

Lekcja 2: Start z Java

Krzysztof Gębicz

Podstawowe funkcje w Javie

Wyświetlanie

```
System.out.println("Witaj w  
Javie!"); // wyświetla i  
przechodzi do nowej linii
```

```
System.out.print("Hello ");  
// wyświetla, bez przechodzenia  
do nowej linii
```

```
System.out.printf("Liczba: %d\n",  
10); // wyświetlanie formatowane
```

`System.out.println()` – używamy do większości wyświetleń, np. komunikatów do użytkownika.

`System.out.print()` – przydatne, gdy chcemy kontynuować w tym samym wierszu.

`System.out.printf()` – gdy potrzebujemy formatować tekst, np. liczby z określoną liczbą miejsc po przecinku.

Pobieranie danych

Biblioteka:

```
import java.util.Scanner;
```

```
Scanner sc = new Scanner(System.in);
```

```
System.out.print(" Podaj imię: ");
```

```
String name = sc.nextLine();
```

- `Scanner sc = new Scanner(System.in);` – tworzymy obiekt `Scanner`, który pozwala czytać dane od użytkownika.
- `sc.nextLine()` – wczytuje cały wiersz tekstu do zmiennej typu `String`.
- `System.out.print()` – wyświetla tekst bez przechodzenia do nowej linii.

Inne przydatne metody `Scanner`:

- `nextInt()` – wczytuje liczbę całkowitą (`int`).
- `nextDouble()` – wczytuje liczbę zmiennoprzecinkową (`double`).
- `nextBoolean()` – wczytuje wartość logiczną (`true` / `false`).
- `next()` → wczytuje słowo (ciąg znaków do pierwszej spacji).
- `char c = sc.next().charAt(0);` → wczytuje pierwszy znak z wczytanego słowa.

Instrukcje warunkowe

```
if (warunek) {  
    // kod, gdy warunek jest  
    prawdziwy  
}  
  
else {  
    // kod, gdy warunek jest  
    fałszywy  
}
```

Opis działania:

- `if` sprawdza warunek logiczny.
- Jeśli warunek jest prawdziwy → wykonuje blok `if`.
- Jeśli warunek jest fałszywy → wykonuje blok `else`.

Użycie: Do podejmowania decyzji w programie.

Pętla for

```
for (int i = 0; i < 5; i++)  
{  
    System.out.println("i = " +  
        i);  
}
```

Opis działania krok po kroku:

1. Tworzymy zmienną `i` i ustawiamy ją na 0.
2. Sprawdzamy warunek `i < 5`. Jeśli prawda → wykonujemy kod w pętli.
3. Po wykonaniu kodu wykonujemy `i++` (zwiększamy `i` o 1).
4. Powtarzamy punkt 2, aż warunek będzie fałszywy.

Klasy

```
class Student {  
    // pola (zmienne)  
  
    String name;  
  
    int age;  
  
    // Konstruktor klasy  
  
    Student(String n, int a) {  
        name = n; // ustawienie wartości  
        age = a;  // ustawienie wartości  
    }  
  
}
```

Opis:

- Klasa to szablon obiektu.
- Zawiera pola (dane) i metody (funkcje).
- Konstruktor to specjalna metoda wywoływana przy tworzeniu obiektu, używana do ustawienia początkowych wartości pól.
- W Javie każda klasa zaczyna się od słowa `class` i nazwy klasy (z wielkiej litery).

Opis użycia:

- Pola przechowują informacje o obiekcie.
- Konstruktor jest wywoływany automatycznie, gdy tworzymy obiekt z użyciem `new`.
- Nazwa konstruktora musi być taka sama jak nazwa klasy.

Metody

```
void introduce() {  
    System.out.println("Cześć,  
jestem " + name + ", mam " + age + "  
lat.");  
}  
  
// Metoda zwracająca wiek  
  
int getAge() {  
    return age;  
}
```

Opis działania:

- Metoda – funkcja związana z klasą, może wykonywać działania lub zwracać wartość.
- **void** – metoda nic nie zwraca, np. **introduce()**.
- **return** – zwraca wartość z metody, np. **getAge()**.
- Metody używamy, żeby operować na danych obiektu i wykonywać akcje.

