UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
OBJECT ORIENTED PROGRAMMING
NRC: 9652

WORKSHOP #14CodingReview
DATE: /06/2023
DUE:

Rubric:
1) class modeling   10 pts.
2) Running program   10 pts.
3) Printing/counting/deleting objects from JSON File 10 pts.
4) Saving JSON data 10pts.
5) Code Quality   10 pts.
TOTAL: .... 50 pts.

Evidence
1) class modeling   (Class diagram)
2) Running program   (cmd line executing the program)
3) Printing/counting/deleting objects from JSON File   (according to the roster number)
4) Saving JSON data   (screenshot of the json file)
5) Code Quality   (lines of code with stinky code)

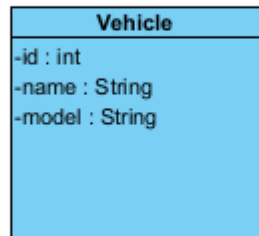1. BAEZ JARAMILLO GABRIEL ALEJANDRO   23/50
        Inspector: Zurita Pablo

Rubric:

1) class modeling	10 pts.	4/10	No main Class
2) Running program	10 pts.	10/10	No problems
3) Printing/counting/deleting objects from JSON File 10 pts.		0/10	No have json
4) Saving JSON data	10pts.	0/10	no saved json
5) Code Quality	10 pts.	9/10	In spanish

TOTAL: .... 50 pts.

Evidence
1) class modeling   (Class diagram)
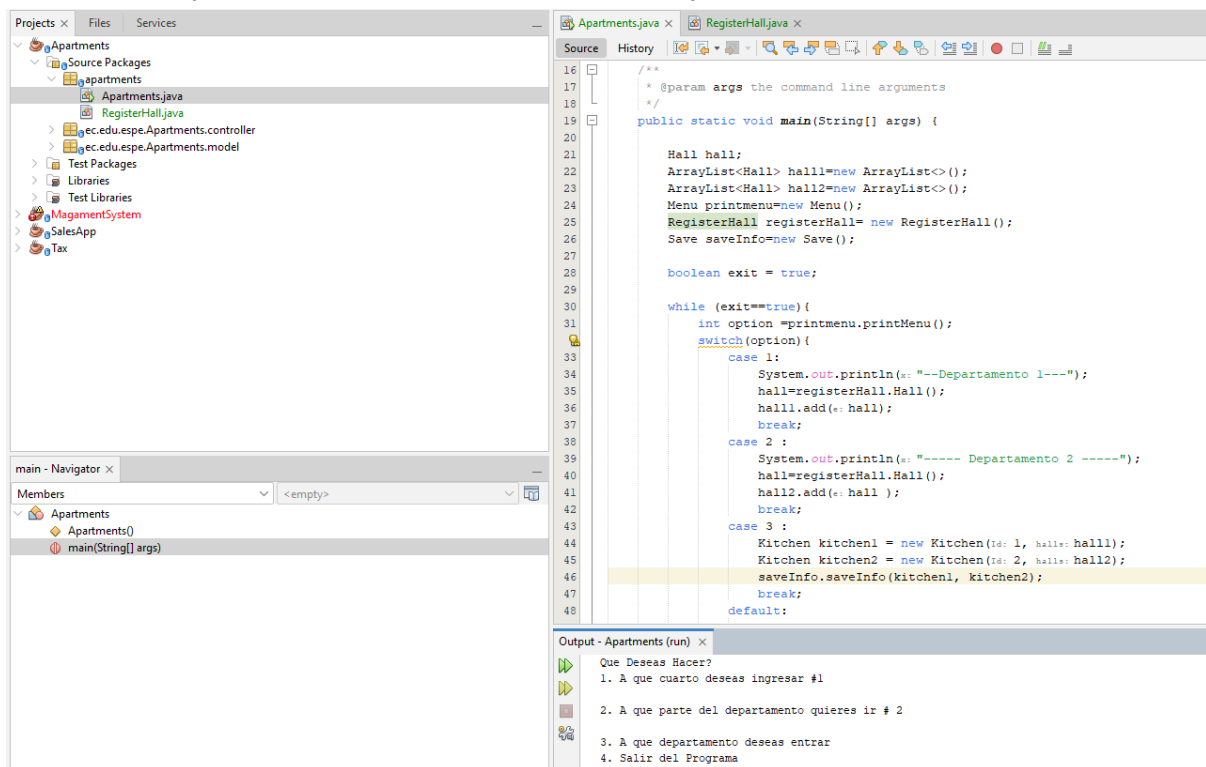

2) Running program   (cmd line executing the program)
3) Printing/counting/deleting objects from JSON File   (according to the roster number)
4) Saving JSON data   (screenshot of the json file)
5) Code Quality   (lines of code with stinky code)
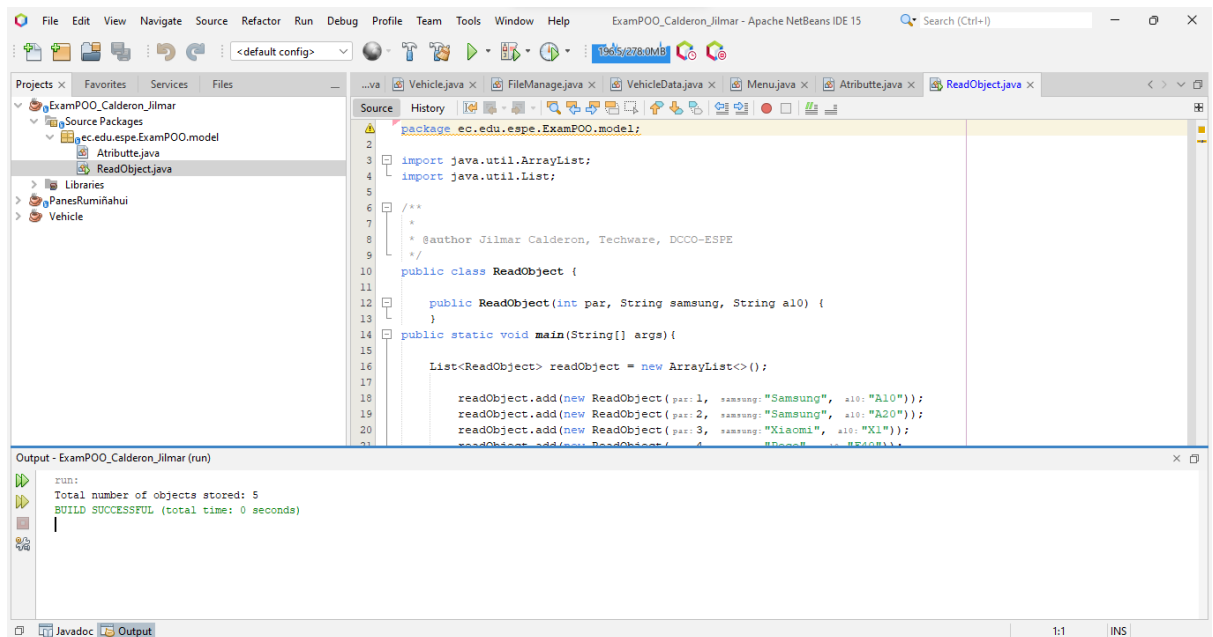

