



# Programowanie - Java

Jakub Mazurek  
kuba@fenix.club

4 listopada 2016

## Spis treści

<b>1</b>	<b>Czym jest program komputerowy?</b>	<b>2</b>
<b>2</b>	<b>Języki programowania</b>	<b>3</b>
2.1	Języki kompilowane . . . . .	4
2.2	Języki interpretowane . . . . .	5
<b>3</b>	<b>Język Java</b>	<b>6</b>
3.1	Maszyna wirtualna Javy . . . . .	6
<b>4</b>	<b>Hello, world!</b>	<b>8</b>
4.1	Kompilacja i wykonywanie kodu Javy . . . . .	8
<b>5</b>	<b>Zadania</b>	<b>9</b>

# 1 Czym jest program komputerowy?

**Program komputerowy** to nic innego jak ciąg instrukcji rozumianych przez procesor, *najczęściej* zapisanych w postaci binarnej (ciąg zer i jedynek).

- Program komputerowy w postaci binarnej nazywamy też **kodem maszynowym**.
- Każdy procesor posiada swój **zestaw instrukcji**, który definiuje, jakie operacje może wykonać dany procesor.
- Kod maszynowy jest dostosowany do konkretnego rodzaju procesora i jego instrukcji, **nie jest uniwersalny**.



Rysunek 1: Supercoder 2000

## 2 Języki programowania

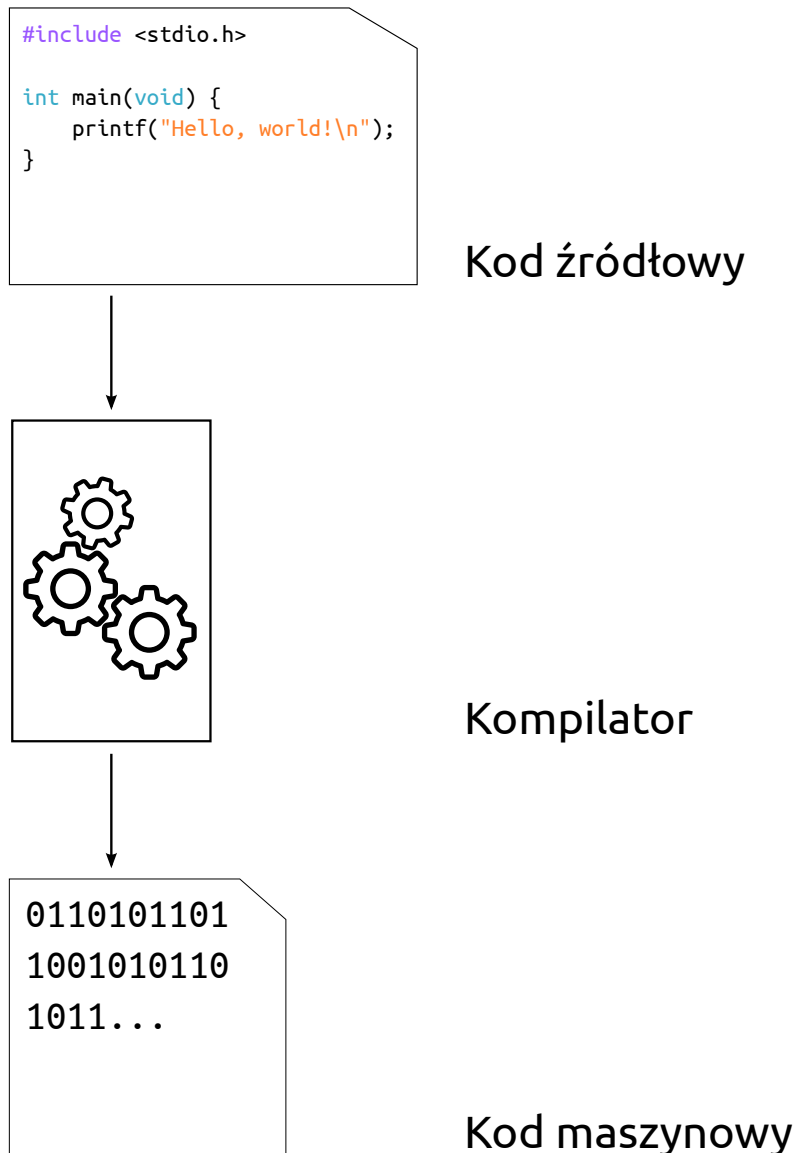
Celem tworzenia i używania języków programowania jest ułatwienie pisania programów komputerowych przez człowieka.

