UNIVERSIDAD DE LAS FUERZAS ARMADAS

ESPE

Object-oriented programming

NRC: 9642

U2 Workshop N° 02

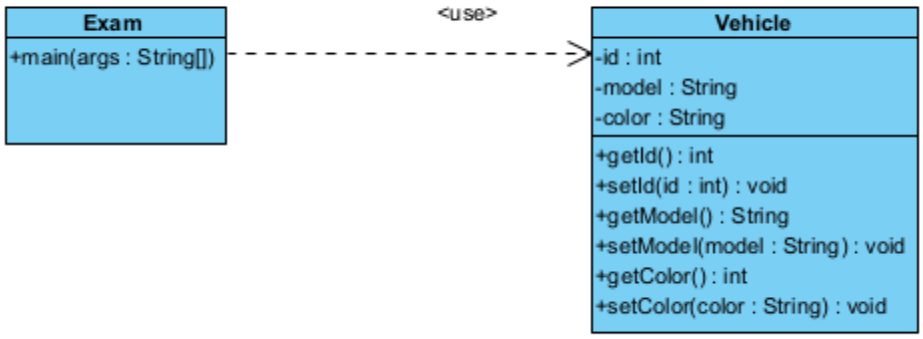
Topic: Code Inspection

ING. Jorge Edison Lascano.

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. RUBEN DARIO BENAVIDES MACIAS (Vehicles, print) /50
   Inspector: Edison Verdesoto
   Rubric:
   1) class modeling 9/10 pts.
   2) Running program 10/10 pts.
   3) Printing/counting/deleting objects from JSON File 9/10 pts.
   4) Saving JSON data 10/10pts.
   5) Code Quality 6/10 pts.
   TOTAL: ..... 44/50 pts.
   Evidence
   1) class modeling (Class diagram)

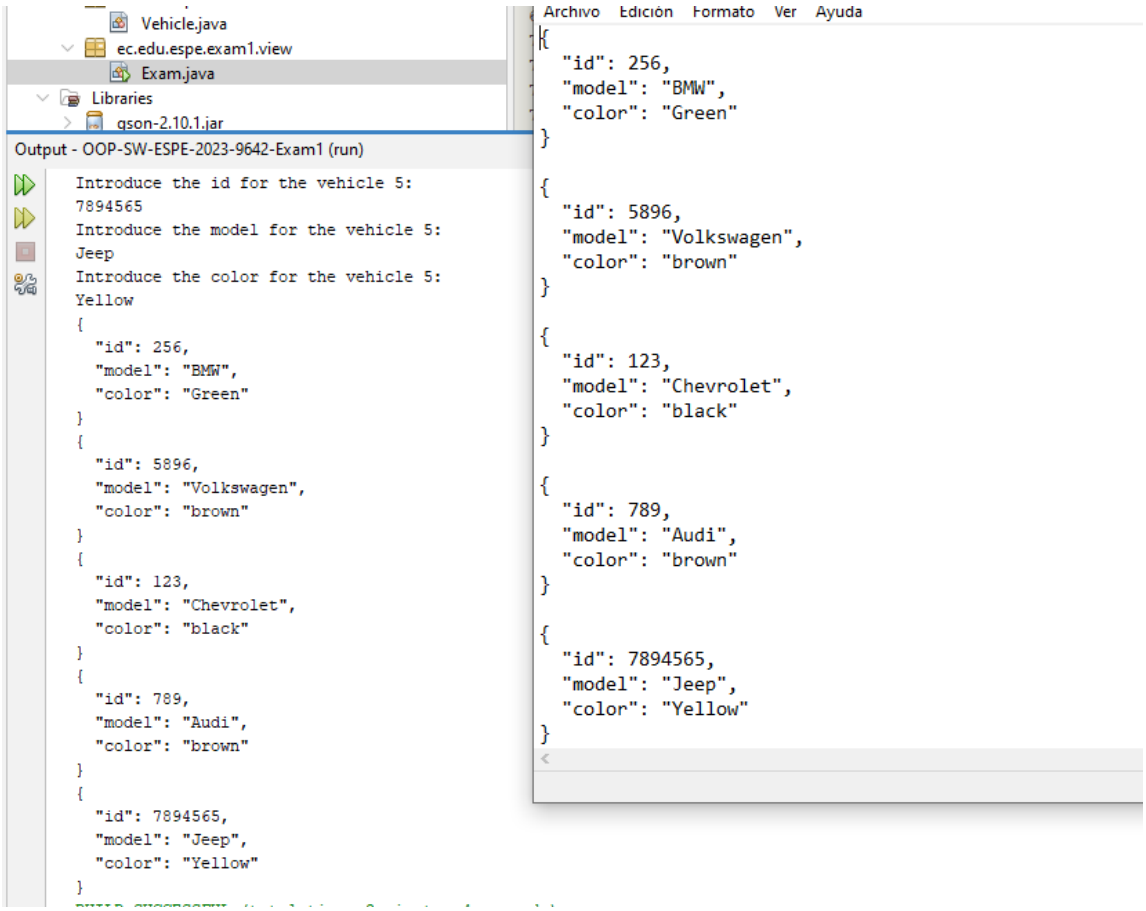


   Main Class: 4/4
   Model Class: 3/4 Missing the attribute
   Dependency: 2/2
   Total: 9/10

2) Running program 10/10 (cmd line executing the program)

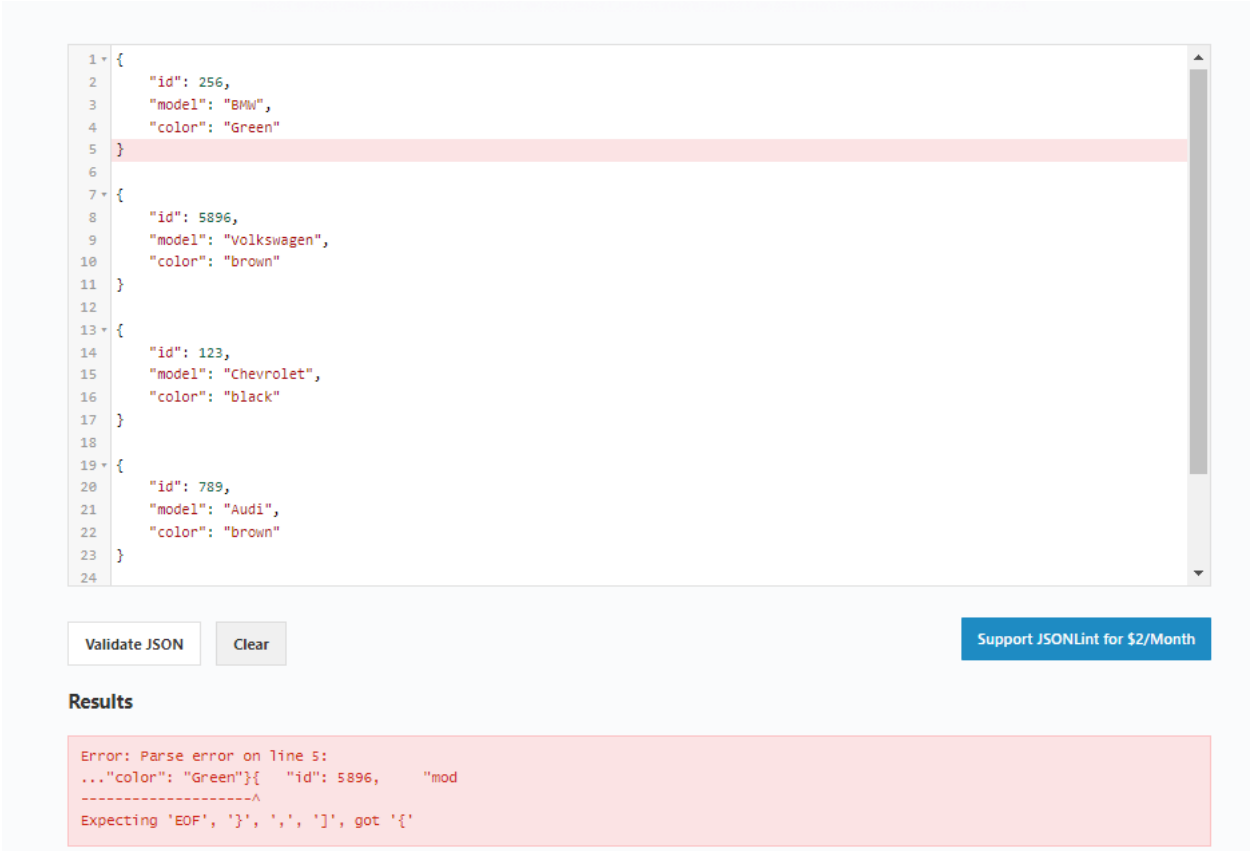
```
Introduce the id for the vehicle 4:
789
Introduce the model for the vehicle 4:
Audi
Introduce the color for the vehicle 4:
brown
Introduce the id for the vehicle 5:
7894565
Introduce the model for the vehicle 5:
Jeep
Introduce the color for the vehicle 5:
Yellow
[
```

Showing option: 5/5



Json Print: 5/5
Total:10/10

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



Validate Json: 9/10, error in line 5.

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

Json File: 10/10

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

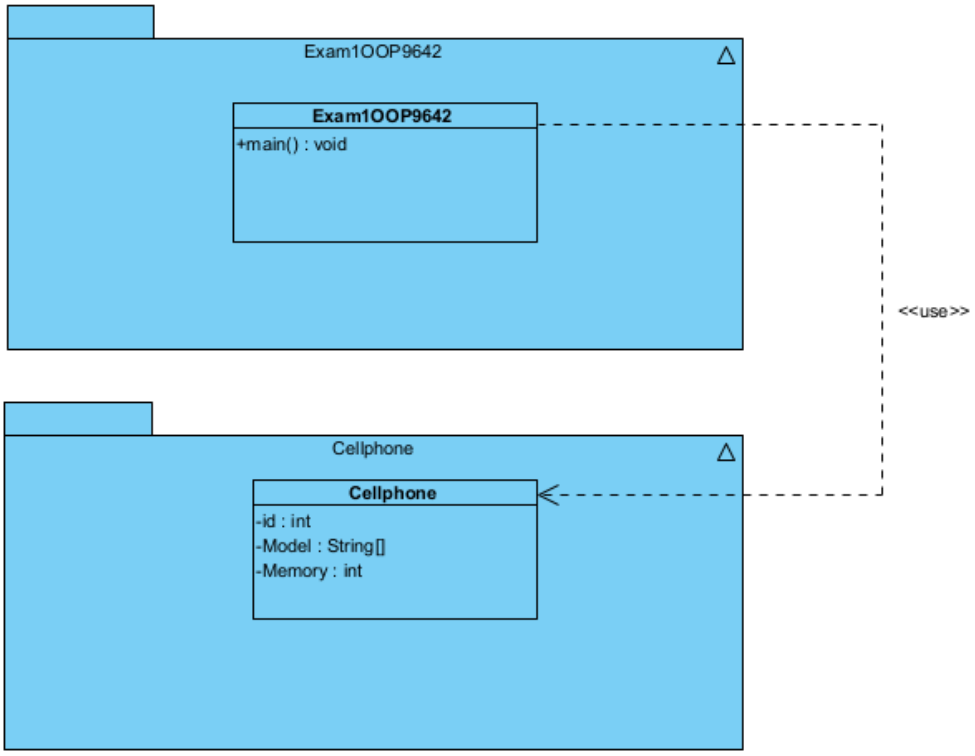
```
public class Exam {
    public static void main(String[] args) {

        ArrayList<Vehicle> vehicles = new ArrayList<Vehicle>();
        int id;
        String model;
        String color;

        for(int i = 0; i < 5; i++){

            Vehicle vehicle = new Vehicle();

            System.out.println("Introduce the id for the vehicle " + (i+1)+ ": " );
            while(true){
                try{
                    Scanner scanner = new Scanner(source: System.in);
                    id = scanner.nextInt();
                    break;
                }catch(Exception e){
                    System.out.println(x: "Error. You just introduce wrong data. Try again: ");
                }
            }


            System.out.println("Introduce the model for the vehicle " + (i+1)+ ": " );
            while(true){
                try{
                    Scanner scanner = new Scanner(source: System.in);
                    model = scanner.nextLine();
                    break;
                }catch(Exception e){
                    System.out.println(x: "Error. You just introduce wrong data. Try again: ");
                }
```

Code Quality: 6/10. Everything is in the main class, no method called.

2. ADONNY MATEO CALERO ARGUELLO (Count computers)
   0/50 Form wasn't answered.
   Inspector: Ruben Benavides
3. PABLO ESTEBAN CARRERA QUINDE
   Inspector: Adonny Calero
   Action: Delete Cellphones

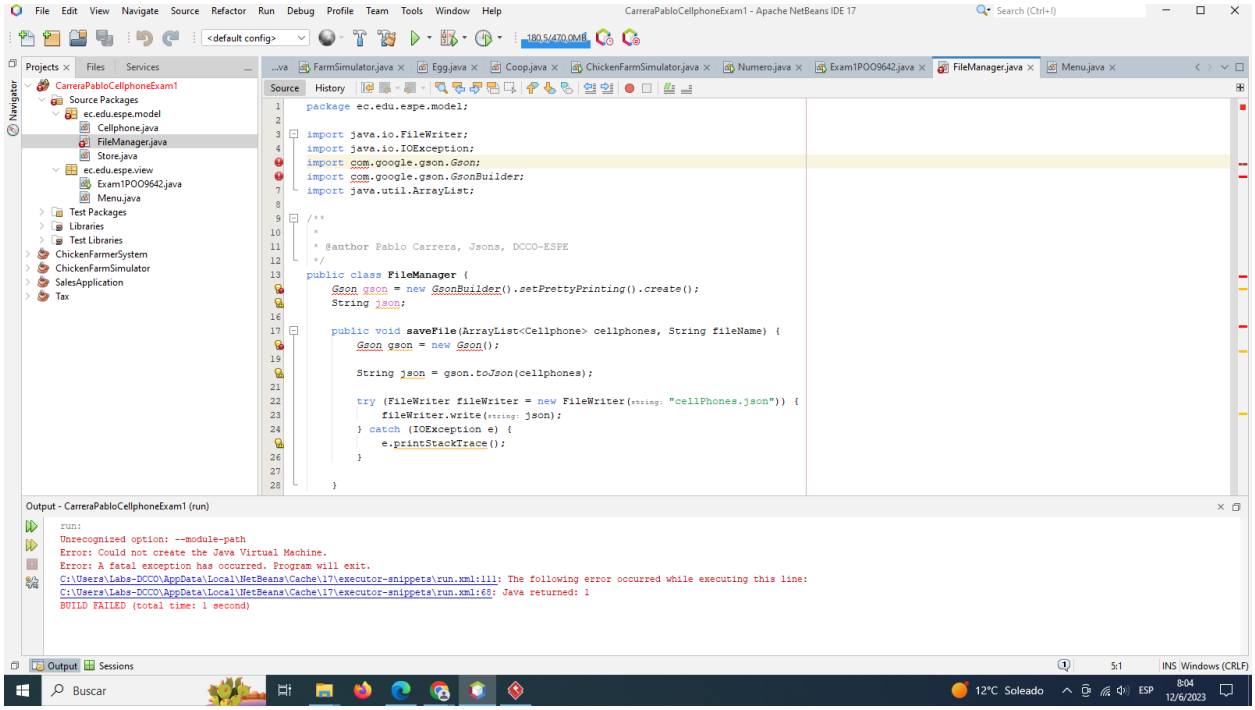
Rubric:
1) class modeling 10 pts.
2) Running program 10 pts.
3) Printing/counting/deleting objects from JSON File 10 pts.
4) Saving JSON data 10pts.
5) Code Quality 10 pts.
TOTAL: ..... 50 pts.