2. CALDERÓN CALVA JILMAR JHOEL
    Inspector: Gabriel Báez   23/50
    Count The total number of objects stored: Computer
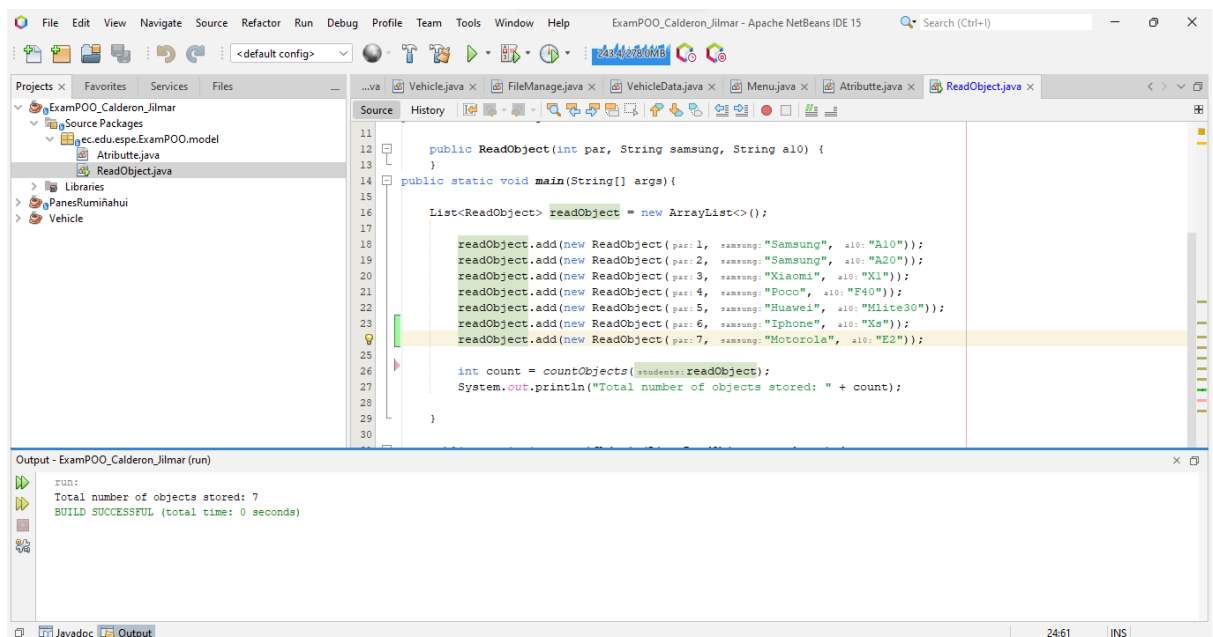   Rubric for the exercise:

1) class modeling   0/10 no class diagram
2) Running program   10/10 run without problems
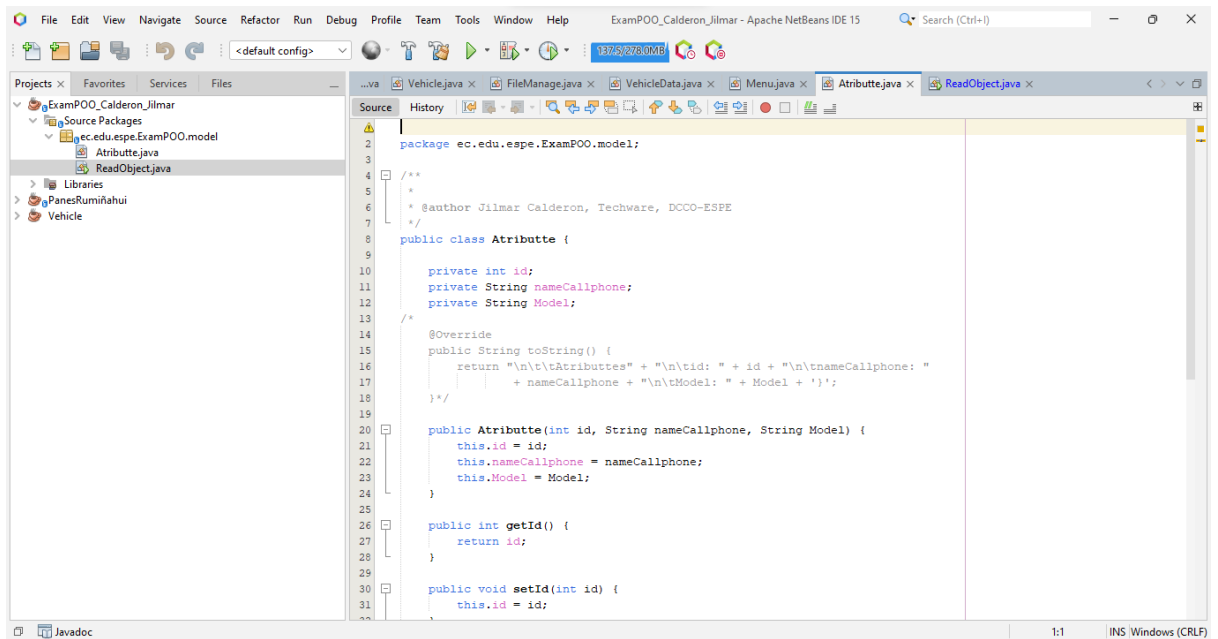

3) Printing/counting/deleting objects from JSON File   5/10

4) Saving JSON data   0/10 doesn't have JSON
5) Code Quality   8/10 has commented code
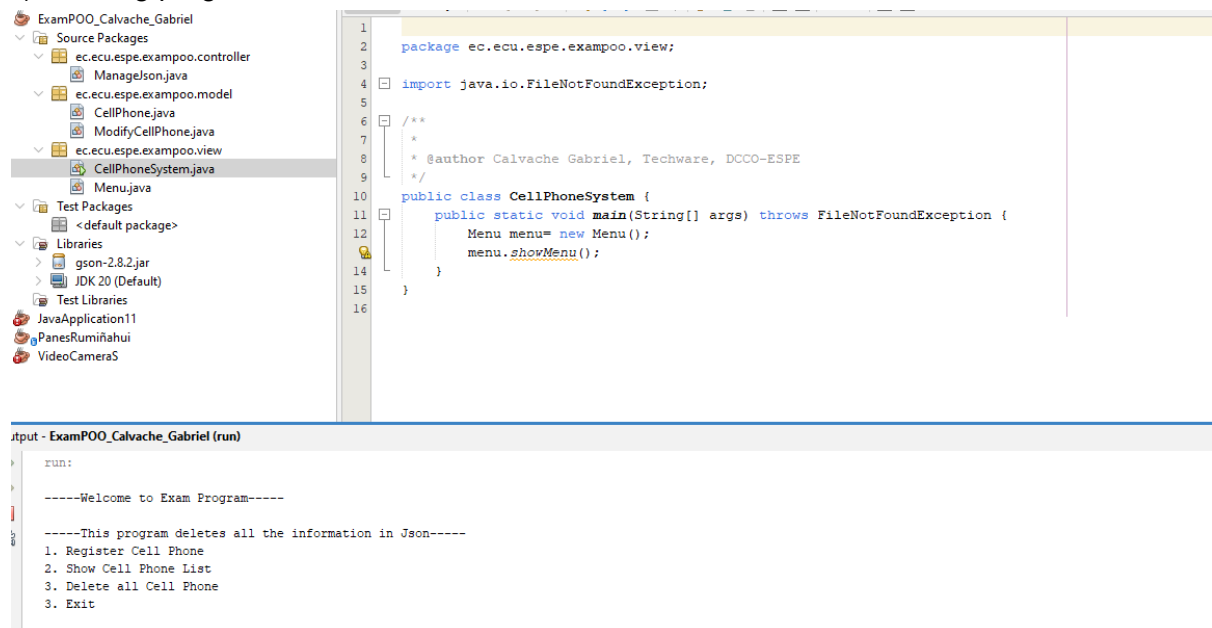
3. CALVACHE BONIFAZ GABRIEL ALEXANDER
   Inspector: Calderon Jilmar   40/50
   Delete all objects: Cellphone.

Rubric:

1) class modeling   0/10
2) Running program   10/10.

