2021-2022 academic year

THE DOCUMENTATION OF THE PROJECT FUNDAMENTAL PROGRAMMING II

-Creators:

Fuad Garibli: 50%;

Nemat Rahimov: 50%;

SECTION1: general information about the project;

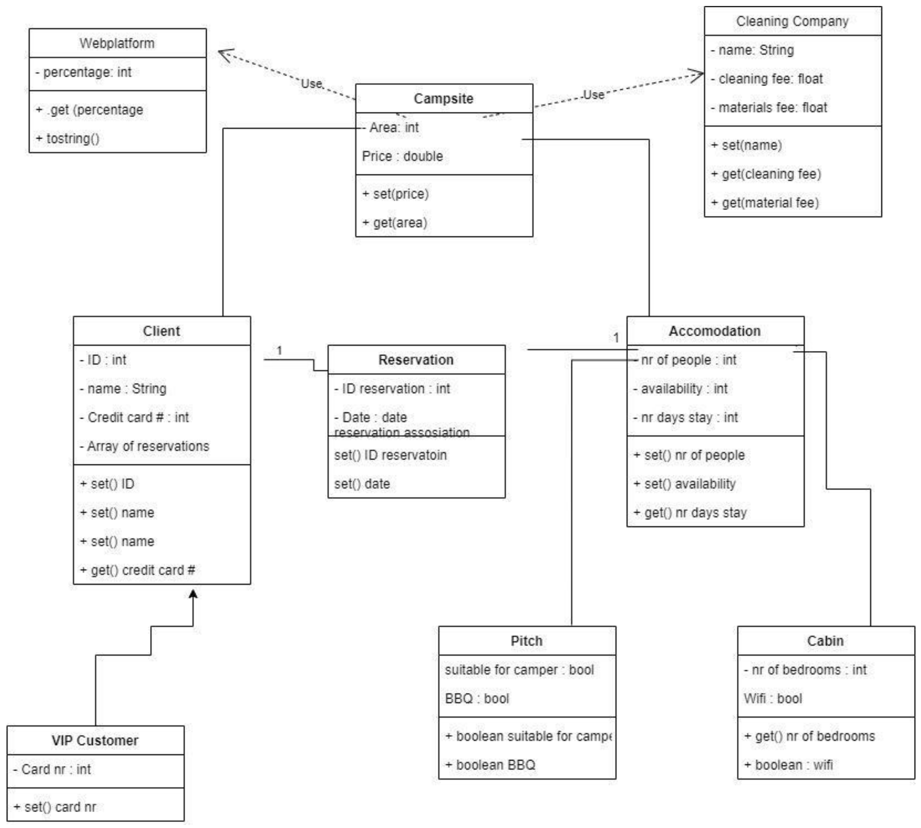
-The project demands us to automate the information system for a tourism complex or campsite. The main functionality that it must support is that of managing reservations for the accommodation available at a tourism complex or *campsite*. Document contains 11 detailed paragraphs, which are the requirements of the application.

The application that is going to be built must be able to support the reservation system, customer registration or login, manage informations about the objects (cabins and pitches), estimated cost of cleaning, web-system support, estimate discount procedures, VIP customer support and customer operations.

Java 17 is used as a main programming tool for the application, while the UML diagram is done with ***Visual Paradigm software***

SECTION2: UML diagrams for classes;

The following UML diagram is used for the creation of application, it was a tool to help to note the classes, associations between them and methods&attributes:

