

Podstawy programowania

Laboratorium 2 - wprowadzenie do zmiennych

mgr inż. Krzysztof Szwarc

krzysztof@szwarc.net.pl

Sosnowiec, 2 marca 2020

Definicja

Zmienna jest elementem programu wykorzystywanym do przechowywania określonej wartości.

W Javie każda zmienna ma określony typ oraz nazwę. Typ określa rodzaj, strukturę i zakres przyjmowanych wartości, a nazwa pozwala na identyfikację zmiennej.

Przed użyciem zmiennej w Javie należy ją stworzyć (zadeklarować). W tym celu wykorzystujemy poniższy zapis:

```
typZmiennej nazwaZmiennej;
```

Istnieje możliwość jednoczesnego zadeklarowania wielu zmiennych poprzez rozdzielenie ich nazw przecinkami np.

`typZmiennej nazwaPierwszejZmiennej, nazwaDrugiejZmiennej;`

W momencie deklaracji można przypisać wartość początkową zmiennej (tzw. inicjalizacja zmiennej) np.

```
typZmiennej nazwaZmiennej = wartoscZmiennej;
```

W Javie występują również stałe (od momentu pierwszego przypisania jej wartość jest niezmienna). Tworzymy je tak samo jak zmienne, jednak przed typem dopisujemy słowo `final`:

```
final typStałej NAZWA_STAŁEJ;
```

Uwagi ogólne

- Wielkość liter ma znaczenie (zapis nazwaZmiennej nie jest równoznaczny z zapisem nazwazmiennej).
- Nazwy klas i zmiennych muszą rozpoczynać się od litery bądź znaku podkreślenia (nazwa może składać się z liter, cyfr i znaku podkreślenia).
- Główny plik zawierający kod źródłowy musi mieć taką samą nazwę jak klasa.
- W uruchamianej klasie musi znajdować się metoda main.

Konwencja nazewnictwa

- Nazwy klas zapisywane są w notacji PascalCase (NazwaKlasy).
- Nazwy metod i zmiennych zapisywane są w notacji camelCase (jakasZmienna).
- Nazwy stałych zapisywane są w notacji SCREAMING_SNAKE_CASE (TO_STALA).

Sposób zapisu

- Komentarz zajmujący jedną linijkę rozpoczynamy od //
- Komentarze zajmujące wiele linii umieszczamy między /*, a */
- Instrukcje umożliwiające automatyczne generowanie dokumentacji umieszczamy między /**, a */

Przykłady

- `int zmienna = 5; // zmienna ma wartość 5`
- `int zmienna = 5; /* zmienna ma wartość 5 */`
- `/** Komentarz dla metody */
public static void main(String[] args)`

Kod powinien się sam komentować!

- Nazwy zmiennych powinny odzwierciedlać reprezentowany byt.

Przykład

Do reprezentacji liczby studentów można wykorzystać nazwę `liczbaStudentow` zamiast `x`.

Kod powinien się sam komentować!

- Nazwy zmiennych powinny odzwierciedlać reprezentowany byt.
- Komentarze nie powinny opisywać oczywistych operacji. Stosujemy je zazwyczaj do argumentowania wybranego podejścia, zrobienia listy TODO oraz opisu licencji.

Przykład

Poniższy komentarz tylko zaciemnia kod:

```
a=1; // przypisanie do zmiennej a wartości 1.
```

Użyteczny komentarz:

```
// stosujemy wzór x, ponieważ zachodzi y.
```

Kod powinien się sam komentować!

- Nazwy zmiennych powinny odzwierciedlać reprezentowany byt.
- Komentarze nie powinny opisywać oczywistych operacji. Stosujemy je zazwyczaj do argumentowania wybranego podejścia, zrobienia listy TODO oraz opisu licencji.
- Należy unikać tzw. magicznych liczb.

Przykład

Dla wyznaczenia średniej arytmetycznej należy zastąpić konstrukcję $(liczbaJeden+liczbaDwa)/2$ poprzez $(liczbaJeden+liczbaDwa)/liczbaZmiennych$.

Kod powinien się sam komentować!

