|  |  |
| --- | --- |
| File:COMSATS new logo.jpg - Wikimedia Commons  OBJECT ORIENTED PROGRAMMING  *Class Assignment - 3*  *File handling* | **submitted by:**  **Shahzaneer Ahmed**  **registration number:**  **sp21-bcs-087**  **submitted to:**  **mA’M sANEEHA aMIR**  **date of submission:**  **may 31, 2022** |

Student Management System

# File Handling

## Source Code

**# File Handling**

- The process of writing content in a file format at the permanent Storage of your Computer and manipulating different operations is Called File Handling.

**# Why File Handling ?**

- To safe Data permanently so that you may have the data even after the execution of

  program

**# File Handling in Java**

- There are many ways of File handling in Java

- For dealing with Object CRUD Operations we use

  `ObjectOutputStream , FileOutputStream , File` for writing

  `ObjectInputStream , FileIntputStream , File` for reading

- The rest of Code writing is explained in the `mini Project Student Management System`

# Person

## Source Code

import java.io.Serializable;

public class Person implements Serializable {

    private String name;

    private String Phone;

    private String gender;

    public Person() {

    }

    public Person(String name, String Phone, String gender) {

*this*.name = name;

*this*.Phone = Phone;

*this*.gender = gender;

    }

    public String getName() {

        return *this*.name;

    }

    public void setName(String name) {

*this*.name = name;

    }

    public String getPhone() {

        return *this*.Phone;

    }

    public void setPhone(String Phone) {

*this*.Phone = Phone;

    }

    public String getGender() {

        return *this*.gender;

    }

    public void setGender(String gender) {

*this*.gender = gender;

    }

    @Override

    public String toString() {

        return "{" +

            " name='" + getName() + "'" +

            ", Phone='" + getPhone() + "'" +

            ", gender='" + getGender() + "'" +

            "}";

    }

}

# Student

## Source Code

import java.io.Serializable;

public class Student extends Person implements Serializable {

    private double GPA;

    private int semester;

    private char section;

    private Department department;

    public Student() {

*super*();

    }

    public Student(String name , String phone , String gender ,double GPA, int semester, char section, Department department) {

*super*(name, phone, gender);

*this*.GPA = GPA;

*this*.semester = semester;

*this*.section = section;

*this*.department = department;

    }

    public double getGPA() {

        return *this*.GPA;

    }

    public void setGPA(double GPA) {

*this*.GPA = GPA;

    }

    public int getSemester() {

        return *this*.semester;

    }

    public void setSemester(int semester) {

*this*.semester = semester;

    }

    public char getSection() {

        return *this*.section;

    }

    public void setSection(char section) {

*this*.section = section;

    }

    public Department getDepartment() {

        return *this*.department;

    }

    public void setDepartment(Department department) {

*this*.department = department;

    }

    @Override

    public String toString() {

        return *super*.toString() + "{" +

            " GPA='" + getGPA() + "'" +

            ", semester='" + getSemester() + "'" +

            ", section='" + getSection() + "'" +

            ", department='" + getDepartment() + "'" +

            "}";

    }

}

# Department

## Source Code

import java.io.Serializable;

public class Department implements Serializable {

    private String name;

    private String location;

    public Department() {

    }

    public Department(String name, String location) {

*this*.name = name;

*this*.location = location;

    }

    public String getName() {

        return *this*.name;

    }

    public void setName(String name) {

*this*.name = name;

    }

    public String getLocation() {

        return *this*.location;

    }

    public void setLocation(String location) {

*this*.location = location;

    }

    @Override

    public String toString() {

        return "{" +

            " name='" + getName() + "'" +

            ", location='" + getLocation() + "'" +

            "}";

    }

}

# MyObjectOutputStream

## Source Code

import java.io.IOException;

import java.io.ObjectOutputStream;

import java.io.OutputStream;

*//! Object writing class*

public class MyObjectOutputStream extends ObjectOutputStream{

*// Our custom Object Output Stream class -> which is used to input object data*

*// without writing the Stream headers.*

    public MyObjectOutputStream() throws IOException{

*super*();

