

Національний технічний університет України
«Київський політехнічний інститут»
Факультет інформатики и обчислювальної техніки
Кафедра обчислювальної техніки

Лабораторна робота №3
«Організація мультипоточної обробки даних в Java»

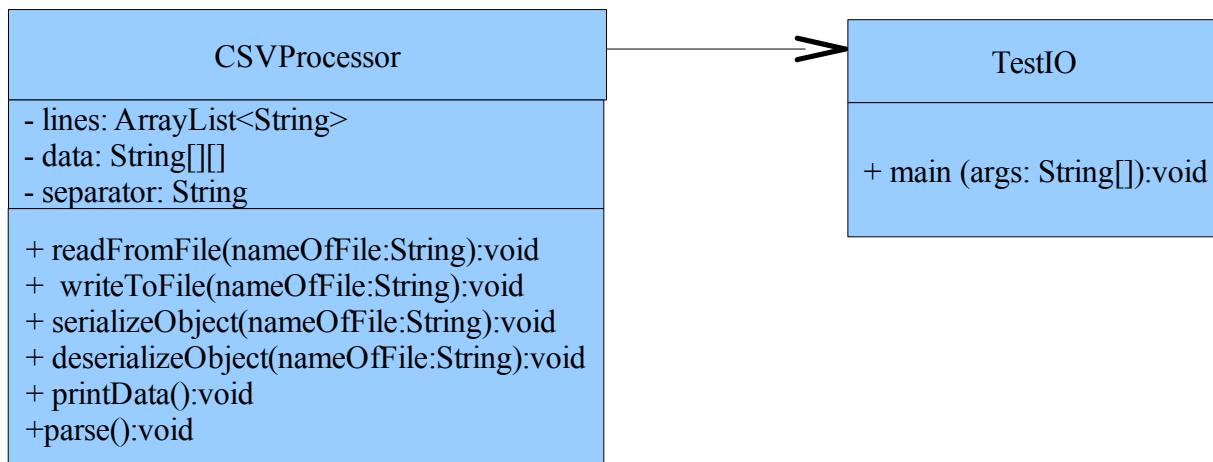
Виконав: ст. гр. ІО-92
Петрук В.О.
Перевірів: Болдак А. А.

Мета: Отримання базових навичок з використання мультитотокового середовища в Java. Поглиблене вивчення механізму виключень. Ознайомлення з засобами Java для розбору текстових даних.

Завдання

1. Вивчити механізм виключень (Exceptions) в Java. Вміти створювати нові класи виключень, інстанціювати виключення, маскувати місце виникнення виключення та його тип. Знати призначення інтерфейсу Throwable.
2. Ознайомитись з засобами розбору даних. Вміти використовувати класи StreamTokenizer, StringTokenizer, Scanner.
3. Вивчити механізм організації множини легких процесів - потоків (threads). Вміти створювати потоки за допомогою класу Thread та інтерфейсу Runnable.
4. Вивчити механізми синхронізації у мультитотоковому середовищі.
5. Вивчити стани потоку. Знати причини та засоби запобігання блокуванню потоків. Вміти керувати станами потоків, змінювати їх пріоритет.
6. Створити клас виключень CSVParseException. Обґрунтувати його розміщення в ієрархії виключень Java.
7. В класі CSVProcessor створити метод parse(), який виконує розбір завантажених в попередній роботі даних (з агрегованого об'єкту класу ArrayList) в двовимірний масив. Під час розбору перехоплювати всі виключення. В разі виникнення виключення, або невідповідності даних формату (не є CSV, не вистачає значень або є зайві, тощо) генерувати виключення класу CSVParseException.
8. Модифікувати тестовий клас TestIO. В методі main() додати створення нового потоку розбору даних (за допомогою метода parse()), при зупинці основного потоку до отримання результатів розбору.

UML-діаграма



Класс CSVProcessor

```
package com.lab111.Lab3;
```

```
import java.util.ArrayList;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;

/**
 *A class for the reading, writing CSV-files
 *from the array and serializing/deserializing objects
 * @author yozhik
 */
@SuppressWarnings("serial")
public class CSVProcessor implements Serializable{
```

```

/**
 * array for store strings of CSV-file
 */
private ArrayList<String> lines=new ArrayList<String>();
/**
 * massive of parsing data from CSVFile
 */
private String[][] data;
/**
 * separator in CSV-file's lines
 */
private String separator=",";
/**
 * A method for the reading CSV-files
 * @param nameOfFile
 * @throws IOException
 */
public void readFromFile(String nameOfFile) throws
FileNotFoundException,IOException{
    BufferedReader file=new BufferedReader(new FileReader(nameOfFile));
    String line="";
    while ((line=file.readLine())!=null){
        lines.add(line);
    }
    if (file!=null)
        file.close();
}
/**
 * The method for the writing CSV-files
 * from the array
 * @param nameOfFile
 * @throws IOException
 */
public void writeToFile(String nameOfFile) throws FileNotFoundException,IOException{
    BufferedWriter file=new BufferedWriter(new FileWriter(nameOfFile));
    for(int i=0; i<lines.size();i++){
        file.write(lines.get(i)+"\n");
    }
    if(file!=null){
        file.close();
    }
}
/**
 * Method for the serializing object to the file
 * @param nameOfFile
 * @throws IOException
 */
public void serializeObject(String nameOfFile) throws
FileNotFoundException,IOException{
    ObjectOutputStream object=new ObjectOutputStream(new
FileOutputStream(nameOfFile));
    object.writeObject(lines);
    if(object!=null)
        object.close();
}
/**
 * The method for the deserializing object from the file
 * @param nameOfFile
 * @throws IOException, ClassNotFoundException
 */
@SuppressWarnings("unchecked")
public void deserializeObject(String nameOfFile) throws FileNotFoundException,
IOException, ClassNotFoundException{
    ObjectInputStream file=new ObjectInputStream(new FileInputStream(nameOfFile));
    ArrayList<String> object=(ArrayList<String>)file.readObject();
    for( int i=0; i<object.size();i++){
        lines.add(object.get(i));
    }
}

