UNIVERSIDADE TÉCNICA DO ATLÂNTICO | CAMPUS DO MAR

Resumo do método random java

Programação Orientado aos objetos |5 abril de 2023

Docente:
Paulo Silva

Discente:
José Carlos Costa

Classe random é usada para gerar um fluxo de números pseudoaleatórios. A classe usa uma semente de 48 bits, que é modificada usando uma fórmula congruente linear. Se duas instâncias de Random forem criadas com a mesma semente e a mesma sequência de chamadas de método for feita para cada uma, elas gerarão e retornarão sequências idênticas de números.

A interface RandomGenerator foi projetada para fornecer um protocolo comum para objetos que geram sequências aleatórias ou pseudoaleatórias de números ou valores booleanos. Essa sequência pode ser obtida invocando repetidamente um método que retorna um único valor escolhido de forma pseudoaleatória ou invocando um método que retorna um fluxo de valores escolhidos de forma pseudoaleatória.

Idealmente, dado um intervalo de valores implícita ou explicitamente especificado, cada valor seria escolhido independentemente e uniformemente desse intervalo. Na prática, pode-se ter que se contentar com alguma aproximação à independência e uniformidade.

Para valores int, long e boolean, se não houver especificação explícita de intervalo, o intervalo incluirá todos os valores possíveis do tipo. Para valores float e double, primeiro um valor é sempre escolhido uniformemente do conjunto de 2^w valores entre 0,0 (inclusivo) e 1,0 (exclusivo), onde w é 23 para valores float e 52 para valores double; então, se um intervalo explícito foi especificado, o número escolhido é dimensionado computacionalmente e convertido para parecer ter sido escolhido de maneira aproximadamente uniforme a partir desse intervalo explícito.

Cada método que retorna um fluxo produz um fluxo de valores, cada um dos quais é escolhido da mesma maneira que para um método que retorna um único valor pseudo-aleatoriamente escolhido.

Métodos:

Método construtor random () que cria um gerador de números aleatórios usando uma única longsemente esta semente é o valor inicial do estado interno do gerador de números, a invocação:

```
new Random(seed)
```

ou

```
Random rnd = new Random();

rnd.setSeed(seed);
```

doubles() Retorna um fluxo efetivamente ilimitada de double valores pseudoaleatórios, cada um entre zero e um :

doubles(double randomNumberOrigin, double randomNumberBound) Retorna um fluxo efetivamente ilimitado de double valores pseudoaleatórios, cada um em conformidade com a origem fornecida e o limite.

doubles (long streamSize) Retorna um fluxo produzindo o stream Sizenúmero fornecido de double valores pseudoaleatórios, cada um entre zero e um.

doubles(long streamSize,double randomNumberOrigin, double randomNumberBound) Retorna um fluxo produzindo o stream Sizenúmero fornecido de double valores pseudoaleatórios, cada um em conformidade com a origem

ints() Retorna um fluxo efetivamente ilimitado de int valores pseudoaleatórios.

ints(int randomNumberOrigin, int randomNumberBound) Retorna um fluxo efetivamente ilimitado de int valores pseudoaleatórios, cada um em conformidade com a origem fornecida

ints(long streamSize) Retorna um fluxo produzindo o stream Sizenúmero fornecido de int valores pseudoaleatórios.

longs() Retorna um fluxo efetivamente ilimitado de long valores pseudo-aleatórios.

longs(long streamSize) Retorna um fluxo produzindo o stream Sizenúmero fornecido de longvalores pseudo-aleatórios.

longs(long randomNumberOrigin, long randomNumberBound) Retorna um fluxo efetivamente ilimitado de long valores pseudoaleatórios, cada um em conformidade com a origem fornecida e o limite.

longs(long streamSize, long randomNumberOrigin, long randomNumberBound) Retorna um stream produzindo o número dado streamSizede pseudorandom long, cada um em conformidade com a origem dada

next(int bits) Gera o próximo número pseudoaleatório.

nextBoolean() Retorna o próximo valor pseudoaleatório distribuído uniformemente booleanda sequência desse gerador de números aleatórios.

nextBytes(byte[] bytes) Gera bytes aleatórios e os coloca em uma matriz de bytes fornecida pelo usuário.

nextDouble() Retorna o próximo valor pseudoaleatório distribuído uniformemente double entre 0.0e 1.0a partir desta sequência do gerador de números aleatórios.

nextGaussian() Retorna o próximo valor distribuído pseudoaleatório gaussiano ("normalmente") double com média 0.0e desvio padrão 1.0da sequência deste gerador de números aleatórios.

setSeed(long seed) Define a semente deste gerador de números aleatórios usando uma única long semente.