

Materiał pomocniczy do kursu „Podstawy programowania”

Autor: Grzegorz Góralski

ggoralski.com

# Matematyka, statystyka, wyjątki, pliki

Biblioteki matematyczne i statystyczne, obsługa  
wyjątków, zapis i odczyt plików tekstowych

# Przypadek...

- \* Liczby **pseudolosowe** można otrzymać np. używając:
  - \* `java.util.Random`
  - \* `Math.random()`

# java.util.Random

<http://download.oracle.com/javase/6/docs/api/java/util/Random.html>

```
import java.util.Random;
Random randomGenerator = new Random();

System.out.println("boolean: " + randomGenerator.nextBoolean());
// prawda lub fałsz

System.out.println("int: " + randomGenerator.nextInt());
// liczba całkowita z zakresu int

System.out.println("int [0-10]: " + randomGenerator.nextInt(11));
// liczba całkowita z zakresu 0-10 (do liczby w nawiasie -1)

System.out.println("long: " + randomGenerator.nextLong());
// liczba całkowita z zakresu long

System.out.println("float: " + randomGenerator.nextFloat());
// ułamek float z zakresu 0-1 (bez 1)

System.out.println("double: " + randomGenerator.nextDouble());
// ułamek double z zakresu 0-1 (bez 1)
```

# Math.random()

<http://download.oracle.com/javase/6/docs/api/java/lang/Math.html>

ułamek double z zakresu 0-1 (bez 1)

```
System.out.println("Math.random(): "+Math.random());
```



# Hazardzista

\* Napisz program symulujący zachowanie hazardzisty.

Zasady:

1. Symulujemy zadaną liczbę sesji gier (np. 100)
2. Każdą sesję gracz rozpoczyna z określoną sumą pieniędzy (np. 100 zł)
3. W każdej grze stawka wynosi 1 zł, tyle samo wynosi wygrana
4. Szansa wygranej i przegranej jest taka sama
5. Każda sesja kończy się gdy graczowi skończą się pieniądze, lub uzyska z góry założoną sumę wygranych (np. 200 zł)
6. Po zakończonej określonej wcześniej liczbie sesji, program podaje procent sesji wygranych.

# Hazardzista - kod

```
int poziomZasobow = 100;
int cel = 1000;
int liczbaGier=100;
double wygrane=0;
double procentWygranych=0;
for (int i = 0; i < liczbaGier; i++) {
    int zasoby = poziomZasobow;
    while (zasoby>0 && zasoby<cel) {
        double wynik = Math.random();
        if(wynik<0.5) zasoby--;
        else zasoby++;
    }
    if (zasoby==cel) wygrane++;
}
procentWygranych=(wygrane/liczbaGier)*100;
System.out.println("Wygranych: "+procentWygranych+"%");
```

# Wybrane inne metody i stałe klasy **Math**

<http://download.oracle.com/javase/6/docs/api/java/lang/Math.html>

```
System.out.println("Zaokrąglenie (round):      "+Math.round(3.49));
System.out.println("Zaokrąglenie (rint)       "+Math rint(3.49));
System.out.println("Zaokrąglenie w dół:         "+Math.floor(3.49));
System.out.println("Zaokrąglenie w górę:        "+Math.ceil(3.49));
System.out.println("Liczba mniejsza:          "+Math.min(3.49, 3.51));
System.out.println("Liczba większa:          "+Math.max(3.49, 3.51));
System.out.println("Wartość absolutna:       "+Math.abs(-3.49));
System.out.println("Potęga:                  "+Math.pow(2.0, 3.0));
System.out.println("Pierwiastek kwadratowy:  "+Math.sqrt(9.0));

System.out.println("Pi:                      "+Math.PI);
System.out.println("e:                       "+Math.E);
```

# Statystyka z [org.apache.commons.math](http://commons.apache.org/math)

[http://commons.apache.org/math/download\\_math.cgi](http://commons.apache.org/math/download_math.cgi)

1. Pobierz z powyższego adresu plik commons-math-X.X.zip
2. Rozpakuj go
3. W Eclipse dodaj do wcześniej utworzonego projektu plik commons-math3-3.0.jar:
  1. prawym klawiszem myszy kliknij na ikonie projektu
  2. Wybierz: Properties
  3. W oknie wybierz: Java Build Path -> Libraries -> Add External Jars...
  4. Wybierz plik commons-math3-3.0.jar
  5. Warto też dodać dokumentację, przez:
    1. W projekcie otwórz katalog: Referenced libraries
    2. Kliknij (PKM) na commons-math3-3.0.jar o
    3. Wybierz Properties-> Javadoc Location -> wybieramy katalog /docs/apidocs/ z katalogu w którym rozpakowaliśmy pobrany plik.
4. Dokumentacja online:

