

Materiał pomocniczy do kursu „Podstawy programowania”

Autor: Grzegorz Góralski

ggoralski.com

Obliczenia, zmienne

Proste działania, zmienne, rodzaje zmiennych,
proste operacje i działania na zmiennych.

Proste obliczenia

- * Stwórz projekt „Proste obliczenia”.
- * Stwórz klasę ProsteObliczenia z metodą *main()*;
- * W metodzie main wpisz kod i uruchom program:

```
System.out.println(2+2);
```

- * Program wyświetlił:

4

- * Co się stało?

program obliczył 2+2, a następnie wynik dodawania został wyświetlony przez komendę `System.out.println()`;

Proste obliczenia

- * Dodaj linię:

```
System.out.println("Wynik działania 2+2 to "+"4");
```

- * Program wyświetlił:

```
Wynik działania 2+2 to 4
```

- * Co się stało?

program połączył ze sobą dwa teksty i wyświetlił je razem

Proste obliczenia

- * Dodaj linię:

```
System.out.println("Wynik działania 2+2 to "+4);
```

- * Program wyświetlił:

```
Wynik działania 2+2 to 4
```

- * Co się stało?

program wyświetlił tekst i liczbę

ale czy 4 została potraktowana jako liczba czy jako znak?

Proste obliczenia

- * Dodaj linię:

```
System.out.println("Wynik działania 2+2 to "+2+2);
```

- * Program wyświetlił:

```
Wynik działania 2+2 to 22
```

- * Co się stało?

Program wyświetlił tekst i obie liczby, nie dodał ich do siebie.

Liczby zostały potraktowane jako znaki.

- * Jak więc można by wyświetlić w jednej linii tekst i wynik dodawania?

Proste działania

- ✱ Dodaj kod:

```
int wynik;  
wynik = 2+2;  
System.out.println("Wynik działania 2+2 to "+wynik);
```

- ✱ Program wyświetlił:

```
Wynik działania 2+2 to 4
```

- ✱ Co się stało?

Stworzyliśmy zmienną, przypisaliśmy jej wynik dodawania, a następnie wyświetliliśmy tekst i wartość zmiennej „wynik”.

Zmienna zmienną jest

- * Dodaj kod:

```
wynik = 4+4;  
System.out.println("Wartość zmiennej po działaniu 4+4 to "+wynik);
```

- * Program wyświetlił:

Wartość zmiennej po działaniu 4+4 to 8

- * Co się stało?

Zmienne, jak sama zawa wskazuje, mogą zmieniać swoją wartość, co powyżej uczyniliśmy.

Zmienna zmienną jest

- * Dodaj kod:

```
int liczba = 10;  
wynik = liczba+4;  
System.out.println("Wartość zmiennej po działaniu liczba+4 to "+wynik);
```

- * Program wyświetlił:

Wartość zmiennej po działaniu liczba+4 to 14

- * Co się stało?

Do wartości zmiennej „liczba” zostało dodane 4, następnie wynik tego działania stał się wartością zmiennej „wynik”

Zmienne podlegają działaniom

- * Dodaj kod:

```
wynik = wynik+4;  
System.out.println("Wartość zmiennej po działaniu wynik+4 to "+wynik);
```

- * Program wyświetlił:

Wartość zmiennej po działaniu wynik+4 to 12

- * Co się stało?

Program do wartości zmiennej „wynik” dodał 4 a następnie zmienił wartość zmiennej na wynik tego działania.

Czy tak się da z tekstem?

- ✱ Dodaj kod:

```
String tekst1, tekst2;
```

```
tekst1="Witaj";
```

```
tekst2="świecie";
```

```
System.out.println("Wartość zmiennej po działaniu tekst1+tekst2 to "+tekst1+tekst2);
```

- ✱ Program wyświetlił:

Wartość zmiennej po działaniu tekst1+tekst2 to Witajświecie

- ✱ Hmm.. chyba nie o to nam chodziło? Zmodyfikujmy kod:

```
System.out.println("Wartość zmiennej po działaniu tekst1+tekst2 to "+tekst1+" "+tekst2);
```

- ✱ Teraz już lepiej:

Wartość zmiennej po działaniu tekst1+tekst2 to Witaj świecie

- ✱ Wniosek: Pamiętaj o spacjach (o ile sa potrzebne) przy łączeniu znaków.

Typy zmiennych

LICZBY

rodzaj wartości	typ	zakres przechowywanych wartości
liczby całkowite	byte	od -128 do 127
liczby całkowite	short	od -32 768 do 32 767
liczby całkowite	int	od -2 147 483 648 do 2 147 483 647
liczby całkowite	long	od -9 223 372 036 854 775 808 do 9 223 372 036 854 775 807
liczby zmiennoprzecinkowe	float	od -3,4e38 do 3,4e38*
liczby zmiennoprzecinkowe	double	od -1,8e308 do 1,8e308

UWAGA: W kodzie, używamy w liczbach zmiennoprzecinkowych, kropki zamiast przecinka!

