

Metody programowania

Wprowadzenie do platformy Java

Dr inż. Andrzej Grosser

Częstochowa, 2013

Spis treści

1. Wprowadzenie do środowiska programowania i języka	5
1.1. Wstęp	5
1.2. Historia Javy	5
1.3. Anatomia pierwszej aplikacji	7
Literatura	9

1. Wprowadzenie do środowiska programowania i języka

1.1. Wstęp

Jednym z najważniejszych języków programowania używanych współcześnie jest Java. Można to chociażby zauważyć, śledząc ranking języków programowania TIOBE¹, gdzie Java z reguły występuje na jednym z pierwszych miejsc. Dlatego też ten popularny język został wybrany jako podstawa dla tego kursu.

Java jest statycznie typowanym, opartym na klasach, obiektywnym językiem programowania. Termin statycznie typowany oznacza, że Java posiada system typów umożliwiający wykrycie wielu błędów już na etapie kompilacji. Natomiast obiektywność oznacza, że prawie wszystko jest w Javie obiektem (oprócz typów podstawowych takich jak `int`). Klasy zorganizowane są w drzewo, którego korzeniem jest klasa `Object`. Język umożliwia stosowanie dziedziczenia i polimorfizmu.

Język posiada szereg udogodnień dla programisty. Dla przykładu zawiera wsparcie do automatycznego odśmiecania pamięci. Zawiera również wbudowane wsparcie dla programowania współbieżnego. Dodatkowym atutem języka jest bogata biblioteka standardowa i ogromna liczba oprogramowania zewnętrznego. Wszystko to pozwala na wielokrotne zwiększenie efektywności tworzenia oprogramowania w porównaniu do innych bardziej tradycyjnych języków takich jak na przykład C.

Większość przygotowanego tekstu bazuje na pozycjach [4], [2], [3] i dokumentacji Javy.

1.2. Historia Javy

Na początku lat 90 dwudziestego wieku firma Sun Microsystems[1] stwierdziła, że dostępne języki programowania nie za bardzo nadają się do tworzenia oprogramowania dla sprzętu elektronicznego (między innymi interaktywnej telewizji). Brakowało języka, który

¹<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

byłby niezawodny, a także, który pozwalałby na uruchomienie już istniejących programów na nowych urządzeniach. Dlatego też zespół Jamesa Goslinga, Mike'a Sheridana i Patricka Naughtona stworzył projekt języka, który umożliwiłby osiągnięcie wymienionych wcześniej celów. Pierwotnie miał on się nazywać Oak (dąb), gdyż tego rodzaju drzewo rośnie obok biura projektantów języka. Okazało się jednak, że istniał wcześniej język programowania o takiej nazwie, stąd też zmieniono nazwę języka na Java.

Szybko stało się jasne, że utworzony język programowania jest na tyle perspektywiczny, że może znaleźć zastosowanie do tworzenia oprogramowania niekonicznie dla sprzętu elektronicznego. Jednym z pierwszych eksperymentalnych zastosowań były aplikacje sieciowe - kluczowa okazała się niezależność od platformy tworzonego oprogramowania pozwalająca na łatwe przenoszenie programów pomiędzy systemami i bezpieczeństwo tworzonego oprogramowania. Wynikiem tych badań było powstanie przeglądarki WWW WebRunner - w całości napisana w Javie - a później HotJava, która umożliwiała uruchamianie appletów. Rozpowszechniły one wiadomość o nowym języku programowania i pozwoliły na dalszy jego rozwój.

Pierwsza wersja Javy została udostępniona przez Sun Microsystem w 1995. Nowy język szybko stał się niezwykle popularny, doskonale trafił w istniejącą lukę na rynku. Pozycję Javy ugruntowało wyjście w 1998 roku Javy 2. Rozwój języka polegał wtedy przede wszystkim na dodawaniu nowych elementów do biblioteki standardowej i polepszeniu wydajności maszyny wirtualnej. Dopiero wypuszczenie na rynek w 2004 platformy Java 1.5 dodało nowe elementy do języka (przede wszystkim typy generyczne i adnotacje). Po wyjściu Javy 1.6 w 2006 roku rozwój języka nieco przyhamował. W tym okresie pewnej stagnacji, było jednak kilka wydarzeń, na które warto zwrócić uwagę, jednym z najważniejszych z nich było wyjście w listopadzie 2006 roku wyszła pierwszej wersji Javy na częściowo uwolnionej licencji.

