

Ćwiczenia 2

Pierwszy program

Andrzej Abramowski

Polsko-Japońska Akademia Technik Komputerowych

8 października 2017 r.

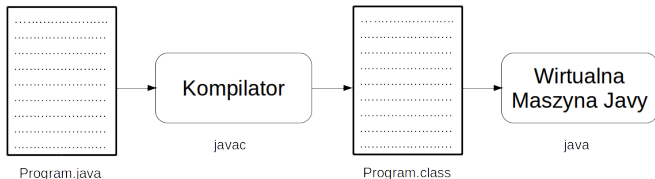
Pierwszy program

```
// Dwa slashe oznaczaja komentarz
// Program jest zamkniety w klasie
public class Hello {
    // Wykonywana jest zawartosc funkcji main
    public static void main(String[] args) {
        // Wypisz na konsoli "Witaj swiecie!"
        System.out.println("Witaj swiecie!");
    }
}
```

Ważne

- Nazwy klasy zwyczajowo zaczynają się od dużej litery
- Nazwa pliku musi odpowiadać nazwie klasy
- Plik musi posiadać rozszerzenie .java

Uruchamianie programu w języku Java



Dwustopniowy proces:

- 1 Kompilacja programu tekstowego za pomocą polecenia `javac`
- 2 Uruchomienie skompilowanego kodu za pomocą polecenia `java`

Ważne

Do pisania programów w Javie konieczna jest instalacja rozszerzenia JDK (Java Development Kit).

Uruchamianie programu w języku Java

Należy wykonać kolejne czynności:

- 1 Stworzenie programu w dowolnym edytorze tekstowym
- 2 Zapisanie treści programu pod nazwą zgodną z nazwą publicznej klasy i rozszerzeniem .java (dla naszego programu będzie to Hello.java)
- 3 Kompilacja programu poleceniem `javac Hello.java`
- 4 Utworzenie przez kompilator pliku `Hello.class` (w przypadku nieudanej kompilacji zostaną wypisane komunikaty opisujące błędy, należy je poprawić i spróbować ponownie)
- 5 Uruchomienie programu za pomocą polecenia `java Hello` (bez rozszerzenia)

Ważne

Dobrze jest wprowadzić w treść programu błędy i zapoznać się z najbardziej powszechnymi komunikatami kompilatora (usunięcie średnika, nawiasów klamrowych, brak deklaracji zmiennych itp.).

```
public class VariableTest {  
    public static void main(String[] args) {  
        // Deklarujemy zmienna x  
        // int oznacza liczbe calkowita (ang. integer)  
        int x;  
        // Przypisujemy jej wartosc 1 (inicjacja)  
        x = 1;  
        System.out.println(x);  
        // Zwiekszmy wartosc x o 1  
        x = x + 1;  
        System.out.println(x);  
    }  
}
```

Podstawowe typy zmiennych

- typ logiczny boolean
- typy całkowitoliczbowe:
 - byte (8 bitów)
 - short (16 bitów)
 - integer (32 bity)
 - long (64 bity)
- typy rzeczywiste:
 - float (32 bity)
 - double (64 bity)
- typ znakowy char (16 bitów)

Czym jest bit?

Definicje

- bit - najmniejsza jednostka informacji w informatyce, przyjmuje wartość 0 albo 1
- bajt - najmniejsza adresowalna jednostka informacji pamięci komputerowej, prawie zawsze składa się z 8 bitów

Systemy liczbowe wykorzystywane w praktyce

- System dziesiętny:
$$5103 = 5 \cdot 10^3 + 1 \cdot 10^2 + 0 \cdot 10^1 + 3 \cdot 10^0$$
- System binarny:
$$1001 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$
- System szesnastkowy:
$$AE08 = 10 \cdot 16^3 + 14 \cdot 16^2 + 0 \cdot 16^1 + 8 \cdot 16^0$$

Kod uzupełnień do dwóch (U2)