    }

    public MyObjectOutputStream(OutputStream o) throws IOException{

*super*(o);

    }

    public void writeStreamHeader(){}

}

# OperationsStorage

## Source Code

import java.io.EOFException;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.ObjectInputStream;

import java.io.ObjectOutputStream;

import java.io.OutputStream;

import java.util.ArrayList;

public class OperationsStorage {

*//\* for update and delete operation we need to safe the objects in an arraylist first and then manipulate them  (Update or delete) and then write them into the*

*//\* the file again therefore the following array list is created to store the objects*

*// \* And to reduce the complexity of code we have added the object to list as soon as they are written in the file so that we can manipuate them afterwards*

    private ArrayList<Student> a = new ArrayList<>();

*//! Create*

    public void writeToFile(Student s) {

*// file object*

        File f = new File("Students.ser");

*// Object for writing class (ObjectOutputStream)*

*// Why this object is NULL?*

*// Answer : Because when even the constructor is called, the object (File)*

*// is created but we want to keep it in the try block*

        ObjectOutputStream oos = null;

*// write to file*

        try {

            if (f.exists()) {

                oos = new MyObjectOutputStream(new FileOutputStream(f, true));

                oos.writeObject(s); *// write object to file*

                a.add(s); *// add object to array list*

            } else {

                oos = new ObjectOutputStream(new FileOutputStream(f, true));

                oos.writeObject(s); *// it will write the object to the file.*

                a.add(s); *// add object to array list*

            }

        } catch (IOException e) {

            e.printStackTrace();

        }

        catch (Exception e) {

            System.err.println("Cannot Write Object");

        }

*// For closing File*

        if (oos != null) {

            try {

                oos.close();

            } catch (IOException e) {

                e.printStackTrace();

            }

        }

    }

*//! Read*

    public void readAll() {

        ObjectInputStream oo = null;

        try {

            oo = new ObjectInputStream(new FileInputStream("Students.ser"));

            while (true) {

*// Reading object is below*

                Student s = (Student) oo.readObject();

                System.out.println(s.toString());

            }

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        }

        catch (EOFException e) {

            return;

        }

        catch (FileNotFoundException e) {

            e.printStackTrace();

        }

        catch (IOException e) {

            e.printStackTrace();

        }

        finally {

            try {

                oo.close();

            }

            catch (IOException e) {

            }

        }

    }

*//! Search*

    void searchStudentByName(String Name) {

        ObjectInputStream oo = null;

        try {

            oo = new ObjectInputStream(new FileInputStream("Students.ser"));

            while (true) {

*// Reading object is below*

                Student s = (Student) oo.readObject();

                if (s.getName().equalsIgnoreCase(Name)) {

                    System.out.println(s.toString());

                    break;

                }

            }

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        }

        catch (EOFException e) {

            return;

        }

        catch (FileNotFoundException e) {

            e.printStackTrace();

        }

        catch (IOException e) {

            e.printStackTrace();

        }

        finally {

            try {

                oo.close();

                System.err.println("File Closed");

            } catch (IOException e) {

            }

        }

    }

    void searchStudentsByDepartment(String department) {

        ObjectInputStream oo = null;

        try {

            oo = new ObjectInputStream(new FileInputStream("Students.ser"));

            while (true) {

*// Reading object is below*

                Student s = (Student) oo.readObject();

                if (s.getDepartment().getName().equalsIgnoreCase(department)) {

                    System.out.println(s.toString());

                    break;

                }

            }

        } catch (ClassNotFoundException e) {

            e.printStackTrace();

        }

        catch (EOFException e) {

            return;

        }

        catch (FileNotFoundException e) {

            e.printStackTrace();

        }

        catch (IOException e) {

            e.printStackTrace();

        }

        finally {

            try {

                oo.close();

                System.err.println("File Closed");

            } catch (IOException e) {

            }

        }

    }

*// We are using Sequential File handling so here we cannot have as such Direct update or Delete Methods for the objects*

*//  here we will use the intuition of Arraylist first for reading the objects and storing them then manipulating in the case of updation and removing the object in the case of Deletion*

