

KURS PROGRAMOWANIA W JAVIE

BINARNA REPREZENTACJA LICZBY CAŁKOWITEJ

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie 1.

Utwórz plik tekstowy o nazwie `username.java`, gdzie *username* to Twój identyfikator w sieci (jeśli zaczyna się od cyfry to dopisz na początku identyfikatora literę *x*). Plik ten powinien zawierać program w Javie wypisujący do standardowego strumienia wyjściowego `System.out` Twoje imię i nazwisko.

Wskazówka.

Plik `username.java` można utworzyć przy pomocy jakiegokolwiek edytora tekstu. Plik źródłowy kompilujemy poleceniem:

```
> javac username.java
```

Kompilator tworzy plik binarny o nazwie `username.class`. Skompilowany program uruchamiamy poleceniem:

```
> java username
```

Program w Javie powinien zawierać publiczną klasę o takiej samej nazwie jak nazwa pliku (dlatego nazwa pliku musi zaczynać się od litery):

```
public class username
{
    treść klasy
}
```

Klasa ta powinna zawierać publiczną statyczną metodę `main`:

```
public static void main (String[] args)
{
    treść metody
}
```

Wykonanie programu polega na wywołaniu tej metody.

Standardowy strumień wyjściowy jest skojarzony z obiektem o nazwie `out`, będącym publicznym polem statycznym w klasie `System`. Obiekt `out` posiada między innymi metody `print` oraz `println`, które wypisują do standardowego strumienia wyjściowego napis (obiekt klasy `String`) podany im jako parametr. Metoda `println` powoduje dodatkowo wypisanie znaku przejścia do nowego wiersza.

Zadanie 2.

Napisz program wypisujący kolejno wszystkie parametry z jakimi zostanie on wywołany. Każdy parametr wypisz w osobnym wierszu. Dla przykładu, jeżeli utworzysz program o nazwie `parametry`, to powinien on działać następująco:

```
> java parametry abc 123 "Java to język programowania" -3.14
abc
123
Java to język programowania
-3.14
```

Wskazówka.

Parametry programu są umieszczone w tablicy napisów przekazanej metodzie `main` jako parametr. Obiekt tablicy posiada pole publiczne `length` mówiące o jej rozmiarze.

Zadanie 3.

Napisz program wypisujący binarną reprezentację liczb podanych mu jako parametry wywołania. Dla każdego *parametru* program powinien wypisać jeden wiersz postaci

$$parametr = b$$

gdzie *parametr* jest liczbą całkowitą typu `int` przekazaną do programu poprzez parametr wywołania, zaś *b* jest jego 32-bitową reprezentacją dwójkową. Jeżeli podany *parametr* nie jest zapisem żadnej liczby, to program powinien wypisać wiersz

parametr *parametr* nie jest poprawnym zapisem dziesiętnym liczby

Program powinien również poprawnie przetwarzać liczby ujemne. Dla przykładu, jeżeli utworzysz program o nazwie `liczby`, to powinien on działać następująco:

[illegible]

Wskazówka.

Klasa `Integer` posiada metodę statyczną `parseInt`, która zamienia podany jej napis na liczbę całkowitą. Gdy konwersja jest niemożliwa, to zgłaszany jest wyjątek `NumberFormatException`. Wyjątki obsługuje się przy pomocy instrukcji `try-catch`:

```
try
{
    n = Integer.parseInt(arg[indeks]);
    // ...
}
catch (NumberFormatException ex)
{
    System.out.println ("parametr "+
        args[indeks]+" nie jest poprawnym zapisem dziesiętnym liczby");
}
```