- Języki programowania pozwalają programiście wydawać polecenia dla komputera, nie używając bezpośrednio kodu maszynowego - wprowadzają **warstwę abstrakcji**.
- Ze względu na "bliskość" do kodu maszynowego wyszczególniamy języki **nisko-** i **wysokopoziomowe**.
- **Kompilator** to program, którego zadaniem jest tłumaczenie danego języka programowania do kodu wykonywalnego.

Jednym z podstawowych podziałów w rodzajach języków programowania jest podział na języki **kompilowane** i **interpretowane**.

## 2.1 Języki kompilowane

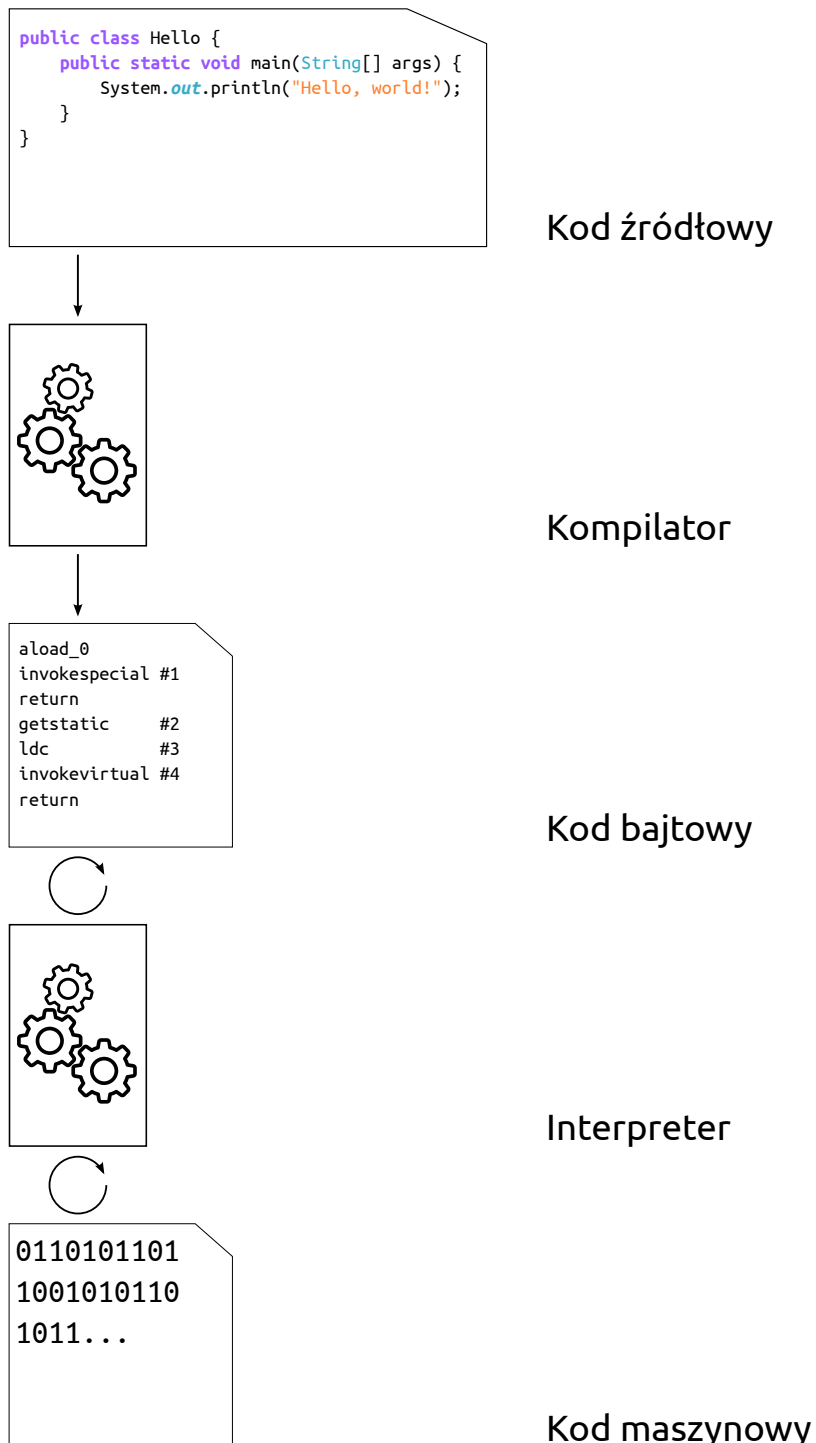
Języki kompilowane (np. C, C++, Objective-C, Go) to takie języki, które są tłumaczone **bezpośrednio** do kodu maszynowego. W związku z tym zyskują one na wydajności, ale tracą na przenośności.



