

# KURS JĘZYKA JAVA

## LICZEBNIKI PORZĄDKOWE PO ANGIELSKU

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

**Zadanie.**

Napisz program, który liczby naturalne (nieujemne liczby całkowite) przekazane do programu poprzez argumenty wywołania będzie przekształcał na liczebniki porządkowe i wypisywał je na standardowe wyjście `System.out` w języku angielskim (również w postaci skróconej). Oto przykład działania takiego programu (program o nazwie `Liczebniki`):

```
$> java Liczebniki 5 21 102 1012 253 2250 3000 2000000
fifth (5th)
twenty-first (21st)
one hundred and second (102nd)
one thousand and twelfth (1012th)
two hundred and fifty-third (253rd)
two thousand two hundred and fiftieth (2250th)
three thousandth (3000th)
two millionth (2000000th)
$>
```

Wykorzystaj w swoim programie tablicowane liczebniki składowe (jedności, nastki, dziesiątki, itd.):

```
public class Liczebniki
{
    private static String[] jed_główne = {"zero", "one", "two", ...};
    private static String[] jed_porząd = {"zeroth", "first", "second", ...};
    private static String[] nast_główne = {"ten", "eleven", "twelve", ...};
    private static String[] nast_porząd = {"tenth", "eleventh", "twelfth", ...};
    private static String[] dzies_główne = {"zero", "ten", "twenty", ...};
    private static String[] dzies_porząd = {"zeroth", "tenth", "twentieth", ...};
    ...
}
```

Program ma konwertować na postać słowną dowolne liczby całkowite typu `int`. Napisz `String` można przekonwertować liczbę `int` korzystając z klasy opakującej `Integer` i autoboxingu:

```
int x = new Integer(napis);
```

Jeśli konwersja nie będzie możliwa to zostanie zgłoszony wyjątek `NumberFormatException`. Jeśli w wyniku poprawnie przeprowadzonej konwersji otrzymamy liczbę ujemną, to należy zgłosić wyjątek instrukcją `throw`:

```
throw new IllegalArgumentException("liczba ujemna " + x + " jest błędem");
```

Twój program powinien sobie skutecznie poradzić także z liczbą 0.

#### **Uwaga.**

Program należy skompilować i uruchomić z wiersza poleceń! Podczas kompilacji użyj opcji `-encoding` jeśli będzie to konieczne (aby kompilator umiał poprawnie przetłumaczyć literały znakowe zaszyte w programie do *unikodu*). Jeśli twój program był napisany w pliku kodowanym zgodnie ze standardem *ISO-8859-2* to wywołanie kompilatora będzie następujące:

```
C:\Documents\MyJavaProg\> javac -encoding iso-8859-2 LiczbySloownie.java
```

Jeśli uruchamiasz program w konsoli pod Windowsami użyj opcji `-Dfile.encoding=cp852` (przed uruchomieniem sprawdź jakiego kodowania używa konsola poleceniem `chcp`). Pozwoli to na prawidłowe wyświetlanie polskich znaków diakrytycznych. Przykładowe uruchomienie programu może wyglądać w takim przypadku następująco:

```
C:\Documents\MyJavaProg\> java -Dfile.encoding=cp852 LiczbySloownie 17 23 abc
```