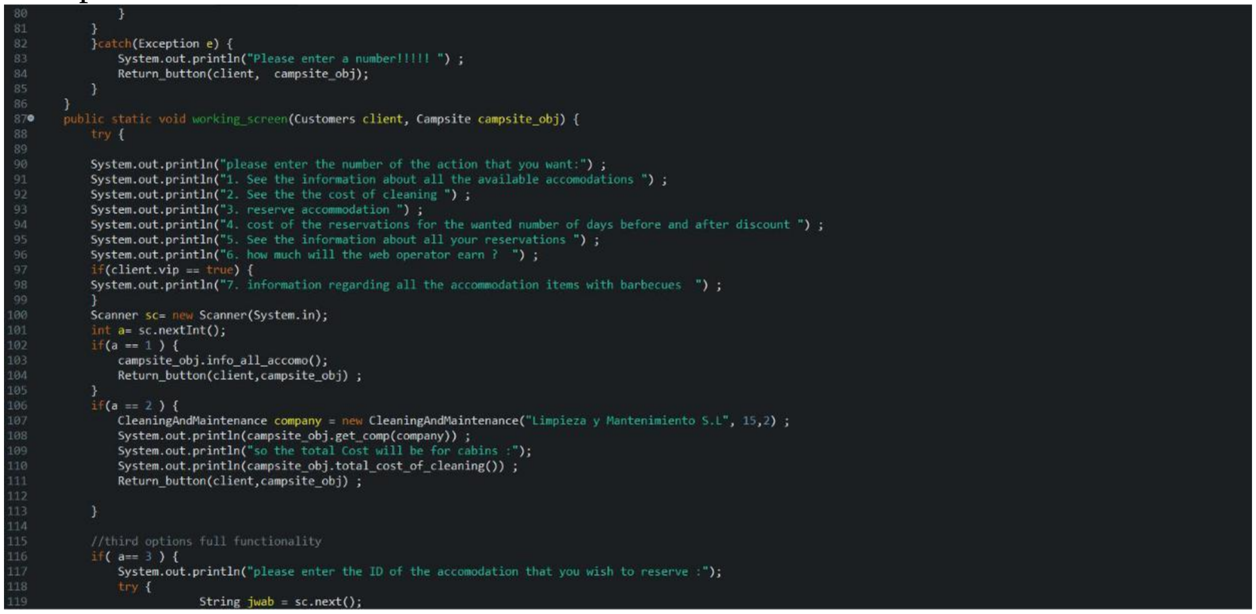


SECTION3: Code design;

While implementing the diagrams, we have used some stuff in order the code to work perfectly. Some data structures and object oriented programming components like exceptions, dependency, association and inheritance. Interface and etc are used to design the code. The code contains 10 classes, including main;

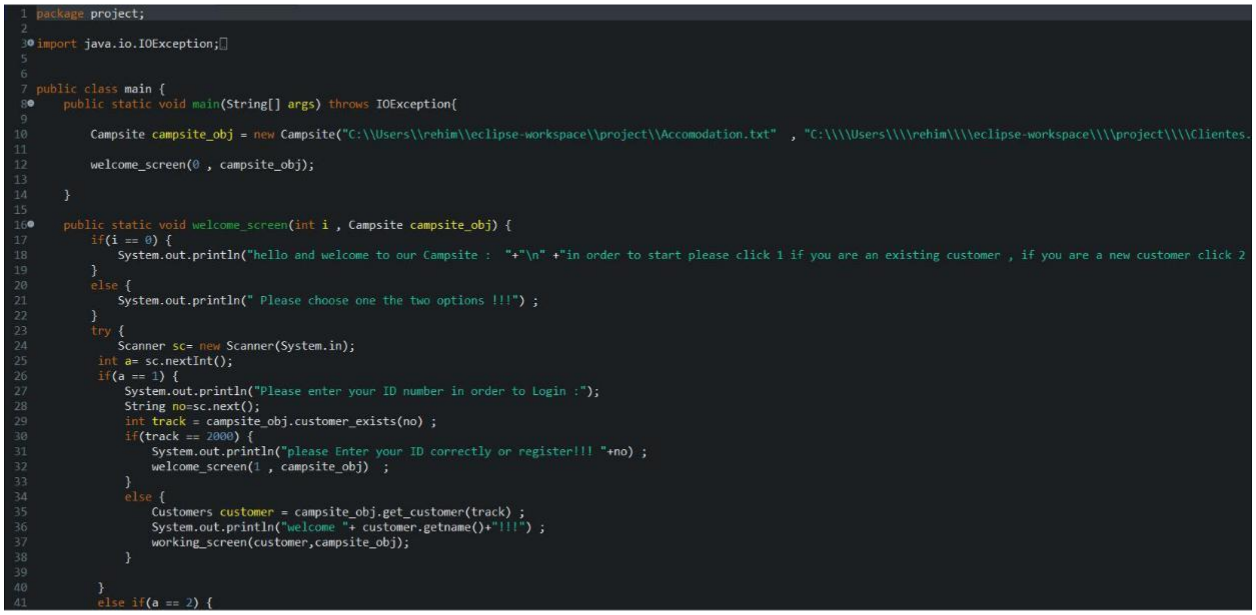
a) Main class:

Main class contains menu part and exceptions for the code. In the first lines of the class, we can see there is a ".txt" file reading command line. Exceptions are defined here.