Evidence
1) class modeling (Class diagram)



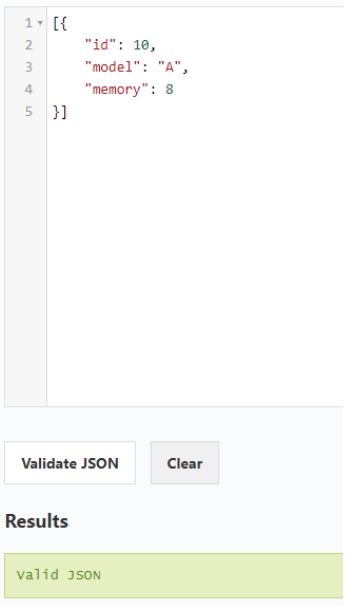
Grade(main): 4/4
Grade(class): 3/4
Grade(Dependency) : 2/2
Total: 9/10

2) Running program (cmd line executing the program)



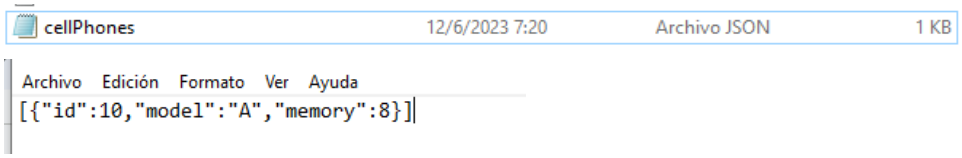
Total: 0/10

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



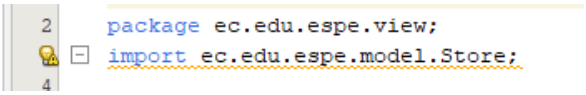
Validate Json: 10/10

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

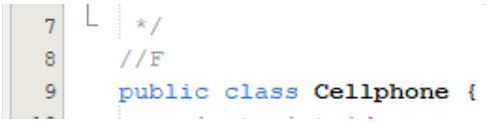


Total: 5/10 The program does not run.

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



unnecessary headers



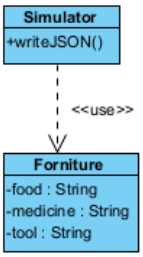
Unnecessary comments

Total: 8/10
Total Exam: 32/50

4. YESHUA AMADOR CHILIQUINGA AMAYA /10 - **Print**

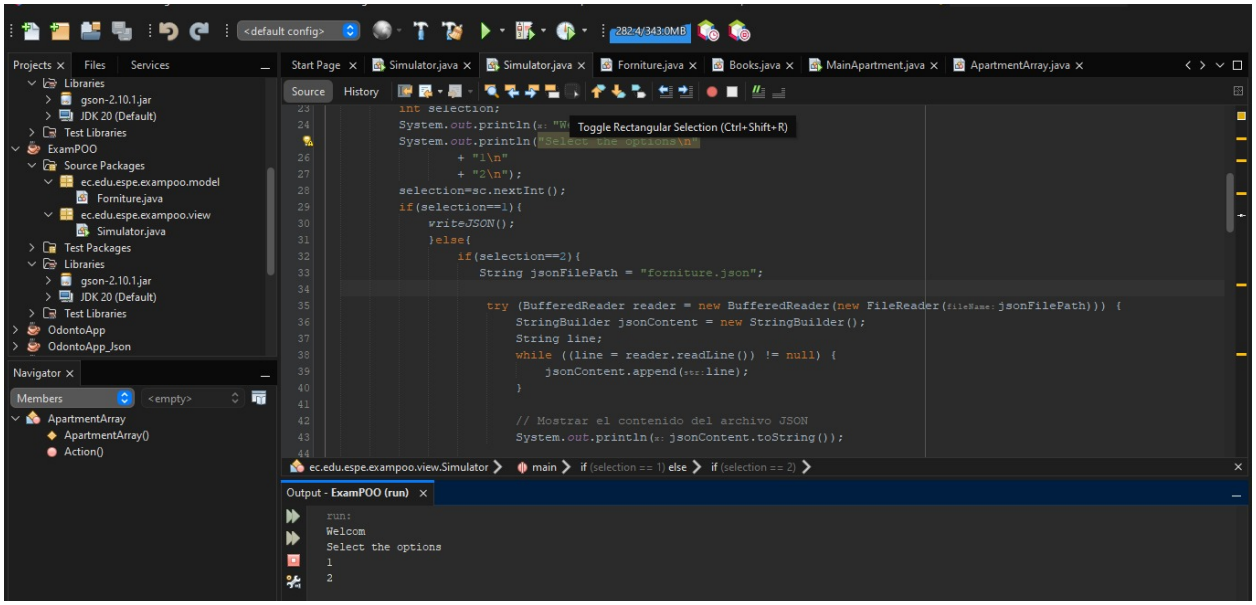
Rubric:
1) class modeling /10 pts.
2) Running program /10 pts.
3) Printing/counting/deleting objects from JSON File /10 pts.
4) Saving JSON data /10pts.
5) Code Quality /10 pts.
TOTAL: ..... /50 pts.

Evidence
1) class modeling (Class diagram)



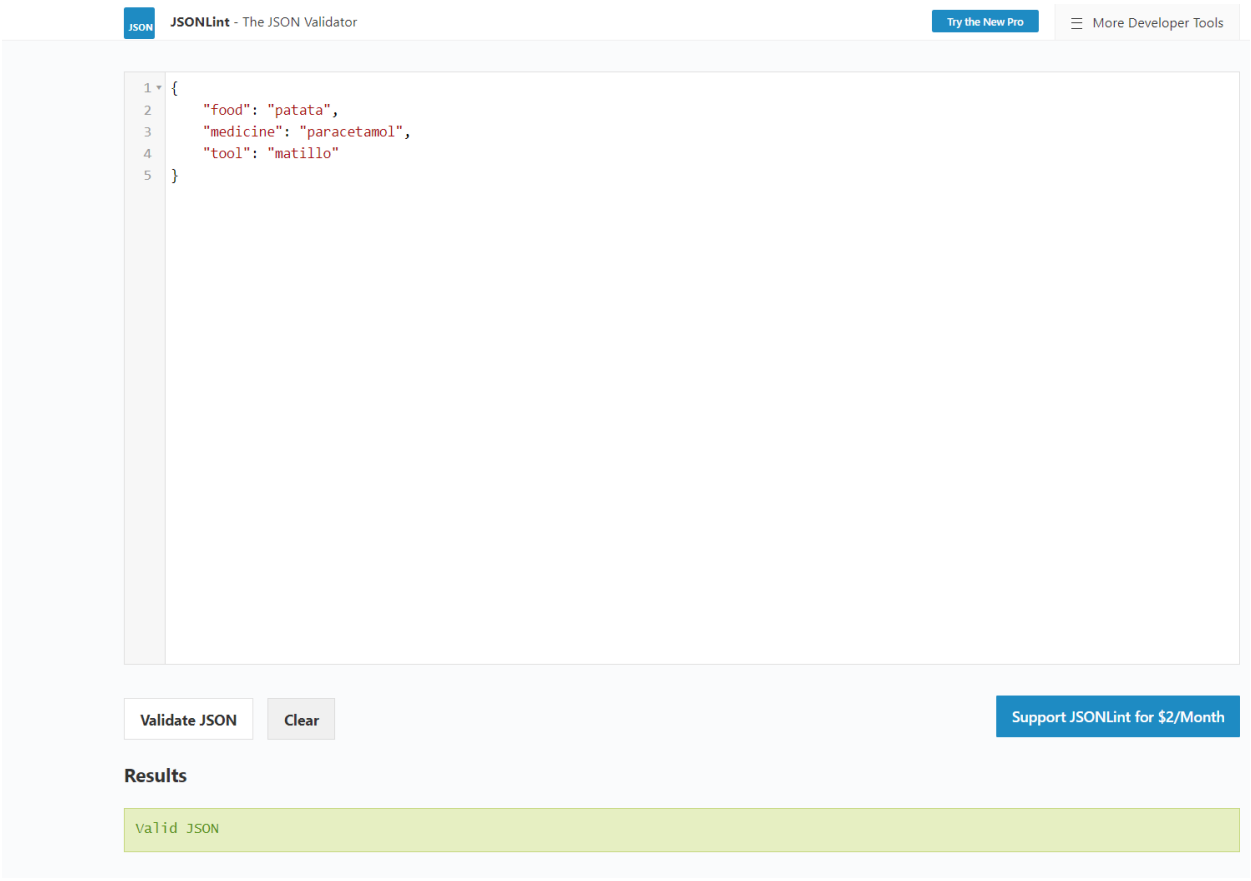
Main Class : 4/4
Second Class: 3/4 - no methods
Dependency: 2/2
Total: 9/10

2) Running program (cmd line executing the program)



2.1 showing option 5/5 pts.
2.2 action with external modify 5/5
Total: 10/10

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



Json: 10/10
Total: 10
4) Saving JSON data (screenshot of the json file)

C: > Users > DELL > Desktop > EXAM1PracticalExerciseYeshuaChiliquinga > ExamPOO > {} forniture.json > ...

```json
{
  "food": "patata",
  "medicine": "paracetamol",
  "tool": "A"
};
{
  "food": "papa",
  "medicine": "sabutamol",
  "tool": "B"
};
{
  "food": "zanahoria",
  "medicine": "ibuprofeno",
  "tool": "C"
};
{
  "food": "carne",
  "medicine": "Icterix",
  "tool": "D"
}
```

Printing Json File

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

```java
/**
 *
 * @author Yeshua Chiliquinga, JSons; DCCO-ESPE
 */
public class Forniture {
    private String food;
    private String medicine;
    private String tool;

    public Forniture(String food, String medicine, String tool) {
        this.food = food;
        this.medicine = medicine;
        this.tool = tool;
    }

    public String getFood() {
        return food;
    }

    public void setFood(String food) {
        this.food = food;
```

Class Furniture: Everything is OK
Class Simulator: 5 pt

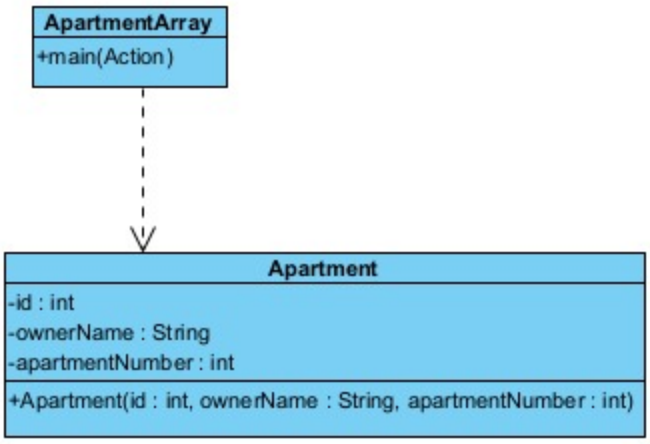
```java
*/
public class Simulator {
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(source: System.in);
        int selection;
        System.out.println(x: "Welcom");
        System.out.println("Select the options\n"
                + "1\n"
                + "2\n");
        selection=sc.nextInt();
        if(selection==1){
            writeJSON();
            }else{
                if(selection==2){
                   String jsonFilePath = "forniture.json";

                    try (BufferedReader reader = new BufferedReader(new FileReader(fileName: jsonFilePath))) {
                        StringBuilder jsonContent = new StringBuilder();
                        String line;
                        while ((line = reader.readLine()) != null) {
                            jsonContent.append(str: line);
                        }

                        // Mostrar el contenido del archivo JSON
                        System.out.println(x: jsonContent.toString());

                    } catch (IOException e) {
                        System.out.println("Error al leer el archivo JSON: " + e.getMessage());
                    }
                }
        }
```

The class contains: Everything the Jsonwrite and everything in the same class
Total:
Inspector: Pablo Carrera

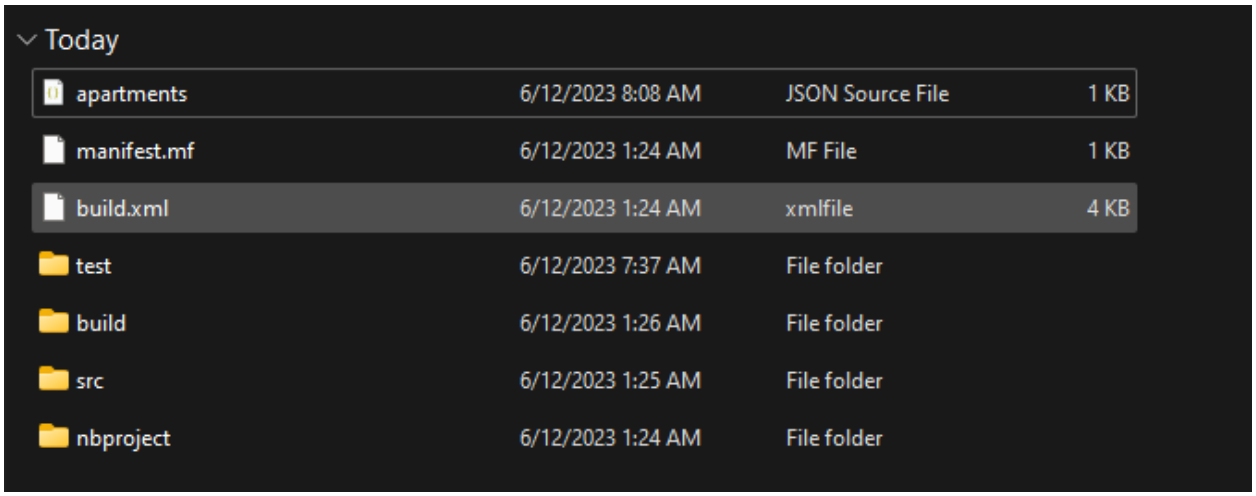
5. PAMELA NAOMI CHIPE ZAMBRANO (Apartmets) 46/50

Inspector: Yeshua Chiliquinga
Rubric:
1) class modeling 10 pts.
Class name: 4/4.
Class topic: 4/4
Relations: 2/2
2) Running program 10 pts.
Run program: 5/5
Run option: 5/5
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) No use void print



Main class : 4
Name class 1
Id 1
Variables 2
Use 2
Total: 10
2) Running program (cmd line executing the program) 10/10
3) Printing/counting/deleting objects from JSON File (according to the roster number) 8/5
4) Saving JSON data (screenshot of the json file)10/10



yes, it has the json structure
5) Code Quality (lines of code with stinky code) 8/10
many spaces between code

6. JOAN OSWALDO COBEÑA ZAMBRANO (Sports) /50
   Inspector: Pamela Chipe

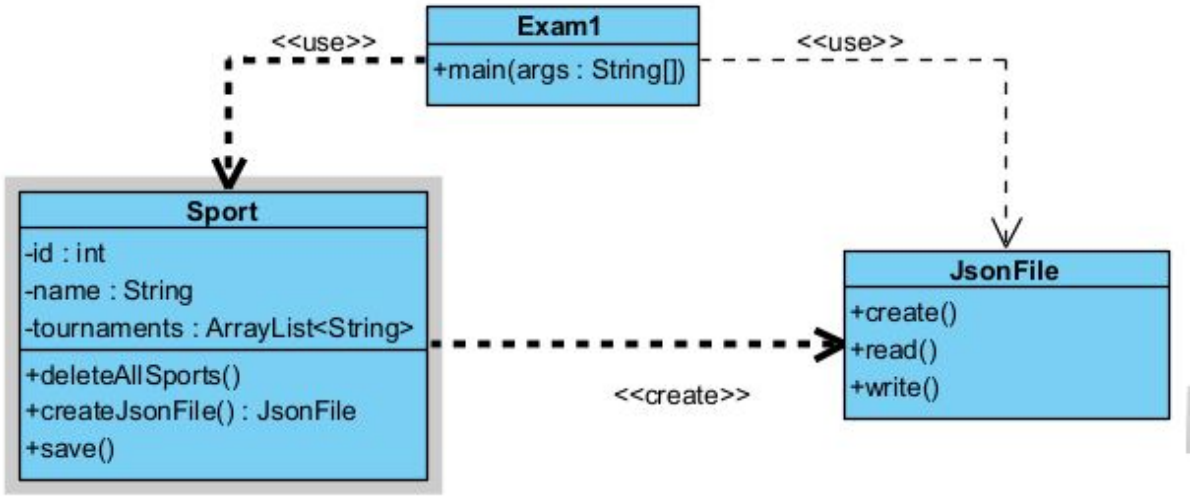
Rubric:

1) class modeling 10/10 pts.
   - Main class 4/4
   - Object class 4 /4
   - Dependency 2 /2
2) Running program 9/10 pts. run the code
   - showing option 5/5 pts.
   - action with external modify 4/5
3) Printing/counting/deleting objects from JSON File 10/ 10 pts.
4) Saving JSON data 10 /10pts.
5) Code Quality 9/10 pts. in delete it doesn't work quite correctly

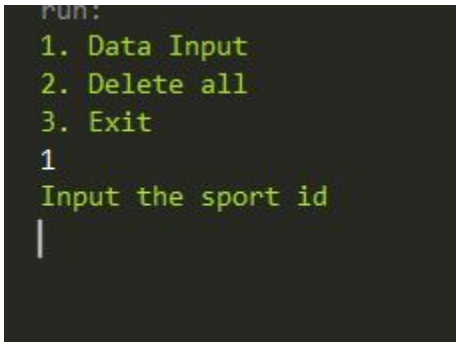
TOTAL: .....48/ 50 pts.

