

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра «Електронних обчислювальних машин»



Звіт
з лабораторної роботи № 7
з дисципліни: «Кросплатформенні засоби програмування»
на тему: «ПАРАМЕТРИЗОВАНЕ ПРОГРАМУВАННЯ»
Варіант 13

Виконав: ст.гр. КІ-34
Окілка Я. А.

Прийняв:
доцент кафедри ЕОМ
Іванов Ю. С.

Мета: оволодіти навиками параметризованого програмування мовою Java.

Завдання

Словник (тип даних)

Виконання:

```
package com.company;

import java.util.*;

/**
 *
 * Це клас який містить статичний метод main який являється точкою входу в
 програму
 * @author Okilka Yaroslav
 */

public class VVocabulary {
    /**
     * Метод main є точкою входу в програму
     * @param args
     */

    public static void main(String[] args){
        Vocabulary <? super Words> MyVocabulary = new Vocabulary <Words>(); //
створення нового об'єкту класу

        MyVocabulary.AddData(new ProWord("washdd", "вош" , "мити"));
        MyVocabulary.AddData(new Word("tree" , "дерево"));
        //MyVocabulary.AddData(new Word("horsecdf" , "хорс"));
        // MyVocabulary.AddData(new ProWord("truck", "трак" , "вантажівка"));

        Words res = MyVocabulary.findMax();
        System.out.print("The longest word: \n");
        res.print();

    }
}

/**
 * Клас реалізує Словник
 * @author Okilka Yaroslav
 * @param <T>
 */
class Vocabulary <T extends Words>{    //параметризований клас
    private ArrayList<T> arr;

    public Vocabulary(){ //конструктор
        arr = new ArrayList<T>();
    }

    public T findMax(){
        if (!arr.isEmpty())
        {
            T max = arr.get(0);
```

```

        for (int i=1; i< arr.size(); i++)
        {
            if ( arr.get(i).compareTo(max) > 0 )
                max = arr.get(i);
        }
        return max;
    }
    return null;
}

public void AddData(T data){
    arr.add(data);
    System.out.print("Element added: ");
    data.print();
}

}

/**
 * Це інтерфейс який описує 2 методи
 * @author Okilka Yaroslav
 */
interface Words extends Comparable<Words>{
    public int get_WordLength();
    public void print();
}

/**
 * Цей клас моделює словник з транскрипцією
 * @author Okilka Yaroslav
 */
class ProWord implements Words
{
    private String Word;
    private int WordSize;
    private String Translations;
    private String Transcript;

    public ProWord(String Word, String Transcript, String Translations){ //
конструктор
        this.Word = Word;
        this.Transcript = Transcript;
        this.Translations = Translations;
        WordSize = Word.length();
    }

    /**
     * Метод повертає слово
     * @return Word
     */
    public String get_Word(){
        return Word;
    }

    /**
     * Метод встановлює значення поля Word
     * @param Word
     */
    public void set_Word(String Word){
        this.Word = Word;
    }
}

```

```

/**
 * Метод повертає довжину слова
 */
public int get_WordLenght() {
    WordSize = Word.length();
    return Word.length();
}

/**
 * Метод приймає значення Transcript
 * @param Transcript
 */
public void set_Transcrip (String Transcript) {
    this.Transcript = Transcript;
}

/**
 * метод повертає значення параметра Transcript
 * @return Transcript
 */
public String get_Transcript() {
    return Transcript;
}

/**
 * Метод приймає значення Translations
 * @param Translations
 */
public void setTranslations(String Translations) {
    this.Translations = Translations;
}

/**
 * метод повертає значення параметра Translations
 * @return Translations
 */
public String getTranslations() {
    return Translations;
}

public int compareTo(Words p) {
    Integer s = Word.length();
    return s.compareTo(p.get_WordLenght());
}

/**
 * Вивід інформації про слово
 */
public void print() {
    System.out.print("Word: " + Word + ", Size: " + WordSize + ",
Translations: " + Translations + ", Transcript: " + Transcript + ";\n");
}
}

/**
 *Клас моделює словник без транскрипції
 * @author Okilka Yaroslav
 */

class Word implements Words {
    private String Word;
    private int WordSize;

