

Klasy Math i String

funkcje matematyczne, aplikacje wykonujące obliczenia, łańcuchy znaków, obiekty klasy String, operatory konkatencji, przeciążanie operatorów konkatencji, metody klasy String, porównywanie stringów, przekształcanie liczb w łańcuchy znaków, metody konwersji łańcuchów do postaci liczbowej, przykłady operacji na stringach.



Funkcje matematyczne

- Funkcje matematyczne Javy są zawarte w klasie **Math**.
- Funkcje te są zadeklarowane jako statyczne, więc można ich używać bez tworzenia egzemplarza obiektu.
- Funkcje matematyczne – metody klasy Math -zwykle operują na liczbach rzeczywistych typu **double** i zwracają wartości typu **double**.

Wybrane funkcje matematyczne

Metoda	Opis
<code>abs(x)</code>	Wartość bezwzględną liczby x
<code>min(a, b)</code>	Mniejsza z liczb a i b
<code>max(a, b)</code>	Większą z liczb a i b
<code>ceil(double x)</code>	Najmniejszą liczbą całkowitą większą lub równą x
<code>floor(double x)</code>	Największą liczbą całkowitą mniejszą lub równą x
<code>exp(double x)</code>	Liczba e podniesiona do potęgi x
<code>log(double x)</code>	Logarytm naturalny liczby x
<code>pow(double x, double n)</code>	Liczba x podniesiona do potęgi n

Wybrane funkcje matematyczne

Metoda	Opis
<code>random()</code>	Generuje liczbę losową z zakresu od 0.0 do 1.0
<code>round(float x)</code>	Zwraca liczbę całkowitą powstałą z zaokrąglenia liczby typu float
<code>sqrt(double x)</code>	pierwiastek kwadratowy z liczby x
<code>sin(double alfa)</code>	sinus kąta alfa (podanego w radianach)
<code>cos(double alfa)</code>	cosinus kąta alfa (w radianach)
<code>tan(double alfa)</code>	tangens kąta alfa (w radianach)
<code>toRadians(double alfa)</code>	Zamienia stopnie na radiany
<code>toDegrees(double alfa)</code>	Zamienia radiany na stopnie

Problemy z zaokrągleniem liczb

- Załóżmy, że mamy zadeklarowane dwie zmienne `double x = 12.2, y = 12.8;`
- Jaką wartość przyjmą zmienne `a, b, c, d, e, f` w wyniku wykonania następujących instrukcji?

- `int a = (int) x; // 12`
- `int b = (int) y; // 12`
- `double c = (int) x; // 12.0`
- `double d = (int) y; // 12.0`
- `double e = Math.round(x); // 12.0`
- `double f = Math.round(y); // 13.0`

Ćwiczenie 1

- Napisz program obliczający wartość wyrażenia określonego wzorem:

$$w = 2 \cos 15^\circ (1 + \sqrt{17.681})^2$$

Odp. $w = 52.335$

- Wynik ma być podany z dokładnością do trzech miejsc po przecinku.

Wskazówki

1. Do zamiany stopni na radiany można użyć funkcji `Math.toRadians()` lub skorzystać z liczby π :
 1. `double pi = Math.PI;`
2. Wynik można zaokrąglić do trzech miejsc po przecinku za pomocą następującej metody:
 1. `w = Math.round(w*1000)/1000.0;`

Rozwiązanie

```
class Liczby
{
    public static void main(String args[])
    {
        double a = Math.cos(Math.toRadians(15));
        double b = 1+Math.sqrt(17.681);
        double w = 2*a*b*b;
        w=Math.round(w*1000)/1000.0;
        System.out.println("w= " + w);
    }
}
```

Ćwiczenie 2

- Napisz program obliczający wartość wyrażenia określonego wzorem:

$$w = \sqrt[3]{2 \log 521 + \frac{750000000}{128 \cdot 422}}$$

Odp. $w = 24.038$

Wskazówki

1. Liczbę 750000000 można zapisać w postaci:
double $x = 7.5e8$;
2. W klasie Math nie ma funkcji $\log_{10}x$. Można obliczyć jej wartość, stosując wzór:

$$\log_{10} x = \frac{\log_e x}{\log_e 10}$$

Rozwiązanie

```
class Liczby
{
    public static void main(String args[])
    {
        double a = 2*Math.log(521)/Math.log(10);
        double b = 7.5e8/128/422;
        double w = Math.pow(a+b, 1/3.0);
        w = Math.round(w*1000)/1000.0;
        System.out.println("w = " + w);
    }
}
```

Ćwiczenie 3

- Napisz program, który dla wylosowanej liczby rzeczywistej x , gdzie $x \in [-2, 2]$ oblicza wartość funkcji $f(x)$ określonej wzorem:

$$f(x) = \sqrt{\frac{2x+1}{1-x}}$$

Program powinien sprawdzać, czy wylosowana liczba x należy do dziedziny funkcji. W przypadku wylosowania liczby nie należącej do dziedziny funkcji, powinien być wyświetlony odpowiedni komunikat.