```

```

    }
    if(file!=null)
        file.close();
}
/**
 * method for the parsing data form CSVFile's lines
 * to the double array 'data'
 */
public void parse() throws CSVParseException{
    String[] headlines=lines.get(0).split(separator);
    data=new String[lines.size()][headlines.length];
    data[0]=headlines;
    for(int i=1; i<lines.size(); i++){
        data[i]=lines.get(i).split(separator);
        if ( (data[i].length)!= (data[i-1].length))
            throw new CSVParseException("Wrong format of CSV-file!");
    }
}
/**
 * method for the printing parsing data
 * from CSVFile in table form
 */
public void printData(){
    int[] L00=new int[data[0].length];
    /**
     * lengthOf0verviews has max value
     */
    for(int i=0; i<data.length;i++){
        for(int j=0; j<L00.length; j++){
            if(L00[j]<data[i][j].length()){
                L00[j]=data[i][j].length();
            }
        }
    }
    int resLength=0;
    for(int i=0;i<L00.length;i++){
        L00[i]=L00[i]+4;
        resLength+=L00[i];
    }
    resLength+=data[0].length+1;//общая длина учитывая разделители
    for(int i=0; i<resLength;i++){
        System.out.print("_");
    }
    System.out.println();
    for(int i=0;i<data.length;i++){
        System.out.print("|");
        for(int j=0; j<data[0].length;j++){
            String field=data[i][j];
            int pr_length=(L00[j]-field.length())/(2);
            int last_length=L00[j]-field.length()-pr_length;
            for(int k=0;k<pr_length;k++){
                System.out.print(" ");
            }
            System.out.print(field);
            for(int k=0;k<last_length;k++){
                System.out.print(" ");
            }
            System.out.print("|");
        }
        if(i==0){
            System.out.println();
            for(int k=0; k<resLength;k++){
                System.out.print("-");
            }
        }
        System.out.println();
    }
}

```

```

        for(int k=0; k<resLength;k++){
            System.out.print("Bï");
        }
    }
}

Класс TestIO
package com.lab111.Lab3;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStreamReader;
/**
 * Class implements the user interaction and
 * class CSVProcessor. All exceptions are
 * processed in this class.
 * @author yozhik
 */
public class TestIO{
    /**
     * Main class
     * @param args
     * @throws IOException
     * @throws ClassNotFoundException
     */
    public static void main (String[] args) throws FileNotFoundException,IOException,
    ClassNotFoundException, CSVParseException{
        CSVProcessor csv=null;
        File file=new File("serialized.dat");
        if (file.exists()){
            csv=new CSVProcessor();
            System.out.print("Deserializing file 'serialized.dat' ... ");
            try{
                csv.deserializeObject("serialized.dat");
            }catch(FileNotFoundException e1){
                System.out.println("FileNotFoundException!!! The object was not
deserialized!!!");
                throw e1;
            }
            catch(ClassNotFoundException e2){
                System.out.println("ClassNotFoundException!!! The object was not
deserialized!!");
                throw e2;
            }
            catch(IOException e3){
                throw e3;
            }
            System.out.println("Done!");
        }
        else{
            BufferedReader consoleIn=new BufferedReader(new
InputStreamReader(System.in));
            int counter=0;
            boolean done=true;
            while(done){
                try{
                    csv=new CSVProcessor();
                    System.out.println("Enter name of CSV-file:");
                    String nameOfFile = consoleIn.readLine();
                    System.out.print("Reading data from file '"+nameOfFile+" ...
");

                    csv.readFromFile(nameOfFile);
                    System.out.println("Done!");
                    System.out.print("Serializing of array to the file
'serialized.dat' ... ");

                    csv.serializeObject("serialized.dat");
                    System.out.println("Done!");
                }
            }
        }
    }
}

```

```

        done=false;
    }catch(FileNotFoundException e1){
        System.out.println("Failed!");
        System.out.println("Wrong name of file! Please try again");
        counter++;
        if (counter==3)
            throw e1;
    }
    catch(IOException e2){
        throw e2;
    }
}
if (csv!=null){
}
// parsing lines from CSV-file
final CSVProcessor parseCSV=csv;
Runnable parseRun = new Runnable() {
    @Override
    public void run() {
        try {
            System.out.print("Parsing the array ... ");
            parseCSV.parse();
            System.out.println("Done!");
        }catch (CSVParseException e) {
            System.out.println("Failed!");
            System.out.println("Wrong format of CSV-file. Please, point
right CSV-file");
        }
    }
};
Thread parseThread = new Thread(parseRun);
parseThread.start();
try {
    parseThread.join();
} catch (InterruptedException e) {
    System.out.println("Interrupted!");
}
//Printing data in table format
try{
    csv.printData();
}catch(ArrayIndexOutOfBoundsException e){
    System.out.println("Wrong format of table!");
}
}
}

```

Клас CSVParseException

```

package com.lab111.Lab3;
/**
 * Signals that a data format error has occurred.
 * @author yozhik
 */
@SuppressWarnings("serial")
public class CSVParseException extends Exception{
    CSVParseException(){
        super();
    }
    CSVParseException(String message){
        super(message);
    }
}

```