3) Printing/counting/deleting objects from JSON File 10/10

```
Model=papa
Color=blak
Size=10

CellPhone List
Model=white
Color=white
Size=15

CellPhone List
Model=pepa
Color=rosa
Size=12


-----Welcome to Exam Program-----

-----This program deletes all the information in Json-----
1. Register Cell Phone
2. Show Cell Phone List
3. Delete all Cell Phone
3. Exit

        Selected option: 3
All cell phone list delete

-----Welcome to Exam Program-----

-----This program deletes all the information in Json-----
1. Register Cell Phone
2. Show Cell Phone List
3. Delete all Cell Phone
3. Exit

        Selected option: 2
                            Cell Phone List
```

4) Saving JSON data 10/10.

```
1 ▾ [{
2         "model": "papa",
3         "color": "blak",
4         "size": "10"
5     },
6 ▾   {
7         "model": "white",
8         "color": "white",
9         "size": "15"
10    },
11 ▾  {
12        "model": "pepa",
13        "color": "rosa",
14        "size": "12"
15    }
16 ]
```

Validate JSON   Clear

Results

```
valid JSON
```

5) Code Quality   10/10.

```
11      * @author Calvache Gabriel, Techware, DCCO-ESPE
12      */
13     public class ModifyCellPhone {
14
15         public void readCellPhone() throws FileNotFoundException {
16             ManageJson manageJson = new ManageJson();
17             File archivo = new File(pathname: "data\\cellphone.json");
18             ArrayList<CellPhone> listCellPhone = new ArrayList<>();
19             System.out.println(x: "\t\t\t\tCell Phone List");
20
21             if (archivo.exists() && archivo.length() == 0) {
22                 System.out.println(x: "There are no cell phones\n\n");
23             } else {
24                 listCellPhone = manageJson.readJson();
25                 for (CellPhone cellPhone : listCellPhone) {
26                     System.out.println(x: cellPhone.toString());
27                 }
28             }
29         }
30
31         public void deleteCellPhone() throws FileNotFoundException {
32             ManageJson manageJson = new ManageJson();
33             File archivo = new File(pathname: "data\\cellphone.json");
34             ArrayList<CellPhone> listCellPhone = new ArrayList();
35
36             if (archivo.exists() && archivo.length() == 0) {
37                 System.out.println(x: "There are no cell phones \n\n");
38             } else {
39                 listCellPhone = manageJson.readJson();
40                 Scanner readData = new Scanner(source: System.in);
41
42                 listCellPhone.removeAll(c: listCellPhone);
43                 ManageJson.writteJson(cellPhone: listCellPhone);
44                 System.out.println( "All cell phone list delete");
```

TOTAL: .... 40/50.

4. CASIGNIA RUIZ DIEGO ALEJANDRO
   inspector: Calvache Gabriel   40/50
   Print All objects
   Rubric for the exercise:
         1) class modeling   0/10
         None
         2) Running program   10/ 10

```
Output - exam1 (run)

  run:

                    --- Forniture ---

  Menu:
  1. Add new forniture
  2. Print All
  3. Finish

            Option:
```

3) Printing/counting/deleting objects from JSON File   10/10

```
run:

                --- Forniture ---

Menu:
1. Add new forniture
2. Print All
3. Finish

        Option: 2


        Forniture list

        ID: 1
        Price: 123.0
        Stock=Yes

        ID: 2
        Price: 85.0
        Stock=No

        ID: 3
        Price: 67.0
        Stock=No

        ID: 4
        Price: 150.0
        Stock=Yes

        ID: 5
        Price: 124.0
        Stock=No

        ID: 6
        Price: 223.0
        Stock=YES

        ID: 7
        Price: 176.5
        Stock=No

        ID: 8
        Price: 89.0
        Stock=Yes

        ID: 8
        Price: 89.0
        Stock=Yes

        ID: 9
        Price: 356.0
        Stock=No
```

4) Saving JSON data   10/10

```
 1 ▾ [{
 2          "id": 1,
 3          "price": 123.0,
 4          "stock": "Yes"
 5      },
 6 ▾    {
 7          "id": 2,
 8          "price": 85.0,
 9          "stock": "No"
10      },
11 ▾    {
12          "id": 3,
13          "price": 67.0,
14          "stock": "No"
15      },
16 ▾    {
17          "id": 4,
18          "price": 150.0,
19          "stock": "Yes"
20      }
21  ]
```

Validate JSON   **Clear**

Results

valid JSON

5) Code Quality   10/10

```java
package examl.model;

import examl.controller.EditFornitureDataFile;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Scanner;

/**
 *
 * @author Diego Casignia, Techware, DCCO-ESPE
 */
public class EditForniture {
    public void createForniture() throws FileNotFoundException{
        File file = new File(pathname:"data\\forniture.json");
        EditFornitureDataFile editFornitureDataFile = new EditFornitureDataFile();
        ArrayList<Forniture> listForniture = new ArrayList();
        Scanner readData = new Scanner(source:System.in);
        if (file.exists() && file.length() == 0) {
            listForniture = new ArrayList<>();
        } else {
            listForniture = editFornitureDataFile.readFornitureData();
        }
        System.out.print(s:"\nID: ");
        int id = readData.nextInt();
        System.out.print(s:"Price: ");
        float price = readData.nextFloat();
        System.out.print(s:"Stock: ");
        String stock = readData.next();
        Forniture forniture = new Forniture(id, price, stock);
        listForniture.add(e:forniture);
        editFornitureDataFile.writeFornitureData(forniture:listForniture);
    }

    public void readForniture() throws FileNotFoundException{
        File file = new File(pathname:"data\\forniture.json");
        EditFornitureDataFile editFornitureDataFile = new EditFornitureDataFile();
        ArrayList<Forniture> listForniture = new ArrayList();
        if (file.exists() && file.length() == 0) {
            System.out.println(x:"File Null\n\n");
        } else {
            listForniture = editFornitureDataFile.readFornitureData();
            System.out.println(x:"\n\n\tForniture list");
            for (Forniture forniture : listForniture) {
                System.out.println(x:forniture.toString());
            }
        }
    }
}
```

Total:40/50
5. CEDEÑO MONTOYA RONY STIVEN
6. GALARZA VILLALVA CESAR LUIS   **TOTAL: 19/50**
Inspector: Casignia Diego

Rubric:
1) class modeling   0/10 pts.
2) Running program   10/10 pts.

3) Deleting objects from JSON File   0/10 pts.

4) Saving JSON data   0/10pts.

5) Code Quality   9/10 pts.

    TOTAL: **19/50 pts**.

Evidence:

1) class modeling: does not have the diagram class

2) Running program:
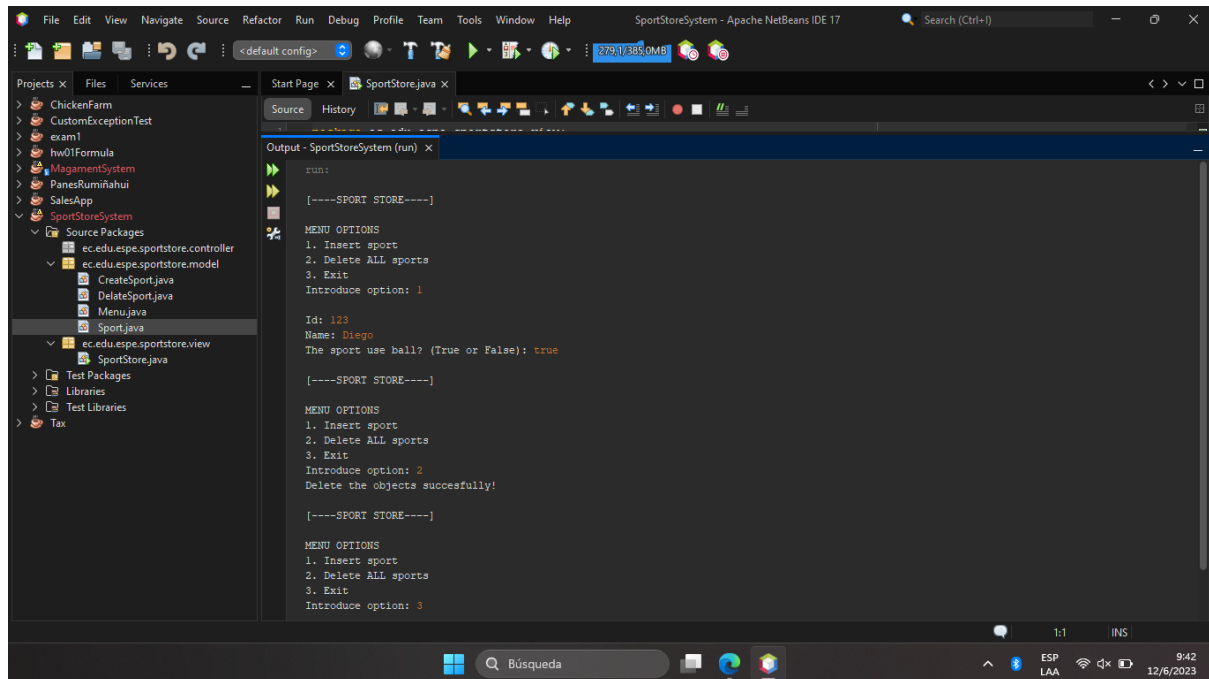
3) Deleting objects from JSON File: does not have the JSON file