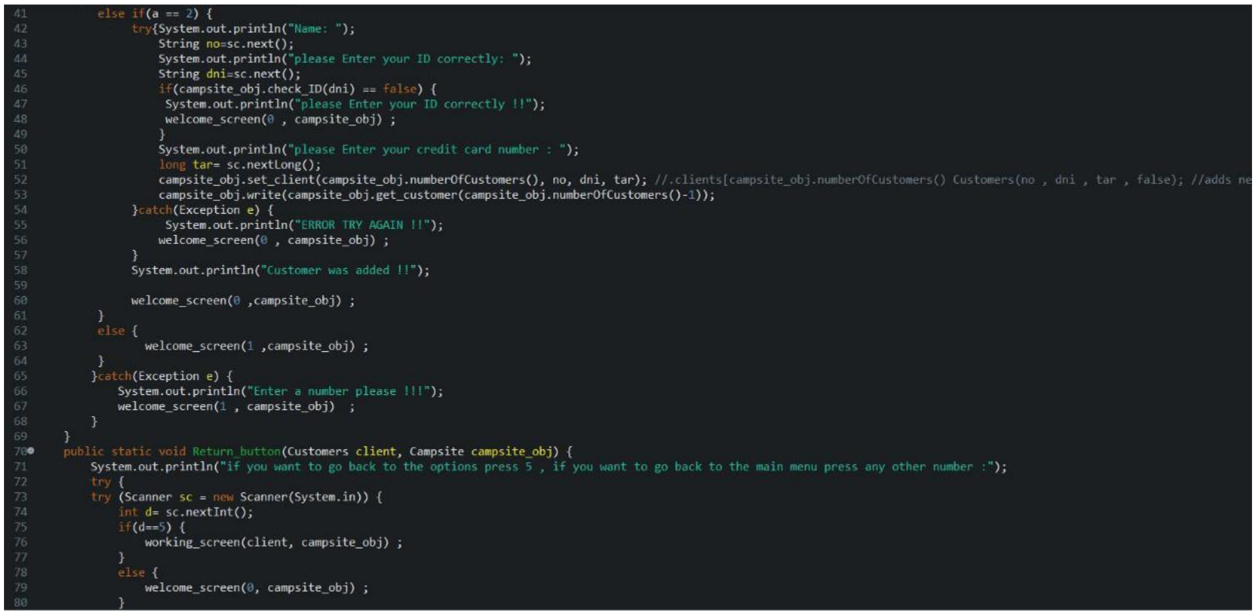
```java
            }
        }
        }catch(Exception e) {
            System.out.println("Please enter a number!!!!! ") ;
            Return_button(client,  campsite_obj);
        }
    }
    public static void working_screen(Customers client, Campsite campsite_obj) {
        try {

        System.out.println("please enter the number of the action that you want:") ;
        System.out.println("1. See the information about all the available accomodations ") ;
        System.out.println("2. See the the cost of cleaning ") ;
        System.out.println("3. reserve accommodation ") ;
        System.out.println("4. cost of the reservations for the wanted number of days before and after discount ") ;
        System.out.println("5. See the information about all your reservations ") ;
        System.out.println("6. how much will the web operator earn ?  ") ;
        if(client.vip == true) {
        System.out.println("7. information regarding all the accommodation items with barbecues  ") ;
        }
        Scanner sc= new Scanner(System.in);
        int a= sc.nextInt();
        if(a == 1 ) {
            campsite_obj.info_all_accomo();
            Return_button(client,campsite_obj) ;
        }
        if(a == 2 ) {
            CleaningAndMaintenance company = new CleaningAndMaintenance("Limpieza y Mantenimiento S.L", 15,2) ;
            System.out.println(campsite_obj.get_comp(company)) ;
            System.out.println("so the total Cost will be for cabins :");
            System.out.println(campsite_obj.total_cost_of_cleaning()) ;
            Return_button(client,campsite_obj) ;

        }

        //third options full functionality
        if( a== 3 ) {
            System.out.println("please enter the ID of the accomodation that you wish to reserve :");
            try {
                        String jwab = sc.next();
```

The screenshot of menu option codding



```java
package project;

import java.io.IOException;


public class main {
    public static void main(String[] args) throws IOException{

        Campsite campsite_obj = new Campsite("C:\\Users\\rehim\\eclipse-workspace\\project\\Accomodation.txt"  , "C:\\\\Users\\\\rehim\\\\eclipse-workspace\\\\project\\\\Clientes.

        welcome_screen(0 , campsite_obj);

    }

    public static void welcome_screen(int i , Campsite campsite_obj) {
        if(i == 0) {
            System.out.println("hello and welcome to our Campsite :  "+"\n" +"in order to start please click 1 if you are an existing customer , if you are a new customer click 2
        }
        else {
            System.out.println(" Please choose one the two options !!!") ;
        }
        try {
            Scanner sc= new Scanner(System.in);
         int a= sc.nextInt();
         if(a == 1) {
             System.out.println("Please enter your ID number in order to Login :");
             String no=sc.next();
             int track = campsite_obj.customer_exists(no) ;
             if(track == 2000) {
                 System.out.println("please Enter your ID correctly or register!!! "+no) ;
                 welcome_screen(1 , campsite_obj)  ;
             }
             else {
                 Customers customer = campsite_obj.get_customer(track) ;
                 System.out.println("welcome "+ customer.getname()+"!!!") ;
                 working_screen(customer,campsite_obj);
             }

         }
         else if(a == 2) {
```

