

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

dr Agnieszka Zbrzezny

## 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. W tablicy typu `ArrayList` zawierającej ciągi znaków zamień wszystkie ciągi na zapisane wielkimi literami. Wykonaj to za pomocą

- (a) iteratora,
- (b) pętli przechodzącej po wartościach indeksu
- (c) metody `replaceAll` z interfejsu `Collections`.

6. 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>`.

7. Zadanie ze slajdu nr 17.

8. Zaimplementuj znaną starożytnym Grekom metodę obliczania liczb pierwszych nazywaną sitem Eratostenesa. Metoda ta tworzy zbiór wszystkich liczb pierwszych nie większych od *n*. Wczytaj liczbę *n* 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**.

9. Zaimplementuj listę zadań do zrobienia. Każde zadanie ma priorytet (będący dodatnią liczbą jednocyfrową) oraz opis. Zadanie jest tym pilniejsze do wykonania im mniejszą liczbą jest jego priorytet. Gdy użytkownik wpisze polecenie **dodaj priorytet opis**, program dodaje nowe zadanie. Gdy użytkownik wprowadzi polecenie **następne**, program usuwa najbardziej pilne zadanie. Polecenie **zakończ** kończy pracę programu. W rozwiązaniu użyj kolejki priorytetowej.