4) Saving JSON data: does not have the JSON file

5) Code Quality: the method starts with a capital letter
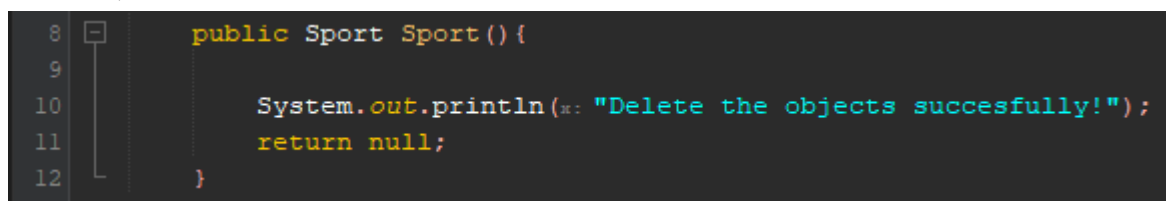
7. LUDEÑA QUICHIMBO EDISON WILFRIDO   **17/50**

Inspector: Galarza César

Topic: Soccer

      1) Print All objects

Rubric:

1) class modeling   8/10 pts.   //-2 pts for each missing method

2) Running program   0/10 pts.   //Doesn't have main class

3) Printing objects from JSON File   0/10 pts.   //Doesn't executed = Doesn't create the JSON

4) Saving JSON data   0/10pts.   //Doesn't executed = Doesn't create the JSON

5) Code Quality   9/10 pts.   //-1 pts unused library/variable

TOTAL: ....   17/50 pts.

Evidence

1) class modeling   (Class diagram)
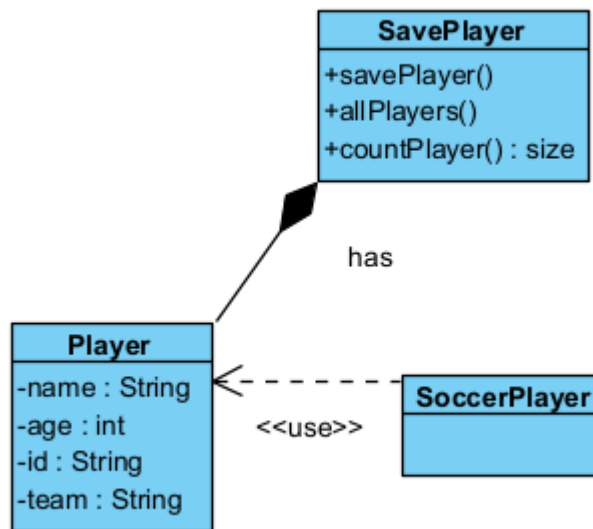
2) Running program   (cmd line executing the program)
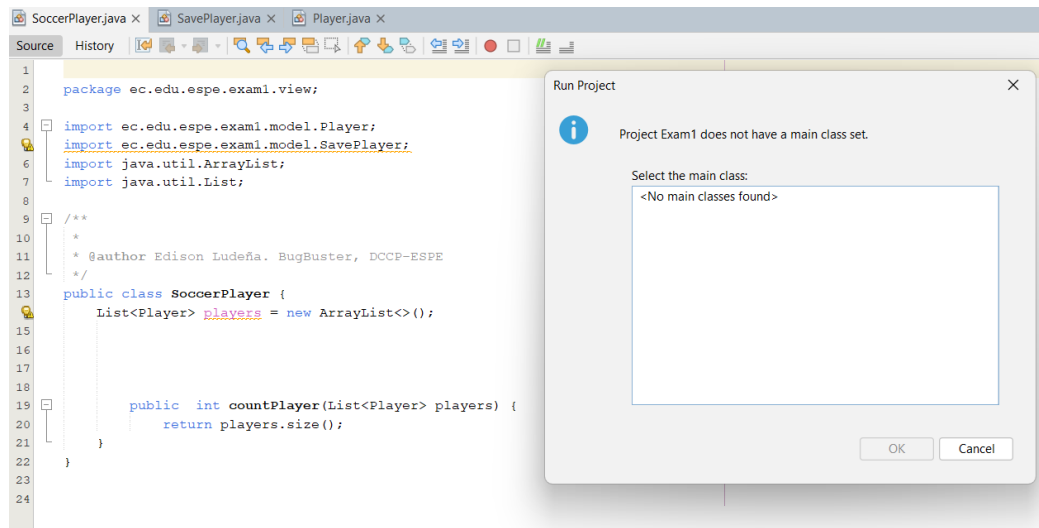
3) Printing/counting/deleting objects from JSON File   (according to the roster number)

```
public class SavePlayer {
    public static void savePlayerToJson(List<Player> players, String filename) {
        try (FileWriter writer = new FileWriter(fileName: filename)) {
            Gson gson = new GsonBuilder().setPrettyPrinting().create();
            gson.toJson(src:players, writer);
            System.out.println("The Soccer Players were saved in " + filename);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

4) Saving JSON data   (screenshot of the json file)

```
        try (FileWriter writer = new FileWriter(fileName: filename)) {
            Gson gson = new GsonBuilder().setPrettyPrinting().create();
            gson.toJson(src:players, writer);
            System.out.println("The Soccer Players were saved in " + filename);
```

5) Code Quality   (lines of code with stinky code)

```
1
2    package ec.edu.espe.exam1.view;
3
4    import ec.edu.espe.exam1.model.Player;
     import ec.edu.espe.exam1.model.SavePlayer;
6    import java.util.ArrayList;
7    import java.util.List;
8
9    /**
10    *
11    * @author Edison Ludeña. BugBuster, DCCP-ESPE
12    */
13    public class SoccerPlayer {
         List<Player> players = new ArrayList<>();
```

8. MIRANDA CREAMER ALISON SOLANGE
Inspector: Ludeña Edison   /50
2)Object size
Rubric:
1) class modeling   6/10 pts. //wrong relations, and no methods
2) Running program   0/10 pts. //Doesn't run the program
3) Printing/counting/deleting objects fromJSON File 0/10 pts. //Doesn't have counting
4) Saving JSON data 0/10pts. //Doesn't create the Json
5) Code Quality   8/10 pts . //Commented code, trash code(class player)
TOTAL: .... 14/50 pts.
Evidence

1) class modeling   (Class diagram)
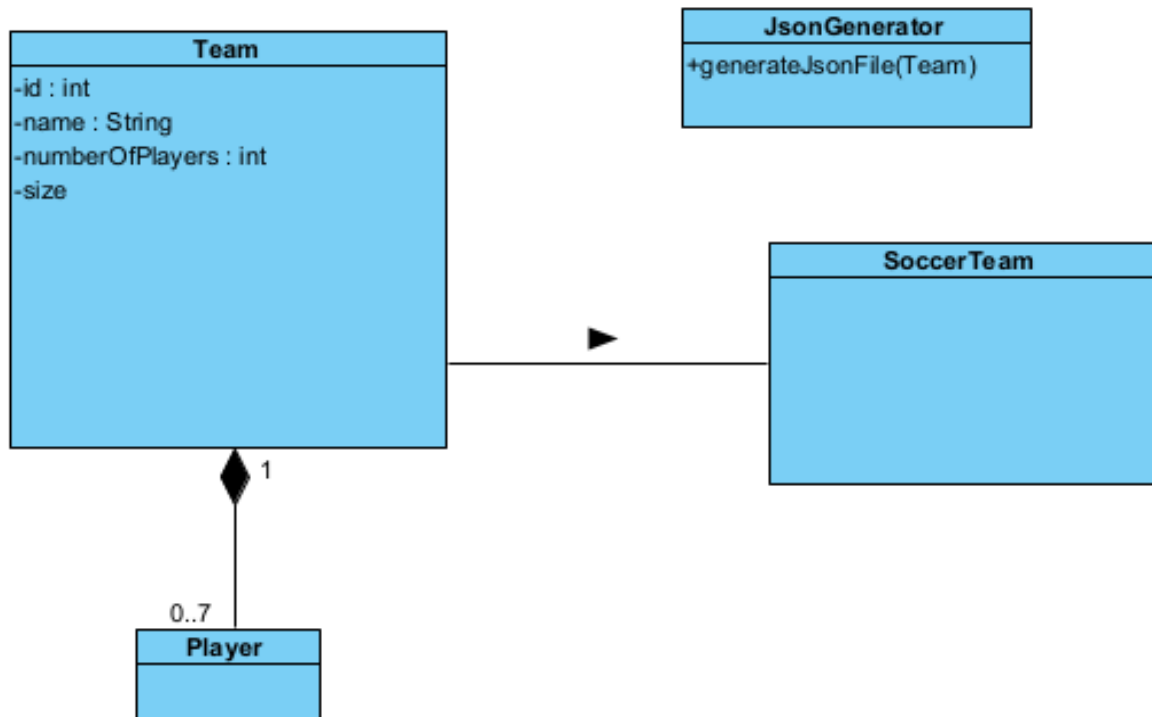
2) Running program   (cmd line executing the program)

3) Printing/counting/deleting objects from JSON File(according to the roster number)