Użycie: Każdy obiekt klasy może wywoływać swoje metody np. **s.introduce()**.

Tworzenie obiektów i użycie metod

```
public class Main {  
    public static void main(String[] args) {  
        // Tworzenie obiektów klasy Student  
        Student uczen1 = new Student("Imie", 20);  
        Student uczen2 = new Student("Imie", 22);  
  
        // Wywoływanie metod obiektów  
        uczen1.introduce();  
        uczen2.introduce();  
  
        // Pobieranie danych przez metodę zwracającą  
wartość int wiek = uczen1.getAge();  
        System.out.println("Wiek ucznia: " + wiek);  
    }  
}
```

Opis działania:

- `new Student(...)` – tworzy nowy obiekt klasy `Student`.
- Każdy obiekt ma własne pola i może wywoływać swoje metody.
- Wynik metody można przechowywać w zmiennej lub używać od razu w `System.out.println()`.

Zadanie 1 Kalkulator dwóch liczb

Napisz program, który:

1. Pyta użytkownika o dwie liczby.
2. Pyta, jakie działanie chce wykonać (+, -, *, /).
3. Wykonuje działanie i wyświetla wynik.

Wskazówka:

- użyj `Scanner` do wczytania danych,
- porównuj znak działania za pomocą `if` i `equals()`,
- pamiętaj o sprawdzeniu dzielenia przez 0.

Podaj pierwszą liczbę: 10

Podaj drugą liczbę: 5

Wybierz działanie (+, -, *, /): *

Wynik: 50

Zadanie 2 Tabliczka mnożenia

Napisz program, który:

1. Pyta użytkownika o liczbę `n`.
2. Wypisuje tabliczkę mnożenia dla tej liczby od 1 do 10.

Wskazówka:

- użyj pętli `for (int i=1; i<=10; i++)`,
- w każdym kroku wypisz `n * i = wynik`.

Podaj liczbę: 7

$$7 * 1 = 7$$

$$7 * 2 = 14$$

...

$$7 * 10 = 70$$

Zadanie 3 Liczby parzyste i nieparzyste

Napisz program, który:

1. Pyta użytkownika, ile liczb wylosować.
2. Losuje tyle liczb z przedziału od 1 do 10.
3. Wypisuje każdą wylosowaną liczbę.
4. Oblicza sumę liczb parzystych i sumę liczb nieparzystych.
5. Wyświetla wynik.

Wskazówka:

na początku programu dodaj bibliotekę do losowania: `import java.util.Random;`

- utwórz obiekt losujący:

```
Random rand = new Random();
```

- losowanie liczby od 1 do 10:

```
int x = rand.nextInt(10) + 1;
```

- sprawdzaj parzystość warunkiem `if (x % 2 == 0)`.

Zadanie 4 Średnia arytmetyczna i największa liczba

Treść:

1. Program pyta użytkownika, ile liczb wylosować.
2. Losuje te liczby z przedziału 1–100 i wypisuje je wszystkie.
3. Oblicza i wypisuje:
 - średnią arytmetyczną,
 - największą liczbę.

Ile liczb wylosować? 5

Wylosowane liczby: 12 87 45 23
56

Średnia = 44.6

Największa liczba = 87

Zadanie 5 Klasa Prostokąt – pole, obwód i rodzaj

Utwórz klasę `Rectangle` z polami: `width` (szerokość) i `height` (wysokość).

Dodaj metody:

- `area()` -> zwraca pole prostokąta,
- `perimeter()` -> zwraca obwód prostokąta,
- `type()` -> zwraca informację, czy prostokąt jest kwadratem (szerokość = wysokość) czy zwykłym prostokątem.

W `main`:

- wczytaj szerokość i wysokość od użytkownika,
- utwórz obiekt `Rectangle`,
- wypisz pole, obwód i rodzaj.

Podaj szerokość: 5

Podaj wysokość: 3

Pole = 15

Obwód = 16

Rodzaj: Prostokąt