

# Java Platform – wprowadzenie

## Platformy Technologiczne

Michał Piotrowski  
Michał Wójcik

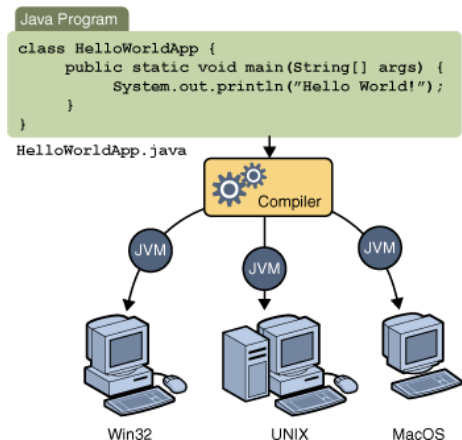
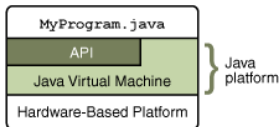
Katedra Architektury Systemów Komputerowych  
Wydział Elektroniki, Telekomunikacji i Informatyki  
Politechnika Gdańska

4 marca 2011

# Java Platform

Java Platform — platforma softwarowa działająca nad platformami hardwarowymi:

- JVM — Java Virtual Machine,
- Java Application Programming Interface.



## Rodzaje Java Platform:

- Java Platform, Standard Edition (Java SE):
  - wcześniej J2SE,
  - aktualna wersja 6 (1.6),
  - wersja 7 w produkcji;
- Java Platform, Enterprise Edition (Java EE):
  - wcześniej J2EE,
  - aktualna wersja 6 (1.6),
  - tworzenie aplikacji serwerowych;
- Java Platform, Micro Edition (Java ME):
  - wcześniej J2ME,
  - przeznaczona na urządzenia mobilne,
  - CLDC — Connected Limited Device Configuration,
  - CDC — Connected Device Configuration.

[1] *Code Conventions for the Java Programming Language.*

http:

[//www.oracle.com/technetwork/java/codeconv-138413.html](http://www.oracle.com/technetwork/java/codeconv-138413.html).

[2] *Core Java Technologies Tech Tips.*

<http://java.sun.com/developer/JDCTechTips/>

<http://blogs.sun.com/CoreJavaTechTips/>.

[3] *Java SE 6 Documentation.*

<http://download.oracle.com/javase/6/docs/>,.

[4] *The Java Tutorials.*

<http://download.oracle.com/javase/tutorial/index.html>.

[5] Bruce Eckel.

*Thinking in Java.*

4th edition.

## Przypomnienie:

- źródła (klasy) umieszczane w oddzielnych plikach, których nazwy odpowiadają nazwom klas,
- klasy grupowane w pakietach (katalogi) (obowiązkowo!),
- biblioteki pakowane jako archiwa JAR,
- archiwum JAR jako sposób dystrybucji aplikacji (ustawienie głównej klasy w pliku MANIFEST:
  - `java -jar aplikacja.jar;`
- dokumentacja generowana poleceniem `javadoc`,
- inicjalizacja pól klasy przy deklaracji pól lub:
  - w konstruktorze dla pól obiektu,
  - w sekcji `static {...}` dla pól statycznych.

## Najważniejsze zalecenia:

- javadoc do komentowania kodu i generowania dokumentacji,
- jedna deklaracja lub wyrażenie w linii,
- klasy i interfejsy nazywamy wielkimi literami, każde kolejne słowo zaczyna się od wielkiej litery:
  - `MojaKlasa;`
- nazwy metod i zmiennych zaczynamy małą literą, kolejne słowa wielkimi literami:
  - `mojaMetoda(int argumentLiczbowy) {...},`
  - `double takaSobieZmienna;`
- nazwy stałych piszemy wyłącznie wielkimi literami, kolejne słowa oddzielamy podkreśleniem:
  - `public static final int BARDZ_WAZNA_STALA = 5;;`
- nazwy paczek powinny składać się wyłącznie z małych liter, odwrócona nazwa domeny (`pl.gda.pg.eti.*`).

JavaBeans — komponenty, których można wielokrotnie używać, klasy Java o specjalnej charakterystyce:

- domyślny publiczny konstruktor bez parametrów,
- serializowalne,
- brak publicznych pól,
- dostęp do pól poprzez metody pośredniczące `get` i `set`,
- wystąpienie metod:
  - `void setCecha(int cecha);`
  - `int getCecha();`

oznaczają że klasa posiada właściwość `cecha` typu `int` dostępną do odczytu i zapisu.

Właściwości (properties) — definiują stan bean wpływając na jego zachowanie lub wygląd.

Klasyfikacja:

- writable — możliwość czytania i zapisu (get i set),
- read only — możliwość jedynie odczytu (get),
- hidden — możliwość jedynie zapisu (set).

Typy:

- simple — pojedyncza wartość, zmiany niezależne od zmiany innych właściwości (set, get, is),
- indexed — tablica właściwości (set, get, set(index), get(index)),
- bound — wspiera informowanie innych zainteresowanych obiektów o zmianach (PropertyChangeListener),
- constrained — wspiera informowanie zainteresowanych o zmianach oraz zgłaszanie przez nich weta (VetoableChangeListener).

# Konfiguracja aplikacji

## Properties:

- pary klucz/wartość (String),
- `java.util.Properties` extends `java.util.Hashtable`,
- `Properties.getProperty(String key);`,  
`Properties.setProperty(String key, String Value);`,
- `Properties.load(InputStream/Reader in);`,  
`Properties.store(OutputStream/Writer out, String comment);`,
- cykl życia aplikacji:
  - star aplikacji: ładowanie domyślnych, ładowanie zapisanych, inicjalizacja,
  - działanie: ustawianie i modyfikacja,
  - zamknięcie: zapisanie.

## Zmienne środowiskowe:

- `String System.getenv(String name);`.

## Zmienne klasy System:

- `String System.getProperty(String key);`.