

## Wprowadzenie

### Czym jest Java?

- Zorientowany obiektowo język programowania
  - prosta składnia
  - rozbudowane biblioteki
- Zaprojektowana przez firmę Sun
- Język o składni podobnej do C++
- Zaprojektowana "od zera", początkowo pod nazwą Oak
- Początkowo pomyślana do zastosowań w urządzeniach elektronicznych codziennego użytku
- Przyjęła się jako język do zastosowań sieciowych (szczególnie World Wide Web)

### Charakterystyka języka Java (1)

- Java jest prosta
  - składnia podobna do C++
  - nie ma w Javie:
    - przeciążania operatorów (poza jednym wyjątkiem)
    - plików nagłówkowych i preprocesora
    - operacji arytmetycznych na wskaźnikach
    - struktur i unii
    - wzorców
    - niejawniej konwersji typów
    - dziedziczenia wielobazowego
- Java jest zorientowana obiektowo
  - klasy i interfejsy

### Charakterystyka języka Java (2)

- Java jest kompilowana, ale niezależna od architektury
  - kompilacja do kodu pośredniego - bajtkodu (ang. *byte codes*)
  - kod pośredni jest interpretowany przez Java Virtual Machine (JVM)
  - definicja języka i maszyna wirtualna są w pełni wyspecyfikowane
  - nie ma elementów niezdefiniowanych lub zależnych od implementacji (np. typy proste są takie same na wszystkich maszynach)
  - pełna przenaszalność kodu źródłowego i wynikowego
- Java jest wielowątkowa
  - wątki Javy w miarę możliwości przekładane są na wątki systemu operacyjnego
  - możliwość synchronizacji

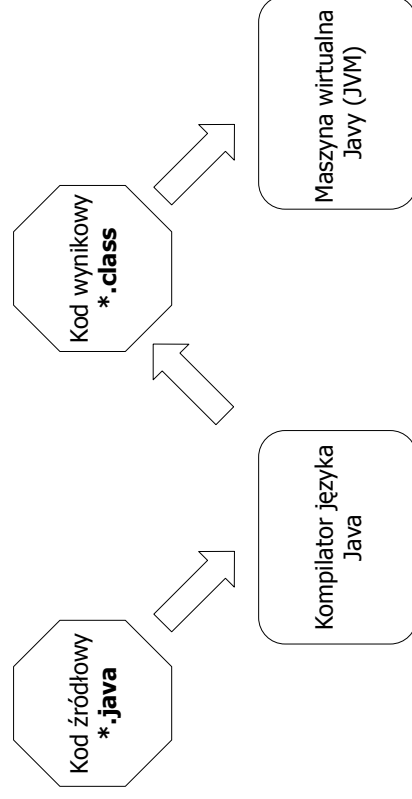
## Charakterystyka języka Java (3)

- Java nadzoruje pamięć
  - nie ma wskaźników takich jak w C++
  - dostęp do obiektów przez *referencje*
    - referencje zachowują się jak "bezpieczne wskaźniki"
    - każdy dostęp do pamięci jest kontrolowany
  - odśmiecanie pamięci (ang. garbage collection)
- Java jest odporna na błędy i bezpieczna
  - ściśle określona forma kodu pośredniego
  - kontrola dostępu do pamięci
  - obsługa błędów w oparciu o wyjątki
  - ograniczenie dostępu do zasobów (Security Manager)

## Charakterystyka języka Java (4)

- Java jest niewielka
  - pomyślana dla małych systemów
- Java jest stosunkowo szybka
  - szybka w porównaniu z innymi językami interpretowanymi
  - wolniejsza od C (może to nie mieć znaczenia w programach wykorzystujących komunikację sieciową lub często oczekujących na reakcję użytkownika)
  - JVM może wykorzystywać kompilatory typu Just-In-Time, aby poprawić efektywność przetwarzania
- Java jest rozszerzalna
  - istnieje możliwość wykorzystywania bibliotek napisanych w innych językach

