

# Podstawy programowania

Ćwiczenie 2

## Zmienne i stałe

W językach programowania występuje pojęcie zmiennych oraz stałych. Są one podstawowymi elementami służącymi do przechowywania danych, pozwalając uzyskać do nich dostęp poprzez unikalną, nadaną im przez programistę nazwę.

Zmienna to symboliczna nazwa dla obszaru pamięci, który przechowuje pewną dynamiczną wartość.

```
int liczbaCalkowita = 10;
double liczbaRzeczywista = 3.14;
String tekst = "Hello, World!";
```

Stała to nazwany obszar pamięci, który przechowuje pewną niezmienną w trakcie działania programu wartość.

```
final double PI = 3.141592653589793;
final int LICZBA_MAX = 100;
```

W skład deklaracji zmiennej lub stałej wchodzi:

- typ danych, który określa rodzaj danych jakie może przechowywać (np. int, double, String),
- nazwa, identyfikująca ją w programie,
- operator przypisania (=), który przypisuje wartość prawej strony wyrażenia do zmiennej/stałej po nazwie,
- wartość początkowa, w przypadku zmiennej, może to być początkowa wartość przypisana do zmiennej a w przypadku stałej, wartość ta nie może być zmieniana po pierwszym jej ustawieniu.

Dodatkowo, stałe poprzedzone są słowem kluczowym final, co oznacza, że wartość tej zmiennej nie może być zmieniana po pierwszym ustawieniu.

## Typy danych

W języku programowania, typy danych określają rodzaj danych, które mogą być przechowywane w zmiennych/stałych. Określa, jakie operacje można wykonywać na nich oraz ile pamięci zajmuje w pamięci komputera. Java posiada kilka podstawowych typów danych:

### Typy proste

Są to podstawowe typy danych, które reprezentują proste wartości. Są one bezpośrednio obsługiwane przez język Java. Są to np.:

- int, reprezentuje liczby całkowite,
- double, reprezentuje liczby rzeczywiste,
- boolean, reprezentuje wartości logiczne true/false,
- char, reprezentuje pojedynczy znak.

```
int integerValue = 42;
double doubleValue = 3.14;
boolean booleanValue = true;
char charValue = 'A';
```

## Typy referencyjne

To typy, które reprezentują obiekty i są przechowywane w pamięci jako referencje do obiektów. Są to np.:

- String, reprezentuje łańcuchy znaków,
- Integer, Double, Boolean, są klasami „opakującymi” dla typów prymitywnych.

```
String text = "Hello, World!";
Integer integerValue = Integer.valueOf(0);
Double doubleValue = Double.valueOf(42);
Boolean booleanValue = Boolean.valueOf(false);
```

Ze względu na wydajność oraz oznaczenie samego użycia klasy jako konwertera typu jako przestarzałe, zaleca się używanie valueOf, poszczególnych klas opakujących.

## Funkcje (metody)

Funkcje (inaczej metody) są fragmentami kodu, które wykonują określone zadanie. Mogą przyjmować argumenty, przetwarzać je oraz wykonywać pewne operacje a następnie, za pomocą słowa kluczowego return, zwracać ich wynik.

```
public static void main(String[] args) {
    System.out.println("Hello, World!");
    System.out.println(add(5, 5));
}

public static int add(int a, int b) {
    int sum = a + b;
    return sum;
}
```

- add, nazwa metody,
- int (przed nazwą), oznacza, że metoda zwraca wartość całkowitą,
- (int a, int b), lista parametrów, które metoda przyjmuje. W tym przypadku są to dwie liczby całkowite,
- return sum, oznacza, że metoda zwraca wynik operacji. W tym przypadku sumę dwóch argumentów.

Metody pozwalają na strukturyzację programów, dzięki czemu kod jest bardziej czytelny, zorganizowany i łatwiejszy do zarządzania. Działa to również z zasadą tzw. enkapsulacji, ponieważ metody pozwalają na ukrywanie szczegółów implementacji i udostępnianie tylko interfejsu (czyli sygnatury) zewnętrznym częściom programu.

## Instrukcje proste i złożone

W językach programowania, instrukcje proste oraz złożone są podstawowymi składnikami kodu.

