
KURS PROGRAMOWANIA W JAVIE

ROZKŁAD LICZB CAŁKOWITYCH NA CZYNNIKI PIERWSZE

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Zadanie.

Napisz program, który wypisze na standardowym wyjściu `System.out` rozkład zadanych (poprzez argumenty wywołania) liczb całkowitych na czynniki pierwsze. Rozkład każdej z nich ma być wypisany w osobnym wierszu: najpierw liczba, potem znak `=` i dalej czynniki pierwsze poodzielane znakiem `*`.

Można założyć, że parametrami wywołania programu będą liczby całkowite nieprzekraczające typu `long`. Jeśli program wywołano bez żadnego parametru, to należy wypisać na standardowym strumieniu dla błędów `System.err` instrukcję obsługi programu.

Programując to zadanie zdefiniuj klasę usługową `LiczbyPierwsze`. Klasa ta powinna posiadać tylko dwie metody statyczne: `czyPierwsza()` testująca pierwszość liczby i `naCzynnikiPierwsze()` tworząca rozkład liczby na czynniki pierwsze. Wykorzystaj do tego celu *sito Eratostenesa*, modyfikując je w ten sposób, że dla każdej liczby pamiętaj jej najmniejszy dzielnik pierwszy (będzie to trik pomocny przy wyliczaniu rozkładu liczby na czynniki pierwsze). Wymienione funkcje powinny udzielać poprawnych odpowiedzi dla wszystkich liczb typu `long` (weź pod uwagę, że nie możesz utworzyć tak dużego sita, więc zastanów się jak ten problem obejść algorytmicznie). Klasa `LiczbyPierwsze` powinna być tak zdefiniowana, aby nie można było stworzyć jej instancji.

```
public class LiczbyPierwsze
{
    protected final static int POTEGA2 = 21 ;
    protected final static int[] SITO = new int[1<<POTEGA2] ;
    // ... potrzebny jest blok inicjalizacyjny dla sita ...

    public static boolean czyPierwsza (long x)
    { /* ... */ }

    public static long[] naCzynnikiPierwsze (long x)
    { /* ... */ }
}
```

Rozkład liczby ujemnej na czynniki pierwsze ma się rozpoczynać od czynnika `-1`. Rozkład liczb `-1`, `0` i `1` ma być tożsamościowy.

Wskazówka.

Liczba złożona n posiada co najmniej jeden dzielnik pierwszy, który jest $\leq \sqrt{n}$.

Uwaga.

Twój program powinien sobie skutecznie poradzić z liczbą `-9223372036854775808` oraz `9223372036854775783`.

Uwaga.

Program należy skompilować i uruchomić z wiersza poleceń! Podczas kompilacji użyj opcji `-encoding`, aby kompilator umiał poprawnie przetłumaczyć literały znakowe zaszyte w programie do *unikodu*.