```
public class SoccerTeam {
    public static void main(String[] args) {
        Team team [];
        ArrayList<Team> team = new ArrayList<>();

        System.out.println("team size --> " + team.size());
        team = new Team(team);
        team.add(team);
    }
}
```

4) Saving JSON data   (screenshot of the json file)
5) Code Quality   (lines of code with stinky code)

```
public Team(ArrayList<Team> team) {
    throw new UnsupportedOperationException( message: "Not supported yet."); // Generated from nbfs://nbhost/SystemFileSystem/Templa
}
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package ec.edu.espe.soccerteam.model;

/**
 *
 * @author Alison Miranda, Bug Busters, DCCO-ESPE
 */
public class Player {

}
```

9. MORENO NAVARRETE PAUL SEBASTIAN   26 /50

        Inspector: Miranda Alison
        3) Delete all objects
                1) Class modeling   0/10   (Wrong diagram)

Nombre	Última modificación	Tamaño del archivo
ScreenShotOfExcercise	-	133 KB
UMLExam1P27	-	1 MB
University	-	558 KB
UseClassOfExcercise	-	1 MB

                2) Running program   10/10 (Runs)

3) Printing/counting/deleting objects from JSON File   0/10
(Doesn´t delete)

4) Saving JSON data   10/10 (Valid Json)

5) Code Quality   6/10   (Menu in spanish, Class name "Save" is a verb, Class name "RegisterUniversity" is a verb, Packages named wrong)

```java
public  int printProductMenu(){
    Scanner readOption = new Scanner( source:System.in);
    System.out.println( x:"<----Informacion Universitaria---->");
    System.out.println( x:"Que deseas hacer ?");
    System.out.println( x:"1. Ingresar informacion de universidad\n");
    System.out.println( x:"2. Borrar la informacion guardada\n");
    System.out.println( x:"3. Salir\n");
```

RegisterUniversity.java

Save.java

University
Source Packages
ec.edu.espe.ferrinventory.controller

```
package ec.edu.espe.ferrinventory.controller;
```
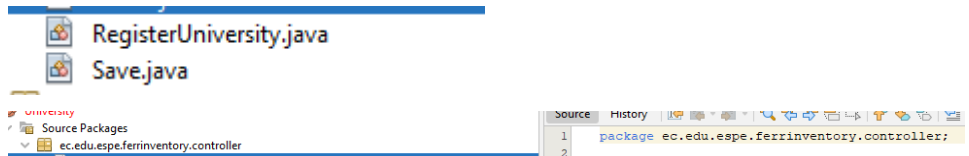
Inspector: Moreno Paul

1)Print all objects

Rubric:

1) class modeling   6/10
2) Running program   0/10
3) Printing/counting/deleting objects from JSON File   0/10
4) Saving JSON data 0 /10
5) Code Quality   4/10

Evidence

1) class modeling   (wrong label, no operations, wrong attribute syntax)
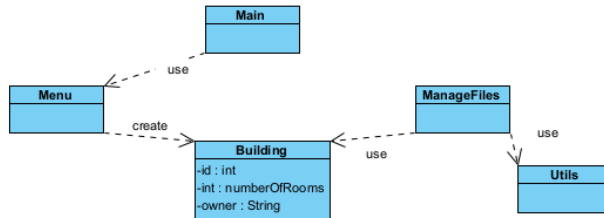
2) Running program   (no print)   yase :c
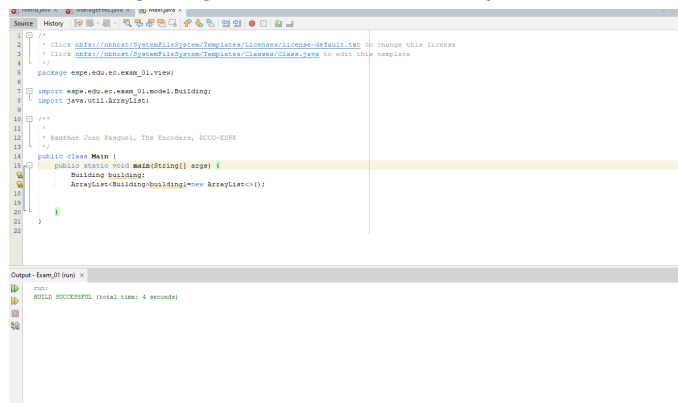
3) Printing/counting/deleting objects from JSON File   (nothing)
4) Saving JSON data   (nothing)
5) Code Quality   (name of class in plural,no use code,no use object, no use variable, error use try/catch)

```java
18  L */
19     public class ManageFiles {
20         public static void createJsonFile(Scanner scan, ArrayList<Building> building) throws nameException, localNameException, idException {
21             String fileName;
22             int enter = 0;
23             System.out.println(x:">>Ingress the name of the file you want to create: ");
24             fileName = scan.next();
25             do {
                   enterStore(scan, building);
                   enterStoreTwo(scan, building);
                   enterStoreThree(scan, building);
                   saveJsonFile(building, fileName);
30             } while (enter == 1);
31         }
32
33     }
       public static void enterBuilding (Scanner scan, ArrayList<Store> store)throws nameException{
35         Store storeInfo = new Store();
           enterStoreData(scan, store);
           if("ssssssss".equals(storeInfo.getName())){
38             throw new nameException("That name doesn't exists");
           }
40     }
41
       public static void enterStoreTwo(Scanner scan, ArrayList<Store> store) throws localNameException {
43         Store storeInfo = new Store();
           enterStoreDataTwo(scan, store);
           if ("aaaaaaa".equals(storeInfo.getlocalName())) {
46             throw new localNameException("That local name doesn't exists");
           }
48     }
49
       public static void enterStoreThree(Scanner scan, ArrayList<Store> store) throws idException {
51         Store storeInfo = new Store();
           enterStoreDataThree(scan, store);
           if (storeInfo.getId()!=10) {
54             throw new idException("That id is incorrect because don't have 10 digits");
           }
56     }
```

```java
16     public  int printMenu(){
17         Scanner readOption = new Scanner( source:System.in);
18         System.out.println( x: "<----WELCOME TO PRINT BUILDINGS XD---->");
19         System.out.println( x: "What do you want to do ?");
20         System.out.println( x: "1.File Creation \n");
21         System.out.println( x: "2. Add Buildings\n");
22         System.out.println( x: "3. Print Buildings\n");
23         System.out.println( x: "4. Exit\n");
24         option = readOption.nextInt();
25         return option;
           try{
27             System.out.println( x: "Digit an option: ");
               option = scan.nextInt();
```

```java
14     public class Main {
15         public static void main(String[] args) {
               Building building;
               ArrayList<Building>building1=new ArrayList<>();
18
```

11. RAMOS YANCHALIQUIN SEGUNDO JAVIER
    Inspector: Pasquel Juan   7/50

    2)Object size : Apartments

Rubric:
1) class modeling   0/10 pts.
2) Running program   0/10 pts.
3) Printing/counting/deleting objects from JSON File 0/10 pts.
4) Saving JSON data 0/10pts.
5) Code Quality   7/10 pts.

TOTAL: .... 7 pts.

Evidence
1) class modeling
(No Class Diagram)
2) Running program
(No running program)