- Nazwy zmiennych powinny odzwierciedlać reprezentowany byt.
- Komentarze nie powinny opisywać oczywistych operacji. Stosujemy je zazwyczaj do argumentowania wybranego podejścia, zrobienia listy TODO oraz opisu licencji.
- Należy unikać tzw. magicznych liczb.
- Zaleca się nadawanie angielskich nazw.

Typy danych w Javie

- Typ arytmetyczny.
- Typ znakowy.
- Typ logiczny.
- Typ obiektowy (referencyjny).

Typ arytmetyczny całkowitoliczbowy

Typ arytmetyczny całkowitoliczbowy jest wykorzystywany do reprezentowania liczb całkowitych. W zależności od zakresu (i zajętości pamięci) wyróżniamy:

Typ	Liczba bitów	Domyślna wartość (dla pola)
byte	8	0
short	16	0
int	32	0
long	64	0L

Do zmiennych typu arytmetycznego całkowitoliczbowego możemy przypisywać wartości w formacie dziesiętnym, binarnym (z przedrostkiem 0b), ósemkowym (z przedrostkiem 0) oraz szesnastkowym (z przedrostkiem 0X).

Przykłady inicjalizacji

```
int number = 10;  
int number = 0b1010;  
int number = 012;  
int number = 0XA;  
long number = 100000L;  
long number = 100000l;
```

Typ arytmetyczny zmiennoprzecinkowy

Typ arytmetyczny zmiennopozycyjny (zmiennoprzecinkowy) jest wykorzystywany do reprezentowania liczb rzeczywistych zapisanych za pomocą notacji naukowej. W zależności od zakresu (i zajętości pamięci) wyróżniamy:

Typ	Liczba bitów	Domyślna wartość (dla pola)
float	32	0.0f
double	64	0.0d

Mogą one przyjmować również wartość NaN (nie jest liczbą, np. $\frac{0.0}{0.0}$) oraz nieskończoność (np. $\frac{1.0}{0.0}$). Liczby zmiennopozycyjne bez przyrostka traktowane są jako typ double.

Typ arytmetyczny zmiennoprzecinkowy - inicjalizacja

Przykłady inicjalizacji

```
float number = 10.0F;
```

```
float number = 10.0f;
```

```
float number = 10;
```

```
double number = 10.0d;
```

```
double number = 10.0D;
```

```
double number = 10.0;
```

```
double number = 10;
```

Maksymalna i minimalna wartość

Aby otrzymać maksymalną lub minimalną wartość, która może być reprezentowana przez dany typ arytmetyczny możemy skorzystać odpowiednio z pola `MAX_VALUE` lub `MIN_VALUE` klasy reprezentującej dany typ (uwaga - liczy zmiennoprzecinkowe).

Przykład:

```
int min = Integer.MIN_VALUE;  
int max = Integer.MAX_VALUE;  
double min = Double.MIN_VALUE;  
double max = Double.MAX_VALUE;
```

Char

Typ znakowy **char** jest używany do reprezentacji jednego znaku kodu Unicode i zajmuje 16 bitów. Jako domyślną wartość przyjmuje `'\u0000'`.

Przykłady inicjalizacji:

```
char character = 'a';
```

```
char character = '\u0061';
```

Znaki specjalne

- 1 \t - tabulacja.
- 2 \b - backspace.
- 3 \n - nowa linia.
- 4 \r - powrót karetki.
- 5 \" - cudzysłów.
- 6 \' - znak apostrofu.
- 7 \\ - lewy ukośnik.

Boolean

Typ logiczny **boolean** przyjmuje jedną z dwóch wartości: true oraz false i jest wykorzystywany do sprawdzania warunków logicznych. Nie można konwertować go na wartości całkowitoliczbowe.