Dalszy rozwój Javy był już związany z firmą Oracle, która przejęła Sun Microsystem w 2010 roku. Przede wszystkim należy tutaj wspomnieć o wypuszczeniu Javy 1.7, które dodało szereg drobnych usprawnień do języka, jak chociażby wychwytywanie wielu wyjątków w jednym bloku czy łańcuchy w instrukcji wyboru switch-case. Nie stanowi to jednak końca rozwoju języka, niezwykle interesująco prezentuje się Java 1.8, która ma wnieść do języka między innymi wyrażenia lambda.

1.3. Anatomia pierwszej aplikacji

Naszą pracę z językiem Java zaczniemy od prawie najprostszej aplikacji, na jej przykładzie omówimy najważniejsze elementy programu napisanego w Javie. Program ma za zadanie wyświetlenie na ekranie napisu - Witaj świecie. Jego cały kod źródłowy został zamieszczony poniżej (małe liczby z lewej strony kodu nie są częścią kodu, ale pokazują jedynie numery wierszy):

```
1 package first ;
2
3 /**
4  *
5  * @author Andrzej Grosser
6  */
7 public class First {
8
9     /**
10      * @param args the command line arguments
11      */
12     public static void main(String[] args) {
13         System.out.println("Witaj_świecie");
14     }
15 }
```

Omówmy po kolei co można zaobserwować w ciekawszych liniach kodu.

1. Pierwsza linia oznacza, że elementy definiowane w pliku będą składowymi pakietu `first`.
2. W liniach od 3 do 6 i 9 do 11 występuje specjalny, zaczynający się od znaków `/**` komentarz (umożliwiający tworzenie dokumentacji) opisujący tworzony przez nas program. W komentarzu można zauważyć tagi `@author` i `@params` dodające informacje dla generatora dokumentacji o autorze kodu i parametrach funkcji `main()`.
3. Linie od 7 do 15 zawierają publiczną klasę `First`. Definicja klasy w Javie zaczyna się od słów kluczowych `public` i `class` po których zapisana jest nazwa klasy (w tym

przypadku oczywiście First) a następnie pomiędzy otwierającym i zamykającym nawiasem klamrowym zawarte są definicje pól i metod składowych. Proszę zwrócić uwagę, że nie ma średnika, jak w C++, po definicji klasy.

4. Definicja klasy zawiera definicję jednej tylko metody - `main()`. Jest to publiczna, statyczna, nie zwracająca niczego (`void`) metoda pobierająca na liście argumentów tablicę łańcuchów znaków. Publiczna oznacza, że jest ona dostępna z zewnątrz pakietu zaś statyczna, że nie ma potrzeby tworzenia obiektu dla uruchomienia metody. Metoda `main()` jest traktowana specjalnie (jeżeli jest opisana tak jak przedstawiono wcześniej), od jej wykonania zaczyna się wykonywanie programów Javy. Parametrami przekazywanymi do funkcji są parametry wywołania programu (tylko parametry wywołania programu, bez nazwy aplikacji - różnica w porównaniu do C++). Ciało metody jest zawarte między parą nawiasów klamrowych.
5. W linii 13 na rzecz obiektu `out` z pakietu `System` (włączanego automatycznie do programu) zostaje wywołana metoda `println()`, która ma za zadanie wyświetlenie napisu Witaj świecie (napisy w Javie są zapisywane między parą cudzysłowów). Instrukcje tak, jak w C++ kończą się średnikami.

Literatura

- [1] History of the java programming language. http://en.wikibooks.org/wiki/Java_Programming/History.
- [2] P. Deitel and H. Deitel. *Java - How to program*. Prentice Hall, 2011.
- [3] B. Eckel. *Thinking in Java*. Helion, 2006.
- [4] C. S. Horstmann and G. Cornell. *Core Java Volume I-Fundamentals*. Prentice Hall, 2011.