

Laboratorium 0 - Podstawy programowania

Zadania:

1. Stwórz pakiet pl.polsl.ip.podstawy
2. Stwórz klasę Main oraz odpowiednią metodę uruchamieniową main

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

3. Stwórz metodę statyczną

```
public static void liczbyParzyste(int start, int koniec) {  
  
}
```

Której zadaniem jest wypisanie na ekranie wszystkich liczb parzystych z przedziału od start do koniec. W tym celu skorzystaj z pętli for oraz warunku if. Aby sprawdzić czy dana liczba podzielna jest przez 2 wykorzystaj operator reszty z dzielenia %, i każdorazowo sprawdź czy reszta z dzielenia jest równa 0

4. W metodzie main() umieść wywołanie funkcji liczby parzyste. I sprawdź które z liczb z przedziału 10 do 19 są liczbami parzystymi.
5. Rozbuduj wcześniej zbudowany program, tak aby sprawdzał czy liczby z danego zakresu są podzielne przez liczbę k , gdzie k podawana jest jako argument wywołania funkcji/metody
6. Korzystając ze zbudowanej metody dla liczb z zakresu 10 do 19 sprawdź które liczby są podzielne przez 3
7. Stwórz metodę która będzie znajdowała wszystkie liczby pierwsze z podanego zakresu. Liczby pierwsze to takie które podzielne są jedynie przez 1 i przez samą siebie. Wykorzystaj swój program i znajdź wszystkie liczby pierwsze z zakresu 1 do 100
8. Stwórz funkcję liczącą silnię korzystając z pętli oraz rekurencyjnie. W pierwszym przypadku metoda powinna być zadeklarowana jako

```
public static int silniaPetla(int b) , a w drugim  
public static int silniaRekurencja(int b)
```

Przy czym do realizacji pierwszej części wykorzystaj pętlę, a do realizacji drugiej wykorzystaj wywołanie rekurencyjne. Wywołanie rekurencyjne to takie, w którym funkcja wywołuje samą siebie.

9. W metodzie main, dokonaj wywołania obydwu funkcji silnia obliczając ją dla wartości 4, 7, 10, 13, przy czym wywołanie funkcji umieść w pętli, która iteruje po tych wartościach.
10. W metodzie main stwórz tablicę liczb typu int[] tab = {5, 3, 1, 4, 6, 2, 7}. Następnie stwórz funkcję `public static int[] sortowanieBabelkowe(int[] tab)`, której zadaniem jest implementacja algorytmu sortowania bąbelkowego. Jej wynikiem powinna być nowa tablica liczb typu int zawierająca posortowane wartości. Algorytm sortowania bąbelkowego:
„[*] Polega na porównywaniu dwóch kolejnych elementów i zamianie ich kolejności, jeżeli zaburza ona porządek, w jakim się sortuje tablicę. Sortowanie kończy się, gdy podczas kolejnego przejścia nie dokonano żadnej zmiany.”

[*] – Algorytm sortowanie bąbelkowego – źródło wikipedia:
http://pl.wikipedia.org/wiki/Sortowanie_babelkowe

„[*] Ciąg wejściowy [4, 2, 5, 1, 7]. Każdy wiersz symbolizuje wypchnięcie kolejnego największego elementu na koniec ("wypłynięcie największego bąbelka"). Niebieskim kolorem oznaczono końcówkę ciągu już posortowanego.”

$$\begin{array}{l}
 \underbrace{[4, 2, 5, 1, 7]}_{4 > 2} \rightarrow \underbrace{[2, 4, 5, 1, 7]}_{4 < 5} \rightarrow \underbrace{[2, 4, 5, 1, 7]}_{5 > 1} \rightarrow \underbrace{[2, 4, 1, 5, 7]}_{5 < 7} \\
 \underbrace{[2, 4, 1, 5, 7]}_{2 < 4} \rightarrow \underbrace{[2, 4, 1, 5, 7]}_{4 > 1} \rightarrow \underbrace{[2, 1, 4, 5, 7]}_{4 < 5} \\
 \underbrace{[2, 1, 4, 5, 7]}_{2 > 1} \rightarrow \underbrace{[1, 2, 4, 5, 7]}_{2 < 4} \\
 \underbrace{[1, 2, 4, 5, 7]}_{1 < 2}
 \end{array}$$