

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

**ЛАБОРАТОРНА РОБОТА №2**  
«Організація вводу/виводу засобами Java (java.io, java.nio)»

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

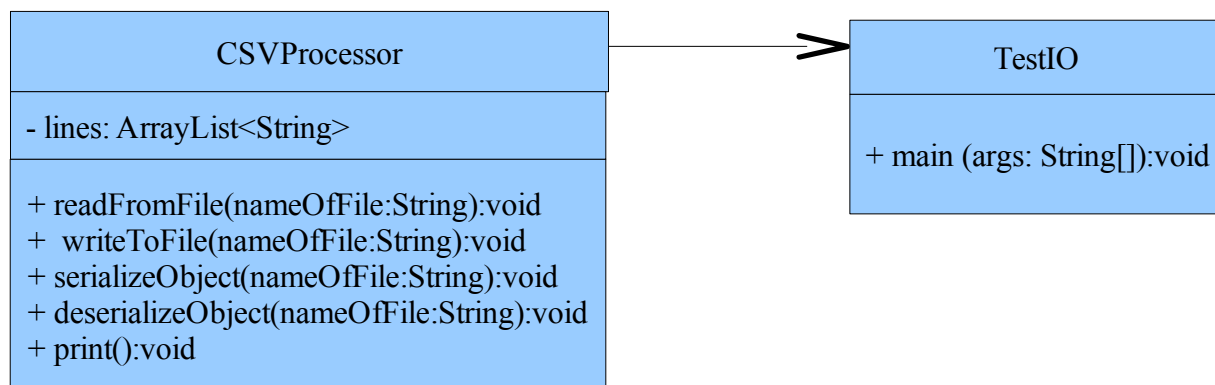
Київ 2011

**Мета:** Отримання базових навичок з використання мови XML. Вивчення функціональних можливостей пакетів java.io та java.nio. Ознайомлення з механізмом підтримки стійкості даних. Отримання навичок з організації вводу/виводу та серіалізації.

### Завдання

1. Вивчити структуру і функціональні можливості пакетів вводу/виводу java.io та java.nio. Володіти концепцією потоків вводу/виводу (байтових та символьних), механізмом використання буферів і каналів.
2. Ознайомитись з засобами підтримки стійкості даних. Вільно володіти механізмом серіалізації (інтерфейси Serializable та Externalizable)
3. Ознайомитись з механізмом виключень (Exceptions). Вміти перехоплювати виключення.
4. Створити клас CSVProcessor для роботи з CSV-файлами. У класі CSVProcessor реалізувати методи завантаження/вивантаження CSV-файлу в/з агрегований об'єкт класу ArrayList, в якому знаходяться рядки CSV-файлу. Реалізувати методи серіалізації/десеріалізації агрегованого об'єкту класу ArrayList в файлі serialized.dat.
5. Створити тестовий клас TestIO, в методі main() якого виконати наступні дії (за допомогою класу CSVProcessor і класів з пакетів java.io, java.nio):
  - А – перевірити чи існує файл serialized.dat, якщо існує – виконати десеріалізацію даних і перейти до пп. Д;
  - Б – ввести з консолі ім'я CSV-файлу;
  - В – перехоплюючи виключення вводу/виводу спробувати завантажити CSV-файл, в разі виникнення виключень повторювати пп. Б і В тричі після чого викинути виключення на рівень вище (перервавши виконання програми)
  - Г – виконати серіалізацію даних в файл serialized.dat;
  - Д – вивести на консоль дані з агрегованого об'єкту класу ArrayList.

UML-діаграма



### Клас CSVProcessor

```
package com.lab111.Lab2;
```

```
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
 */
public class CSVProcessor implements Serializable{
    /**
     * array for store strings of CSV-file
     */
    private ArrayList<String> lines=new ArrayList<String>();
    /**
     * 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
     */
    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();
    }
    /**
     * The method for the printing strings
     * of the CSV-file to the console
     */
    public void print(){
        for(int i=0; i<lines.size();i++){
            System.out.println(lines.get(i));
        }
    }
}

```

## Класс TestIO

```
package com.lab111.Lab2;
```

```

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{
        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();
                csv.readFromFile(nameOfFile);
                System.out.println("Reading data from file
"+nameOfFile+" ... Done!");
                System.out.print("Serializing of array to the file
"+nameOfFile+" ... ");
                csv.serializeObject("serialized.dat");
                System.out.println("Done!");
                done=false;
            }catch(FileNotFoundException e1){
                System.out.println("Wrong name of file! Please try again");
                counter++;
                if (counter==3)
                    throw e1;
            }
            catch(IOException e2){
                throw e2;
            }
        }
    }
    if (csv!=null){
        System.out.println("CSV-file contents:");
        csv.print();
    }
}
}

```