

Testowanie generatora liczb losowych.

```
class test {
    public static void main (String [] args) {
        int n=5;
        for (int i=0;i<n;i++)
            System.out.println(i);
    }
}
```

1) Oblicz średnią generowanych liczb. Oblicz drugi moment. Zweryfikuj wykonane oszacowania obliczając te średnie analitycznie (całkowanie po jednorodnym rozkładzie gęstości prawdopodobieństwa $p(x)=1$ dla $0<x<1$ oraz $p(x)=0$ dla pozostałych x).

Odpowiedź: $\langle x \rangle = \frac{1}{2}$, $\langle x^2 \rangle = \frac{1}{3}$

Jakiego odchylenia od średniej można się spodziewać dla wygenerowanych $n=1000$ liczb?

Odpowiedź: Obliczenie odchylenia standardowego dla zmiennej $S_n = \frac{1}{n} \sum_{i=1}^n x_i$ daje wynik $\sigma = \sqrt{\text{var}(S_n)} = \sqrt{\langle S_n^2 \rangle - \langle S_n \rangle^2} = \frac{1}{\sqrt{12n}}$ czyli dla $n=1000$ otrzymujemy $\sigma \approx 0.01$

Wskazówka:

- dla zmiennych niezależnych $\text{var}(A+B)=\text{var}(A)+\text{var}(B)$
- $\text{var}(cA)=c^2\text{var}(A)$, gdzie c - stała

2) Sprawdź niezależność otrzymanych liczb losowych. Np. sprawdź

czy $\langle x_i x_{i+1} \rangle = \langle x_i \rangle \langle x_{i+1} \rangle$.

3) Wykorzystaj klasę Random (pakiet util).

4) W wydruku liczby zmiennoprzecinkowe zaokrąglij do 3 cyfr po przecinku.

```
class TEST {
    public static void main (String [] args) {
        int n=1000;
        double suma=0.0;
        double aver;
        for (int i=0;i<n;i++)
            {suma=suma+Math.random();}
        aver=suma/n;
        System.out.println("Średnia="+aver);
    }
}
```

Prosty aplet:

```
import java.awt.*;
import java.applet.*;
public class First_Applet extends Applet {
    public void init(){
        add (new Label("To jest mój pierwszy aplet"));
        add(new Button ("Przycisk START/STOP"));
        add(new TextField("podaj tekst..."));
    }
}
```

Rysowanie w apletach bazuje na przedefiniowaniu metody paint:

```
import java.applet.Applet;

import java.awt.*;

public class Draw_square extends Applet {

    int applet_size=300;

    public void init(){

        setSize(applet_size,applet_size);

    }

    public void paint(Graphics g){

        g.drawRect(10, 40, 20, 20);

    }

}
```

Utwórz aplet, który jest kwadratem 300x300 i w którym rysujemy 1000 punktów o losowo wybranych współrzędnych.

```
import java.applet.Applet;
import java.awt.*;
public class Rand_test_plot extends Applet {
    int applet_size=300;
    int n=1000;
    public void init(){
        setSize(applet_size,applet_size);
        add (new Label("Liczby losowe"));
    }
    public void paint(Graphics g){
        for (int i=0;i<n;i++){
            int x1=(int)(applet_size*Math.random());
            int y1=(int)(applet_size*Math.random());
            g.drawRect(x1, y1, 1, 1);
        }
    }
}
```

Odseparowanie rysowania od reszty apletu umożliwia nam klasa Canvas. Utwórz klasę Wykres dziedziczącą po klasie Canvas z konstruktorem dwuargumentowym (wielkość płótna, ilość punktów, które są rysowane). W aplecie utwórz obiekt Wykres(300,1000) i dodaj go do apletu.

Generator liczb losowych o lepszych parametrach niż Math.random() znajduje się w pakiecie java.util:

```
import java.util.Random;
class TestGen {
    public static void main (String [] args) {
        Random randomGenerator = new Random();
        int n=1000;
        double suma=0.0;
        for (int i=0;i<n;i++)
{suma=suma+randomGenerator.nextDouble()*randomGenerator.nextDouble();}
        double aver;
        aver=suma/n;
        System.out.println(aver);
        aver=Math.round(aver*1000);
        System.out.println(aver/1000);
    }
}
```

5) Czy zamiast

```
aver=Math.round(aver*1000);
System.out.println(aver/1000);
```

można napisać

```
System.out.println(Math.round(aver*1000)/1000);
```