Some exception examples from the code

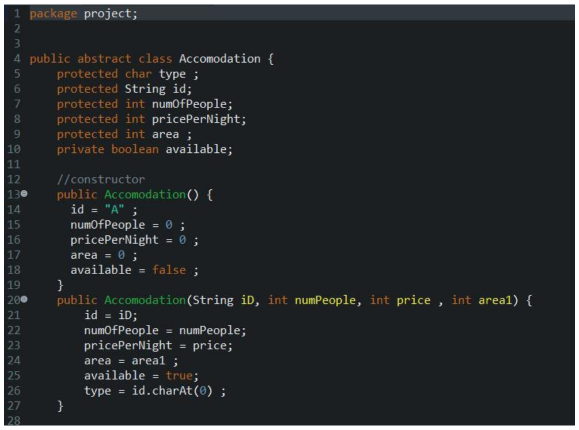


```java
        else if(a == 2) {
            try{System.out.println("Name: ");
                String no=sc.next();
                System.out.println("please Enter your ID correctly: ");
                String dni=sc.next();
                if(campsite_obj.check_ID(dni) == false) {
                 System.out.println("please Enter your ID correctly !!");
                 welcome_screen(0 , campsite_obj) ;
                }
                System.out.println("please Enter your credit card number : ");
                long tar= sc.nextLong();
                campsite_obj.set_client(campsite_obj.numberOfCustomers(), no, dni, tar); //.clients[campsite_obj.numberOfCustomers() Customers(no , dni , tar , false); //adds ne
                campsite_obj.write(campsite_obj.get_customer(campsite_obj.numberOfCustomers()-1));
            }catch(Exception e) {
                 System.out.println("ERROR TRY AGAIN !!");
                welcome_screen(0 , campsite_obj) ;
            }
            System.out.println("Customer was added !!");

            welcome_screen(0 ,campsite_obj) ;
        }
        else {
              welcome_screen(1 ,campsite_obj) ;
        }
       }catch(Exception e) {
           System.out.println("Enter a number please !!!");
           welcome_screen(1 , campsite_obj)  ;
       }
    }
    public static void Return_button(Customers client, Campsite campsite_obj) {
        System.out.println("if you want to go back to the options press 5 , if you want to go back to the main menu press any other number :");
        try {
        try (Scanner sc = new Scanner(System.in)) {
            int d= sc.nextInt();
            if(d==5) {
                working_screen(client, campsite_obj) ;
            }
            else {
                welcome_screen(0, campsite_obj) ;
            }
```

Some exception examples (2)

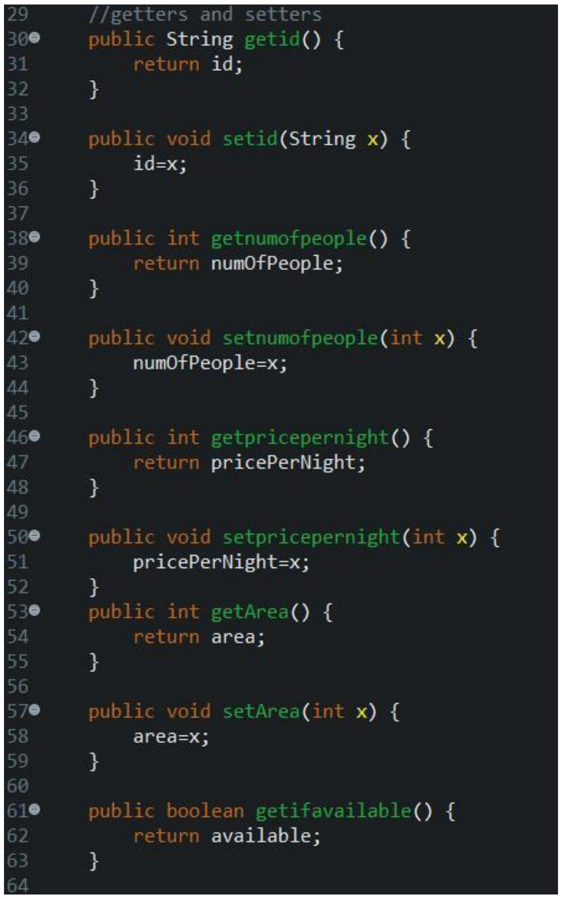
B)Acommodation class:

Acommodation class contains 5 protected and 1 private attributes, one constructor method, one toString() method, and getters&setters. This class is inherited by 2 other classes: Pitches and Cabins. The screenshot of the code is provided bellow:



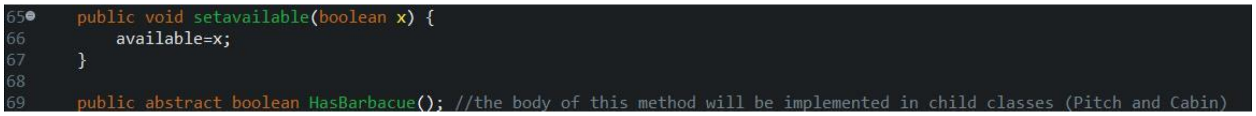
```java
package project;


public abstract class Accomodation {
    protected char type ;
    protected String id;
    protected int numOfPeople;
    protected int pricePerNight;
    protected int area ;
    private boolean available;

    //constructor
    public Accomodation() {
      id = "A" ;
      numOfPeople = 0 ;
      pricePerNight = 0 ;
      area = 0 ;
      available = false ;
    }
    public Accomodation(String iD, int numPeople, int price , int area1) {
        id = iD;
        numOfPeople = numPeople;
        pricePerNight = price;
        area = area1 ;
        available = true;
        type = id.charAt(0) ;
    }
```

Screenshoot of constructor and attributes;



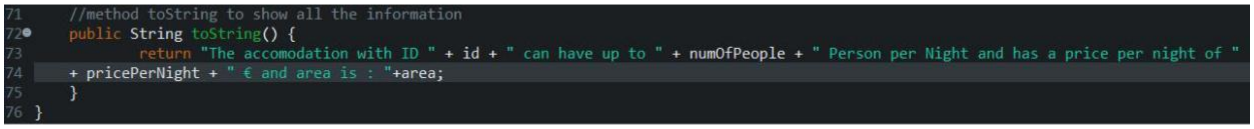
```java
    //getters and setters
    public String getid() {
        return id;
    }

    public void setid(String x) {
        id=x;
    }

    public int getnumofpeople() {
        return numOfPeople;
    }

    public void setnumofpeople(int x) {
        numOfPeople=x;
    }

    public int getpricepernight() {
        return pricePerNight;
    }

    public void setpricepernight(int x) {
        pricePerNight=x;
    }
    public int getArea() {
        return area;
    }

    public void setArea(int x) {
        area=x;
    }

    public boolean getifavailable() {
        return available;
    }

```

Setters and getters



```java
    public void setavailable(boolean x) {
        available=x;
    }

    public abstract boolean HasBarbacue(); //the body of this method will be implemented in child classes (Pitch and Cabin)
```