*//! UPDATE*

    void updateCGPA(Student currentStudent, double newCGPA) throws Exception {

*// firstly manimupating the desired object for updating the CGPA*

        for (int i = 0; i < a.size(); i++) {

            if (a.get(i).getName().equalsIgnoreCase(currentStudent.getName())) {

                a.get(i).setGPA(newCGPA);

            }

        }

*//\* now again writing the Arraylist Objects to the file. first time it will create the file again and only then it will append!*

*// file object*

        File f = new File("Students.ser");

*// Object for writing class (ObjectOutputStream)*

        ObjectOutputStream oos = null;

*// write to file - code*

        int counter = 0;

        for (int i = 0; i < a.size(); i++) {

            if (counter > 0) {

*// System.out.println("other Times here you will append");*

                oos = new MyObjectOutputStream(new FileOutputStream(f, true));

                oos.writeObject(a.get(i));

            } else {

*// System.out.println("first Time here you will create new File");*

                oos = new ObjectOutputStream(new FileOutputStream(f));

                oos.writeObject(a.get(i)); *// it will write the object to the file.*

                counter++;

            }

        }

*// For closing File*

        if (oos != null) {

            oos.close();

        }

    }

*// !DELETE*

    void deleteStudent(Student toBeDeletedStudent) throws Exception {

*// removing the specified object from the arraylist*

            for (int i = 0; i < a.size(); i++) {

                if (a.get(i).getName().equalsIgnoreCase(toBeDeletedStudent.getName())) {

                    a.remove(i);

                }

            }

*// now again writing the Arraylist objects in the file first time we will create a new file and then we will append*

            File f = new File("Students.ser");

*// Object for writing class (ObjectOutputStream)*

        ObjectOutputStream oos = null;

*// write to file*

        int counter = 0;

        for (int i = 0; i < a.size(); i++) {

                if (counter > 0) {

*// when you are running it for the second and afterwards iterations you will append the file*

                    oos = new MyObjectOutputStream(new FileOutputStream(f, true));

                    oos.writeObject(a.get(i));

                }

                else {

*// for the first time you will create a new file*

                    oos = new ObjectOutputStream(new FileOutputStream(f));

                    oos.writeObject(a.get(i)); *// it will write the object to the file.*

                    counter++;

                }

        }

*// For closing File*

        if (oos != null) {

            oos.close();

        }

    }

}

# Runner

## Source Code

import java.io.EOFException;

import java.io.FileNotFoundException;

public class Runner {

    public static void main(String[] args) {

        OperationsStorage o = new OperationsStorage();

        Department cs = new Department("Computer Science", "CUI Islamabad");

        Department cyber = new Department("Cyber Security", "CUI Islamabad");

        Student s = new Student("Shahzaneer Ahmed", "0316-4606490", "Male", 3.23, 3, 'B', cs);

        Student s2 = new Student("Laiba Imran", "0316-XXXXXXX", "Female", 3.23, 3, 'A', cyber);

        Student s3 = new Student("Rabbiya Tabassum", "0316-XXXXXXX", "Female", 3.48, 3, 'A', cyber);

        Student s4 = new Student("Raheem Arif", "0316-57532236", "Male", 3.56, 3, 'A', cyber);

        o.writeToFile(s);

        o.writeToFile(s2);

        o.writeToFile(s3);

        o.writeToFile(s4);

        o.readAll();

*// try {*

*//     o.updateCGPA(s2, 4);*

*// }*

*// catch (EOFException e) {*

*// }*

*// catch (FileNotFoundException e) {*

*//     e.printStackTrace();*

*// }*

*// catch (Exception e) {*

*//     e.printStackTrace();*

*// }*

*// o.readAll();*

        try {

            o.deleteStudent(s);

        } catch (Exception e) {

            e.printStackTrace();

        }

        o.readAll();

*// o.searchStudentByName("Laiba Imran");*

*// o.searchStudentByName("Shahzaneer Ahmed");*

*// o.searchStudentsByDepartment("Computer Science");*

    }

}