|  |  |
| --- | --- |
| File:COMSATS new logo.jpg - Wikimedia Commons  **Class Assignment 3** | **Subject:**  **object oriented programming**  **submitted by:**  **Daoud Hussain**  (Sp21-bcs-102)  **Class:**  **bcs-3b**  **submitted to:**  **mam saneeha amir**  **date of submission:**  **June 05 , 2022** |

Student Class

import java.util.\*;

import java.io.Serializable;

public class Student extends Person implements Serializable{

private String gpa;

private String semester;

private String section;

private Department typeDepartment;

public Student(){

//Default Argument Constructor

Department myDept = new Department();

}

public Student(String name, String phone, String gender, String gpa, String semester, String section, Department myDept){

super(name, phone, gender);

this.gpa = gpa;

this.semester = semester;

this.section = section;

this.typeDepartment = myDept;

}

//Setters

public void setGPA(String gpa){

if(gpa != "")

this.gpa = gpa;

}

public void setSemester(String semester){

if(semester != "")

this.semester = semester;

}

public void setSection(String section){

if(section != "")

this.section = section;

}

public void setDepartment(Department deptObj){

if(deptObj != null)

this.typeDepartment = deptObj;

}

//Getters

public String getGPA(){

return this.gpa;

}

public String getSection(){

return this.section;

}

public String getSemester(){

return this.semester;

}

public Department getTypeDepartment(){

return this.typeDepartment ;

}

//Method to display values

public String toString(){

String text = super.toString()+ "\nGPA: " + gpa + "\nSemester: " + semester + "\nDepartment Name: " + typeDepartment.getName() + "\nLocation: " + typeDepartment.getLocation();

return text;

}

}

Department Class

import java.util.\*;

import java.io.Serializable;

public class Department implements Serializable{

private String name;

private String location;

public Department(){

//Default Argument Constructor

}

public Department(String name, String location){

this.name = name;

this.location = location;

}

//Setters

public void setName(String name){

if(name != "")

this.name = name;

}

public void setlocation(String location){

if(location != "")

this.location = location;

}

//Getters

public String getName(){

return this.name;

}

public String getLocation(){

return this.location;

}

}

Person class

import java.util.\*;

import java.io.Serializable;

public class Person implements Serializable{

private String name;

private String phone;

private String gender;

public Person(){

//Default Argument Constructor

}

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

this.name = name;

this.phone = phone;

this.gender = gender;

}

//Setters

public void setName(String name){

if(name != "")

this.name = name;

}

public void setPhone(String phone){

if(phone != "")

this.phone = phone;

}

public void setGender(String gender){

if(gender != "")

this.gender = gender;

}

//Getters

public String getName(){

return this.name;

}

public String getGender(){

return this.gender;

}

public String getPhone(){

return this.phone;

}

//Method to display values

public String toString(){

String text = "Name: " + name + "\nGender: " + gender + "\nPhone: " + phone ;

return text;

}

}

MyObjectOutputStream class

import java.io.IOException;

import java.io.ObjectOutputStream;

import java.io.OutputStream;

public class MyObjectOutputStream extends ObjectOutputStream{

public MyObjectOutputStream() throws IOException{

super();

}

public MyObjectOutputStream(OutputStream o) throws IOException{

super(o);

}

public void writeStreamHeader(){}

}

FileOperation class

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.util.ArrayList;

public class FileOperation{

public void writeAFile(Student s){

try {

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

if(f.exists()){

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

myObject.writeObject(s);

myObject.close();

}else{

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

myObject.writeObject(s);

myObject.close();

}

} catch (Exception e) {

e.printStackTrace();

}

}

public void readAFile(){

try {

ObjectInputStream redFile = new ObjectInputStream(new FileInputStream("Student.ser"));

try {

while(true){

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

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

}

}catch (ClassNotFoundException e) {

System.out.println("Sorry the targeted class is not found");

}catch(EOFException e){

System.out.println("Sorry the targeted file has reached to end ");

}

}catch(FileNotFoundException e){

System.out.println("Sorry the targeted file is not found ");

}

catch(IOException e){

System.out.println("Couldn’t perform the task due to some error");

}

}

public String SearchAStudentByName(String name) throws IOException{

String str = "";

ObjectInputStream obj = new ObjectInputStream(new FileInputStream("Student.ser"));

try {

while(true){

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

if(s.getName().equals(name)){

str += "\n"+s.toString();

}

}

}catch (ClassNotFoundException e) {

System.out.println("Sorry! the targetted class is not Found\n");

}catch(EOFException e){

obj.close();

System.out.println("Sorry! the targetted file has ended\n");

}catch(IOException e){

System.out.println("Couldn't perform operation due to some error");

}

return str;

}

public String updateGpaByName(String name, String gpa) throws IOException{

ArrayList<Student> list = new ArrayList<Student>();

boolean flag = false;

ObjectInputStream obj = new ObjectInputStream(new FileInputStream("Student.ser"));

try {

while(true){

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

if(s.getName().equals(name)){

s.setGPA(gpa);

flag = true;

}

list.add(s);

}

}

catch(FileNotFoundException e){

System.out.println("Sorry! the targetted file is not found\n");

}

catch(IOException e){

System.out.println("Couldn't perform the task due to some error\n");

}

catch(ClassNotFoundException e){

System.out.println("Sorry! the targetted class is not found\n");

}

try {

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

f.delete();

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

writeAFile(list.remove(i));

}

}

catch (Exception e) {

System.out.println("Some error occured");

}

if(flag){

return ("Updated the file successfully!");

}

else{

return ("Couldn't Update the file successfully!");

}

}

public String removeByName(String name) throws IOException{

ArrayList<Student> list = new ArrayList<Student>();

boolean flag = false;

ObjectInputStream obj = new ObjectInputStream(new FileInputStream("Student.ser"));

try {

while(true){

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

list.add(s);

}

}

catch(SecurityException e){

System.out.println("Security");

}

catch(FileNotFoundException e){

System.out.println("File not found");

}

catch(EOFException e){

obj.close();

System.out.println("End of file");

}

catch(IOException e){

System.out.println("Io exception");

}

catch (Exception e) {

System.out.println("Some error occured");

}

try {

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

f.delete();

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

if(name.equals(list.get(i).getName())){

list.remove(i);

System.out.println("iNside loop if");

}else{

writeAFile(list.remove(i));

}

}

}catch(SecurityException e){

System.out.println("Security");

}catch (Exception e) {

System.out.println("Some error occured");

}

if(flag){

return ("Removed successfully!");

}

else{

return ("Couldn't remove successfully!");

}

}

}

Runner Class

import java.io.EOFException;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

public class Runner {

public static void main(String[] args) {

FileOperation myFile = new FileOperation();

Department stuDept = new Department("CS", "Academic Block III");

Student student1 = new Student("Daoud", "0348-3016704", "Male", "3.04", "3rd", "3B", stuDept);

Student student2 = new Student("Hussain", "0345-5828223", "Male", "3.48", "3rd", "3B", stuDept);

// myFile.writeAFile(student1);

// myFile.writeAFile(student2);

// myFile.readAFile();

// System.out.println(myFile.SearchAStudentByName("Daoud"));

// System.out.println(myFile.updateGpaByName( "Daoud", "3.48"));

// System.out.println(myFile.removeByName( "Daoud"));

//After all the operations

myFile.readAFile();

}

}

------------------------------

(End of File)