* 3,4e38 oznacza $3,4 \cdot 10^{38}$

Typy zmiennych cd.

INNE

rodzaj wartości	typ	zakres przechowywanych wartości
prawda/fałsz	boolean	prawda lub fałsz (FALSE, TRUE)
jeden znak	char	znaki alfanumeryczne (litery, cyfry) , znaki przestankowe, i in.
łańcuch znaków	String	ciągi (łańcuchy) znaków - tekst

Deklarowanie zmiennych

`int` `liczba;`

rodzaj zmiennej nazwa zmiennej

- ✱ Można zadeklarować wiele zmiennych tego samego typu na raz:

```
String tekst1, tekst2;
```

- ✱ Można zadeklarować zmienną i przypisać jej od razu wartość:

```
int liczba = 10;
```

Stałe

- * Dodaj kod:

```
final double PI = 3.146;
```

```
PI = 4;
```

- * Eclipse zgłasza błąd, po uruchomieniu wyświetlił się komunikat:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
```

```
    The final local variable PI cannot be assigned. It must be blank and  
    not using a compound assignment
```

```
    at ProsteObliczenia.main(ProsteObliczenia.java:47)
```

- * Co się stało?


Stała, jak sama nazwa wskazuje, nie może się zmieniać.

Stosujemy je wtedy gdy mamy do czynienia z wartościami które nie powinny się zmieniać (jak np. Pi).

Stałe

```
final double PI = 3.146;
```

wartość nie może
się zmienić



- * Raz zadeklarowana stała z przypisaną wartością nie może jej już zmienić

Konwencje nazw (klas, metod, zmiennych, stałych)

- * Mogą składać się z liter i cyfr, niektórych innych znaków (np. _), nie zaleca się polskich znaków
- * Nazwa powinna być opisowa (o ile to ma sens): np. raczej `długosc` ,zamiast `d`.
- * Nazwa klasy zaczyna się od wielkiej litery (np. `Chromosom`), dalej małe
- * Nazwa metody, zmiennej zaczyna się od małej litery, dalej małe
- * Jeśli nazwa składa się z wielu słów, to pierwsze słowo pisze się wg. powyższych zasad, następne zaczynają się z wielkiej litery: np. `GatunekRosliny` (klasa) , `LiczbaChromosomow` (zmienna), `wypiszDaneZBazyChromosomow` (metoda)
- * Nazwa stałej składa się wyłącznie z wielkich liter, jeśli zawiera kilka słów, oddzielone są podkreślnikiem (_), np. `MASA_ATOMOWA_WEGLA`
- * **WIELKOŚĆ MA ZNACZENIE:** `LICZBA`, `liczba`, `Liczba` - to trzy **różne** zmienne

Operatory arytmetyczne

znak	znaczenie	przykład	wynik
+	dodawanie	$20+12$	22
-	odejmowanie	$7.0-5.0$	2.0
*	mnożenie	$3*7$	21
/	dzielenie	$5.0/2.0$	2.5
%	dzielenie modulo (reszta z dzielenia)	$5.0\%2.0$	1.0

Działania arytmetyczne

- * Wpisz i wykonaj następujący kod:

```
double wynikDzialania;  
wynikDzialania= 10+3;  
System.out.println("10+3= "+wynikDzialania);  
wynikDzialania = 10-3;  
System.out.println("10-3= "+wynikDzialania);  
wynikDzialania = 10*3;  
System.out.println("10*3= "+wynikDzialania);  
wynikDzialania = 10/3;  
System.out.println("10/3= "+wynikDzialania);  
wynikDzialania = 10%3;  
System.out.println("10%3= "+wynikDzialania);
```

- * Otrzymujemy:

10+3= 13.0

10-3= 7.0

10*3= 30.0

10/3= 3.0

10%3= 1.0

zaraz, zaraz, czy tu nie
powinniśmy otrzymać:
10/3= 3.333333 ?



Działania arytmetyczne

- * Dopisz taki kod:

```
wynikDzialania= 10.0+3.0;  
System.out.println("10.0+3.0= "+wynikDzialania);  
wynikDzialania = 10.0-3.0;  
System.out.println("10.0-3.0= "+wynikDzialania);  
wynikDzialania = 10.0*3.0;  
System.out.println("10.0*3.0= "+wynikDzialania);  
wynikDzialania = 10.0/3.0;  
System.out.println("10.0/3.0= "+wynikDzialania);  
wynikDzialania = 10.0%3.0;  
System.out.println("10.0%3.0= "+wynikDzialania);
```

- * Otrzymujemy:

10.0+3.0= 13.0

10.0-3.0= 7.0

10.0*3.0= 30.0

10.0/3.0= 3.33333333333333333335

10.0%3.0= 1.0

- * Wniosek: Pamiętaj o przecinkach (a właściwie kropkach), kiedy istotne jest to co po nich.