private | Y | N | N | N |
| default | Y | Y | N | N |
| protected | Y | Y | Y | N |
| public | Y | Y | Y | Y |

Phạm vi truy cập private

Đây là phạm vi truy cập hạn chế nhất, tất cả các thuộc tính và phương thức sẽ chỉ được truy xuất từ bên trong lớp đó. Ví dụ nếu bạn truy xuất một thuộc tính hoặc phương thức được khai báo với từ khóa `private` từ một lớp khác thì chương trình sẽ báo lỗi giống như sau:

```
class Student {
    private String name;
}

public class Entry {
    public static void main(String[] args) {
        Student s = new Student();
        s.name = "Codelearn";
    }
}
```

The field Student.name is not visible
2 quick fixes available:
[Change visibility of 'name' to 'package'](#)
[Create getter and setter for 'name'...](#)

Để chương trình trên có thể chạy được bạn cần thay đổi phạm vi truy cập của thuộc tính `name` trong lớp `Student` từ `private` sang `default`, `protected` hoặc `public`.

Phạm vi truy cập mặc định (default)

Khi khai báo một thuộc tính hoặc một phương thức mà không khai báo phạm vi truy cập thì chương trình sẽ hiểu là bạn đang khai báo với phạm vi truy cập mặc định. Các biến và phương thức có phạm vi truy cập mặc định chỉ có thể được truy xuất trong cùng một package. Ví dụ trong cùng package OOP bạn có thể truy xuất tới thuộc tính của lớp `Student` từ lớp `Entry` giống chương trình sau (kéo dài kích thước của cửa sổ bài tập để xem rõ hơn):



```

3 public class Student {
4     String name;
5 }
6

3 public class Entry {
4     public static void main(S
5         Student s = new Stude
6         s.name = "Codelearn";
7         System.out.println(s.
8     }
9 }
10

```

Problems @ Javadoc Declaration Console

<terminated> Entry [Java Application] C:\Program Files\Java\jdk-14\bin\javaw.exe (Apr 1, 2020, 12:04:05 AM – 12:04:05 AM)
Codelearn

Nhưng nếu 2 lớp này khác package thì chương trình sẽ báo lỗi:

```

Student.java
1 package package1;
2
3 public class Student {
4     String name;
5 }
6

Entry.java
1 package OOP;
2
3 import package1.Student;
4
5 public class Entry {
6     public static void main(
7         Student s = new Stud
8         s.name = "Codelearn"
9         Sy
10    }
11 }
12

```

The field Student.name is not accessible in package OOP. 2 quick fixes available:

- Change visibility of 'name' to public
- Create getter and setter for 'name'

Trong trường hợp trên, để chương trình chạy được bạn cần thay đổi phạm vi truy cập từ **default** sang **public**.

Lưu ý: khi muốn sử dụng một lớp từ package khác thì bạn cần **import** lớp này vào bằng câu lệnh **import <tên_package>.<tên_lớp>;** giống như ví dụ trên.

Phạm vi truy cập protected

Đây là phạm vi truy cập tương tự với phạm vi truy cập **default** nhưng ít hạn chế hơn ở chỗ các thuộc tính và phương thức có phạm vi truy cập này có thể được truy xuất từ một lớp con của package khác (bạn sẽ hiểu rõ hơn ở các bài về tính kế thừa).



Để tìm được truy xuất và các thuộc tính của package này.

Đọc tới đây bạn đã hiểu về các phạm vi truy cập trong Java và đã có thể làm được bài này.

Hướng dẫn

Phương thức `display()` trong lớp `Student` được khai báo với từ khóa `private` nên không thể được truy xuất từ phương thức `main()` của lớp `Entry`. Do đó đáp án của bài này là `"Compile Time Error"` (lỗi biên dịch).

[Làm bài](#)[← Bài học trước](#)[Bài tiếp theo →](#)

Giáo trình

Bình luận

[Gửi bình luận](#)