Evidence

1) class modeling (Class diagram)



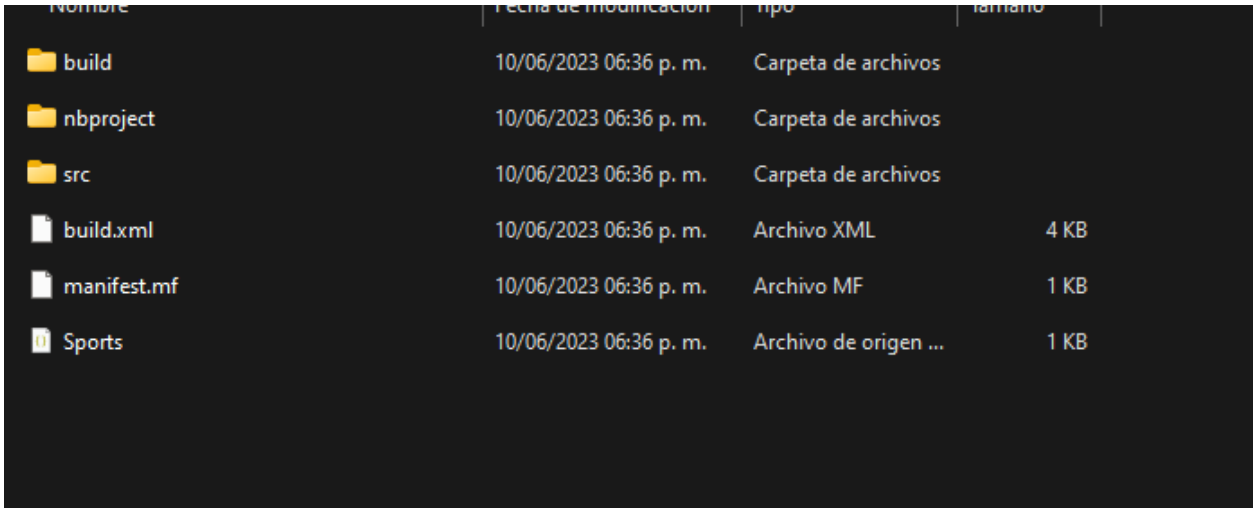
2) Running program (cmd line executing the program)

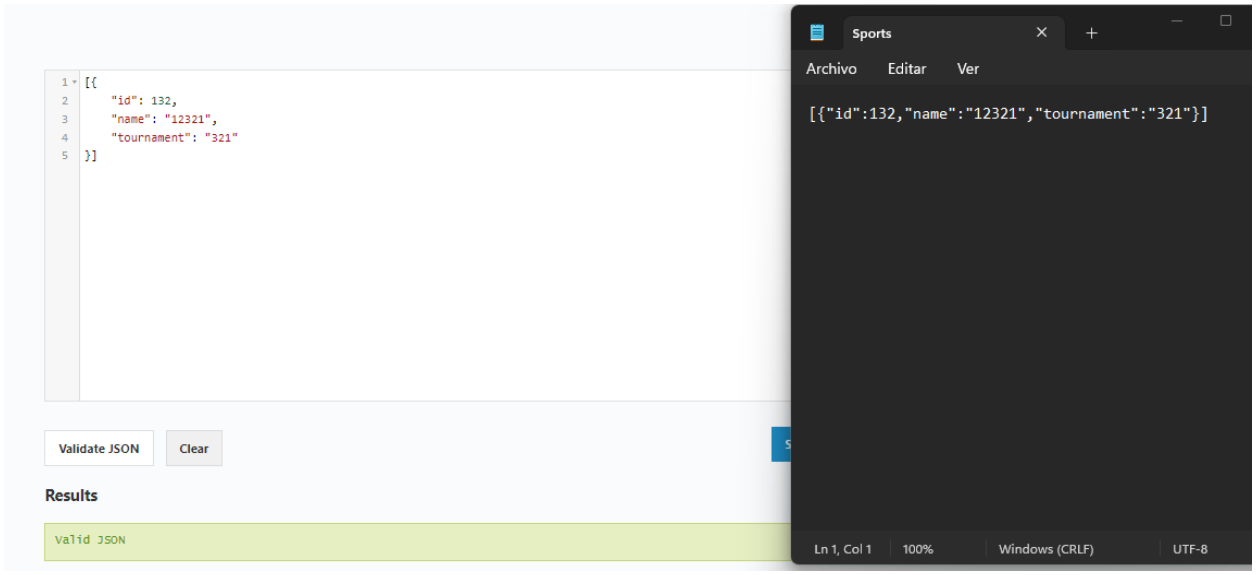


3) Printing/counting/deleting objects from JSON File (according to the roster number)

```
1. Data Input
2. Delete all
3. Exit
1
Input the sport id
1
Input the sport name
Football
Input the sport tournament
Champions
1. Data Input
2. Delete all
3. Exit
```

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





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

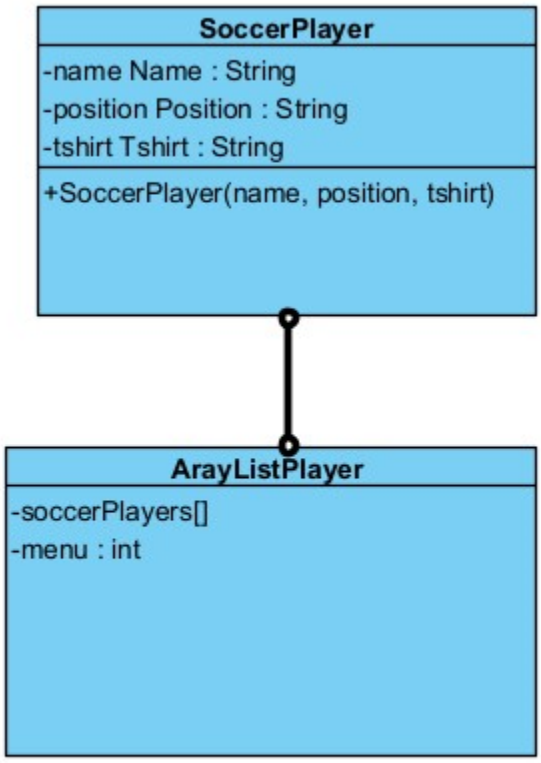
7. ANABEL DAVILA MOLINA Print Soccer players

Rubric:

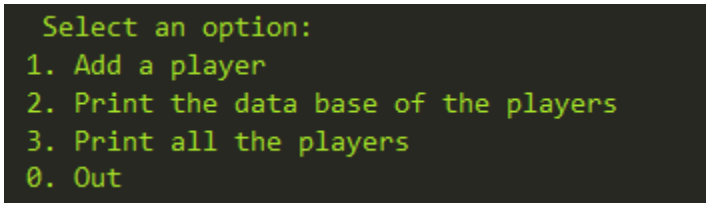
1) class modeling 7/10pts.
1.1 main class ¾ pts.
1.2 object class ¾ pts.
1.3 dependency ½ pts.
2) Running program 6/10 pts.
2.1 showing option 5/5 pts.
2.2 action with external modify 1/5
3) Printing/counting/deleting objects from JSON File 10/10 pts.
4) Saving JSON data 9/10 pts. Diagram no-concordance.
5) Code Quality 6/10 pts.
TOTAL: ..... 38 pts.

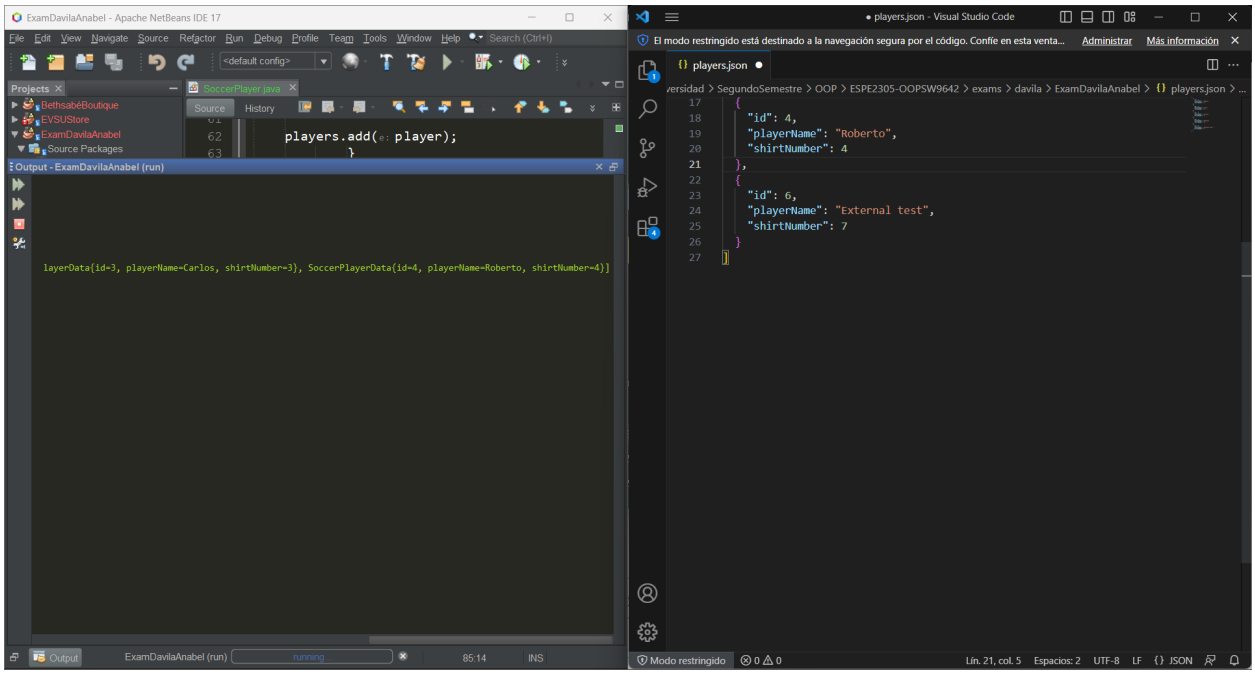
Evidence

1) class modeling (Class diagram)



2) Running program (cmd line executing the program)



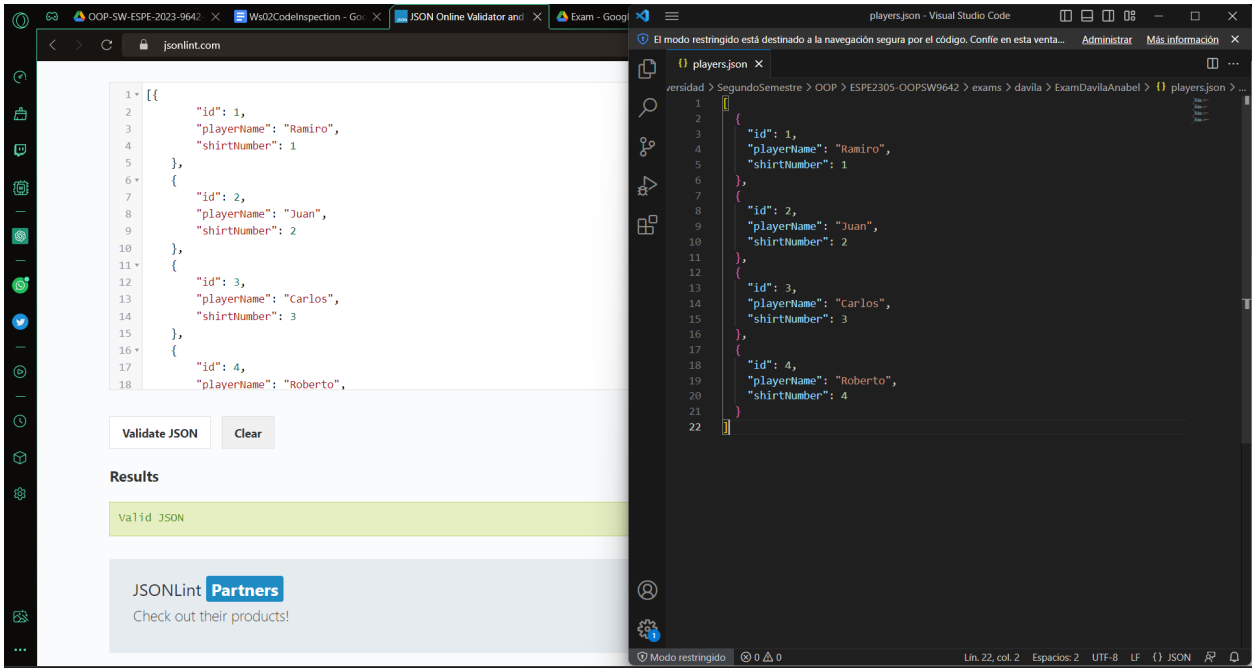


Default creations and no modify.

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

```
4
Add the name of the player
Test4
Add the shirt number of the player
4
 Select an option:
1. Add a player
2. Print the data base of the players
3. Print all the players
0. Out
1
Add the id of the player
5
Add the name of the player
Test5
Add the shirt number of the player
5
 Select an option:
1. Add a player
2. Print the data base of the players
3. Print all the players
0. Out
2
Data base file created
 Select an option:
1. Add a player
2. Print the data base of the players
3. Print all the players
0. Out
3
```

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



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

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this li
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
```

Unnecessary headers

```
*
* @author Anabel Davila,Kill Chain, DCOO-ESPE
*/
public class SoccerPlayer {

    //creamos un método para hacer el menú
public void SoccerPlayerData(){
```

Unnecessary comments

```java
public class SoccerPlayer {

    //creamos un método para hacer el menú
 public void SoccerPlayerData(){

    SoccerPlayerData player;
    ArrayList<SoccerPlayerData> players = new ArrayList<>();


    Scanner collectData = new Scanner(source: System.in);

    player = new SoccerPlayerData (id: 1, playerName: "Ramiro", shirtNumber: 1);
    players.add(e: player);
    player = new SoccerPlayerData (id: 2, playerName: "Juan", shirtNumber: 2);
    players.add(e: player);
    player = new SoccerPlayerData (id: 3, playerName: "Carlos", shirtNumber: 3);
    players.add(e: player);
    player = new SoccerPlayerData (id: 4, playerName: "Roberto", shirtNumber: 4);
    players.add(e: player);

    int menu;


    do{


        System.out.println(x: " Select an option: ");
        System.out.println(x: "1. Add a player ");
        System.out.println(x: "2. Print the data base of the players");
        System.out.println(x: "3. Print all the players");
```

Menu options in the SoccerPlayer class, instead of main.

