Wprowadzenie do programowania w języku Java

Materiały dla początkujących

Przygotowane na potrzeby zajec indywidualnych

16 grudnia 2025

Spis treści

1 Wprowadzenie

Java jest językiem programowania wysokiego poziomu, służącym do tworzenia aplikacji webowych, mobilnych (Android), serwerowych, desktopowych i wielu innych.

Najważniejsze cechy:

- obiektowość,
- przenośność ("Write Once, Run Anywhere"),
- duża społeczność i bogaty ekosystem,
- bezpieczeństwo i stabilność.

2 Instalacja i narzędzia

2.1 Java Development Kit (JDK)

Do programowania w Javie potrzebny jest JDK (Java Development Kit). Pobierz aktualną wersję ze strony:

https://adoptium.net

Po instalacji sprawdź wersję:

```
java -version
javac -version
```

2.2 Środowisko programistyczne (IDE)

Polecane narzędzia:

- IntelliJ IDEA (najpopularniejsze),
- Eclipse,
- VS Code z rozszerzeniami Java.

3 Pierwszy program w Javie

Każdy program zaczyna się od klasy zawierającej metodę `main`.

3.1 Przykład: Hello World

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

Kompilacja:

```
javac Main.java
```

Uruchomienie:

```
java Main
```

4 Podstawy składni

4.1 Zmienne i typy danych

Java posiada typy proste:

- `int`
- `double`
- `boolean`
- `char`

Przykład:

```
int wiek = 25;
double cena = 19.99;
boolean deszcz = false;
char znak = 'A';
```

4.2 Operatory

- Aritmetyczne: `+ - * / %`
- Porównania: `== != > < >= <=`
- Logiczne: `&& || !`

5 Instrukcje sterujące

5.1 If / else

```
int x = 10;

if (x > 5) {
    System.out.println("Wi ksze ni  5");
} else {
    System.out.println("Mniejsze lub r wne 5");
}
```

5.2 Switch

```
int day = 3;

switch(day) {
    case 1: System.out.println("Poniedzia ek"); break;
    case 2: System.out.println("Wtorek"); break;
    case 3: System.out.println(" roda "); break;
    default: System.out.println("Inny dzie ");
}
```

5.3 Pętle

for:

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

while:

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

6 Tablice

```
int[] numbers = {1, 2, 3, 4};

System.out.println(numbers[0]);    // 1
System.out.println(numbers.length); // 4
```

7 Programowanie obiektowe (OOP)

Java jest silnie obiektowa — wszystko opiera się na klasach i obiektach.

7.1 Tworzenie klasy

```
public class Osoba {
    String imie;
    int wiek;

    void przedstawSie() {
        System.out.println("Jestem␣" + imie);
    }
}
```

7.2 Tworzenie obiektu

```
Osoba o = new Osoba();
o.imie = "Jan";
o.wiek = 30;
o.przedstawSie();
```

7.3 Konstruktory

```
public class Osoba {
    String imie;
    int wiek;

    public Osoba(String imie, int wiek) {
        this.imie = imie;
        this.wiek = wiek;
    }
}
```

8 Dziedziczenie i interfejsy

8.1 Dziedziczenie

```
class Zwierze {
    void dzwiek() {
        System.out.println("Zwierz ␣wydaje␣d  wi  k");
    }
}

class Pies extends Zwierze {
    void dzwiek() {
        System.out.println("Hau␣hau!");
    }
}
```

8.2 Interfejsy

```
interface Drukowalne {
    void drukuj();
}

class Dokument implements Drukowalne {
    public void drukuj() {
        System.out.println("Drukuj ␣dokument");
    }
}
```

9 Wyjątki

```
try {
    int x = 5 / 0;
} catch (ArithmeticException e) {
    System.out.println("B    d :␣" + e.getMessage());
}
```

10 Kolekcje (Java Collections Framework)

10.1 List

```
import java.util.*;

List<String> lista = new ArrayList<>();
lista.add("A");
lista.add("B");

System.out.println(lista.get(0));
```

10.2 Map

```
Map<String, Integer> mapa = new HashMap<>();
mapa.put("Jan", 30);
mapa.put("Ala", 25);

System.out.println(mapa.get("Jan"));
```

11 Programowanie funkcyjne: lambdy i streamy

```
List<Integer> liczby = List.of(1,2,3,4,5);

liczby.stream()
      .filter(n -> n % 2 == 0)
      .forEach(System.out::println);
```

12 Praca z plikami

```
import java.nio.file.*;
import java.io.IOException;

try {
    Files.writeString(Path.of("plik.txt"), "Hello␣Java!");
    String tekst = Files.readString(Path.of("plik.txt"));
    System.out.println(tekst);
} catch (IOException e) {
    e.printStackTrace();
}
```

13 Projekt dla pierwsze lekcji

Zaproponowany projekt: **Prosty system kontaktów w konsoli**

Wymagania:

- klasa `Kontakt` (imię, nazwisko, telefon),
- lista kontaktów,
- możliwość dodania, usunięcia, wyszukania kontaktu,
- zapis i odczyt z pliku.