```
Output - Apartments (run)                                              ×  ⬚
   run:
   Exception in thread "main" java.lang.RuntimeException: Uncompilable code -
      symbol:   class hall
      location: class apartments.Apartments
            at apartments.Apartments.main(Apartments.java:1)
   C:\Users\jdpas\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:1
   C:\Users\jdpas\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:6
   BUILD FAILED (total time: 1 second)
```

3) Printing/counting/deleting objects from JSON File
(There's no json)
4) Saving JSON data
(There's no json)
5) Code Quality
-Save isn't a noun

```
  */
public class Save {


}
```

-The menu is in spanish

```
  */
public class Menu {
    int option;
    public  int printMenu(){
        Scanner readOption = new Scanner(source:System.in);
        System.out.println(x: "<----Apartments---->\n");
        System.out.println(x: "Que Deseas Hacer?");
        System.out.println(x: "1. A que cuarto deseas ingresar #1\n");
        System.out.println(x: "2. A que parte del departamento quieres ir # 2\n");
        System.out.println(x: "3. A que departamento deseas entrar");
        System.out.println(x: "4. Salir del Programa\n");
        option = readOption.nextInt();
        return option;
    }
}
```

-The attribute category doesn't have lower camel-case

```
private String name;
private String Category;
private int stock;
```

RUEDA VILATUÑA JUAN SEBASTIAN
        Inspector: Ramos Javier   /50
        2)Object size

Rubric:
1) class modeling   0/10 pts.
2) Running program   0/10 pts.
3) Printing/counting/deleting objects from JSON File 0/10 pts.
(There's no json)
4) Saving JSON data 0 pts.
(There's no json)
5) Code Quality   8 pts.
TOTAL: .... 8/50 pts.

Evidence
1) class modeling
        (No Class diagram)
2) Running program   (cmd line executing the program)
No running program

```
Source   History   [toolbar icons]                                    ⊞

 1      package ec.edu.espe.exam1.controller;
 2
 3      import com.google.gson.Gson;
 4      import com.google.gson.GsonBuilder;
 5      import ec.edu.espe.exam1.model.Topic;
 6      import java.io.File;
 7
 8      import java.io.IOException;
 9      import java.io.PrintWriter;
10
11
12
13      /**
14       *
15       * @author juan rueda, The encoders; DCCO-ESPE
16       */
17
18      public class Eliminate {
19          public void sasveInfo(Topic topic1) {
               Gson gson=new GsonBuilder().setPrettyPrinting().create();
               String gsonTopic1 = gson.toJson(Topics1);
```

```
Output - Exam1Exercise (run)  ×                                         —

  run:
  Error occurred during initialization of boot layer
  java.nio.file.InvalidPathException: Illegal char <?> at index 76: C:\Users\JAVIER\OneDrive\Desk
  C:\Users\JAVIER\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:111: The following er
  C:\Users\JAVIER\AppData\Local\NetBeans\Cache\17\executor-snippets\run.xml:68: Java returned: 1
  BUILD FAILED (total time: 3 seconds)
```

3) Printing/counting/deleting objects from JSON File   (according to the roster number)
(There's no json)
4) Saving JSON data   (screenshot of the json file)
(There's no json)
5) Code Quality   (lines of code with stinky code)

- The sasveInfo isn't not

```
    public void sasveInfo(Topic topic1) {
```

- Must be a noun not a verb

```
    public class EliminateTopics {
        ArrayList <topic> topics = new ArrayList<>;
```

12. SANMARTIN GALAN JOSE MIGUEL

Inspector: Rueda Juan   /50

Print object:Shoe

Rubric:

1) class modeling   8/10 pts.
2) Running program   10/10 pts.
3) Printing/counting/deleting objects from JSON File 10/10 pts.
4) Saving JSON data   10/10pts.
5) Code Quality   9/10 pts. Menu in Spanish

TOTAL: .... 47/50 pts.

Evidence
1) class modeling   (Class diagram)

```
          ┌─────────────────┐
          │      Shoe       │
          ├─────────────────┤
          │ -id : int       │        ┌──────────────────────────┐
          │ -brand : String │        │        ShoeRecord        │
          │ -Size : int     │        ├──────────────────────────┤
          └─────────────────┘────────│ -shoe : Array<shoe>      │
                                      ├──────────────────────────┤
                                      │ +record()                │
          ┌─────────────────┐         │ +saveToJson()            │
          │     ExamP1      │         │ +printAll()              │
          ├─────────────────┤         │ +print()                 │
          │ +print()        │ <<use>> └──────────────────────────┘
          │ +record()       │<- - - - - - - -
          │                 │
          │                 │
          │                 │
          └─────────────────┘
```

2) Running program   (cmd line executing the program)

```
-------Shoes Controller-----

1. Add a Shoe

2. Print List

3. Exit

Choose an Option:
1
Write a ID:
2
Write a Brand:
puma
Write a Size:
10
Students saved to shoes.json
-------Shoes Controller-----
```

3) Printing/counting/deleting objects from JSON File   (according to the roster number)

```
1  [
2    {
3      "id": 1,
4      "brand": "Nike",
5      "size": 10
6    },
7    {
8      "id": 6,
9      "brand": "Puma",
10     "size": 12
11   },
12   {
13     "id": 5,
14     "brand": "Adidas",
15     "size": 8
16   },
17   {
18     "id": 2,
19     "brand": "Mike",
20     "size": 9
21   },
22   {
23     "id": 3,
24     "brand": "Adidas",
25     "size": 11
26   }
27 ]
```

```
2. Print List

3. Exit

Choose an Option:
2
ID: 1
Brand: Nike
Size: 10
-------------
ID: 6
Brand: Puma
Size: 12
-------------
ID: 5
Brand: Adidas
Size: 8
-------------
ID: 2
Brand: Mike
Size: 9
-------------
ID: 3
Brand: Adidas
Size: 11
-------------
-------Shoes Controller-----
```

4) Saving JSON data   (screenshot of the json file)

```
1 ▾ [{
2       "id": 3,
3       "brand": "Rebbok",
4       "size": 42
5  }]
```

| Validate JSON | Clear | | Support JSONLint for $2/Month |

Results

```
Valid JSON
```
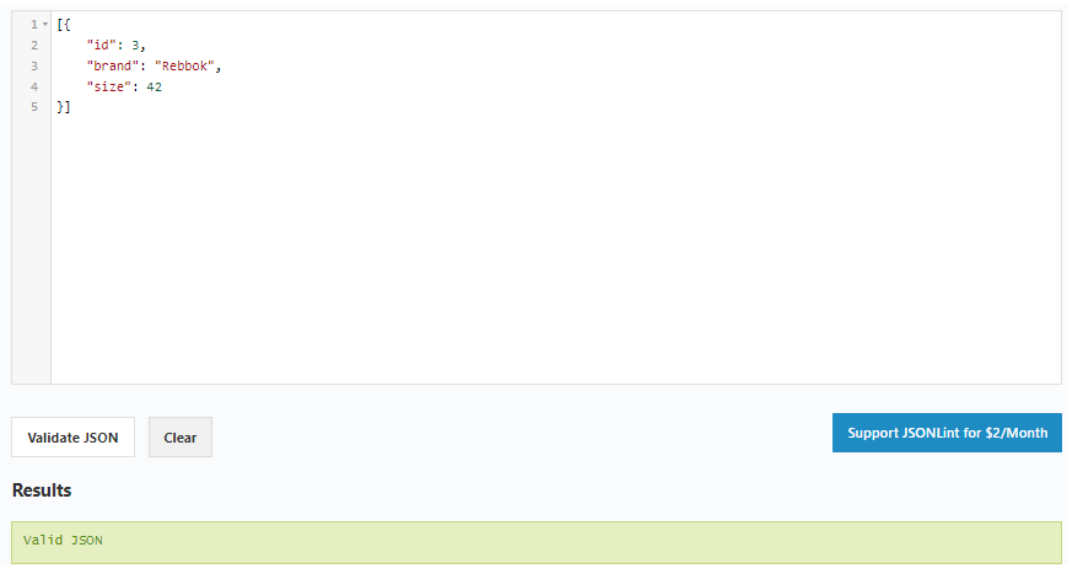
5) Code Quality   (lines of code with stinky code)

