

Laboratorium 2

Celem ćwiczenia jest doskonalenie umiejętności pisania i uruchamiania prostych programów w języku Java. Programy powinny wypisywać komunikaty w oknie konsoli lub w oknie dialogowym utworzonym za pomocą metody `JOptionPane.showMessageDialog()`. Dane powinny być wczytywane oknie dialogowym utworzonym przez metodę `JOptionPane.showInputDialog()`.

Wstęp

Proszę zapoznać się z poniższymi programami przykładowymi, które ilustrują działanie instrukcji sterujących w języku Java:

PrzykladIf.java, PrzykladSwitch.java, PrzykladWhile.java,
PrzykladFor.java, PrzykladBrak.java, Przyklad.Cont.java.

Zadania (Proszę wykonać co najmniej 4 zadania !)

1. Klasyfikacja wzrostu

Napisz program, który w pętli pyta się o wzrost osób, a następnie klasyfikuje go do jednego z poniższych przedziałów:

- i. **niski** - poniżej 150 cm,
- ii. **średni** - od 151 do 180 cm
- iii. **wysoki** - powyżej 180 cm.

Program powinien zakończyć działanie po wprowadzeniu wzrostu mniejszego lub równego zero. Dane wprowadź z klawiatury za pomocą metody

`JOptionPane.showInputDialog`.

Komunikaty o wzroście wyświetlaj na ekranie za pomocą metod

`System.out.println` oraz `JOptionPane.showMessageDialog`.

W przypadku zastosowania standardowego okienka dialogowego, program należy zakończyć wywołaniem metody `System.exit(0);`.

2. Zliczanie wczytywanych z klawiatury liczb

Napisz program, który w pętli wczytuje liczby całkowite i zlicza:

- i. ilość wprowadzonych liczb,
- ii. sumę wprowadzonych liczb,
- iii. wartość średnią wprowadzonych liczb.
- iv. liczbę największą wśród wprowadzonych liczb.

Wprowadzanie powinno zakończyć się gdy:

- ilość wprowadzonych liczb przekroczy 5,
- albo suma wszystkich wprowadzonych liczb przekroczy 100.

Do wczytywania liczb użyj metody `JOptionPane.showInputDialog`,

Po wczytaniu każdej kolejnej liczby program powinien wyświetlać w oknie konsoli ilość oraz sumę dotychczas wczytanych liczb. Po zakończeniu wczytywania program powinien wyświetlić dodatkowo wartość średnią wprowadzonych liczb. oraz liczbę największą .

Dane w oknie konsoli proszę wypisywać za pomocą metody `System.out.println`.

3. Rysowanie trójkątów i liter

Wysokość trójkąta lub litery (ilość wierszy) proszę wczytywać za pomocą metody metody `JOptionPane.showInputDialog`,

- a. Napisz program wyświetlający w oknie konsoli jeden z trójkątów o wysokości zadawanej z klawiatury

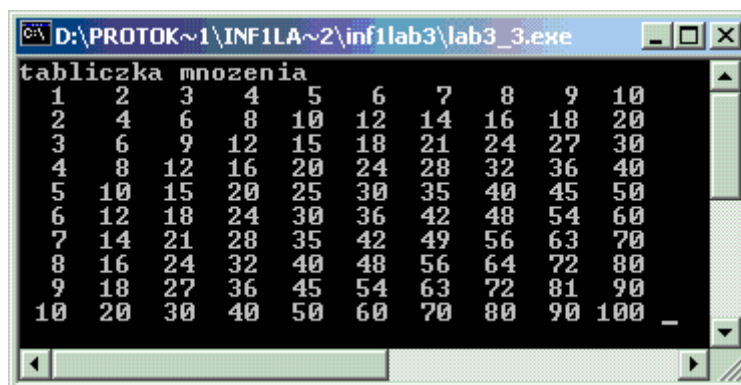
```
*           *           *
**          **          ***
***         ***         *****
****        ****        *******
*****       *****      *********
```

- b. Napisz program wyświetlający na ekranie jedną z liter "M", "N" lub "Z" o wysokości zadawanej z klawiatury

```
#           #           #           #           #####
##          ##          ##          #           #
# #         # #         # #         #           #
# # #       # #       # # #         #           #
# # #       # #       # # #         #           #
# # #       # #       # # #         #####
```