Rysunek 2: Proces kompilacji

## 2.2 Języki interpretowane

Języki interpretowane (np. **Java**, **Python**) nie są kompilowane bezpośrednio do postaci binarnej. Zamiast tego pojawia się forma pośrednia zwana **kodem bajtowym**.



## 3 Język Java

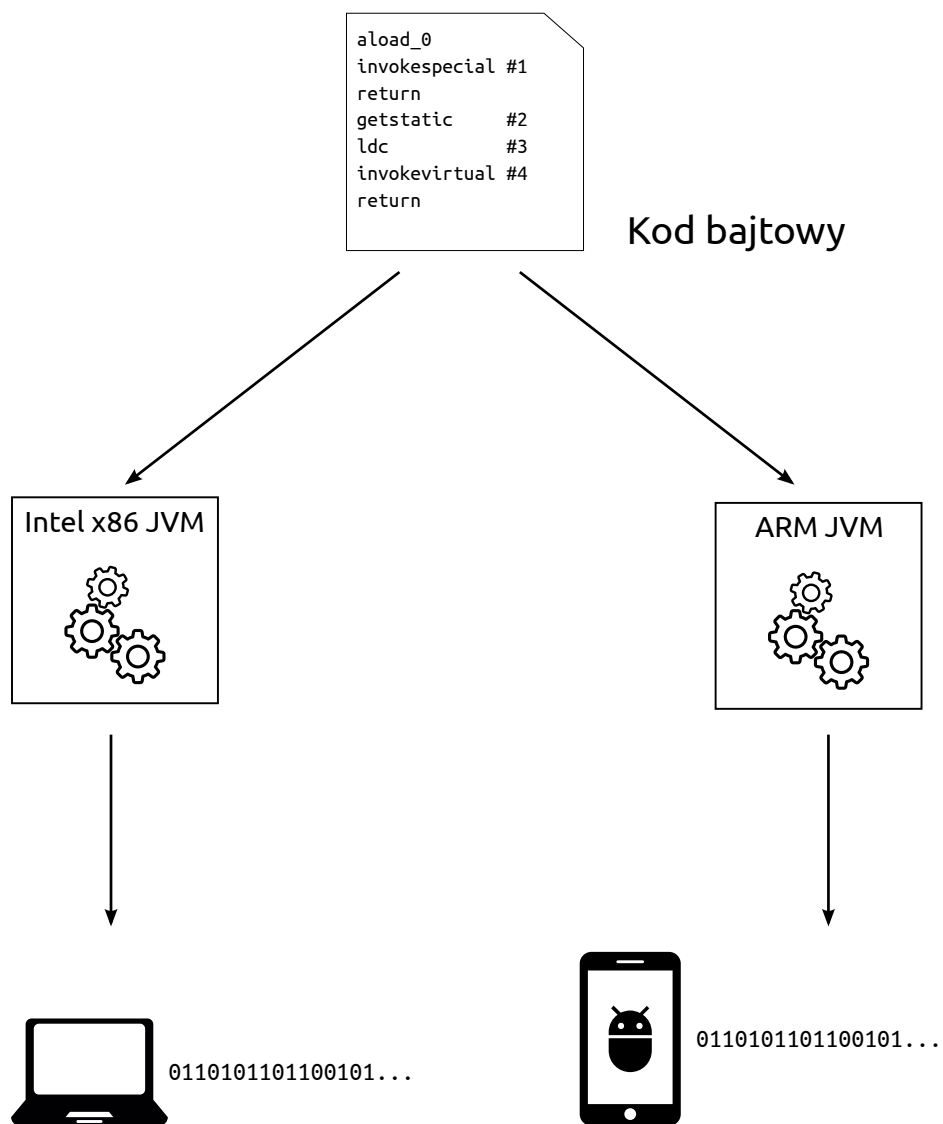
Java jest językiem **interpretowanym**. Kod bajtowy Javy jest wykonywany przez **Java Virtual Machine** - maszynę wirtualną Javy.

### 3.1 Maszyna wirtualna Javy

JVM jest tradycyjnym, kompilowanym programem komputerowym, pisanym na różne urządzenia.