Rozwiązanie

```
class Liczby
{
    public static void main(String args[])
    {
        double x = losuj();
        if ( (1-x)!=0 && (2*x+1)/(1-x)>=0 )
        {
            double w = Math.sqrt((2*x+1)/(1-x));
            System.out.println("x = " + x);
            System.out.println("f(x) = " + w);
        }
        else
        {
            System.out.println("x = " + x);
            System.out.println("x spoza dziedziny");
        }
    }
}
```

Rozwiązanie

```
public static double losuj()  
{  
    double los;  
    if ((int)(Math.random()*2)==0)  
    {  
        los = Math.random()*2;  
    }  
    else  
    {  
        los = -1*Math.random()*2;  
    }  
    return los;  
}
```

Zadania



1. **Napisz program, który dla losowych liczb x i y z zakresu od 0 do 100 podaje wartość funkcji $f(x,y)$ określonej wzorem:**

$$f(x, y) = \frac{\sqrt{1 + x^2 y^2}}{2 \cdot (x^2 + y^2) \sqrt{xy}}$$

2. **Napisz program, który dla wylosowanej liczby rzeczywistej x z zakresu od -10 do 10 oblicza wartość funkcji $f(x)$ określonej wzorem:**

$$f(x) = \begin{cases} x^2 + x + 1 & \text{dla } |x| < 1 \\ \sin(x^2 + x + 1) & \text{dla } |x| > 1 \end{cases}$$

Łańcuchy znaków

- Podstawowym elementem każdego tekstu jest znak, który może być literą, cyfrą, znakiem interpunkcyjnym lub innym symbolem.
- Do przechowywania pojedynczych znaków Unicode służą dwubajtowe zmienne typu `char`. Np. deklaracja postaci:
`char ch;`
tworzy zmienną znakową `ch`.
- Podczas deklaracji zmiennej znakowej, można jej nadać wartość początkową:
`char ch = 'a'; // lub poprzez kod znaku: char ch = 97;`
- Łańcuch nazywany też napisem lub stringiem - to ciąg dowolnych znaków, np. słowo, wyrażenie lub zdanie.
- Ciąg znaków ujęty w podwójny cudzysłów traktowany jest przez kompilator jako wartość typu `String`.

Obiekty klasy String

- Klasa String jest **wbudowaną** klasą Javy. Jej obiekty używane są do przechowywania łańcuchów znaków.
- Obiekty klasy String tworzy się bardzo łatwo, np.:
 - `String s;` //utworzenie zmiennej referencyjnej
 - `s = "abc";` //utworzenie obiektu zawierającego napis abc
- Instrukcje te można połączyć, pisząc:
 - `String s = "abc";`
- Nazwa typu String zaczyna się od dużej litery, ponieważ jest to nazwa typu obiektowego.
- Raz utworzone obiekty klasy String nie mogą ulegać zmianom – ich zawartość jest tylko do odczytu.
- Operacje przeprowadzane na stringach tworzą nowe obiekty klasy String.
- Nieużywane stringi są automatycznie usuwane z pamięci.

Operacje na łańcuchach

- Operacje na łańcuchach wykonuje się za pomocą:
 - operatorów konkatencji (łączenia):
 - + (*operator konkatencji*)
 - += (*rozszerzony operator konkatencji*)
 - funkcji łańcuchowych.
- Każdy z tych operatorów tworzy nowy obiekt klasy **String**.
- Przykłady:
 - `String napis = "Ala ";`
 - `napis = napis + "ma kota ";`
 - `napis += "i psa";`

Operatory konkatencji

- Operatory konkatencji `+` oraz `+=` są jedynymi operatorami przeciążonymi w Javie - oprócz sklejania łańcuchów, pozwalają również na ich łączenie z liczbami lub obiektami innych klas.
- Przykłady:

```
int n = 10;
```

```
String str, str1, str2, str3, str4;
```

```
str = "napis";
```

```
str1 = str + n; // napis10
```

```
str2 = str + n + n; // napis1010
```

```
str3 = str + (n + n); // napis20
```

```
str4 = n + n + str; // 20napis
```

Metody klasy String

- Operacje na łańcuchach wykonuje się za pomocą funkcji łańcuchowych.
 - `String s="aBCDe", s1, s2;` //obiekty
 - `int dl=s.length();` //dl=5
 - `int k=s.indexOf('C');` //k=2
 - `int l=s.indexOf('Z');` //l=-1
 - `s1=s.toLowerCase();` //s1=abcde
 - `s2=s.toUpperCase();` //s2=ABCDE
 - `boolean b1=s1.equals(s2);` //b1=false
 - `boolean b2=s1.equals(s1);` //b2=true
 - `String s3=s.substring(0,2);` //s3=aB
 - `char znak=s.charAt(2);` //znak=C

Porównywanie stringów

- Do porównywania zawartości dwóch obiektów typu String służy metoda `equals()`.
- Można też zastosować metodę `compareTo()` - zwraca zero, gdy stringi są równe.
- Użycie operatora `==` może prowadzić do błędów. Operator ten pozwala jedynie stwierdzić, czy zmienne referencyjne odwołują się do tego samego obiektu.
- Przykład:

```
String s = "abc";  
s = s+12;  
String s1 = "abc12";  
boolean b1 = s==s1;           //false  
boolean b2 = s.equals(s1);    // true  
int k = s.compareTo(s1);     // 0
```