```
do{


        System.out.println( x: " Select an option: ");
        System.out.println( x: "1. Add a player ");
        System.out.println( x: "2. Print the data base of the players");
        System.out.println( x: "3. Print all the players");
        System.out.println( x: "0. Out");

        menu = collectData.nextInt();
            switch (menu){
            case 1 -> {
                System.out.println( x: "Add the id of the player");
                int id = collectData.nextInt();
                System.out.println( x: "Add the name of the player");
                String playerName = collectData.next();
                System.out.println( x: "Add the shirt number of the player");
                int shirtNumber = collectData.nextInt();
                player = new SoccerPlayerData (id, playerName, shirtNumber);

    players.add( e: player);
            }
```

Bad Indentation.

SoccerPlayer.java

Source | History

```
62          players.add( e: player);
63                  }
64                  case 2 -> {
65
66              Gson gson = new GsonBuilder().setPrettyPrinting().create();
67
68
69              String filePath = "players.json";
70
71              try (FileWriter fileWriter = new FileWriter( fileName: filePath)) {
72
73                  gson.toJson( src: players,  writer: fileWriter);
74                  System.out.println( x: "Data base file created");
75              } catch (IOException e) {
76                  System.out.println("There was an error at creating the file: " + e.getMessage());
77              }
78                  }
79
80                  case 3 -> {
81
82                      System.out.println("These are the players " + players);
83
84
85                  }
86                  }
87              }while(menu != 0);
88
```

JSON File CRUD in the same SoccerPlayer class.

Inspector: Joan Cobeña

8. ANDRES ALEXANDER ESPIN ANDRADE 0/50 Forms wasn't answered

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.

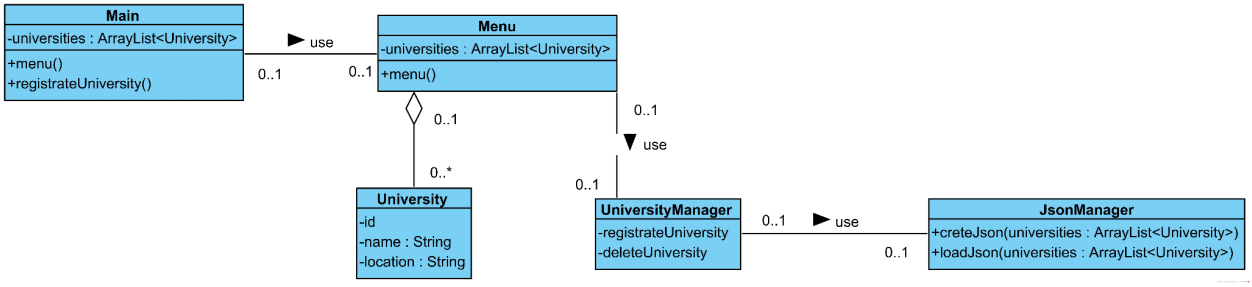
Evidence
1) class modeling (Class diagram)
2) Running program (cmd line executing the program)
3) Printing/counting/deleting objects from JSON File (according to the roster number)
4) Saving JSON data (screenshot of the json file)
5) Code Quality (lines of code with stinky code)
Inspector: Anabel Davila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

9. ALEJANDRO CAETANO FLORES MAYA University (delete)
Inspector: Espin Andres

Rubric:
1) class modeling 10 pts.
Class name: 4/4.
Class topic: 4/4
Relations: 2/2
2) Running program 10 pts.
Run program: 5/5
Run option: 5/5
3) Printing/counting/deleting objects from JSON File 10 pts.
Json Validate: 10/10
4) Saving JSON data 10pts.
Saving data 10/10
5) Code Quality 10 pts.

TOTAL: ..... 50 pts.

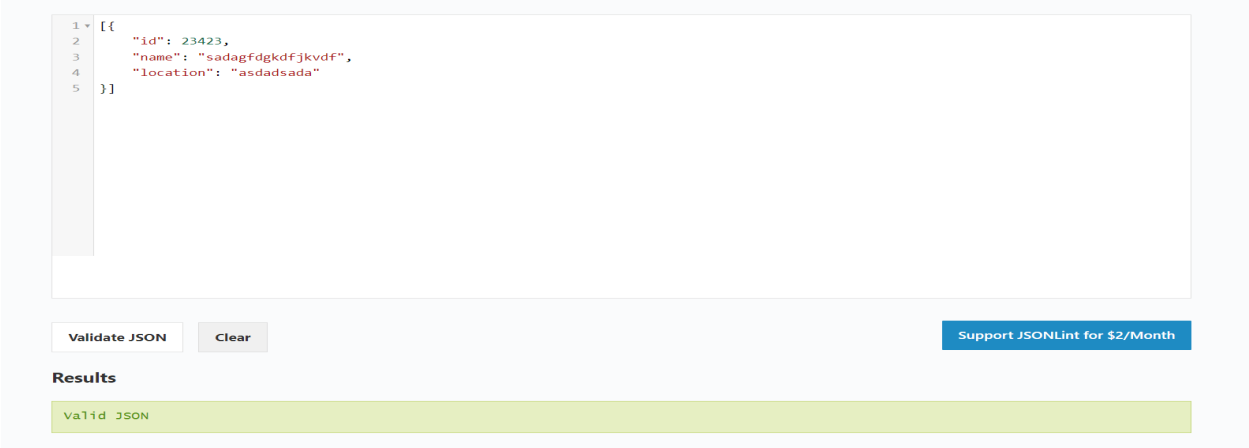
Evidence
1) class modeling (Class diagram)



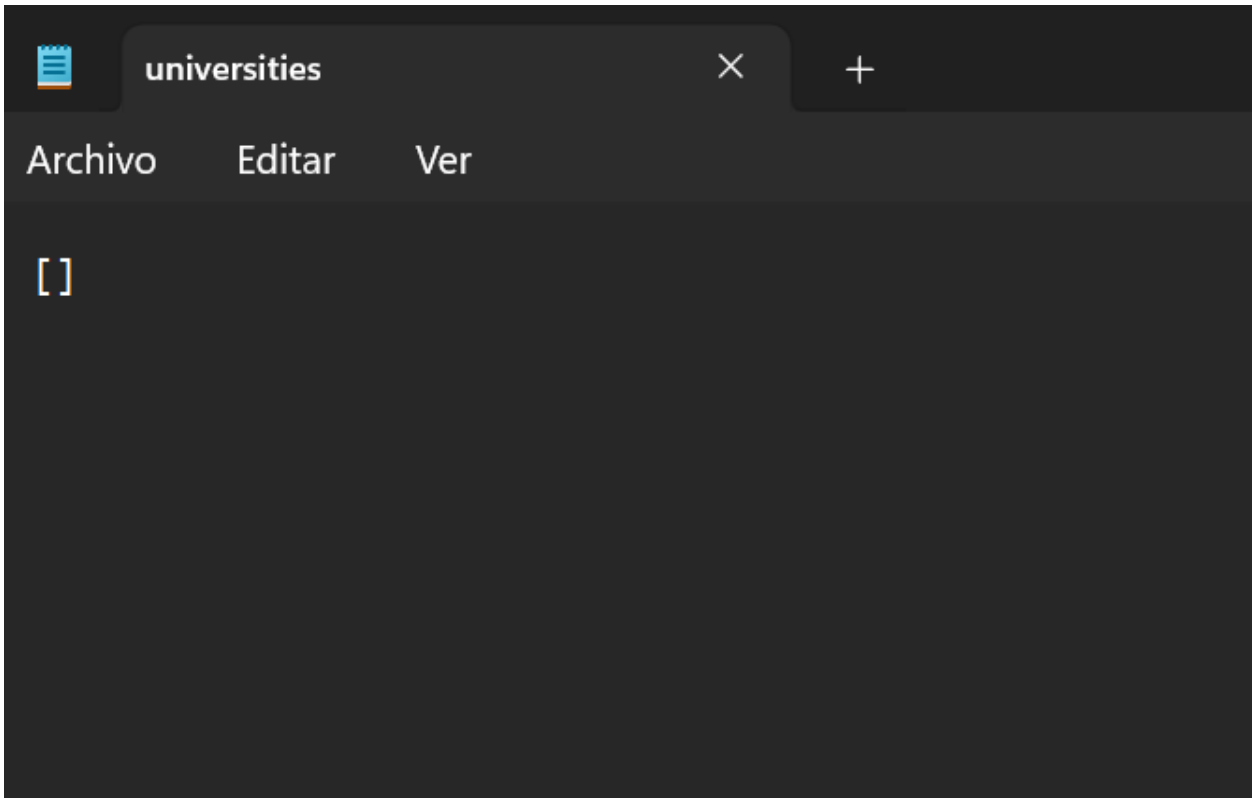
2) Running program (cmd line executing the program)

```
run:
1.Enter university
2.Delete all universities
3.Close
1
Enter university id:
12
Enter university name:
saasdad
Enter university location:
qwdqweqw
University registrated correctly
1.Enter university
2.Delete all universities
3.Close
2
All universities deleted correctly
1.Enter university
2.Delete all universities
3.Close
```

3) Printing/counting/deleting objects from JSON File (Delete)



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



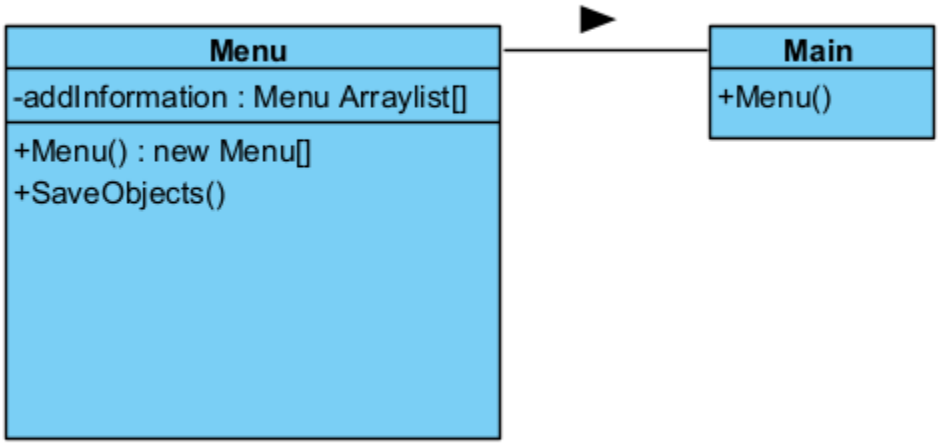
5) Code Quality (lines of code with stinky code)
Clean code 9/10
TOTAL: ..... 49/50 pts.

10. JORDAN ALEXANDER GUAMAN ALCIVAR - Count Buildings
Inspector: Caetano Flores
Rubric:
1) class modeling 5/10 pts.
Main class: 4/4.
Object Class: 0/4 pts.

Dependence: 1/2 pts. Association instead of dependency

2) Running program 10/10 pts.
3) Printing/counting/deleting objects from JSON File 6/10 pts. List the items instead of cout it.
4) Saving JSON data 9/10pts. Missing building name
5) Code Quality 9/10 pts. Bad indentation on line 57 in “TenanPerson”
TOTAL: ..... 39/50 pts.

Evidence
1) class modeling (Class diagram)



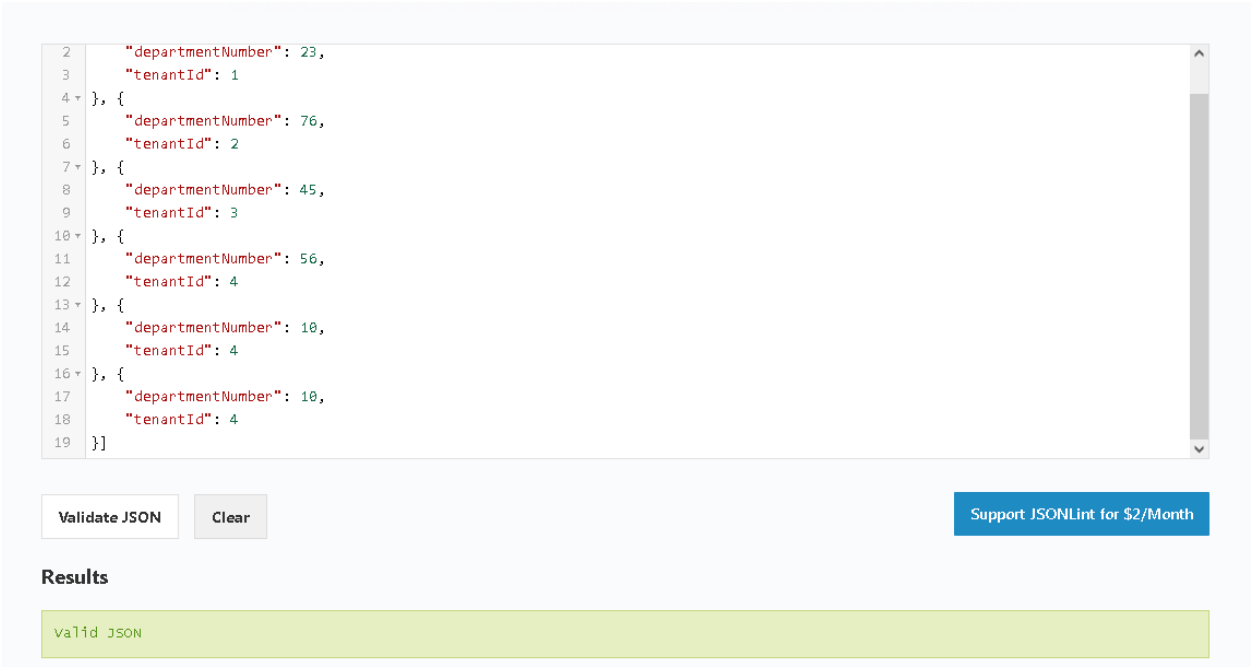
2) Running program (cmd line executing the program)

```
 Select an option:
1. Add a tenant:
2. Print the data base of the tenants in the build :
3. Print all the tenents
0. Out
2
Data base file created,OK
------BUILDING SIMULATOR---------
 Select an option:
1. Add a tenant:
2. Print the data base of the tenants in the build :
3. Print all the tenents
0. Out
3
These are the tenants: [TenantData{departmentNumber=23, tenantId=1, tenantName=null}, TenantData{departmentNumber=76, tenantId=2, tenantName=null},
```

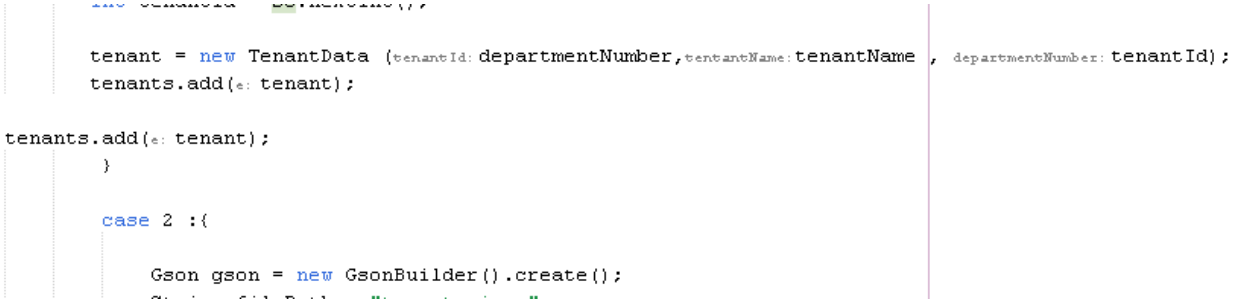
3) Printing/counting/deleting objects from JSON File (according to the roster number)