Maszyna wirtualna jest odpowiedzialna za interpretowanie kodu bajtowego (pośredniego) do kodu maszynowego (binarnego) podczas wykonywania programu napisanego w Javie. Co to oznacza w praktyce?

- Jeśli na dane urządzenie (*architekturę procesora*) istnieje JVM - możemy na nim odpalić Javę.
- Implementacje JVM istnieją na ogromną liczbę różnorodnych urządzeń - od tradycyjnych komputerów, przez telefony z Androidem, aż po Lego Mindstorms.
- Znając Javę, Twój kod będzie rozumiany przez wszystkie te urządzenia i będziesz w stanie je programować!



Rysunek 3: Maszyna wirtualna Javy

## 4 Hello, world!

Przyjrzyjmy się teraz jak wygląda przykładowy kod źródłowy napisany w Javie.

```
1 public class HelloWorld {  
2     public static void main(String[] args) {  
3         System.out.println("Hello, world!");  
4     }  
5 }
```

Po Skompilowaniu i uruchomieniu powyższego kodu źródłowego, komputer wypisze tekst: **Hello, world!** do tzw. wyjścia standardowego (domyślnie: konsoli poleceń). Powyższy program składa się z trzech głównych elementów:

### `public class HelloWorld`

Pierwsza linijka programu to deklaracja tzw. klasy o nazwie **HelloWorld**. Klasy deklarujemy za pomocą słowa kluczowego **class**.

**public** jest tutaj tzw. **modyfikatorem dostępu**, dowiemy się o nich więcej w przyszłości.

### `public static void main(String[] args)`

Linijka nr 2 to deklaracja tzw. **metody** klasy. Nadajemy tej metodzie nazwę **main**, która w Javie ma specjalne znaczenie. Metoda **main** wskazuje, gdzie nasz program powinien zacząć się wykonywać. Kod wewnątrz metody **main** to początek naszego programu.

Metoda **main** przyjmuje jeden argument - jest nim zmienna o nazwie **args**, której typ to: **String[]** (tablica elementów typu **String**). W tym argumentcie znajdują się wszystkie argumenty, które zostały podane do programu z konsoli poleceń.

### `System.out.println("Hello, world!");`

Powyższa linijka to jedyna instrukcja w naszym pierwszym programie (instrukcje możemy rozpoznać po znaku **;** na końcu linii). Ta linijka wywołuje metodę o nazwie **println** z klasy **System.out**.

Metoda **println** przyjmuje jeden argument (w przypadku naszego przykładu jest nim **"Hello, world!"**) i wypisuje go do wyjścia standardowego. Metoda ta interpretuje swój argument jako ciąg znaków, a na końcu argumentu dodaje tzw. znak końca linii (stąd w nazwie **println** mamy sufix **ln**).

### 4.1 Kompilacja i wykonywanie kodu Javy

Aby wykonać kod Javy z poziomu konsoli poleceń:

1. Zapisujemy nasz kod źródłowy w pliku z rozszerzeniem **.java**. W przypadku naszego przykładowego programu, plik musi się nazywać: **HelloWorld.java**.
2. W terminalu (konsoli poleceń) przechodzimy do folderu w którym zapisaliśmy nasz kod źródłowy.



3. Kompilujemy nasz kod źródłowy za pomocą polecenia: `javac HelloWorld.java`
4. Po udanym skompilowaniu, uruchamiamy nasz program wpisując w konsolę: `java HelloWorld`

## 5 Zadania

1. Zmień i uruchom program `HelloWorld.java` tak, aby zamiast `Hello, world!` wypisywał `Brawo ja!`
2. Korzystając z argumentów przekazanych za pomocą tablicy `String[] args`, zmień swój program tak, aby wypisywał pierwszy z argumentów.
  - a) Podpowiedź: aby przekazać argumenty do programu, wywołaj go w ten sposób:  
`java HelloWorld argument1 argument2`
  - b) Podpowiedź: skorzystaj z indeksowania tablicy: `args[i]`
3. Napisz program, który wypisze wszystkie argumenty przekazane w konsoli, niezależnie od ich liczby.
  - a) Podpowiedź: skorzystaj z pętli `for`.

W razie problemów:

- Spójrzcie do dokumentacji Javy.
- Piszcie na Slacku, bezpośrednio do mnie albo na naszym kanale.
- Wyślijcie mi maila ([kuba@fenix.club](mailto:kuba@fenix.club)).
- Zapytajcie przy okazji następnych zajęć - omówimy wspólnie rozwiązania zadań.