숙제: 표준정규분포 확률변량 생성

표준 정규분포에 따르는 확률변수 x의 PDF (probability density function)는 다음과 같이 정의된다.

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, -\infty < x < \infty$$

CDF는 PDF를 적분하여 구할 수 있다.

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$$

위 식에 의해 CDF의 정확한 값을 구하는 것은 어렵기 때문에 역변환(inverse transformation) 방법에 의해 랜덤의 확률변량을 구하는 것은 효과적이지 못하다. 대신에 Box-Muller 방법을 보편적으로 사용한다.

표준 정규분포에 따르는 랜덤의 확률변량을 구하는 또 다른 방법은 중심극한정리(central limit theorem)를 이용하는 것이다. n 개의 $U(0,1)$ 에 따르는 난수를 생성하여 이들을 더하면 이 값은 n 이 클수록 중심 극한 정리(central limit theorem)에 의해 정규분포에 근사한다. 이 성질을 이용하여 정규분포 $N(\mu, \sigma^2)$ 를 갖는 확률변수를 근사적으로 구할 수 있다. 확률변수 X 를 다음과 같다고 하자.

$$X = \sum_{i=1}^{n} U_i$$

X 는 근사적으로 평균 $n/2$, 분산 $n/12$ 을 갖는 정규분포에 따른다. 특히 n 이 12라면 분산의 값이 1로 간단해지고, X 는 근사적으로 N(6, 1)에 따른다. 이를 표준정규분포로 변환하면 다음과 같다.

$$Z = \frac{x - \mu}{\sigma} = \sum_{i=1}^{12} U_i - 6$$

이러한 중심극한정리 방법을 이용하여 다음 문제를 풀어보자.

1) 중심극한 정리에 의한 방법으로 표준 정규분포에 따르는 샘플을 1000개 생성한다.
2) 생성된 1000개 샘플에 대한 히스토그램을 그린다.
3) 카이 제곱 방법으로 적합도 검정을 한다.
4) K-S 검정을 한다.

- 주의: JAVA 의 java.utilRandom 클래스를 이용하여 난수를 생성하고, 초기 seed는 12345로 하라.

```
import java.io.*;
import java.util.Arrays;
import java.util.Random;


public class RandomEx {
  public static void main(String[] args) throws IOException {
     File outputFile = new File("random.txt");
     FileWriter out = new FileWriter(outputFile);

     double a[];
     double r;

     long seed = 12345;
     Random rd = new Random(seed);

     int n = 10;

     a = new double[n];
     for (int i=0;i<n;i++) {
       a[i] = rd.nextDouble();
     }

     Arrays.sort(a);

     String str;
     for (int i=0;i<n;i++) {
       str = Double.toString(a[i]);
       out.write(str, 0, 10);
       out.write("\n");
       System.out.println(a[i]);
     }
     out.close();
  }
}
```

- 주의: 카이 제곱 방법에 의한 적합도 검정은 이론적 빈도수의 계산을 필요로 한다. 즉, CDF의 값을 구할 수 있어야 한다. 그러나, 정확한 값은 구하기 어렵기 때문에 근사적인 CDF의 값을 구할 수 밖에 없다. 근사적인 CDF 값을 구할 수 있는 방법에는 여러 가지가 있다. 이 중 2004년에 George Marsaglia이 제안한 방법은 매우 효율적이다(다음 함수는 F(x)의 값을 근사적으로 구한다).

```
public static double normalCDF(double x) {
   double s = x, t = 0., b = x, q = x*x, i = 1.;

   while(s != t) {
      i += 2.;
      b = b * (q/i);
      t = s;

      s = t + b;
   }

   return .5 + s * Math.exp(-.5 * q - .91893853320467274178);
  }
```