```
These are the tenants: [TenantData{departmentNumber=23, tenantId=1, tenantName=null}, TenantData{departmentNumber=76, tenantId=2, tenantName=null},
```

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



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

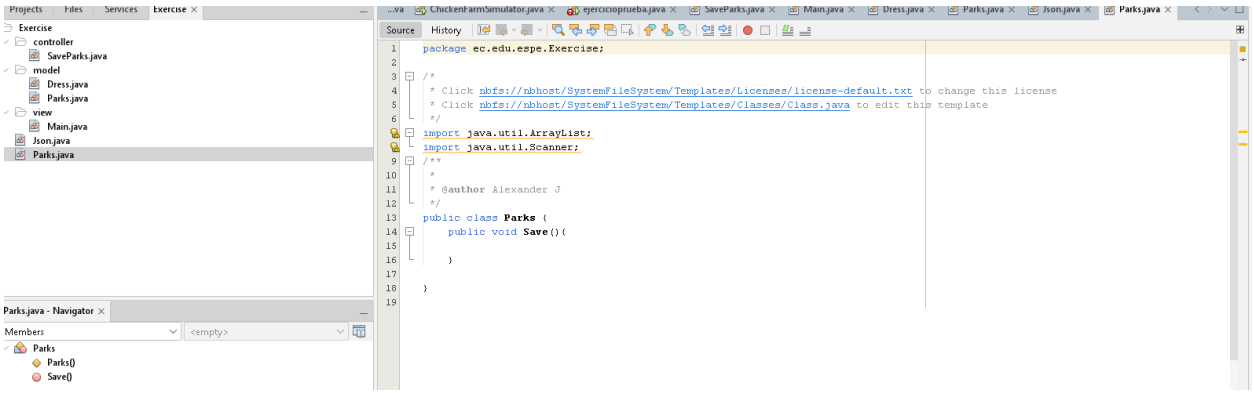


11. CARLOS ANDRES JAYA HERRERA 0/50 Forms wasn't answered
    Inspector: Jordan Guaman
    Rubric:
    1) class modeling 7/10 pts.
    2) Running program 5/10 pts.
    3) Printing/counting/deleting objects from JSON File /10 pts.
    4) Saving JSON data /10pts.
    5) Code Quality /10 pts.
    TOTAL: ..... 50 pts.

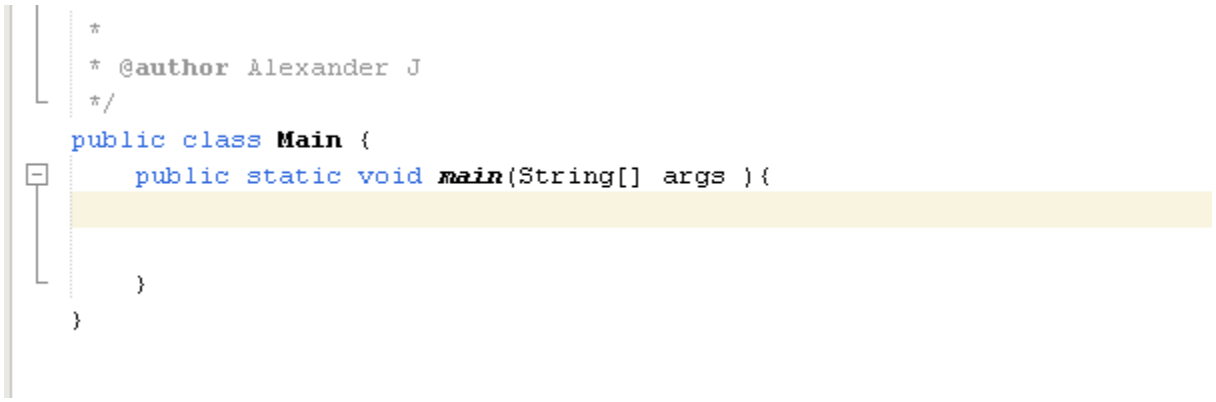
12. BRYAN ALEXANDER JUMBO SALCEDO 18/50

    Inspector: Carlos Jaya

Rubric:
1) class modeling 9 pts. Unused classes



2) Running program 0 pts. The program does not execute, nothing on the main class



3) Printing/counting/deleting objects from JSON File 0 pts.
4) Saving JSON data 0pts.
5) Code Quality 9 pts.Unused code

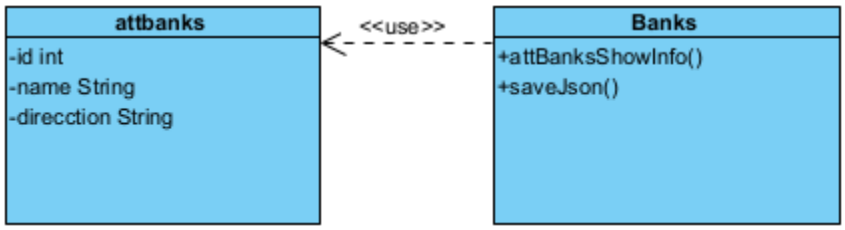
TOTAL: ..... 0 pts.

13. JEFFREY ALEXANDER MANOBANDA CHABLA 0/50 Forms wasn't answered
    Inspector: Bryan Jumbo

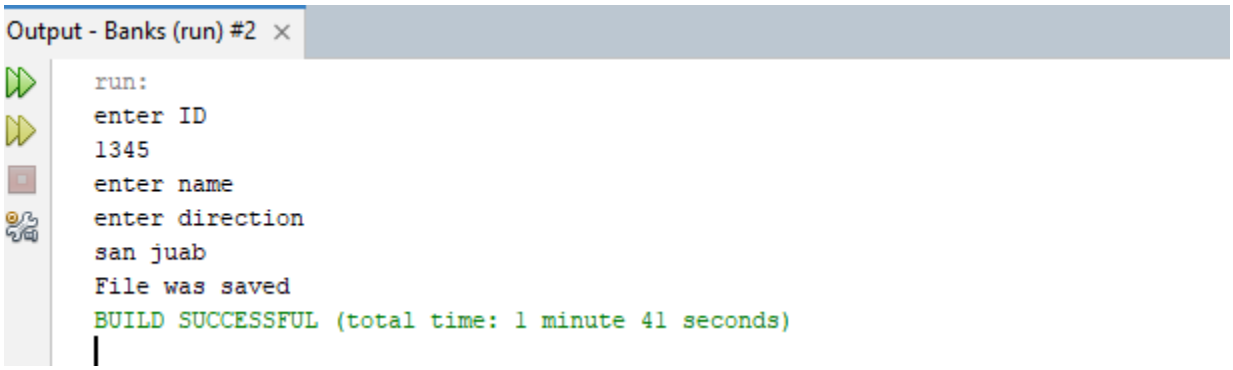
14. JHORDY PAUL MARCILLO MORA
    Inspector: Jeffrey Alexander Manobanda Chabla
    2) Count Banks

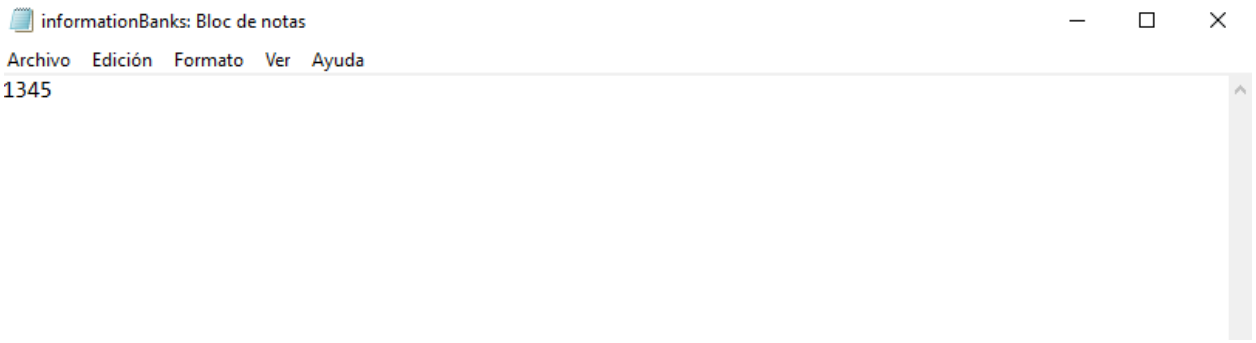
Rubric:
1) class modeling 8/10 pts.
2) Running program 7/10 pts. (No valid Json)

3) Printing/counting/deleting objects from JSON File 5/10 pts. (No valid Json)
4) Saving JSON data 0/10pts. (No valid Json)
5) Code Quality 9/10 pts. (The json file is in the main)
TOTAL: ..... 29 / 50 pts.
Evidence
1) class modeling (Class diagram)
Main class: 3/4.
Object Class: 3/4
Dependence: 2/2

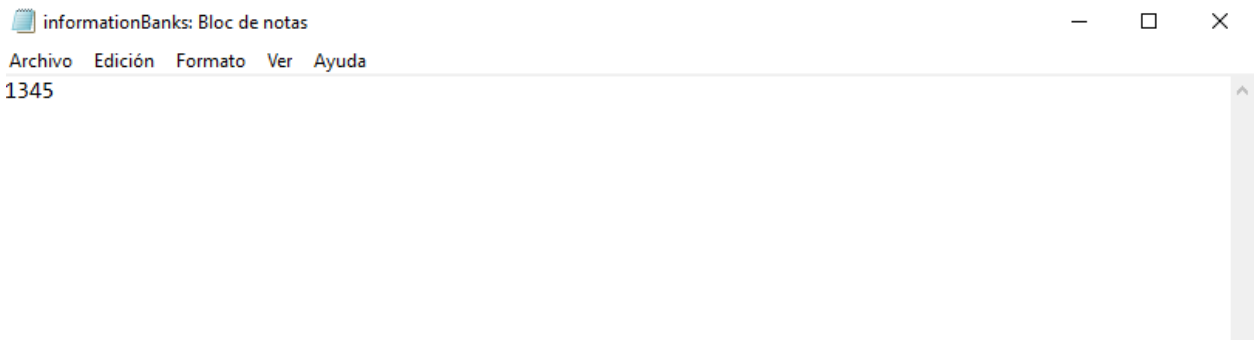


2) Running program (cmd line executing the program)

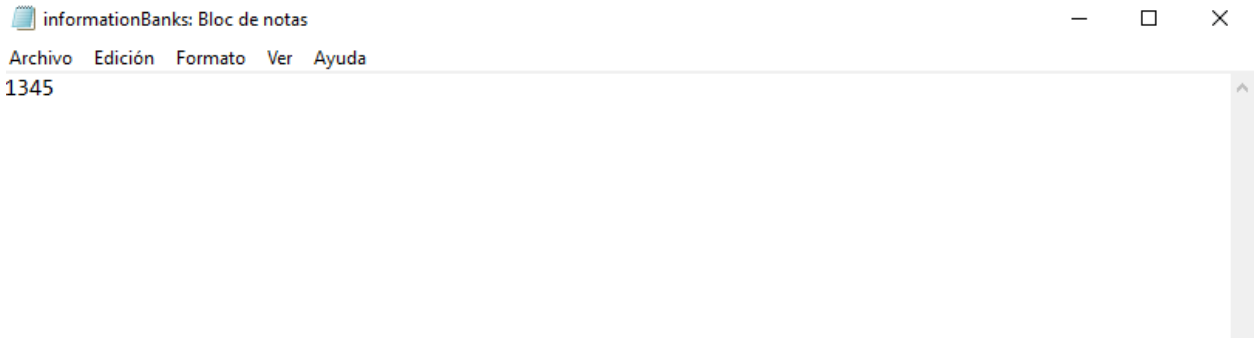




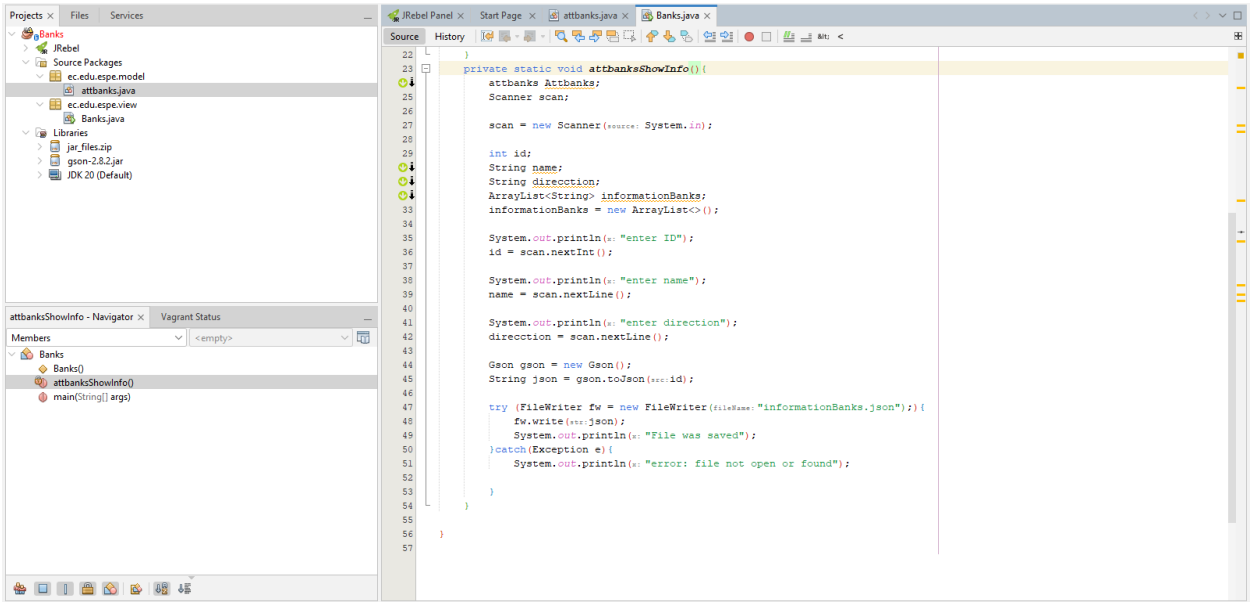