Przykłady inicjalizacji:

```
boolean variable = true;
```

```
boolean variable = false;
```

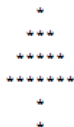
Opis

Do wyświetlenia wartości możemy wykorzystać metody obiektu `out` (należącego do klasy `PrintStream`) - `print` bądź `println` (w zależności od tego czy chcemy, żeby wypisanie wartości zakończyło się przejściem do nowego wiersza). Do łączenia przekazanych wartości możemy użyć operatora `+`. Przykłady:

```
System.out.print(number);  
System.out.print("Wartość");  
System.out.print("Wartość " + number);  
System.out.println(number);  
System.out.println("Wartość");  
System.out.println("Wartość " + number);
```


- 1 Zadeklaruj zmienną o nazwie floatNumber typu float, po czym przypisz jej wartość 0,1. Następnie zainicjuj zmienną o nazwie doubleNumber typu double wartością zmiennej floatNumber i wypisz obie wartości w następujących wierszach.
- 2 Zainicjuj dwie zmienne typu char wartością reprezentującą znak s (z czego jedną za pomocą kodu Unicode) i wypisz je.
- 3 Zainicjuj zmienną typu long wartością 5000000000, po czym wypisz tekst "Zainicjowałem zmienną. Jej wartość to: " i wypisz wartość zmiennej.

- Narysuj choinkę.



- Korzystając wyłącznie z metody `print()` wypisz dowolny napis w dwóch wierszach.
- Zainicjuj stałą typu boolean wartością `true` i spróbuj przypisać jej wartość `false`.
- Zadeklaruj dwie zmienne o nazwie "variable".
- Zadeklaruj zmienną o nazwie "variable" oraz "Variable".
- Umieść komentarz w wybranym programie.

- 10 Wypisz najmniejszą i największą wartość, jaką może przyjąć zmienna typu long.
- 11 Korzystając z generatora na stronie <http://patorjk.com/software/taag/> utwórz napis PP i wypisz go wykorzystując jedno wywołanie metody print (`System.out.print()`).
- 12 Zainicjuj zmienną typu float wartością 100000001 i wypisz ją.
- 13 Zadeklaruj zmienną typu float i wypisz jej wartość (bez inicjalizacji zmiennej).

Sformatowane wyświetlenie wartości

Opis

Aby uniknąć konkatencji (łączenia) tekstu za pomocą operatora `+`, możemy skorzystać z metody `printf`. Przykład:

```
System.out.print("Wartość " + number + ".");  
System.out.printf("%s%d%c", "Wartość ", number, '.');
```

`%d` - liczba całkowita

`%f` - liczba zmiennopozycyjna

`%c` - znak

`%s` - napis

- 1 Korzystając z metody `printf` wyświetl tekst "Wpisałeś liczbę 13", gdzie 13 jest wartością zmiennej `liczba`.

Opis

W celu pobrania wartości od użytkownika musimy utworzyć obiekt klasy Scanner. Aby tego dokonać należy:

- 1 Zaimportować pakiet zawierający jego definicję za pomocą instrukcji `import java.util.Scanner;`
- 2 Utworzyć nowy obiekt klasy Scanner - `Scanner reader = new Scanner(System.in);`
- 3 Wartości, w zależności od pożądanego typu danych, możemy pobierać od użytkownika korzystając z metod utworzonego obiektu - `nextDouble()`, `nextBoolean()`, `nextInt()`,

Przykład

```
6   package javaapplication44;
7
8   import java.util.Scanner;
9
10  public class JavaApplication44 {
11      public static void main(String[] args) {
12          Scanner reader = new Scanner(System.in);
13          double number = reader.nextDouble();
14      }
15  }
```

Program powinien wypisać informację o swoim przeznaczeniu oraz poprosić użytkownika o wpisanie liczby.

- 1 Zadeklaruj zmienną typu `int` i pobierz jej wartość od użytkownika, po czym wyświetl wiadomość "Wpisałeś liczbę" i wypisz pobraną wartość.
- 2 Zadeklaruj zmienną typu `double` i pobierz jej wartość od użytkownika, po czym wyświetl wiadomość "Wpisałeś liczbę" i wypisz pobraną wartość.
- 3 Zadeklaruj zmienną typu `boolean` i pobierz jej wartość od użytkownika, po czym wyświetl wiadomość "Wpisałeś wartość" i wypisz pobraną wartość.

Dziękuję za uwagę