Ogólna postać kodu U2

Umożliwia zapis liczby ze znakiem poprzez przypisanie do współczynnika dla najwyższej potęgi znaku -:

$$x_{n-1}x_{n-2}\dots x_1x_0 = -x_{n-1} \cdot 2^{n-1} + x_{n-2} \cdot 2^{n-2} + \dots + x_1 \cdot 2^1 + x_0 \cdot 2^0$$

Przykład - kolejne wartości typu byte

$$10000000 = -1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

$$10000001 = -1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

...

$$11111111 = -1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$00000000 = -0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$$

$$00000001 = -0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

...

$$01111110 = -0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

$$01111111 = -0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$$

Kod uzupełnień do dwóch (U2)

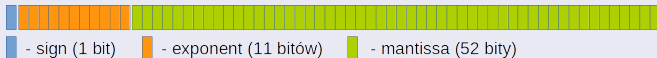
Co trzeba wiedzieć?

- zmienne typów całkowitoliczbowych są przechowywane przez Jave w postaci kodu U2
- zwiększenie o 1 zmiennej przechowującej maksymalną wartość, spowoduje przyjęcie przez zmienną wartości minimalnej
- zmniejszenie o 1 zmiennej przechowującej minimalną wartość, spowoduje przyjęcie przez zmienną wartości maksymalnej

Humor

There are only 10 types of people in the world: those who understand the binary system and those who don't.

Reprezentacja binarna



$$x = (-1)^{sign} \cdot mantissa \cdot 2^{exponent-1023}$$

$$mantissa = 1 + b_{51} \cdot 2^{-1} + b_{50} \cdot 2^{-2} + \dots + b_0 \cdot 2^{-52}$$

Ważne

Jak widać zmienne typu float i double mają ograniczoną dokładność, przez co nie nadają się do operacji matematycznych wymagających wysokiej precyzji ani bezpośredniego porównywania!

- ❶ Przedstaw aplikację deklarującą i inicjującą wszystkie zmienne następujących typów:
 - logicznego
 - liczb całkowitych
 - liczb rzeczywistych
 - znaków
- ❷ Zainicjuj zadeklarowane w poprzednim zadaniu zmienne najmniejszymi i największymi wartościami jakie mogą przyjąć.

Test kodu U2 dla typu int

```
public class IntegerTest {  
    public static void main(String[] args){  
        int min = Integer.MIN_VALUE;  
        System.out.println("Min_int_=_ " + min);  
        System.out.println("Min_int_-_1_=_ " + (min-1));  
        int max = Integer.MAX_VALUE;  
        System.out.println("Max_int_=_ " + max);  
        System.out.println("Max_int_+_1_=_ " + (max+1));  
        int x = -1;  
        System.out.println(Integer.toBinaryString(x));  
    }  
}
```

Ważne

W dokumentacji języka Java można sprawdzić inne metody klasy Integer wyszukując `java.lang.Integer`.

- ❶ Jaki będzie rezultat dodania:
- do zmiennej typu char zmiennej typu int
 - do zmiennej typu int zmiennej typu char
 - do zmiennej typu float zmiennej typu double
 - do zmiennej typu byte zmiennej typu int

```
public class AdditionTest {  
    public static void main(String[] args) {  
        char a = 'a';  
        int b = 7;  
        char c = a + b;  
        int d = a + b;  
    }  
}
```

Promocja numeryczna

Promocja numeryczna

Promocja numeryczna jest to automatyczna konwersja typów argumentów operatorów arytmetycznych do typu wspólnego.

Zasady promocji numerycznej

Zasada działania promocji argumentów dla operatorów dwuargumentowych:

- 1 jeżeli jeden z argumentów jest typu double, drugi przekształcany jest do typu double
- 2 jeżeli jeden z argumentów jest typu float, drugi przekształcany jest do typu float
- 3 jeżeli jeden z argumentów jest typu long, drugi przekształcany jest do typu long
- 4 w pozostałych przypadkach oba argumenty przekształcane są do typu int

```
public class ConversionTest {  
    public static void main(String[] args) {  
        char a = 'a';  
        int b = 7;  
        char c = (char)(a + b);  
        int d = a + b;  
    }  
}
```

Ważne

W języku Java można rzutować wartość danego typu na inny typ za pomocą konstrukcji (typ)wartość.