3) Printing/counting/deleting objects from JSON File (according to the roster number)



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



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



15. JOSUE ISAAC MARIN ALQUINGA(Student) (Delete objects)
Inspector: Jhordy Marcillo

Rubric:

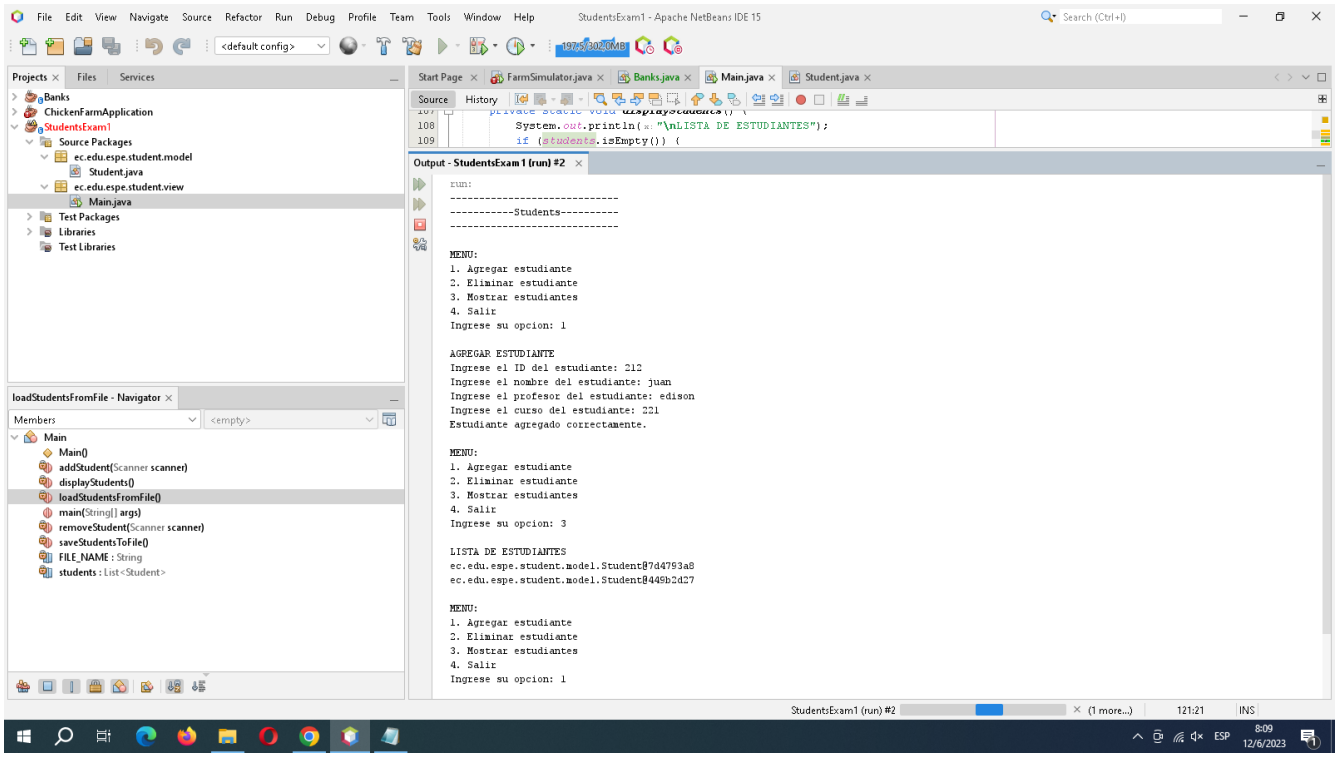
1) class modeling 7/10 pts.
2) Running program 10/10 pts.
3) Printing/counting/deleting objects from JSON File 10 /10 pts.
4) Saving JSON data 8/10pts.

5) Code Quality 7/10 pts.
TOTAL: ..... 42/50 pts.
Evidence

1) Class modeling (Class diagram)

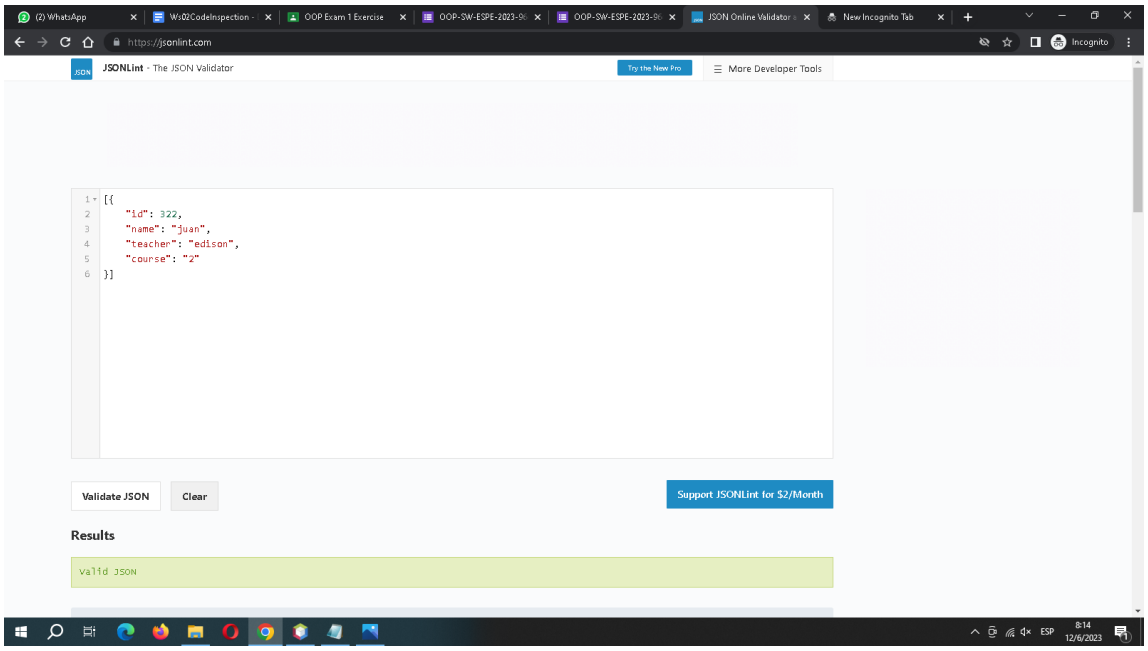
Wrong name, No methods

2) Running program



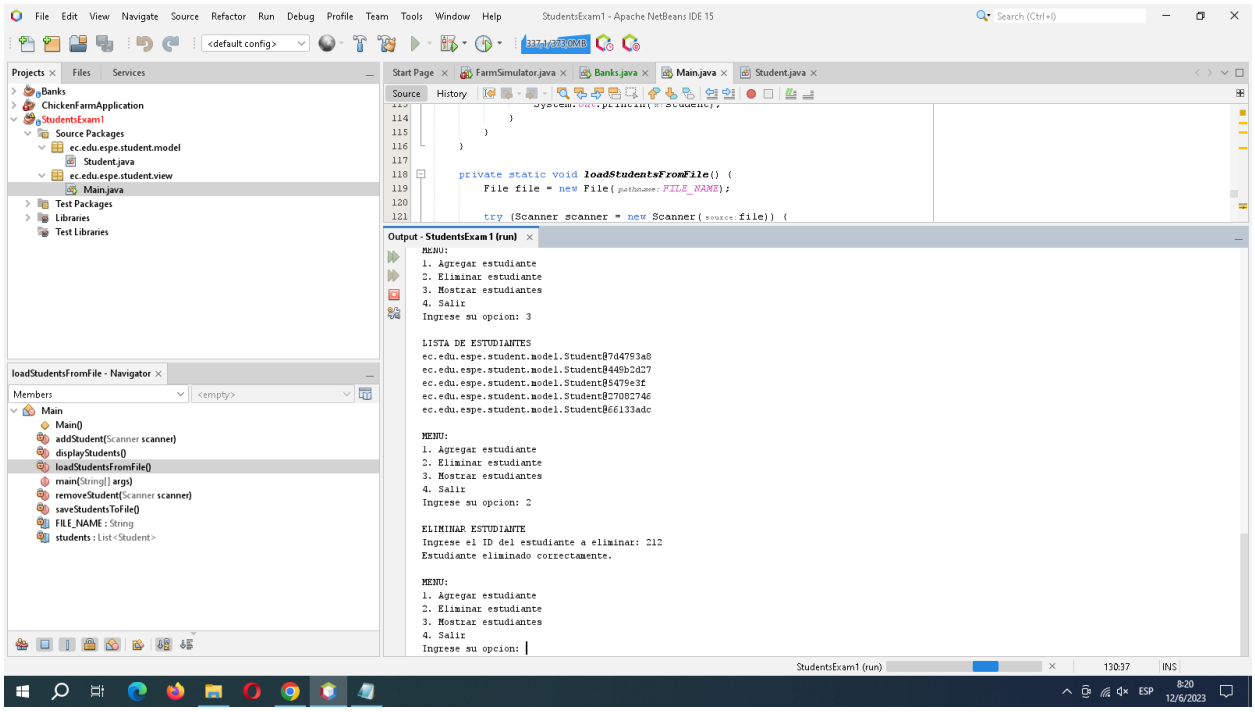
In the program everything runs correctly

3) Printing/counting/deleting objects from JSON File



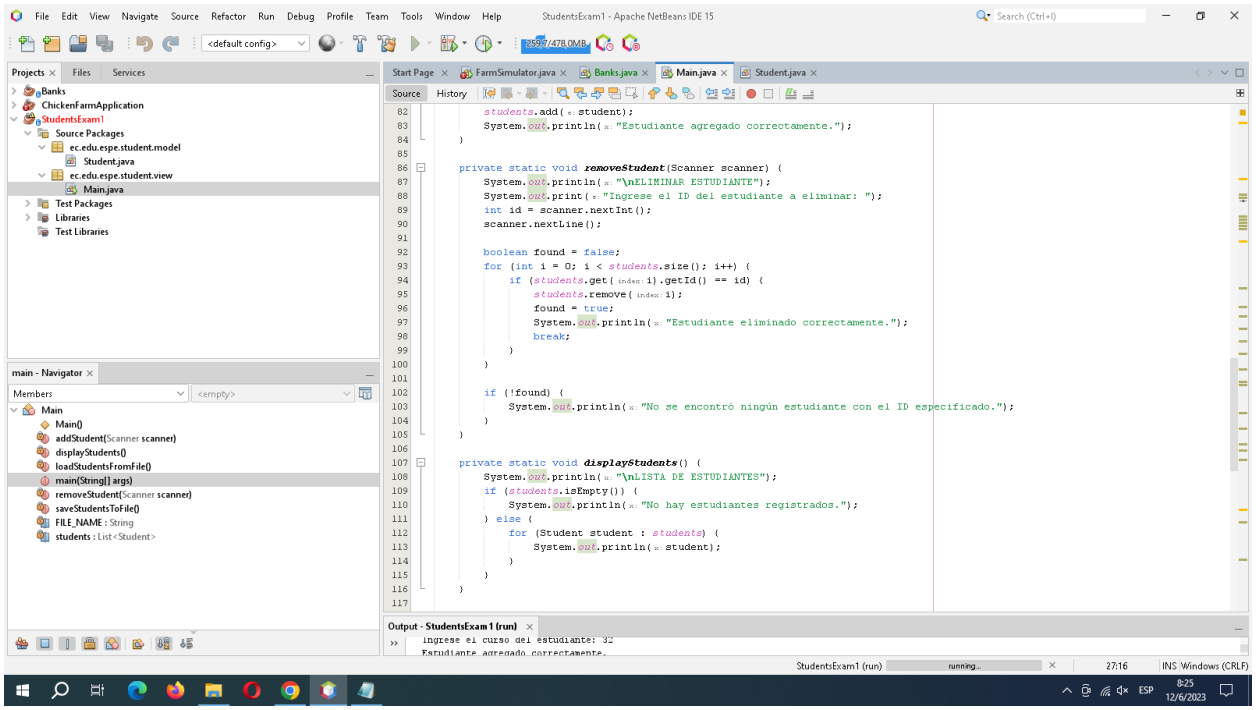
All attributes in json :)

4) Saving JSON data



Display is not displayed correctly

5) Code Quality



Everything is in the same class

16. MESIAS ORLANDO MARISCAL OÑA (Print All objects)

Inspector: Josue Isaac Marin Alquinga

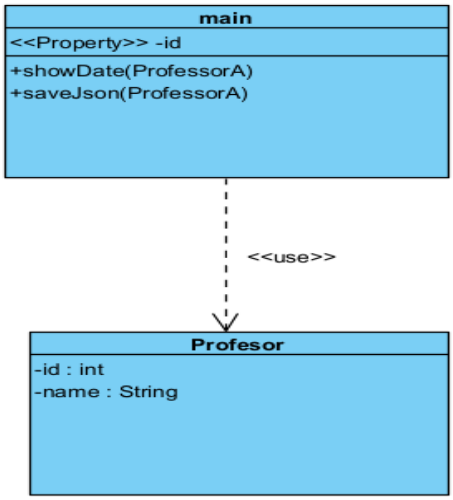
Rubric:

1) class modeling **7/10 pts.**
2) Running program **10/10 pts.**
3) Printing/counting/deleting objects from JSON File **0/10 pts.**
4) Saving JSON data **0/10pts.**
5) Code Quality **8/10 pts.**

TOTAL: ..... **25/50 pts.**

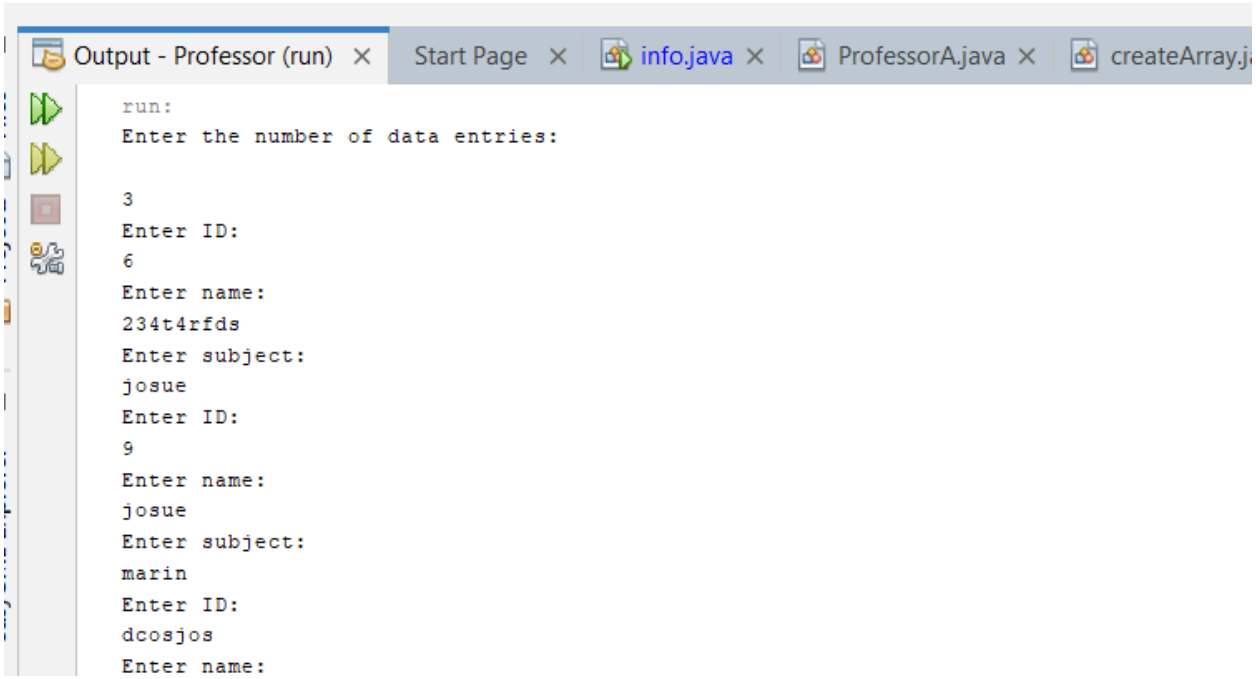
Evidence

1) class modeling (Class diagram)



Name with lowercase. (¾)
has no attributes (2/4)
Dependence(2/2)
Total:7/10

2) Running program (cmd line executing the program)



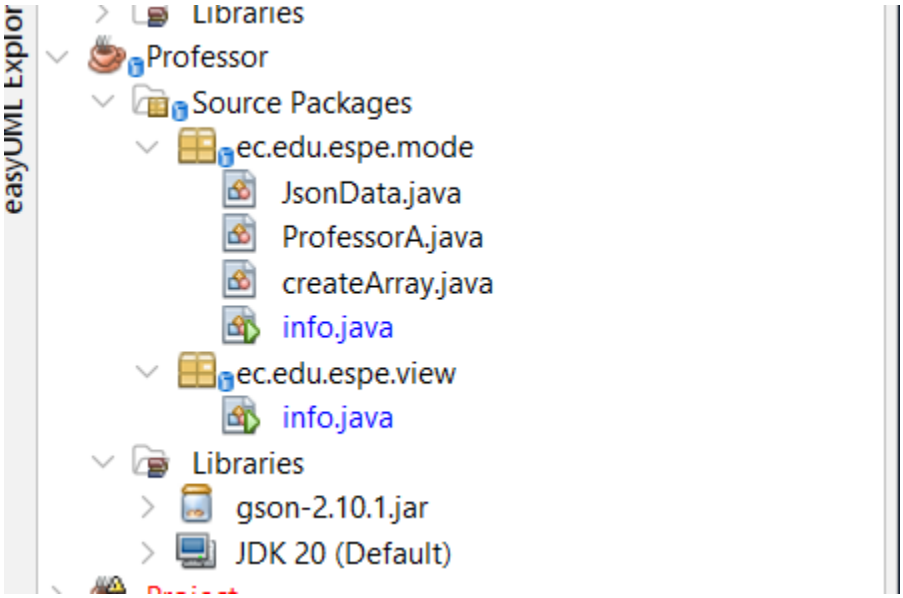
Run the program: 10/10

3) Printing/counting/deleting objects from JSON File (according to the roster number) does not have JSON 0/10

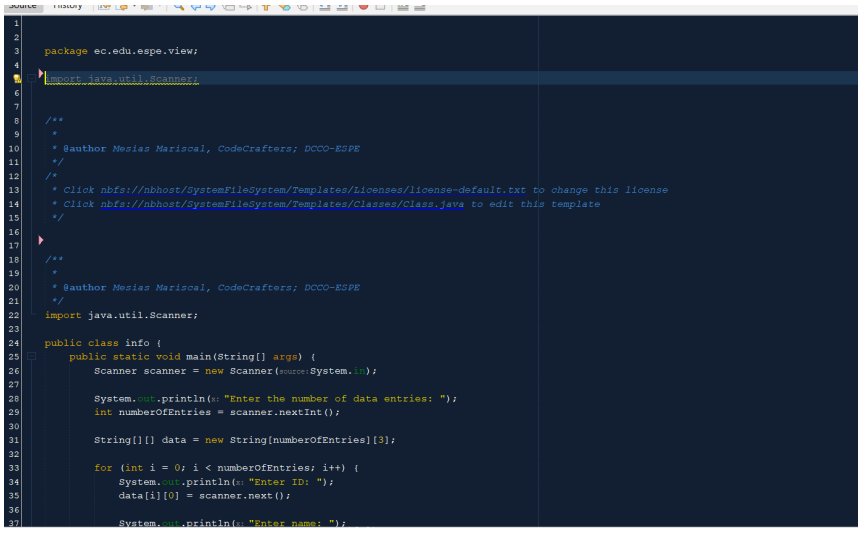
4) Saving JSON data (screenshot of the json file)

0/10

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



Unnecessary classes