Setters and getters (2

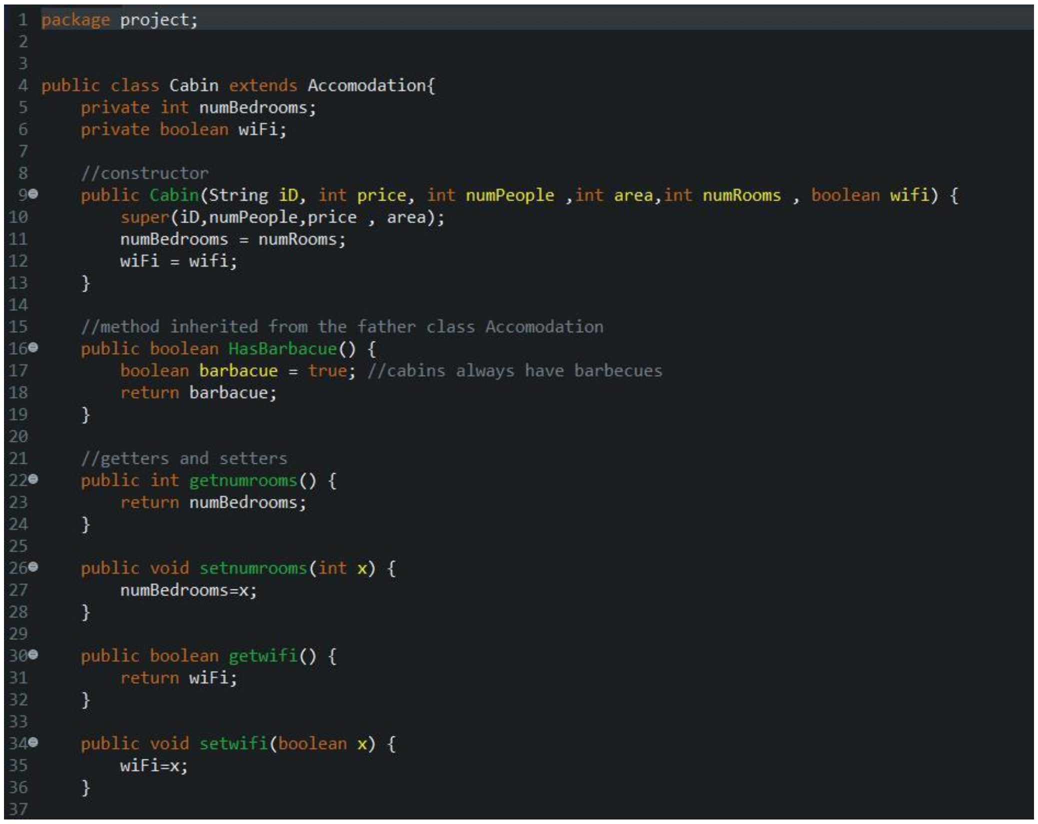


```java
    //method toString to show all the information
    public String toString() {
            return "The accomodation with ID " + id + " can have up to " + numOfPeople + " Person per Night and has a price per night of "
    + pricePerNight + " € and area is : "+area;
    }
}
```

toString() metho

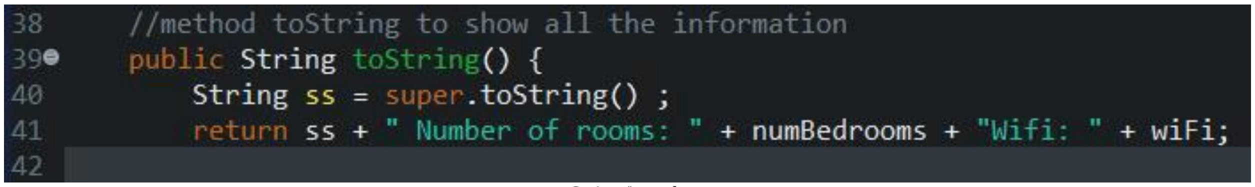
c)Cabin class:

the class inherits from Accommodation class, and has 2 private attributes, 1 constructor, 1 inherited method, 1 toString method and getters&setters. Availability of each cabin, and their personal data is read from the .txt document provided in moodle.



```java
package project;


public class Cabin extends Accomodation{
    private int numBedrooms;
    private boolean wiFi;

    //constructor
    public Cabin(String iD, int price, int numPeople ,int area,int numRooms , boolean wifi) {
        super(iD,numPeople,price , area);
        numBedrooms = numRooms;
        wiFi = wifi;
    }

    //method inherited from the father class Accomodation
    public boolean HasBarbacue() {
        boolean barbacue = true; //cabins always have barbecues
        return barbacue;
    }

    //getters and setters
    public int getnumrooms() {
        return numBedrooms;
    }

    public void setnumrooms(int x) {
        numBedrooms=x;
    }

    public boolean getwifi() {
        return wiFi;
    }

    public void setwifi(boolean x) {
        wiFi=x;
    }

```

Constructor, getters and setters



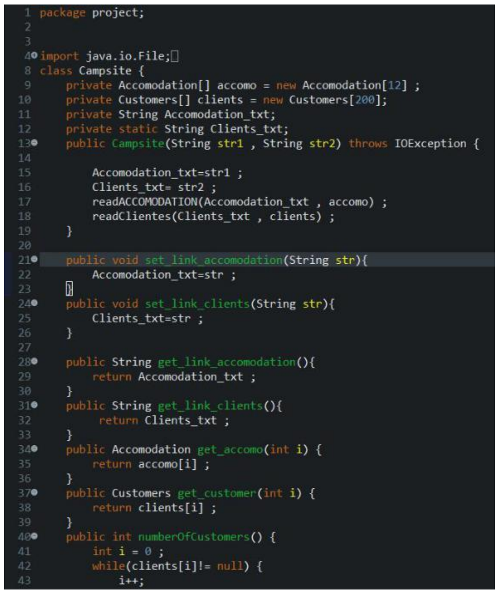
```java
    //method toString to show all the information
    public String toString() {
        String ss = super.toString() ;
        return ss + " Number of rooms: " + numBedrooms + "Wifi: " + wiFi;