Przekształcanie liczb w łańcuchy

- Konwersję liczb do postaci łańcuchów znakowych przeprowadza się za pomocą metody **valueOf()** klasy **String**. Na przykład:
 - **double** x = 12.31;
 - **int** n = 100;
 - **String** s1 = **String.valueOf(x)**;
 - **String** s2 = **String.valueOf(n)**;
- Innym (bardziej kosztownym) rozwiązaniem jest użycie operatorów konkatencji, np:
 - **double** x = 12.31;
 - **int** n = 100;
 - **String** s1 = "" + x;
 - **String** s2 = "" + n;

Przekształcanie łańcuchów w liczby

- Do przekształcania łańcuchów w liczby stosuje się metodę `valueOf()` klasy `Double` lub `Integer`. Na przykład:
 - `String s="3.14", t="10";`
`double x; int i;`
`x=Double.valueOf(s).doubleValue();`
`i=Integer.valueOf(t).intValue();`
- Można też posłużyć się nowszą metodą `parseInt()`:
 - `String t="125";`
`int i = Integer.parseInt(t);`

Przykłady konwersji

```
String s1 = "12.3";
```

```
String s2 = "7.5";
```

```
double a=Double.valueOf(s1).doubleValue();
```

```
double b=Double.valueOf(s2).doubleValue();
```

```
double suma=a+b;
```

```
String wynik = String.valueOf(suma);
```

```
System.out.println(wynik);
```

Ćwiczenie 1

Napisz program wypisujący w pionie:

- a. litery danego słowa, a obok nich odpowiadające im kody;
- b. kolejne wyrazy danego zdania.

Wskazówki

Skorzystaj z następujących metod obiektu String:

- `charAt(i)` - zwraca znak występujący w napisie na i-tej pozycji;
- `indexOf(' ')` - zwraca pozycję pierwszego znaku spacji występującego w napisie;
- `substring(0,i)` - wycina z napisu fragment od pozycji 0 do i.

Rozwiązanie

```
String s = "Astronomia";  
char znak;  
int kod;  
for (int i=0; i < s.length(); i++)  
{  
    znak = s.charAt(i);  
    kod = (int) znak;  
    System.out.println(znak + " - " + kod);  
}
```

```
A - 65  
s - 115  
t - 116  
r - 114  
o - 111  
n - 110  
o - 111  
m - 109  
i - 105  
a - 97
```


Rozwiązanie

```
String s = "Do wakacji jeszcze daleko";
s = s + " ";
int i = s.indexOf(' ');
while (i != -1)
{
    String wyraz = s.substring(0, i);
    if (wyraz != " " && wyraz.length() != 0)
    {
        System.out.println(wyraz);
    }
    s = s.substring(i+1, s.length());
    i = s.indexOf(' ');
}
```

Ćwiczenie 2

Napisz program, który zlicza wyrazy w podanym ciągu znaków. Dodatkowe spacje początkowe i końcowe oraz wielokrotne spacje nie powinny wpływać na wynik.

Na przykład dla napisu:

String s = " Alicja w krainie czarów ";

program powinien wypisać:

Suma wyrazów = 4.

Rozwiązanie

```
public class Napisy
{
    public static void main(String args[])
    {
        String s = "Alicja w krainie czarów";
        System.out.println(s);
        int suma = 0, dl = s.length();
        s = s + " ";
        for (int i=0; i<dl; i++)
        {
            if(s.charAt(i)!=' ' && s.charAt(i+1) == ' ')
                suma = suma+1;
        }
        System.out.println("Suma wyrazów = " + suma);
    }
}
```

Zadania



- **Napisz program, który czyta ciąg znaków, a następnie wypisuje w oddzielnych liniach jego pierwszy i ostatni znak.**
- **Napisz program, który czyta ciąg znaków, a następnie wypisuje jego środkowy znak lub też stosowny komunikat, że takiego znaku nie ma.**
- **Napisz program, który czyta ciąg znaków, a następnie:**
 - **wypisuje co drugi znak z tego ciągu;**
 - **wypisuje w kolejnych liniach po dwa znaki z tego ciągu;**
 - **wypisuje w jednym wierszu jego znaki oddzielone spacjami.**

Zadania



- Napisz program, który czyta ciąg znaków, wypisuje go, a następnie wypisuje ten ciąg znaków w odwrotnej kolejności. Na przykład po wprowadzeniu napisu "atom" program powinien wypisać "atommota".
- Napisz program, który czyta ciąg znaków, a następnie wypisuje jego skrót: po trzy znaki z początku i końca. Na przykład ciąg znaków "Malwina" powinien być skrócony do "Malina". Krótkie ciągi znaków nie ulegają skróceniu.
- Napisz program, który zamienia w danym ciągu znaków wszystkie litery "a" na "b" i na odwrót.
- Napisz program, który podaje liczbę liter 'a' (małych lub dużych) w danym ciągu znaków.