```
package ec.edu.espe.view;

import java.util.Scanner;


/**
 *
 * @author Mesias Mariscal, CodeCrafters; DCCO-ESPE
 */
/*
 * 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
 */


/**
 *
 * @author Mesias Mariscal, CodeCrafters; DCCO-ESPE
 */
import java.util.Scanner;

public class info {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(source:System.in);

        System.out.println(x: "Enter the number of data entries: ");
        int numberOfEntries = scanner.nextInt();

        String[][] data = new String[numberOfEntries][3];

        for (int i = 0; i < numberOfEntries; i++) {
            System.out.println(x: "Enter ID: ");
            data[i][0] = scanner.next();

            System.out.println(x: "Enter name: ");
```

Unnecessary comments

8/10

17. ANTHONY ALAIN MORALES CARVAJAL (add

Inspector: Mesias Mariscal

Rubric:

1) class modeling 7/10

2) Running program 5/10 pts.

3) Printing/counting/deleting objects from JSON File 0/10

No json

4) Saving JSON data 0pts.

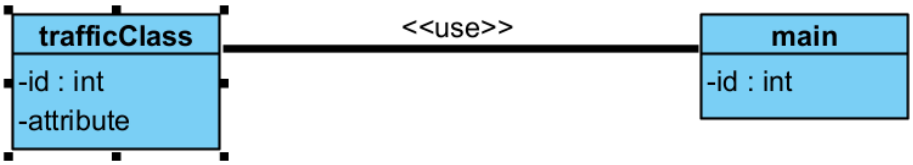
No file

5) Code Quality 10 /10

TOTAL: ..... 22 pts.

Evidence

1) class modeling (Class diagram)



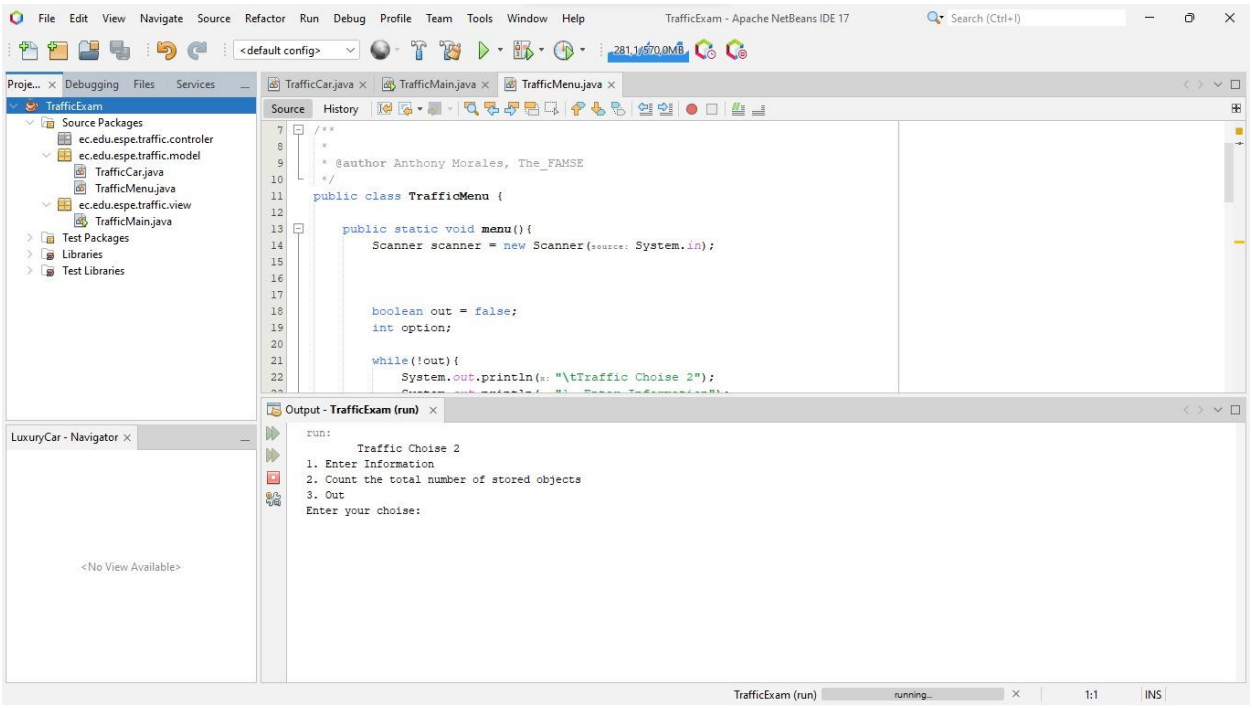
Name with lowercase. (3/4)

has no attributes (2/2)

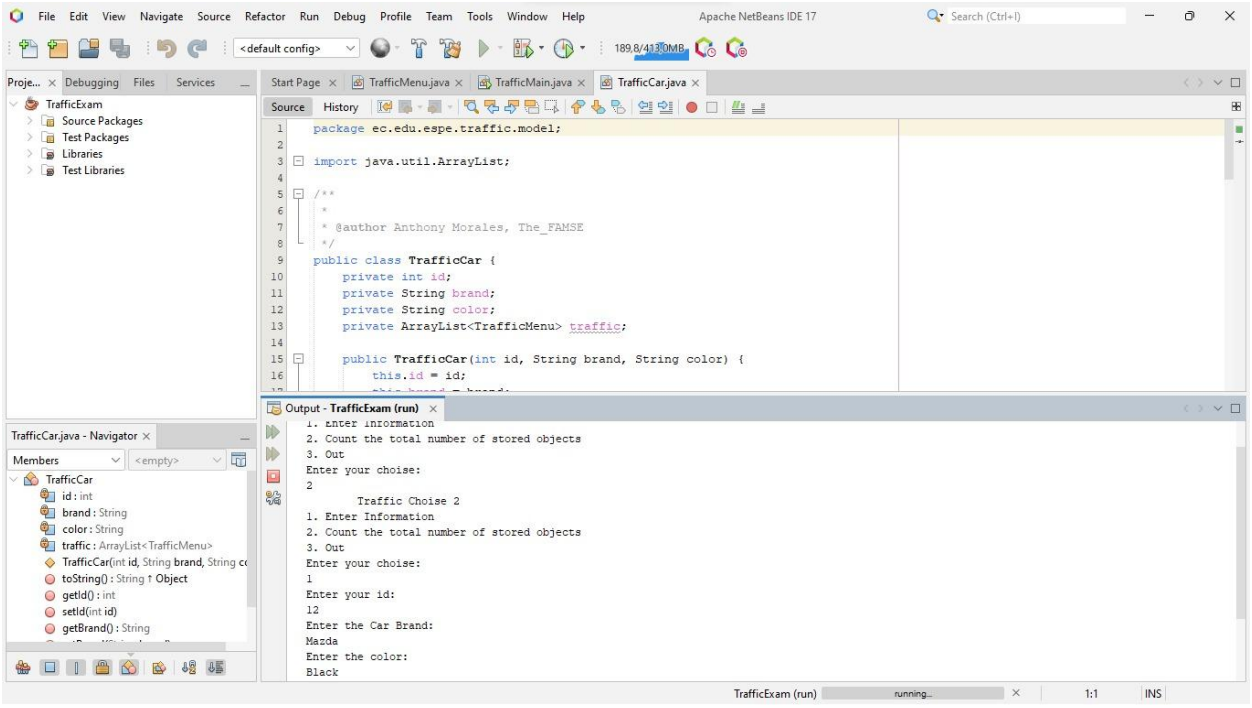
Dependence(2/2)

Total: 7

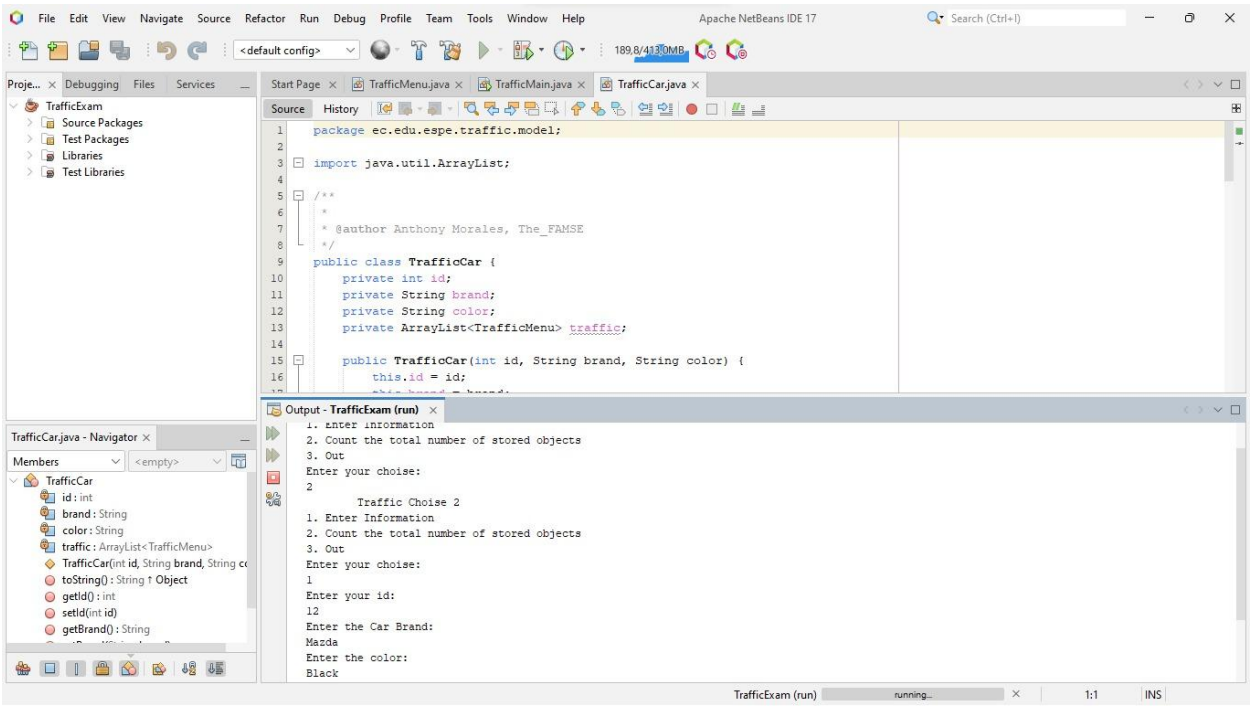
2) Running program (cmd line executing the program)



Option 1



Option 2



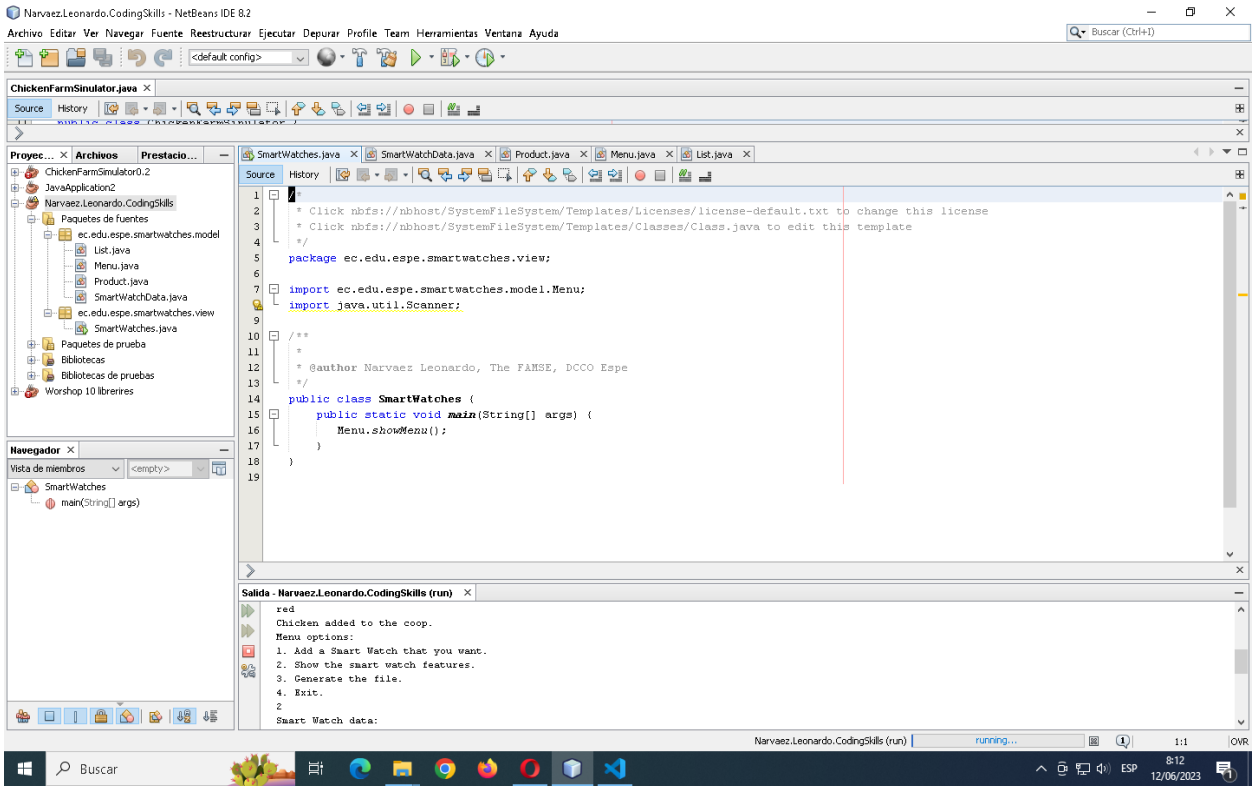
3) Printing/counting/deleting objects from JSON File (according to the roster number)
No json

4) Saving JSON data (screenshot of the json file)
No file

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

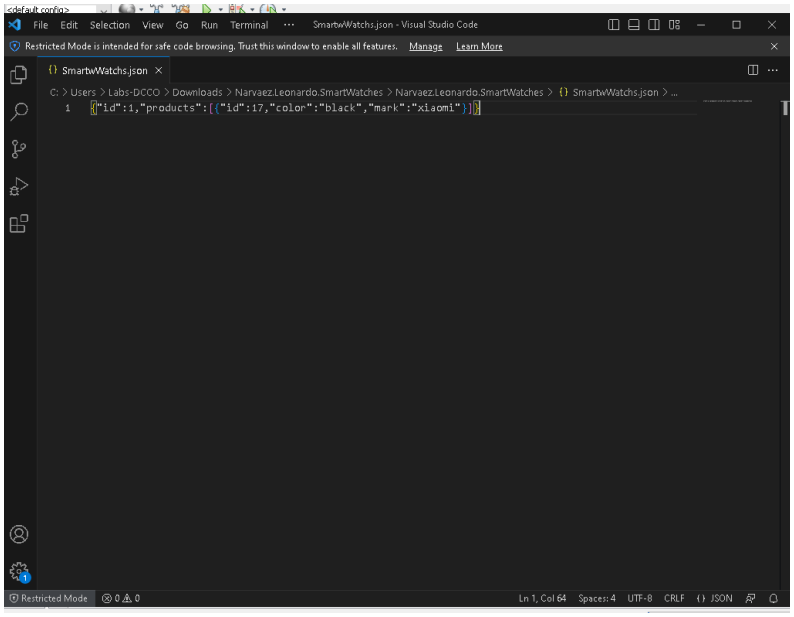
18. LENIN DAVID MORAN PALOMO 0/50 Form wasn't answered. (Trucks/Delete)
Inspector: Anthony Morales

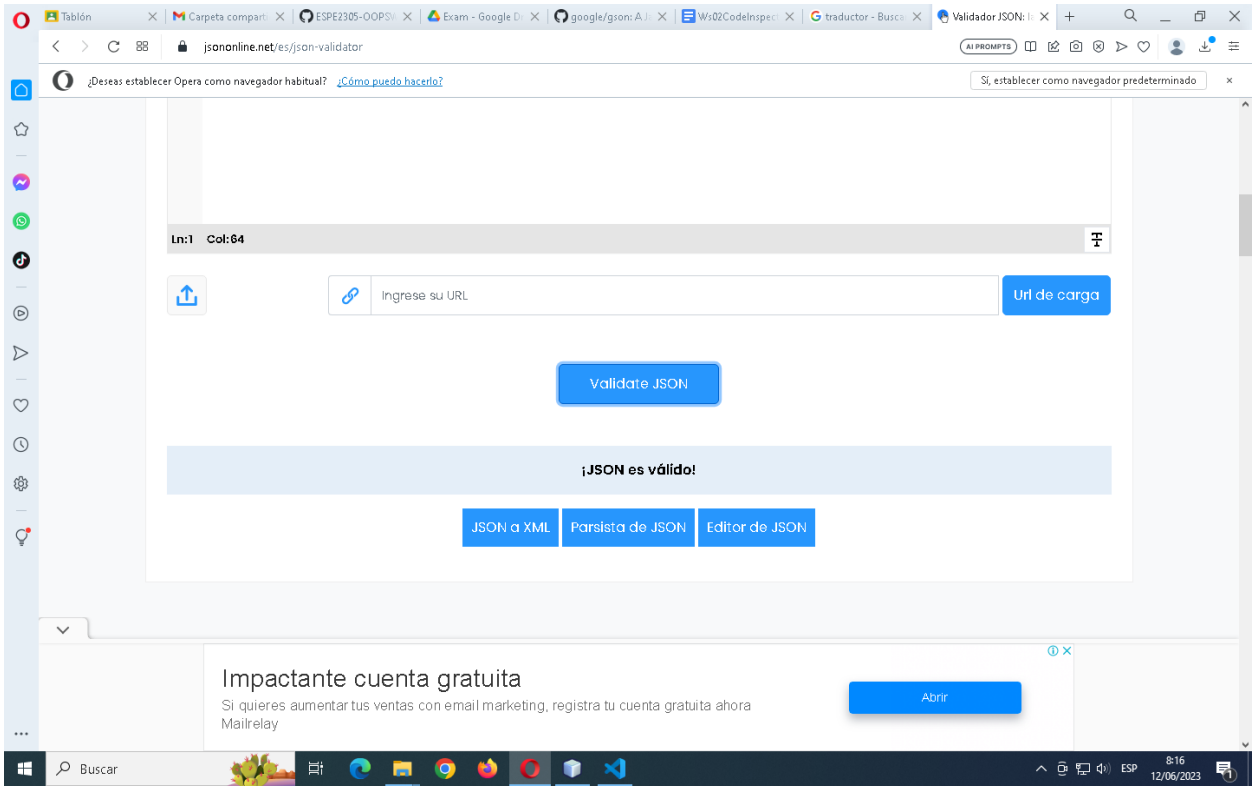
19. LEONARDO VINICIO NARVAEZ CRIOLLO
Inspector: Moran David
Rubric:
1) class modeling **0/10 pts.**
   - The name of the main is not valid because it could be confused
   - no diagram
2) Running program **10/10 pts**.
Evidence



3) Printing/counting/deleting objects from JSON File **10/10 pts.**

Evidence





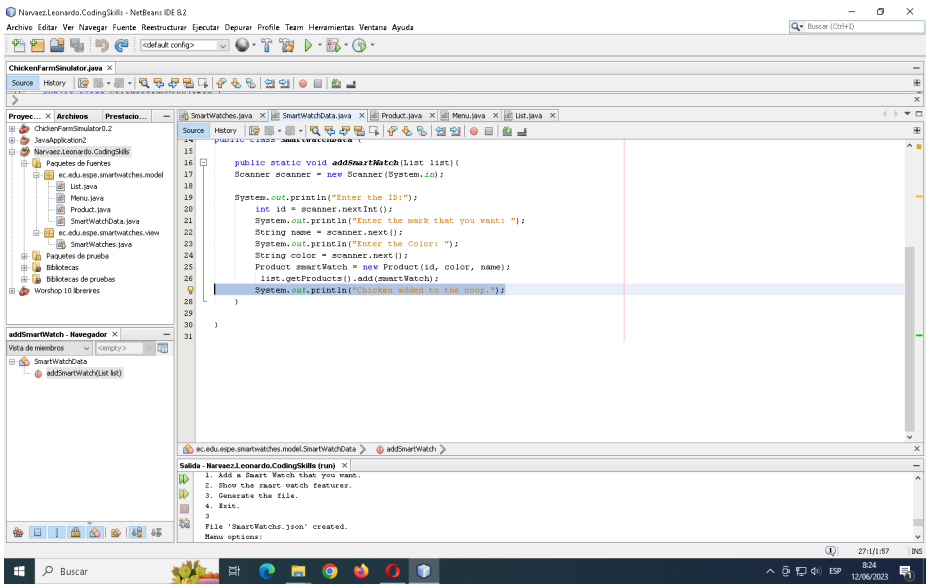
4) Saving JSON data **4/10pts.**

- generates data but not complete and with line break

5) Code Quality **8/10 pts.**

Evidence

- unnecessary line of code
- unnecesary line of code and in revision of the rest is fine



TOTAL: ..... **32/50 pts.**

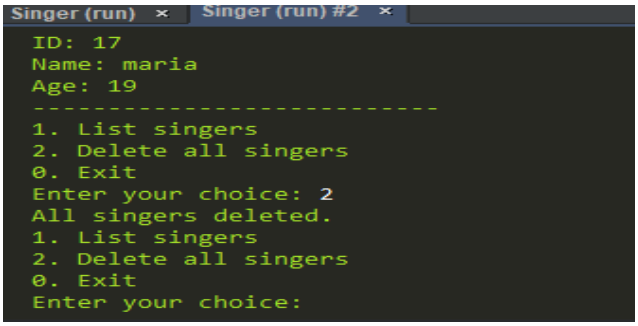