## Instrukcje proste

Są to pojedyncze operacje wykonywane sekwencyjnie. Każda instrukcja prosta, kończy się średnikiem (;). Instrukcje proste wykonują jedną konkretną operację.

```
int a = 5;
int b = 7;
int sum = a + b;
System.out.println("Sum: " + sum);
```

## Instrukcje złożone

Instrukcje złożone to grupy instrukcji prostych, umieszczonych między nawiasami {} (tzw. curly braces). Blok kodu, może zawierać jedną instrukcję złożoną lub wiele instrukcji prostych. Pozwalają one na kontrolę przepływu programu, poprzez grupowanie instrukcji prostych w tzw. blokach warunkowych, pętlach czy też metodach. Pomagają one w organizacji kodu, zapewniając mu większą czytelność i lepszą strukturyzację programów.

```
int number = 10;
if (number > 0) {
    System.out.println("Is positive.");
    System.out.println("It is an complex instruction example.");
} else {
    System.out.println("Is negative.");
}
```

## Instrukcje warunkowe

Instrukcje warunkowe w języku Java, pozwalają na wykonanie różnych bloków kodu w zależności od spełnienia określonego warunku. Najczęściej stosowaną instrukcją warunkową jest if-else.

```
int number = 10;
if (number > 0) {
    System.out.println("Is positive.");
} else if (number < 0) {
    System.out.println("Is negative.");
} else {
    System.out.println("Is equal zero.");
}
```

Instrukcje warunkowe pozwalają programowi na elastyczne reagowanie na różne sytuacje, co jest kluczowe w sterowaniu przepływem programu. Inne rodzaje instrukcji warunkowych to switch oraz operator trójargumentowy, tzw. ternary operator czy też operator Elvisa.

```
int weeeekday = 3;
// Switch
switch (weeeekday) {
    case 1:
        System.out.println("Monday");
        break;
    case 2:
        System.out.println("Tuesday");
        break;
}
```

```

        case 3:
            System.out.println("Wednesday");
            break;
        case 4:
            System.out.println("Thursday");
            break;
        case 5:
            System.out.println("Friday");
            break;
        default:
            System.out.println("Weekend!");
    }

    // Ternary operator
    int number = 7;
    String result = (number % 2 == 0) ? "even" : "odd";
    System.out.println("Number is " + result);

```

## Pętle

W językach programowania pętle służą do wielokrotnego wykonywania pewnego bloku kodu. Pozwalają na automatyzację operacji, które muszą być z jakiegoś powodu powtarzane. W Java, dostępne są różne rodzaje pętli, tj. for, while i do-while.

```

// Pętla "for"
for (int i = 1; i <= 5; i++) {
    System.out.println(i);
}

// Pętla "while"
int i = 1;
while (i <= 5) {
    System.out.println(i);
    i++;
}

// Pętla "do while"
int j = 1;
do {
    System.out.println(j);
    j++;
} while (j <= 5);

```

## Typy wyliczeniowe

Typy wyliczeniowe (tzw. enum) to specjalne typy, które pozwalają na definiowanie zestawu stałych wartości, które są zazwyczaj związane tematycznie. Jest to zbiór stałych, które reprezentują konkretne wartości.

```
enum Colors {  
    RED,  
    GREEN,  
    BLUE  
}  
  
Colors myFavouriteColor = Colors.GREEN;
```

## Tablice

Tablice w języku programowania Java są strukturami danych, które pozwalają przechowywać zbiór elementów tego samego typu pod jednym identyfikatorem. Są przydatne do grupowania danych i umożliwiają dostęp do elementów, poprzez indeksowanie. Indeksowanie w tablicach zaczyna się od zera.

```
int[] integerNumber = new int[5];  
integerNumber[0] = 10;  
integerNumber[1] = 20;  
integerNumber[2] = 30;  
integerNumber[3] = 40;  
integerNumber[4] = 50;  
  
String[] strings = { "A", "B", "C" };  
  
System.out.println("Element 1: " + strings[1]);  
System.out.println("Element 2: " + integerNumber[0]);  
System.out.println("Element 3: " + integerNumber[integerNumber.length - 1]);
```

Tablice reprezentują jednowymiarowe struktury danych, ale istnieją także wielowymiarowe tablice (np. tablice dwuwymiarowe), które pozwalają na organizację danych w bardziej złożone struktury.

## Moduły

## Ćwiczenia