

Programowanie obiektowe II

Laboratorium nr 2

Stefan Sokołowski

Zadanie 1:

Jakie działania arytmetyczne i/lub wywołania metod klasy `Math` dają

- sygnał błędu przy kompilacji?
- wyjątek w czasie działania?
- wynik ewidentnie niepoprawny?
- wartość `NaN`?
- wartość `Infinity`?
- wartość `-Infinity`?

Zademonstrować każde z powyższych zachowań prostym programikiem testowym.

Wypróbować w szczególności różne rodzaje dzielenia przez zero, pierwiastki z liczb ujemnych, logarytmy liczb niedodatnich, tangens liczb postaci $k\pi + \frac{\pi}{2}$ (dla $k \in \mathbb{Z}$), itp.

Jakie muszą być wartości podstawy i wykładnika, żeby

```
Math.pow(podstawa, wykładnik)
```

było poprawną liczbą?

Zadanie 2:

Czy 123456787654321123456787654321 jest liczbą pierwszą?

Napisać w Javie program, który sprawdza, czy dana bardzo duża liczba naturalna jest pierwsza czy złożona. Oczywiście nie da się tego zrobić na zwykłych zmiennych typu `int`¹; dlatego proszę wykorzystać klasę `BigInteger`².

P.S. Powyżej podana przykładowa liczba *jest złożona*:

$$123456787654321123456787654321 = 7 * 17636683950617303350969664903$$

Ale to jest tylko pojedynczy przykład, program ma działać dla dowolnych liczb naturalnych.

¹W zmiennej typu `int` można zmieścić liczby nie większe niż ok. 2 miliardy (dwójka z dziewięcioma zerami).

²Por. docs.oracle.com/en/java/javase/11/docs/api/java.base/java/math/BigInteger.html.

Zadanie 3:

Na wykładzie 2 został omówiony program `P0_DecHex.java` zamieniający liczby dziesiętne na szesnastkowe, czerpiący dane z wiersza rozkazowego, bez komend wczytywania z klawiatury³. Proszę napisać i uruchomić program `HexDec.java` działający w odwrotną stronę, od liczb szesnastkowych do dziesiętnych, działający również bez komend wczytywania.

Zadanie 4 (domowe na punkty):

Proszę rozważyć w Jawie następującą metodę szyfrowania:

1. Klucz jest liczbą naturalną $k \geq 2$.
2. Najpierw tekst (złożony ze znaków ASCII) jest zamieniany na ciąg liczb — kodów ASCII znaków tekstu; np.

M	i	l	c	z	,		s	e	r	c	e	!
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
77	105	108	99	122	44	32	115	101	114	99	101	33

3. Następnie ten ciąg liczb jest zamieniany na pojedynczą liczbę naturalną; np.

$$\begin{array}{lcl} 77, 105, 108 & \rightarrow & ((0 \cdot 256 + 77) \cdot 256 + 105) \cdot 256 + 108 \\ & = & 5073260 \end{array}$$

4. Wynikiem szyfrowania ma być iloczyn tej pojedynczej liczby naturalnej przez klucz: $5073260 \cdot k$.

Oczywiście przy szyfrowaniu dłuższych tekstów powstają ogromne liczby, zbyt duże, żeby je zmieścić w zmiennej typu `int`. Dlatego dla zrealizowania tego zadania trzeba skorzystać z klasy `BigInteger`.

Uwaga:

To jest zadanie dla dwóch osób. Należy doprowadzić do tego, żeby mogły one się porozumiewać przez wysyłanie sobie liczb, kodujących wiadomości. Kiedy już będzie działać, proszę to zadanie zaliczyć w następujący sposób:

1. mailem przysłać mi klucz (liczbę co najmniej 7-cyfrową); w mailu podać imiona, nazwiska i numery indeksu obu autorów;
2. ja tym kluczem zaszyfruję jakiś tekścik i odeślę do nadawców;
3. odesłać mi ten mój tekścik w postaci odszyfrowanej.

Uwaga:

Czas na przysłanie (na s.sokolowski@ans-elblag.pl): 2 tygodnie od zajęć z lab.2 — potem już nie przyjmuję rozwiązań. Liczba punktów zależy od oryginalności i elegancji rozwiązania.

³Patrz iis.ans-elblag.pl/~stefan/Dydaktyka/Prog0biII/Prog02/P0_DecHex.java.

Zadanie 5 (domowe specjalne):

W sposób podobny jak w zadaniu 4 oprogramować metodę kluczy publicznych RSA, omawianą na matematyce dyskretniej.

Uwaga:

Pierwszej osobie, która przyśle mi (na `s.sokolowski@ans-elblag.pl`) poprawne rozwiązanie tego zadania, przysługują punkty do zaliczenia laboratorium. Liczba tych punktów zależy od oryginalności i elegancji rozwiązania.
