

# Zadania z programowania w języku Java dla II roku Informatyki.

## Kolekcje

1. Napisz metodę

```
public static void redukuje(LinkedList<String> pracownicy, int n)
```

która usuwa z listy nazwisk pracowników co  $n$ -tego pracownika. W funkcji `main` przetestuj działanie metody `redukuje`.

2. Zmodyfikuj metodę z poprzedniego zadania, tak aby była ona metodą generyczną ze względu na typ elementów listy.

3. Napisz metodę

```
public static void odwroc(LinkedList<String> lista)
```

która odwraca kolejność elementów listy. W funkcji `main` przetestuj działanie metody `odwroc`.

4. Zmodyfikuj metodę z poprzedniego zadania, tak aby była ona metodą generyczną ze względu na typ elementów listy.

5. Użyj stosu (obiektu klasy `Stack<String>`, aby odwrócić słowa w zdaniu. Czytaj kolejne słowa, aż do napotkania słowa, które kończy się kropką, umieszczając je na stosie. Następnie wypisz kolejne słowa zdejmując je ze stosu. Zakończ program, gdy na wejściu nie ma już żadnych słów. Przykładowo, dla wejścia postaci

**Ala ma kota. Jej kot lubi myszy.**

program powinien wypisać

**Kota ma ala. Myszy lubi kot jej.**

Zadbaj o dużą literę na początku i o kropkę na końcu każdego zmienionego zdania.

6. Wczytaj nieujemną liczbę całkowitą i podziel ją na poszczególne cyfry. Przykładowo, po wczytaniu liczby 2015 program powinien wypisać: 2 0 1 5. Ostatnią cyfrę liczby  $n$  daje wyrażenie  $n \% 10$ . Ponieważ w ten sposób uzyskamy cyfry w odwrotnej kolejności, należy je najpierw umieścić na stosie, a następnie wypisać kolejne cyfry umieszczone na stosie zdejmując je ze stosu.

7. Zaimplementuj znaną starożytnym Grekom metodę obliczania liczb pierwszych nazywaną sitem Eratostenesa. Metoda ta tworzy zbiór wszystkich liczb pierwszych nie większych od  $n$ . Wczytaj liczbę do zmiennej  $n$ , po czym wstaw do zbioru `pierwsze` wszystkie liczby od 2 do  $n$ . Następnie usuń wszystkie wielokrotności liczby 2 (z wyjątkiem 2), potem wszystkie wielokrotności liczby 3 (z wyjątkiem 3) i tak dalej aż do pierwiastka z liczby  $n$ . Po zakończeniu obliczeń wypisz zbiór `primes`.

8. Napisz generyczną statyczną metodę `print`, której argumentem jest dowolny obiekt implementujący interfejs `Iterable<E>`. Metoda `print` wypisuje elementy swojego argumentu oddzielając je przecinkami. W funkcji `main` przetestuj działanie metody `print` dla obiektów kilku różnych klas implementujących interfejs `Iterable<E>`.