20. JOHN MATEO QUISHPE LLUMIQUINGA 0/10 Not sent to the form.
    Inspector: Leonardo Narváez.
21. LEONEL ALEJANDRO TIPAN QUINTO (Singers, Delete objects)
    Inspector: Leonardo Narváez.

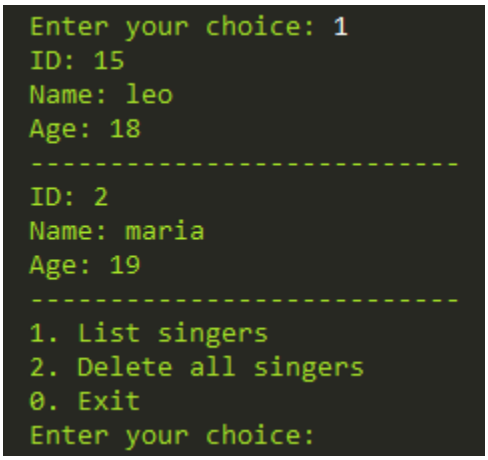
Rubric:

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)
0/10. No class diagram uploaded.
2) Running program (cmd line executing the program)
The program runs very well: 10/10
Run program: 5/5
Run option: 5/5

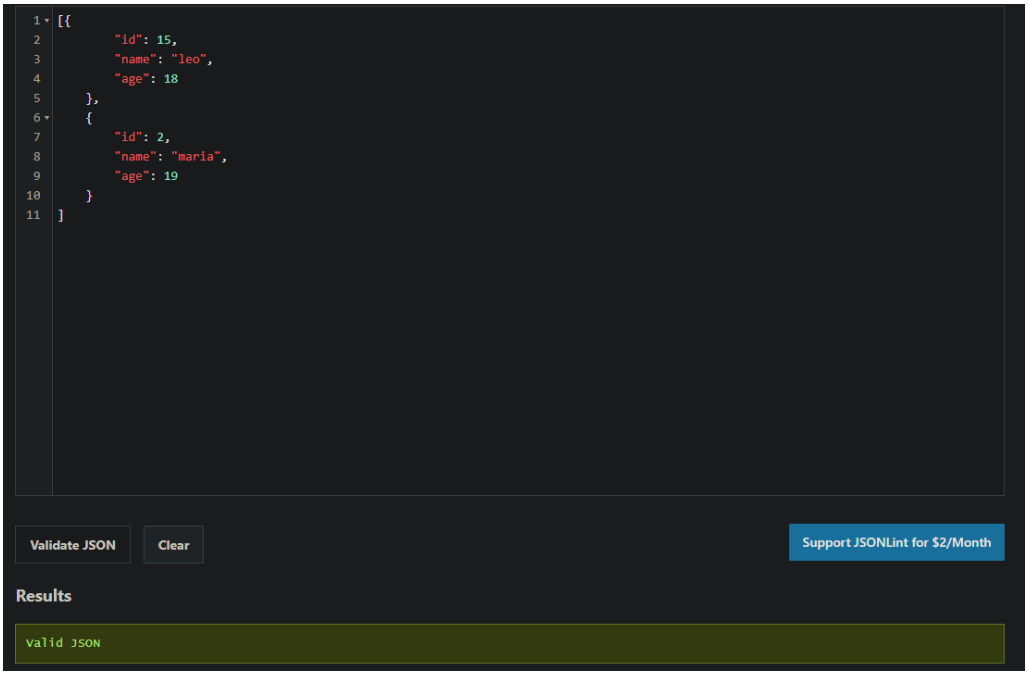




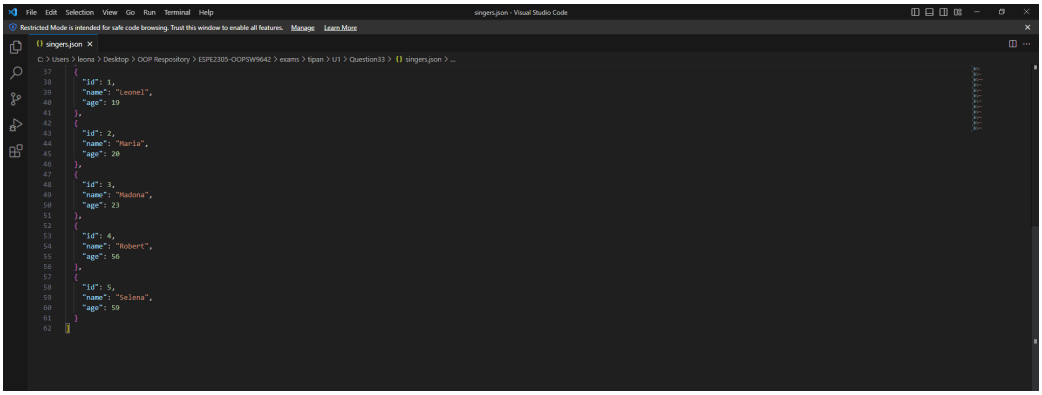
```
Enter the number of singer objects to add: 2
Enter details for singer #1
ID: 15
Name: leo
Age: 18
Singer added successfully.
---------------------------
Enter details for singer #2
ID: 2
Name: maria
Age: 19
Singer added successfully.
```

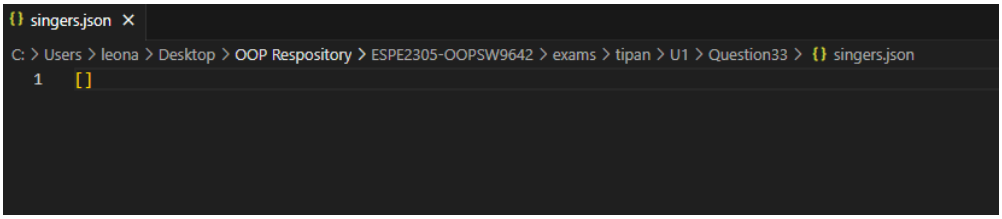
3) Printing/counting/deleting objects from JSON File (according to the roster number)

Json file is generated correctly: 10/10



4) Saving JSON data (screenshot of the json file)
It updates correctly, but deletes everything, and it was only to delete one by one.
7/10.





5) Code Quality (lines of code with stinky code)
Unnecessary comments.
9/10.

```
public Singer(int id, String name, int age) {
    this.id = id;
    this.name = name;
    this.age = age;
}

// Getters and Setters

public int getId() {
    return id;
}
```

Rubric:

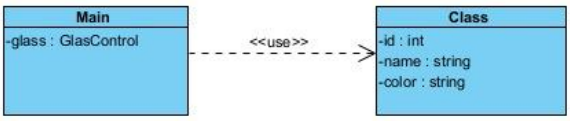
1) class modeling 0/10 pts.
2) Running program 10/10 pts.
3) Printing/counting/deleting objects from JSON File 10/10 pts.
4) Saving JSON data 10/10pts.
5) Code Quality 9/10 pts.
TOTAL: **36/50 pts.**

22. EDISON DAMIAN VERDESOTO SEGOVIA 26 /50 (glasses, print)
Inspector: Leonel Tipan
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) 7/10



Class name: 3/4
Model class: 2/4
Dependency: 2/2

2) Running program (cmd line executing the program) 5/10

```
run:
UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE
Subject: Oriented Objects Programming
NRC: 9642
INSTRUCTOR: Edison Lascano
Practical Exam.

 MENU
GLASSES INPUT
1. Add a glasses
2. Print the list
3. Exit
Choose an option
```

Showing Menu: 5/5
No json print: 0/5

3) Printing/counting/deleting objects from JSON File (according to the roster number) 0/10
No json file: 0/10
4) Saving JSON data (screenshot of the json file) 3/10

```
Scanner sc = new Scanner(source:System.in);
File file = new File (pathname: "./dataBase.json");
```

No Json file created, but exists in the program: 3/10
5) Code Quality (lines of code with stinky code) 10/10