

Like



Share



Subscribe

thuydung

ProIT4All



Giới thiệu

Thư viện xử lý số ngẫu nhiên



Fan page: <http://facebook.com/Proit4All>



Thư viện xử lý số ngẫu nhiên

- ❑ Math hay `java.lang.Math` là một lớp cung cấp hàm `random()` để lấy số ngẫu nhiên từ 0 đến 1
- ❑ Để tạo một số ngẫu nhiên:
 - ❖ sử dụng class `java.util.Random` đã được tích hợp trong JDK.



Thư viện xử lý số ngẫu nhiên

□ **Cú pháp:** tạo mới 1 đối tượng Random:

```
import java.util.Random;
```

```
public class TênClass {
```

```
    Random rd = new Random();
```

```
}
```




Các phương thức của Random

❑ **nextInt():** sẽ trả về 1 số ngẫu nhiên có kiểu int.

➤ **Cú pháp:**

```
Random rd = new Random();
```

```
int number = rd.nextInt(); // trả về 1 số nguyên bất kỳ
```



Các phương thức của Random

❑ **nextInt(bound):** sẽ trả về 1 số ngẫu nhiên có kiểu int từ 0 đến bound -1

➤ *Lưu ý: bound > 0, nếu nhỏ hơn 0 chương trình sẽ trả về exception thuộc lớp `java.lang.IllegalArgumentException`*

➤ **Cú pháp:**

```
Random rd = new Random();
```

```
int number = rd.nextInt(10); // trả về 1 số nguyên bất kỳ từ  
0 đến 9
```



Thư viện xử lý số ngẫu nhiên

□ Phương thức `nextFloat()`

❖ Sẽ trả về 1 số ngẫu nhiên có kiểu float nằm trong phạm vi `[0.0f...1.0f)`.

❖ Cú pháp:

```
Random rd = new Random();
```

```
float floatNumber = rd.nextFloat();
```

```
// trả về 1 số bất kỳ có kiểu là float
```




Thư viện xử lý số ngẫu nhiên

□ Phương thức `nextDouble()`

❖ Phương thức `nextDouble()` sẽ trả về 1 số ngẫu nhiên có kiểu `double` nằm trong phạm vi `[0.0d...1.0d)`.

❖ Cú pháp

```
Random rd = new Random();
```

```
// trả về 1 số bất kỳ có kiểu là double
```

```
double doubleNumber = rd.nextDouble();
```



Thư viện xử lý số ngẫu nhiên

❑ Phương thức nextLong()

❖ Phương thức nextLong() sẽ trả về 1 số ngẫu nhiên có kiểu long.

❑ Cú pháp

```
Random rd = new Random(); // khai báo 1 đối tượng Random  
// trả về 1 số bất kỳ có kiểu long  
long longNumber = rd.nextLong();
```




Thư viện xử lý số ngẫu nhiên

❑ Phương thức `nextBoolean()`

❖ Phương thức `nextBoolean()` sẽ trả về 1 biến ngẫu nhiên có giá trị là `true` hoặc `false`.

❖ Cú pháp

```
Random rd = new Random(); // khai báo 1 đối tượng Random  
// trả về 1 biến bool có giá trị là true hoặc false.  
boolean bool = rd.nextBoolean();
```



Like



Share



Subscribe

Fan page: <http://facebook.com/Proit4All>

DEMO



Tìm hiểu thư viện xử lý số ngẫu nhiên



Java™





Thư viện xử lý số ngẫu nhiên

Ngoài ra, nếu đang làm việc Java 8 trở lên, có thể sử dụng lớp `java.util.concurrent.ThreadLocalRandom` để tạo số ngẫu nhiên cục bộ cho từng luồng một cách hiệu quả hơn trong môi trường đa luồng. Đây là cách sử dụng nó:

```
import java.util.concurrent.ThreadLocalRandom;  
class GenerateRandom {  
    public static void main( String args[] ) {  
        // Generate random integers  
        int int_random =  
            ThreadLocalRandom.current().nextInt();  
        // Print random integers  
        System.out.println("Random Integers: " +  
            int_random);  
    }  
}
```




Thư viện xử lý số ngẫu nhiên

Lấy số ngẫu nhiên từ n đến m

```
public static void main(String args[]) {  
    Random rd = new Random();  
    //lấy số ngẫu nhiên từ n đến m  
    int n = 5;  
    int m = 15;  
    Random random = new Random();  
    int randomInRange = random.nextInt(m - n + 1) + n;  
    System.out.println("Số ngẫu nhiên trong khoảng từ "  
        + n + " đến " + m + ": " + randomInRange);  
}
```

Like



Share



Subscribe

thuydung

ProIT4All



Giới thiệu

Thư viện xử lý số ngẫu nhiên



Fan page: <http://facebook.com/Proit4All>

thuydung

ProIT4All

Fan page: <http://facebook.com/Proit4All>



THANK YOU

<http://youtube.com/@AnhNguyenNgoc>



Java™