Literały typu long i float

```
public class ConversionTest {  
    public static void main(String[] args) {  
        long x = 21L;  
        float y = 20.5F;  
        float z = (float)20.5;  
    }  
}
```

Ważne

W języku Java wartości rzeczywiste są typu double, a wartości całkowite typu int. W celu oznaczenia do wartości typu float dodaje się przyrostek f lub F, a do wartości typu long przyrostek l lub L.

Instrukcja if - else

```
public class IfTest {  
    public static void main(String[] args) {  
        int x = (int)(10 * Math.random());  
        // if oznacza jezeli  
        if (x == 2) {  
            System.out.println("x jest rowny 2.");  
        }  
        // else oznacza w pozostalych wypadkach  
        else {  
            System.out.println("x jest rozny od 2.");  
        }  
    }  
}
```

Ważne

Metoda `Math.random()` losuje liczbę rzeczywistą (dlatego niezbędne jest rzutowanie!) z przedziału `[0, 1)`.

Operatory relacji i operatory logiczne

Operatory relacji

- `==` - argumenty są równe
- `!=` - argumenty są nierówne
- `>` - argument po lewej jest większy
- `>=` - argument po lewej jest większy równy
- `<` - argument po lewej jest mniejszy
- `<=` - argument po lewej jest mniejszy równy

Operatory logiczne

- `!` - zaprzeczenie
- `&&` - logiczne i
- `||` - logiczne lub

❶ Dane są następujące zbiory:

- $A = [-5; \infty]$
- $B = [-\infty; 5]$
- $C = [-10; 10]$

Napisz program jednoznacznie klasyfikujący przynależność danej wartości x do jednego lub kilku z powyższych zbiorów (*Zadeklaruj i zainicjuj zmienną x w programie różnymi wartościami w celu weryfikacji poprawności działania programu lub losuj wartość zmiennej x i uruchom program kilkakrotnie*).

❷ Dane jest wyrażenie: $\text{int } x = 2 * 5 + 3 * 4 - 8$; Jaki jest rezultat tych operacji? Zmodyfikuj to wyrażenie, tak aby rezultatem była wartość 48.

Operator trójargumentowy

```
public class IfTest2 {  
    public static void main(String[] args) {  
        int x = (int)(10 * Math.random());  
        // Deklarujemy zmienne przechowujące łańcuch  
        // znaków (typu String)  
        String text1 = "x jest równy 2.";  
        String text2 = "x jest różny od 2.";  
        // operator ?: wybiera text1 gdy x równa się 2  
        // lub wybiera text2 w pozostałych przypadkach  
        String text = x == 2 ? text1 : text2;  
        System.out.println(text);  
    }  
}
```

- 1 Dane są następujące zbiory:
- $A = (-15; 10] \cup (-5; 0) \cup [5; 10]$
 - $B = [-\infty; -13] \cup (-8; -3)$
 - $C = [-4; \infty]$

Napisz program jednoznacznie klasyfikujący przynależność danej wartości x do jednego lub kilku z powyższych zbiorów (*Zadeklaruj i zainicjuj zmienną x w programie różnymi wartościami w celu weryfikacji poprawności działania programu lub losuj wartość zmiennej x i uruchom program kilkakrotnie*).

Tematy do przypomnienia/przećwiczenia:

- Tworzenie programu w języku Java
- Deklaracja zmiennych
- Inicjacja zmiennych
- Typy zmiennych
- Operatory i ich priorytet
- Promocja numeryczna
- Rzutowanie zmiennych
- Instrukcja warunkowa if-else
- Operator trójargumentowy