4. Rysowanie tabliczki mnożenia

Napisz program, który wyświetla na ekranie tabliczkę mnożenia dla liczb całkowitych od 1 do 10, podobnie jak na zrzucie z ekranu. Osoby ambitne powinny napisać program tak, by liczba wierszy i kolumn była zadawana z klawiatury. za pomocą metody `JOptionPane.showInputDialog`



5. Obliczanie *nwd* oraz *nww*

Proszę wczytywać za pomocą metody metody `JOptionPane.showInputDialog`, dwie liczby całkowite a i b . Dla podanych liczb należy obliczyć i wypisać na ekranie największy wspólny dzielnik *nwd* oraz najmniejszą wspólną wielokrotność *nww*.

Wskazówki:

- *nwd* jest to największa liczba całkowita, dla której:
 $a \% nwd = 0$ oraz $b \% nwd = 0$,
gdzie $x \% y$ jest resztą z dzielenia x przez y .
- *nww* jest to najmniejsza liczba całkowita, dla której:
 $nww \% a = 0$ oraz $nww \% b = 0$.
- Np. dla $a=6$ oraz $b=9$
największy wspólny dzielnik *nwd* wynosi 3,
najmniejsza wspólna wielokrotność *nww* wynosi 18.

6. Obliczanie wartości stałych matematycznych π oraz e

Napisz program, który wczytuje z klawiatury pożądaną dokładność obliczeń *eps* ($eps < 0.1$), a następnie oblicza zadaną dokładnością wartość sumy następujących szeregów:

$$S1 = 4 * \sum_{k=1}^{\infty} \left[(-1)^{k+1} * \frac{1}{2k-1} \right] = 4 * \left[\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots \right]$$

$$S2 = \sum_{k=0}^{\infty} \frac{1}{k!}$$

Wskazówki:

- Sumowanie szeregu należy zakończyć, gdy wartość bezwzględna pojedynczego wyrazu szeregu jest mniejsza niż zadana dokładność obliczeń *eps*.
- Suma szeregu *S1* jest zbieżna do wartości stałej π
- suma szeregu *s2* jest zbieżna do wartości stałej e .

7. Obliczanie funkcji $\pi(n)$

Napisz funkcję, która wczytuje z klawiatury dodatnią liczbę całkowitą n , a następnie dla podanej liczby n oblicza wartość funkcji $\pi(n)$ czyli ilość liczb pierwszych mniejszych lub równych n . Dla $n > 3$ postać funkcji $\pi(n)$ jest następująca:

$$\pi(n) = -1 + \sum_{j=3}^n \left((j-2)! - j \left\lfloor \frac{(j-2)!}{j} \right\rfloor \right)$$

gdzie $\lfloor x \rfloor$ oznacza część całkowitą x zwracaną przez funkcję `floor(x)`, natomiast $x!$ oznacza silnię liczby x .

Wskazówki:

- Funkcja `floor(x)` w języku Java jest obliczana za pomocą metody `Math.floor(x)`.

8. Obliczanie ciągu Hofstadtera

Napisz funkcję, która wczytuje z klawiatury dodatnią liczbę całkowitą n_0 , a następnie generuje ciąg liczb całkowitych zgodnie z następującym wzorem:

$$n_{i+1} = \begin{cases} n_i/2 & \text{jesli } n_i \text{ jest parzyste,} \\ 3*n_i+1 & \text{jesli } n_i \text{ jest nieparzyste.} \end{cases}$$

Obliczenia należy zakończyć, gdy kolejna obliczona liczba n_{i+1} będzie równa 1. Funkcja powinna wyświetlać na ekranie w kolejnych wierszach następujące informacje: numer iteracji $i+1$, wartość n_i , {parzyste|nieparzyste}, wartość n_{i+1}

np. dla $n_0 = 10$ na ekranie powinno pojawić się:

```
1, 10, parzyste, 5
2, 5, nieparzyste, 16
3, 16, parzyste, 8
4, 8, parzyste, 4
5, 4, parzyste, 2
6, 2, parzyste, 1
```

Wskazówki:

- Algorytm ten zaproponował Hofstadter. Co ciekawe, nie ma dowodu, że algorytm zawsze się zatrzymuje. Jaki jest ciąg wynikowy dla liczby 27 ?