```
41                        break;
42                default:
43                    System.out.println(x: "Opción inválida. Intente nuevamente.");
44                    break;
45            }
```

13. SIMBAÑA RIVADENEIRA MICHAEL STEVEN
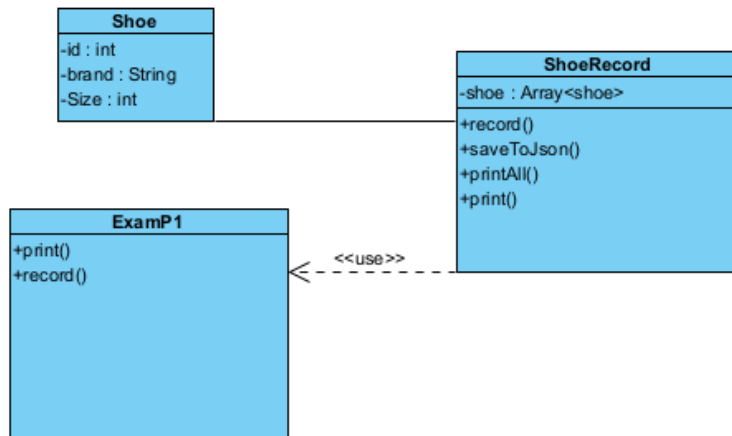   *Inspector:* Sanmartin Jose   /50
    Bank - Count

    Rubric:
    1) class modeling   0/10 pts. No Class Diagram
    2) Running program   10/10 pts.
    3) Printing/counting/deleting objects from JSON File 0/10 pts.Option do not run
    4) Saving JSON data 5/10pts. Create Json but not save information
    5) Code Quality   9/10 pts. Menu in Spanish
    TOTAL: ....24/ 50 pts.

    Evidence
    1) class modeling
            (No Class diagram)
    2) Running program   (cmd line executing the program)

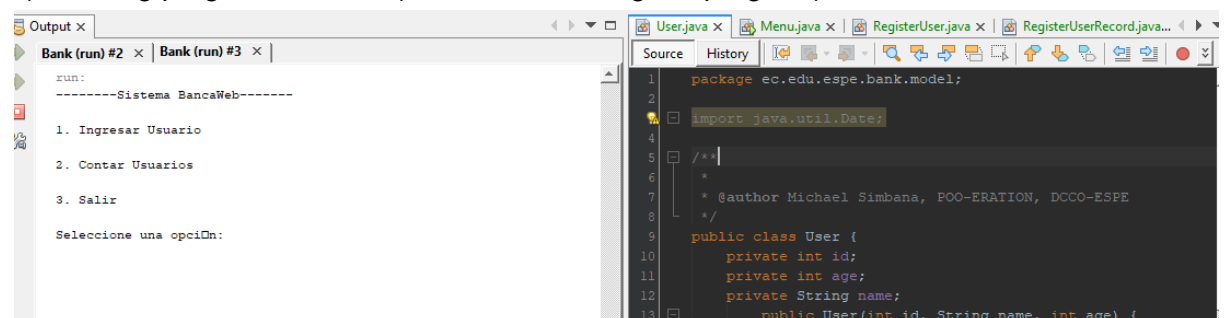3) Printing/counting/deleting objects from JSON File   (according to the roster number)

```
3. Salir

Seleccione una opción: 1
-------Registrar Usuarios-----

Ingrese los datos

Id: 01
Name: Jose
Age: 24
--------Sistema BancaWeb-------

1. Ingresar Usuario

2. Contar Usuarios

3. Salir

Seleccione una opción: 1
-------Registrar Usuarios-----

Ingrese los datos

Id: 02
Name: Michelle
Age: 21
--------Sistema BancaWeb-------

1. Ingresar Usuario

2. Contar Usuarios

3. Salir

Seleccione una opción: 2
BUILD SUCCESSFUL (total time: 1 minute 6 seconds)
```

4) Saving JSON data   (screenshot of the json file)

| Portapapeles | | Organizar | | Nuevo | | Abrir | | Seleccio |

> Este equipo > Descargas > Bank > data

	Nombre	Fecha de modificación	Tipo	Tamaño
	Users	12/6/2023 9:55	Archivo de origen ...	1 KB

```
run:
--------Sistema BancaWeb-------

1. Ingresar Usuario

2. Contar Usuarios

3. Salir

Seleccione una opción: 1
-------Registrar Usuarios-----

Ingrese los datos

Id: 01
Name: Jose
Age: 24
--------Sistema BancaWeb-------
```
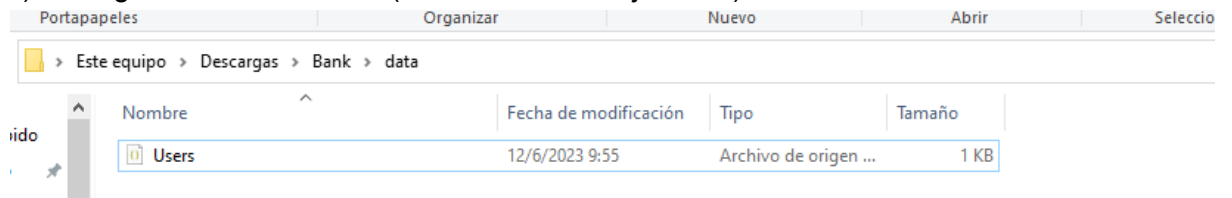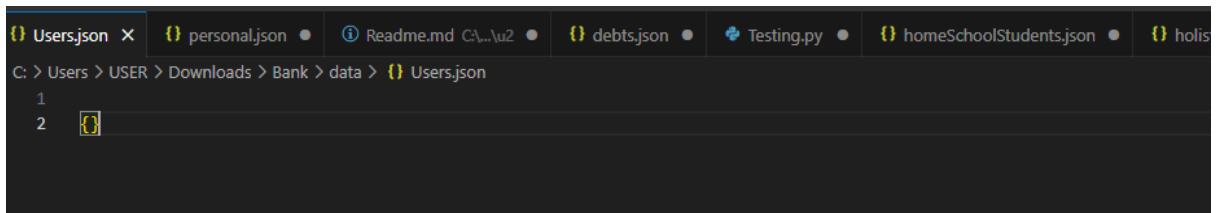
```
{} Users.json  ×    {} personal.json ●    ⓘ Readme.md C:\...\u2 ●    {} debts.json ●    🐍 Testing.py ●    {} homeSchoolStudents.json ●    {} holis

C: > Users > USER > Downloads > Bank > data > {} Users.json
  1
  2    {}
```

5) Code Quality   (lines of code with stinky code)

```java
System.out.println( x: "--------Sistema BancaWeb-------\n");
System.out.println( x: "1. Ingresar Usuario\n");
System.out.println( x: "2. Contar Usuarios\n");
System.out.println( x: "3. Salir\n");
System.out.print( s: "Seleccione una opción: ");
option = optionIn.nextInt();