<http://commons.apache.org/math/api-2.2/index.html>

5. Przewodnik:

<http://commons.apache.org/math/userguide/stat.html>



# Statystyka z `org.apache.commons.math`

```
import ...
```

```
DescriptiveStatistics stats = new DescriptiveStatistics();
for (int i = 0; i < 100; i++) {
    stats.addValue(Math.random());
}
System.out.println(" *** Stosujemy DescriptiveStatistics() ***");
System.out.println("Średnia:           "+stats.getMean());
System.out.println("W. największa:         "+stats.getMax());
System.out.println("W. najmniejsza:        "+stats.getMin());
System.out.println("Odch. standardowe:    "+stats.getStandardDeviation());
System.out.println("Suma:                 "+stats.getSum());
System.out.println("Wariancja:           "+stats.getVariance());

System.out.println(" *** StatUtils ***");
double[] dane = {2.0, 4.0, 5.0, 4.0, 3.0, 3.5, 5.5};
System.out.println("Średnia:           "+StatUtils.mean(dane));
System.out.println("W. największa:         "+StatUtils.max(dane));
System.out.println("W. najmniejsza:        "+StatUtils.min(dane));
System.out.println("Suma kwadratów:      "+StatUtils.sumSq(dane));

...

```

# Statystyka z `org.apache.commons.math`

```
System.out.println(" *** TestUtils ***");
double[] dane1 = {2.0, 4.0, 5.0, 4.0, 3.0, 3.5, 5.5};
double[] dane2 = {2.0, 6.0, 5.0, 6.0, 3.0, 6.5, 5.5};
long[] obserwowane = {30,20};
double[] spodziewane = {37.5, 12.5};
try {
    System.out.println("t-test dwie próbki:      "+
        TestUtils.pairedTTest(dane1, dane2));
    System.out.println("t-test dwie próbki (+p): "+
        TestUtils.pairedTTest(dane1, dane2, 0.05));
    System.out.println("Chi-kwadrat:          "+
        TestUtils.chiSquare(spodziewane, obserwowane));
    System.out.println("Chi-kwadrat (p):        "+
        TestUtils.chiSquareTest(spodziewane, obserwowane));
} catch (IllegalArgumentException e) {
    e.printStackTrace();
}
```

# Obsługa błędów

- \* Na ogół nie chcemy, aby błąd w programie prowadził do przerwania jego działania w sposób nieprzewidziany
- \* W **obsłudze błędów (wyjątków)** pomagają dwie konstrukcje:
  - \* `try ... catch`
  - \* `throws`

# try ... catch

- \* Obsługujemy błędy bezpośrednio w kodzie

```
try {  
    instrukcje  
}  
catch (ExceptionType e) {  
    kod obsługujący wyjątek  
}  
catch (ExceptionType e1) {  
    kod obsługujący inny wyjątek  
}  
finally {  
    kod wykonujący się zawsze na końcu  
    (można ominąć)  
}
```

# throws

- ✱ Przekazujemy wyjątek do kodu który wywołał nasz kod.

```
public void metoda() throws ExceptionType {  
    .....  
}
```

# Zapisanie danych w pliku tekstowym

```
String plik = "dane.txt";

PrintWriter out;
try {
    out = new PrintWriter(plik);
    out.println("Raz");
    out.println("Dwa");
    out.println("Trzy");
    out.close();
} catch (FileNotFoundException e) {
    System.out.println("Brak pliku!");
    e.printStackTrace();
}
...
```

# Odczytanie danych z pliku tekstowego

...

```
File plikDane = new File(plik);
Scanner skaner;
ArrayList<String> dane = new ArrayList<String>();
try {
    skaner = new Scanner(plikDane);
    while (skaner.hasNextLine()) {
        dane.add(skaner.nextLine());
    }
} catch (FileNotFoundException e) {
    System.out.println("Brak Pliku do odczytania!");
    e.printStackTrace();
}

System.out.println("Odczytane dane: "+dane);
System.out.println("Odczytane dane po kolei: ");
for (String d : dane) System.out.println(d);
```

# Zadanie

- \* Zapisz liczby (**double**) do pliku, odczytaj je i podaj średnią



# Zadanie

- \* Zapisz liczby (**double**) do pliku, odczytaj je i podaj średnią

```
String plik = "liczby.txt";
PrintWriter out;
try {
    out = new PrintWriter(plik);
    out.println("1.0");
    out.println("2.0");
    out.println("3.0");
    out.println("4.0");
    out.close();
} catch (FileNotFoundException e) {
    System.out.println("Brak pliku!");
    e.printStackTrace();
}

...
```

# Zadanie

- \* Zapisz liczby (**double**) do pliku, odczytaj je i podaj średnią

```
File plikDane = new File(plik);
Scanner skaner;
ArrayList<String> dane = new ArrayList<String>();
double suma =0;
int l=0;
try {
    skaner = new Scanner(plikDane);

    while (skaner.hasNext()) {
        suma = suma + Double.parseDouble(skaner.next());
        l++;
    }
} catch (FileNotFoundException e) {
    System.out.println("Brak pliku do odczytania!");
    e.printStackTrace();
}
System.out.println("Średnia: " + suma/l);
```