

Rastgele Sayı Üretilmesi ve Random Sınıfı

Rassal sayılar rassal süreçlerin yardımıyla üretilebilirler. Örneğin bir torbadan pul çekmek, bir zarı atmak gibi süreçlerde rastgele sayılar üretilir. Ancak bilgisayar rassal sayıları böyle rassal olaylar sonucunda üretemez. Aritmetik hesaplarla elde eder. Bu nedenle bilgisayarın elde ettiği bu rassal sayılara “yapay rassal sayılar (pseudo random numbers)” denilmektedir.

Olasılık bir teorik limit değeridir. Bir parayı attığımızda yazı gelme olasılığı 0.5 dir. Ancak parayı 10 kez attığımızda 5 kere yazı geleceği garanti değildir. Fakat parayı sonsuz kez atarsak oran 0.5 olacaktır. Zaten olasılık teorisi çok fazla kez deneylenen süreçlerden hareketle geliştirilmiştir.

Java’da rassal sayı üretmek için iki yol vardır:

- java.util paketi içerisindeki Random sınıfını kullanmak
- Math sınıfı içerisindeki random metodunu çağırmak

Random Sınıfı

Random sınıfı java.util paketi içerisinde bildirilmiş bir sınıftır. Sınıfın nextXXX metotları ile rastgele sayılar elde edilebilir. Sınıfın nextInt metotları int türden rastgele sayı döndürür.

```
int nextInt()  
  
int nextInt(int max)
```

Sınıfın parametresiz nextInt metodu [-2147483648, 2147483647] aralığında rastgele sayı döndürür. Sınıfın int parametrelili nextInt metodu [0, max) aralığında rastgele sayı döndürür. nextBoolean metodu true ya da false değerlerinden birini döndürür.

Sınıf Çalışması: Bir paranın yazı gelme olasılığını yaklaşık olarak simüle eden programı yazınız.

Çözüm:

```
package csd;  
  
class App {  
    public static void main(String[] args)  
    {  
        java.util.Random r = new java.util.Random();  
  
        long nThrow = 100000000;  
        long nTail = 0;  
  
        System.out.println("Paralar atılıyor");  
        for (long i = 0; i < nThrow; ++i)  
            nTail += r.nextInt(2);  
  
        double probTail = (double)nTail / nThrow;  
  
        System.out.printf("Yazı Gelme Olasılığı~%f%n", probTail);  
    }  
}
```

Sınıf Çalışması: İki zar atıldığında çift (iki zar da aynı) gelme olasılığını simüle eden programı yazınız.

Çözüm:

```
package csd;
```

```

class App {
    public static void main(String [] args)
    {
        java.util.Random r = new java.util.Random();

        long n = 1000000;
        long count = 0;

        for (long i = 0; i < n; ++i) {
            int dice1 = r.nextInt(6) + 1;
            int dice2 = r.nextInt(6) + 1;

            if (dice1 == dice2)
                count++;
        }

        System.out.printf("p=%f%n", (double)count / n);
    }
}

```

Sınıf Çalışması: [1, 99] aralığında rastgele 10 adet sayı üreten programı yazınız.

Çözüm:

```

package csd;

class App {

    public static void main(String[] args)
    {
        java.util.Random r = new java.util.Random();

        for (int i = 0; i < 10; ++i)
            System.out.println(r.nextInt(99) + 1); //[1, 99] aralığı
    }
}

```

Sınıfın nextDouble() metodu ile [0.0, 1.0) aralığında double türden sayı döndürür. Örneğin:

```

package csd;

class App {
    public static void main(String[] args)
    {
        java.util.Random r = new java.util.Random();

        for (int i = 0; i < 10; ++i)
            System.out.printf("%f%n", r.nextDouble());
    }
}

```

Random sınıfının nextFloat() metodu [0.0, 1.0) aralığında rastgele sayı döndürür. Örneğin:

```

package csd;

class App {
    public static void main(String[] args)
    {
        java.util.Random r = new java.util.Random();

        for (int i = 0; i < 10; ++i)
            System.out.printf("%f%n", r.nextFloat());
    }
}

```

```
}
```

Random sınıfının `nextBoolean()` metodu `true` ya da `false` değerini üretir. Örneğin:

```
package csd;

class App {
    public static void main(String [] args)
    {
        java.util.Random r = new java.util.Random();

        for (int i = 0; i < 10; ++i) {
            boolean val = r.nextBoolean();

            System.out.println(val);
        }
    }
}
```

Random sınıfının maalesef istenilen bir aralıkta rastgele sayı üreten bir metodu yoktur. Bunun için örneğin `[min, max]` aralığında `int` türden rastgele sayı üreten aşağıdaki gibi bir metot yazılabilir:

```
public static int getInt(java.util.Random r, int min, int max)
{
    return r.nextInt(max - min + 1) + min;
}
```

Benzer şekilde `[min, max)` aralığında `int` türden rastgele sayı üreten bir metot da aşağıdaki gibi yazılabilir.

```
public static int getInt(java.util.Random r, int min, int max)
{
    return r.nextInt(max - min) + min;
}
```

Random sınıfının `setSeed()` metodu rasgele sayı üretimi sırasında kullanılacak tohum değerini belirlemekte kullanılır. Örneğin:

```
package csd;

class App {
    public static void main(String [] args)
    {
        java.util.Random r = new java.util.Random();

        r.setSeed(100);

        for (int i = 0; i < 10; ++i) {
            int val = r.nextInt(101);

            System.out.println(val);
        }
    }
}
```

Bir Random nesnesi `new Random()` şeklinde yaratıldığında tohum değeri zamana bağlı olarak verilmiş olur. Dolayısıyla, aşağıdaki program her çalıştırıldığında farklı bir dizilim elde edilecektir. Örneğin:

```
package csd;

class App {
```

```
public static void main(String [] args)
{
    java.util.Random r = new java.util.Random();

    for (int i = 0; i < 10; ++i) {
        int val = r.nextInt(101);

        System.out.println(val);
    }
}
```