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:

```
class Student {
    private String name;
    private int age;

    public static int numberOfStudents;

    public Student(String name, int age) {
        this.name = name;
        this.age = age;
        numberOfStudents++;
    }
}

class Entry {
    public static void main(String[] args) {
        Student s1 = new Student("Manh", 19);
        Student s2 = new Student("Trung", 19);
        Student s3 = new Student("Kien", 19);
        System.out.println(Student.numberOfStudents);
    }
}
```

Lý thuyết

Chắc bạn cũng đã ít nhiều sử dụng lớp `Math` để thực hiện việc tính toán. Bạn có bao giờ thắc mắc là tại sao lại có thể sử dụng được các phương thức và các biến của lớp `Math` mà không cần khởi tạo đối tượng. Ví dụ:

```
class Entry {
    public static void main(String[] args) {
        // Hiển thị ra màn hình căn bậc 2 của 5
```

```
        // Hiển thị ra màn hình số lớn nhất trong 2 số
        System.out.println("Max(345, 43) = " + Math.max(345,  43));
    }
}
```

Kết quả khi chạy chương trình:

```
Sqrt(5) = 2.23606797749979
Pi = 3.141592653589793
Max(345, 43) = 345
```

Qua bài này bạn sẽ hiểu được biến static và phương thức static.

Biến static

Biến static là biến mà bạn có thể sử dụng mà không cần phải khởi tạo đối tượng. Cú pháp để khai báo và sử dụng biến static rất đơn giản, bạn hãy xem ví dụ sau:

```
class Counter{
    // Khai báo biến static có tên là count
    public static int count;
}

class Entry {
    public static void main(String[] args) {
        Counter c = new Counter();
        c.count = 7;
        System.out.println(c.count);
    }
}
```

Kết quả khi chạy chương trình:

```
7
```

Ngoài việc sử dụng mà không cần phải khởi tạo đối tượng thì biến `static` còn có đặc điểm nữa là biến `static` được chia sẻ bởi tất cả các đối tượng trong chương trình (giá trị của biến `static` là giống nhau ở tất cả các đối tượng). Bạn hãy xem ví dụ về biến thông thường và biến static sau đây để hiểu rõ tính chất này:

```
class Counter {
    int count;
```

```
    }
}

class Entry {
    public static void main(String[] args) {
        Counter c1 = new Counter();
        Counter c2 = new Counter();
        Counter c3 = new Counter();
    }
}
```

Kết quả khi chạy chương trình:

```
1
1
1
```

Kết quả này chắc bạn cũng đoán được (do `count` là thuộc tính riêng của mỗi đối tượng nên kết quả sẽ là 3 số 1). Nhưng nếu biến `count` là biến `static` thì tất cả các đối tượng này đều sẽ dùng chung 1 biến `count`:

```
class Counter {
    static int count;

    public Counter() {
        count++;
        System.out.println(count);
    }
}

class Entry {
    public static void main(String[] args) {
        Counter c1 = new Counter();
        Counter c2 = new Counter();
        Counter c3 = new Counter();
    }
}
```

Kết quả khi chạy chương trình:

```
3
```

Chính vì 2 tính chất này nên biến `static` sẽ thường được dùng để lưu thông tin chung cho tất cả các đối tượng và lưu các hằng số (giống như biến `PI` trong thư lớp `Math`).

Lưu ý: biến được khai báo với từ khóa `static` không được coi là thuộc tính do nó không thuộc đối tượng nào.

Phương thức static

Tương tự với biến `static`, phương thức `static` cũng được khai báo với từ khóa `static` và được sử dụng mà không cần tạo ra báo đối tượng. Ví dụ hàm `max()` của lớp `Math` là một phương thức `static` và trông giống như sau:

```
class Math{
    public static int max(int a, int b) {
        return (a >= b) ? a : b;
    }
}

class Entry {
    public static void main(String[] args) {
        System.out.println(Math.max(3, 6));
    }
}
```

Kết quả khi chạy chương trình:

```
6
```

Tới đây chắc bạn cũng đã hiểu được lớp `Math` là lớp chứa các biến và phương thức `static`.

Một số tính chất của phương thức `static`:

- Phương thức `static` có để được gọi mà không cần phải khởi tạo đối tượng.
- Trong cùng một lớp, phương thức `static` chỉ có thể gọi tới phương thức `static` khác, không thể gọi tới phương thức không phải là `static`.
- Trong cùng một lớp, phương thức `static` không thể gọi tới các thuộc tính không phải là static.

Đọc tới đây bạn đã hiểu về biến static và phương thức static, hãy quay lại phần bài tập và chọn đáp án đúng.

Làm bài