```

toString() meth

d)campsite class:

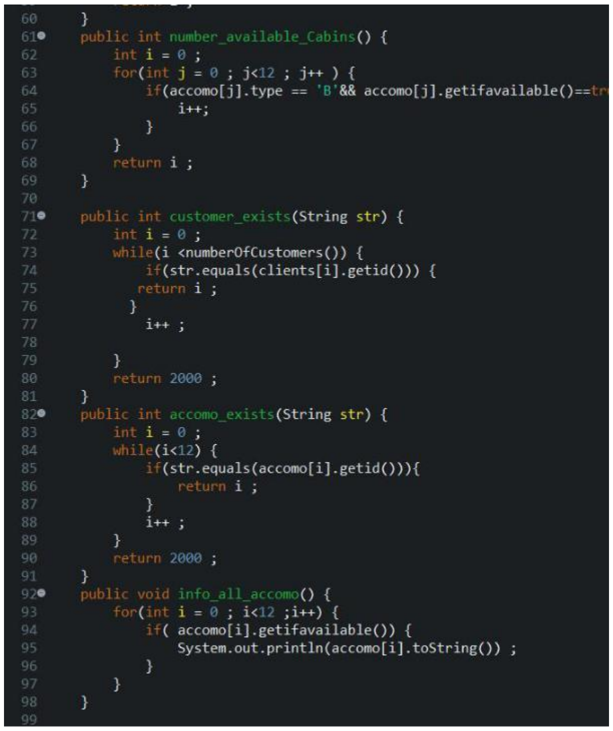
this class uses association with 2 other classes: accommodation and customers. There are 3 private, 1 private static attributes, campsite method that assigns the values of variables to the created objects, set and getters for the attributes.
explanation of methods:

- numberofCustomers(): this function is used to calculate the number of people that have reserved an accommodation in campsite;
-
- `number_available_pitchs()`: this function is used to calculate the number of available pitches;
- number_available_cabins(): this function returns the value of available cabins;
- customer_exists(String str): this function returns the value of the reserved customers for campsite;
- total_cost_of_cleaning(): this function is used to calculate the total cost of cleaning according to the entered informations;
- total_cost_web(Customers c): this function returns the value of total money that has been used for web services and indicate how much the web operator will earn;
- check_ID(String id): checks if the entered ID has any reservations or not;



```java
package project;


import java.io.File;
class Campsite {
    private Accomodation[] accomo = new Accomodation[12] ;
    private Customers[] clients = new Customers[200];
    private String Accomodation_txt;
    private static String Clients_txt;
    public Campsite(String str1 , String str2) throws IOException {

        Accomodation_txt=str1 ;
        Clients_txt= str2 ;
        readACCOMODATION(Accomodation_txt , accomo) ;
        readClientes(Clients_txt , clients) ;
    }

    public void set_link_accomodation(String str){
        Accomodation_txt=str ;
    }
    public void set_link_clients(String str){
        Clients_txt=str ;
    }

    public String get_link_accomodation(){
        return Accomodation_txt ;
    }
    public String get_link_clients(){
         return Clients_txt ;
    }
    public Accomodation get_accomo(int i) {
        return accomo[i] ;
    }
    public Customers get_customer(int i) {
        return clients[i] ;
    }
    public int numberOfCustomers() {
        int i = 0 ;
        while(clients[i]!= null) {
            i++;
```