## Uruchamianie programów w języku Java



## Maszyna wirtualna Javy (JVM)

- Zachowuje się jak "wirtualny komputer" interpretujący bajtkod Javy
- Stanowi bezpieczne środowisko do uruchamiania programów
- Musi być zaimplementowana dla konkretnej platformy
- JVM może wykorzystywać kompilację Just-In-Time (JIT)
  - kompilacja "w locie" bajtkodu do instrukcji maszynowych
  - szczególnie efektywna w przypadku powtarzalnych fragmentów kodu (np. pętle)
- JVM może stanowić samodzielną aplikację lub być wbudowana w inny program np. przeglądarkę

## Typy programów Java

- Samodzielne aplikacje
  - graficzne
  - pracujące w trybie tekstowym
- Aplety
  - małe aplikacje zagnieżdżane w dokumentach HTML
  - w dużym stopniu przyczyniły się do popularności Javy
- Serwlety
  - aplikacje pracujące po stronie serwera
  - mogą stanowić alternatywę dla CGI w serwerach WWW
- JavaBeans
  - komponenty języka Java, zbudowane wg ustalonego wzorca, tworzone z myślą o ich wielokrotnym wykorzystaniu
- Enterprise JavaBeans
  - komponenty języka Java wykorzystujące architekturę rozproszoną

## Programowanie obiektowe (1)

- Abstrakcja
  - identyfikacja obiektów i operacji na nich
  - klasyfikacja podobnych obiektów za pomocą klas
  - zdefiniowanie atrybutów i dopuszczalnych operacji dla poszczególnych klas
- Hermetyczność
  - ukrycie wewnętrznej implementacji klas
  - dostęp do obiektów poprzez wyspecyfikowany interfejs
  - interakcja z obiektami przez wysyłanie *komunikatów* (wywołania publicznych metod)
  - zalety: ochrona i elastyczność kodu

## Programowanie obiektowe (2)

- Dziedziczenie
  - technika wykorzystania istniejących fragmentów kodu
  - polega na tworzeniu nowych klas na bazie już istniejących
  - cechy wspólne dla wszystkich podklas definiowane są w nadklasie
  - podklasa może
    - korzystać z cech nadklasy
    - nadpisywać zachowanie nadklasy
    - dodawać nowe atrybuty i zachowania
- Polimorfizm
  - pozwala w jednolity sposób traktować obiekty klas z hierarchii dziedziczenia przy zachowaniu ich charakterystycznego zachowania
  - od strony technicznej sprowadza się do tzw. późnego wiązania metod przy ich wywołaniu (wybór metody na podstawie typu obiektu)
  - w Javie wszystkie metody zachowują się jak metody wirtualne w C++

## Proste aplikacje w języku Java

- Kod programu w języku Java jest pogrupowany w klasy
- Aplikacje Java posiadają wyróżnioną metodę o nazwie `main()`, od której rozpoczyna się wykonanie programu

```

Hello.java
-----
public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello world");
    }
}

```

## Przekazywanie parametrów wywołania aplikacji Java

- Parametry przekazywane jako tablica obiektów klasy `String`
- Tablica w Javie jest pełnoprawnym obiektem

Hello.java

```
public class Hello {
    public static void main(String[] args) {
        if (args.length > 0)
            System.out.println("1st parameter: " + args[0]);
        else
            System.out.println("No parameters");
    }
}
```

## Java Developer's Kit (JDK)

- Środowisko firmy Sun do tworzenia aplikacji Java
- Zbiór narzędzi wywoływanych z linii poleceń:
  - `javac` - kompilator java
  - `java` - maszyna wirtualna Javy
  - `appletviewer` - przeglądarka appletów
  - `jdb` - debugger klas Java
  - `javadoc` - generator dokumentacji klas w formacie HTML

## Kompilacja i uruchomienie programu

