

Wrocławska Wyższa Szkoła Informatyki Stosowanej

Przedmiot	Programowanie Java (lab)
Semestr	Zima 2017/2018

ZADANIE 1

Deklaracja i inicjalizacja kolekcji, przeglądanie wartości kolekcji, wybór odpowiedniej implementacji

Wymagane wiadomości wstępne:

- Znajomość Java Collections Framework

Przebieg ćwiczenia:

Stwórz program:

1. Zadeklaruj kolekcję zawierającą typy całkowite Integer i wypełnij ją wartościami:
1, 2, 21 13, 8, 7 ,3, 2, 45, 23, 8
2. Wyświetl na konsoli wszystkie parzyste elementy kolekcji.

```
package pl.wsis.java;

public class Zadanie1{

    public static void main(String[] args) {
        Collection<Integer> liczby = .....;
        // dodaj podane w instrukcji elementy do kolekcji

        // wykorzystaj pętlę 'for each' lub iterator, aby przejrzeć elementy kolekcji
        // dla każdego elementu sprawdź czy jest parzysty i wyświetl
        // [UZUPEŁNIJ KOD]
    }
}
```

ZADANIE 2

Odpowiednia implementacja metody equals i hashCode

Wymagane wiadomości wstępne:

- Znajomość Java Collections Framework

Przebieg ćwiczenia:

1. Zaimplementuj odpowiednio klasę Tag, aby poniżej napisany program wyświetlił na konsoli "Dobra implementacja".
2. Pamiętaj, aby zaimplementować konstruktor przyjmujący zmienną typu String.
3. Pamiętaj, aby odpowiednio zaimplementować metody equals i hashCode.

```
package pl.wsis.java;

public class Zadanie2{

    public static void main(String[] args) {
        Set<Tag> tagi = new HashSet<Tag>();
        tagi.add(new Tag("nowy"));
        tagi.add(new Tag("nowy"));
        tagi.add(new Tag("nowy"));
        tagi.add(new Tag("przeczytane"));
        if(tagi.size()==2){
            System.out.println("Dobra implementacja");
        } else {
            System.out.println("Sprawdź jeszcze raz implementację");
        }
    }
}
```

ZADANIE 3

Sortowanie, porównywanie obiektów, sortowanie

Wymagane wiadomości wstępne:

- Znajomość Java Collections Framework

Przebieg ćwiczenia:

1. Wykorzystaj szkielet programu z metodą `generateRandomNumbers()` i utwórz listę z 10 losowymi elementami.
2. Wyświetl pozycję na której znajduje się najmniejszy i największy element listy.
3. Posortuj listę rosnąco ale tak, aby wszystkie parzyste elementy listy występowały przed nieparzystymi:
Przykład:
(4,5,6,7,1,2,3) -> po sortowaniu (2,4,6,1,3,5,7)
4. Wyświetl pozycję na której znajduje się najmniejszy i największy element

```
package pl.wsis.java;

public class Zadanie3 {

    static Random rnd = new Random();

    public static void main(String[] args) {
        // UZUPEŁNIJ KOD
    }

    private static List<Integer> generateRandomNumbers() {
        List<Integer> lista = new ArrayList<Integer>();
        int i=0;
        while(i<10){
            lista.add(rnd.nextInt());
            i++;
        }
        return lista;
    }
}
```

ZADANIE 4Kolekcja Map, korzystanie z metod klasy *String***Wymagane wiadomości wstępne:**

- Znajomość Java Collections Framework

Przebieg ćwiczenia:

Napisz program, który dla kolekcji zawierającej nazwy wszystkich miesięcy wyświetli statystykę częstotliwości występowania liter w nazwach miesięcy,

np: dla "Ala ma kota" wyświetli

a - 4

l - 1

m - 1

k - 1

o - 1

t - 1

1. Zadeklaruj kolekcję i zainicjalizuj kolekcję zawierającą nazwy miesięcy.
2. Przejrzyj elementy kolekcji i dla każdego elementu zaktualizuj dane dotyczące statystyki występowania liter -> do przechowywania statystyki wykorzystaj mapę.
3. Po przejrzaniu wszystkich elementów wyświetl ostateczną statystykę.
4. Korzystając z mapy ze statystyką sprawdź, czy nazwy miesięcy zawierają literę 'ć'.
5. Korzystając z mapy ze statystyką sprawdź, czy istnieje litera występująca dokładnie 13 razy w nazwach wszystkich miesięcy.