switch (option) {
    case 1:
        addUser();
        break;
    case 2:
        break;
    case 4:
        System.out.println( x: "Saliendo del programa...");
        option =3;
        break;
    default:
        System.out.println( x: "Opción inválida. Intente nuevamente.");
        break;
```

14. TIPAN MOREIRA OSWALDO JACINTO
    Inspector: Michael Simbaña   /50
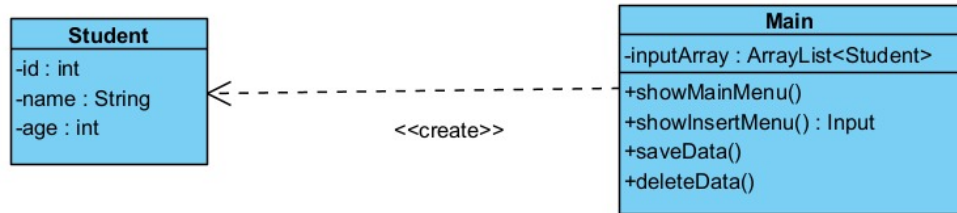   Delete all objects: Students

    Rubric:

1) class modeling   8/10 pts. (Methods in the wrong class/inaccurate dependency)
2) Running program   10/10 pts.
3) Printing/counting/deleting objects from JSON File 10/10 pts.
4) Saving JSON data 10/10pts.
5) Code Quality   9/10 pts.
TOTAL: 47/ 50 pts.

Evidence
1) class modeling   (Class diagram)
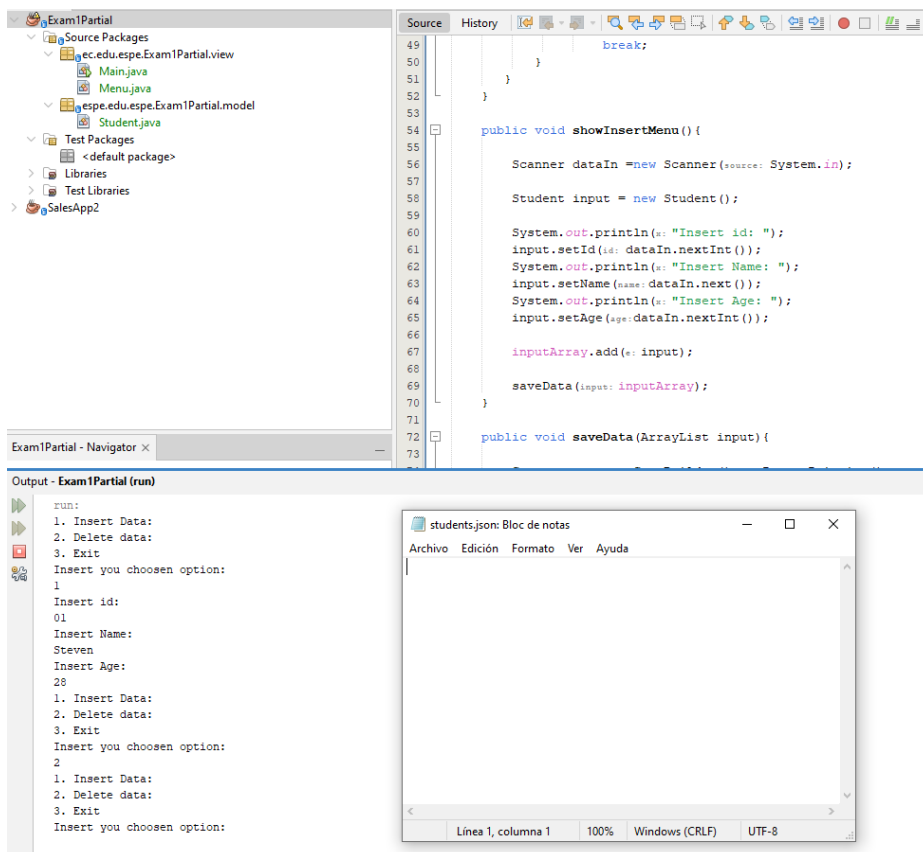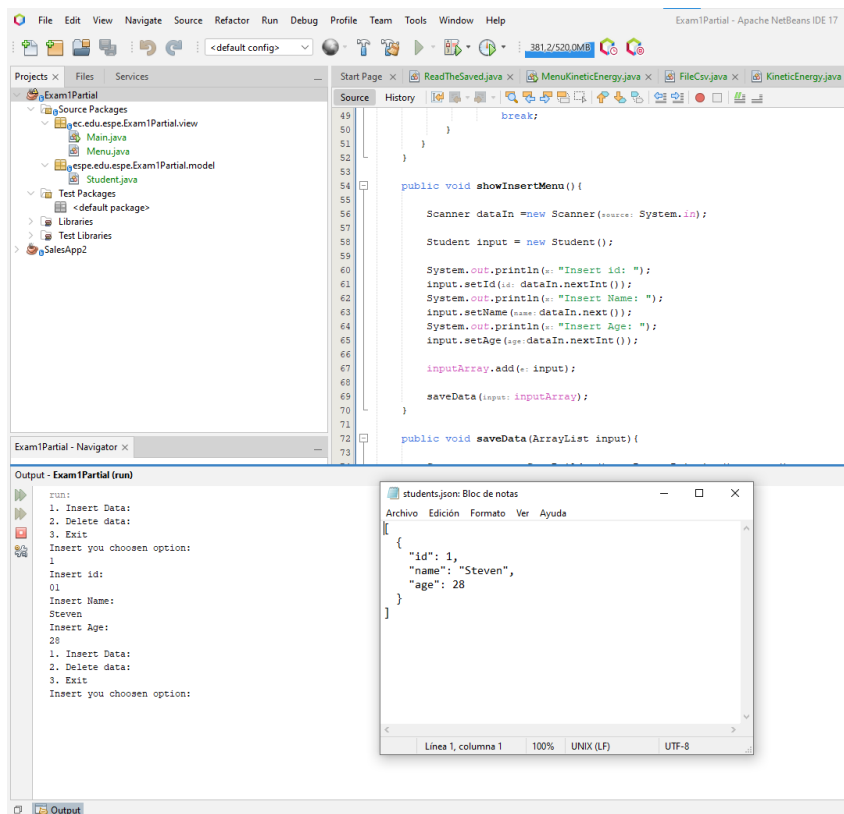

2) Running program   (cmd line executing the program)

3) Printing/counting/deleting objects from JSON File   (according to the roster number)

4) Saving JSON data   (screenshot of the json file)

```
1 ▾ [{
2      "id": 1,
3      "name": "Michael",
4      "age": 25
5 }]
```

Validate JSON **Clear**

Support JSONLint for $2/Month

Results

```
Valid JSON
```

5) Code Quality   9/10(bad name of attribute)

```
try(FileWriter writer2 = new FileWriter(fileName: "data\\students.json", append: false)) {
    writer2.write(str:dataSaver);
} catch (IOException e) {
    System.out.println("Error generating JSON file: " + e.getMessage());
}

try(FileWriter writer = new FileWriter(fileName: "data\\students.json", append: false)){
    writer.write(str:"");
```

15. ZURITA ROSERO PABLO ESTEBAN
    Inspector: Oswaldo Tipán   19/50
    Print objects: Professor

    Rubric for the exercise:
         1) class modeling   4/10 pts. (Menu class missing)

Professor

-id : int
-name : String
-email : String
-phoneNumber : int

+printAll()

2) Running program   10/10 pts. (It actually runs with no problems)


```
Output - Exam1Exercise (run)  ×

run:
-------Menu-------
1. Show all data ingresed
2. Exit
Enter any opcion:
1

-------Menu-------
1. Show all data ingresed
2. Exit
Enter any opcion:
2
Exit program...

-------Menu-------
1. Show all data ingresed
2. Exit
Enter any opcion:
1

-------Menu-------
1. Show all data ingresed
2. Exit
Enter any opcion:
```

3) Printing/counting/deleting objects from JSON File   0/10 pts. (The program doesn't ask for data, the program doesn't print)

Name	Date modified	Type	Siz
build	6/8/2023 10:32 AM	File folder	
nbproject	6/8/2023 10:06 AM	File folder	
src	6/8/2023 10:06 AM	File folder	
test	6/8/2023 10:08 AM	File folder	
build	6/8/2023 10:06 AM	XML Document	
manifest.mf	6/8/2023 10:06 AM	MF File	

```
run:
-------Menu-------
1. Show all data ingresed
2. Exit
Enter any opcion:
1


-------Menu-------
1. Show all data ingresed
2. Exit
Enter any opcion:
2
Exit program...

-------Menu-------
1. Show all data ingresed
2. Exit
Enter any opcion:
1
```

4) Saving JSON data 0/10pts. (Data file missing, No Json saved)

☐ 📁 build	6/8/2023 10:32 AM	File folder	
☐ 📁 nbproject	6/8/2023 10:06 AM	File folder	
☐ 📁 src	6/8/2023 10:06 AM	File folder	
☐ 📁 test	6/8/2023 10:08 AM	File folder	
☐ 📄 build	6/8/2023 10:06 AM	XML Document	
☐ 📄 manifest.mf	6/8/2023 10:06 AM	MF File	

```
-------Menu-------
1. Show all data ingresed
2. Exit
Enter any opcion:
1

-------Menu-------
1. Show all data ingresed
2. Exit
Enter any opcion:
```

5) Code Quality   5/10 pts.

- Junk Code

```java
System.out.println(x: "Hola");
```

- Method name in plural

```java
private static void showAttributes() {
```

- Menu output in Spanish

```java
if (dataOfProfessors.isEmpty()) {
    System.out.println(x: "No hay datos registrados");
} else {
    System.out.println(x: "Datos Ingresados:");
```

- Mixing Spanish and English

```java
int opcionOfMenu;
```

- Unused and empty method

```java
private static void showData(){


}
```

Total: 19/50