```

```

private String Translation;

/**
 * конструктор
 * @param Word - слово
 * @param Translation - переклад
 */
public Word(String Word, String Translation){
    this.Word = Word;
    this.Translation = Translation;
    WordSize = Word.length();
}

/**
 * метод повертає значення поля Word
 * @return Word - слово
 */
public String getWord(){
    return Word;
}

/**
 * метод встановлює значення поля Word
 * @param w - слово
 */
public void setWord(String w){
    Word = w;
}

/**
 * метод встановлює значення поля Translation
 * @param Translation - переклад
 */
public void SetTranslations(String Translation){
    this.Translation= Translation;
}

/**
 * Метод повертає значення поля Translation
 * @return Translation
 */
public String getTranslations(){
    return Translation;
}

public int compareTo(Word p){
    Integer s = WordSize;
    return s.compareTo(p.get_WordLenght());
}

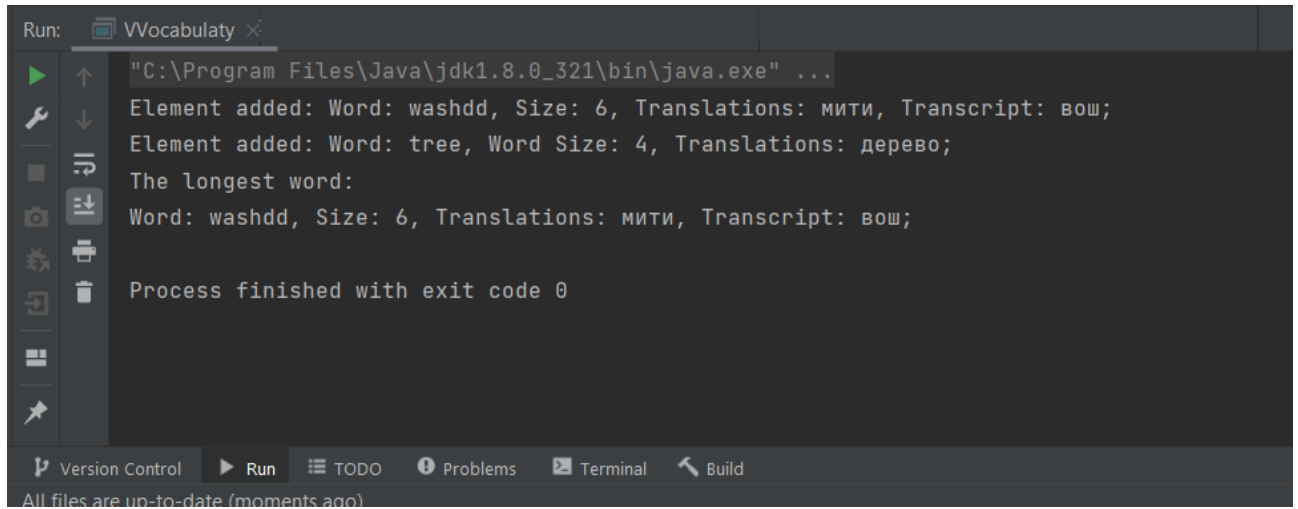
/**
 * метод повертає значення поля WordSize
 */
public int get_WordLenght() {
    WordSize = Word.length();
    return WordSize;
}

/**
 * Метод виводить інформацію про слово
 */
public void print(){
    System.out.print("Word: " + Word + ", Word Size: " + WordSize + ",
Translations: " + Translation + ";\n");
}

```

```
}  
}
```

Вивід програми:



```
Run: VVocabulary x  
"C:\Program Files\Java\jdk1.8.0_321\bin\java.exe" ...  
Element added: Word: washdd, Size: 6, Translations: мити, Transcript: вош;  
Element added: Word: tree, Word Size: 4, Translations: дерево;  
The longest word:  
Word: washdd, Size: 6, Translations: мити, Transcript: вош;  
Process finished with exit code 0  
Version Control Run TODO Problems Terminal Build  
All files are up-to-date (moments ago)
```

Висновки: На даній лабораторній роботі я оволодів навиками параметризованого програмування мовою Java.