Attributes and setters&getters

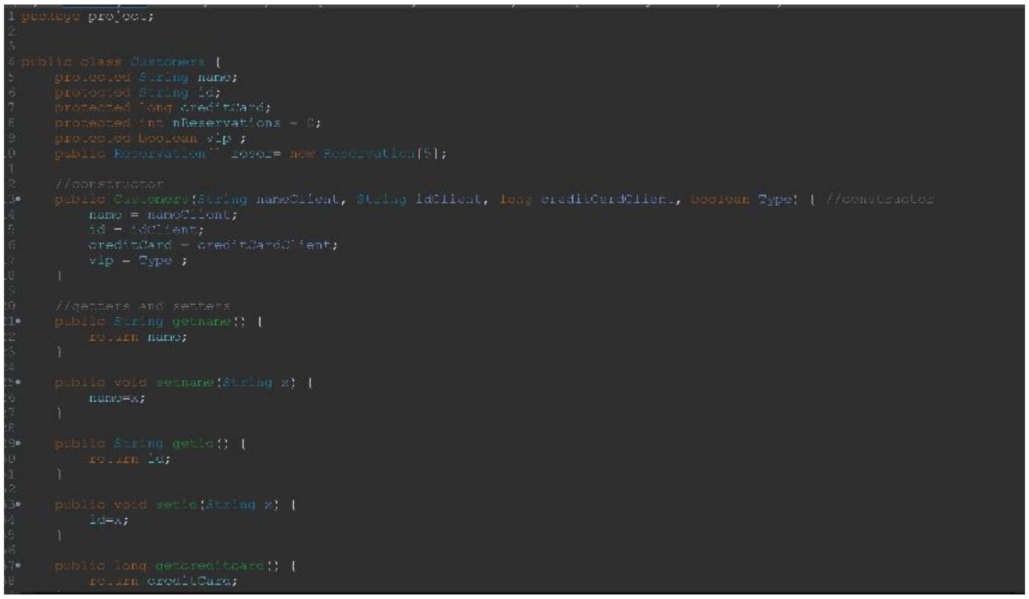


```java
}
public int number_available_Cabins() {
    int i = 0 ;
    for(int j = 0 ; j<12 ; j++ ) {
        if(accomo[j].type == 'B'&& accomo[j].getifavailable()==tr
            i++;
        }
    }
    return i ;
}

public int customer_exists(String str) {
    int i = 0 ;
    while(i <numberOfCustomers()) {
        if(str.equals(clients[i].getid())) {
       return i ;
      }
        i++ ;

    }
    return 2000 ;
}
public int accomo_exists(String str) {
    int i = 0 ;
    while(i<12) {
        if(str.equals(accomo[i].getid())){
            return i ;
        }
        i++ ;
    }
    return 2000 ;
}
public void info_all_accomo() {
    for(int i = 0 ; i<12 ;i++) {
        if( accomo[i].getifavailable()) {
            System.out.println(accomo[i].toString()) ;
        }
    }
}
```

Some functions

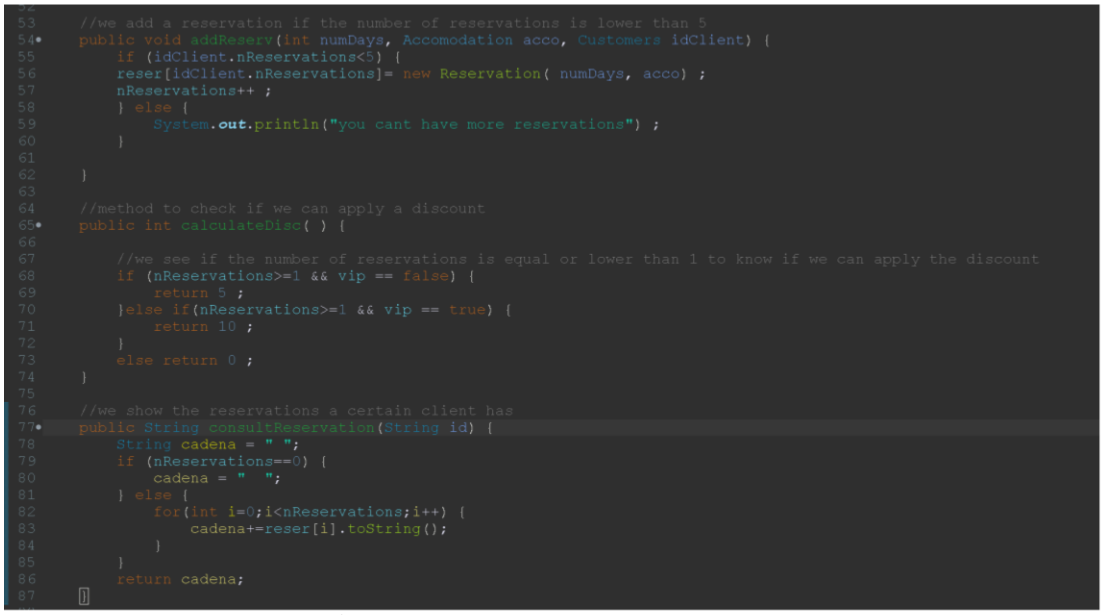
e) Customers class

This class is in an association relationship with Reservation class and Campsite class.Customers class has 5 protected and 1 public attributes.And dependency relationship with VIP customer class.It has constructor method,get and set methods,method for adding a new reservation,method for counting discount,methods for writing informations about reservation and barbecue and toString method.



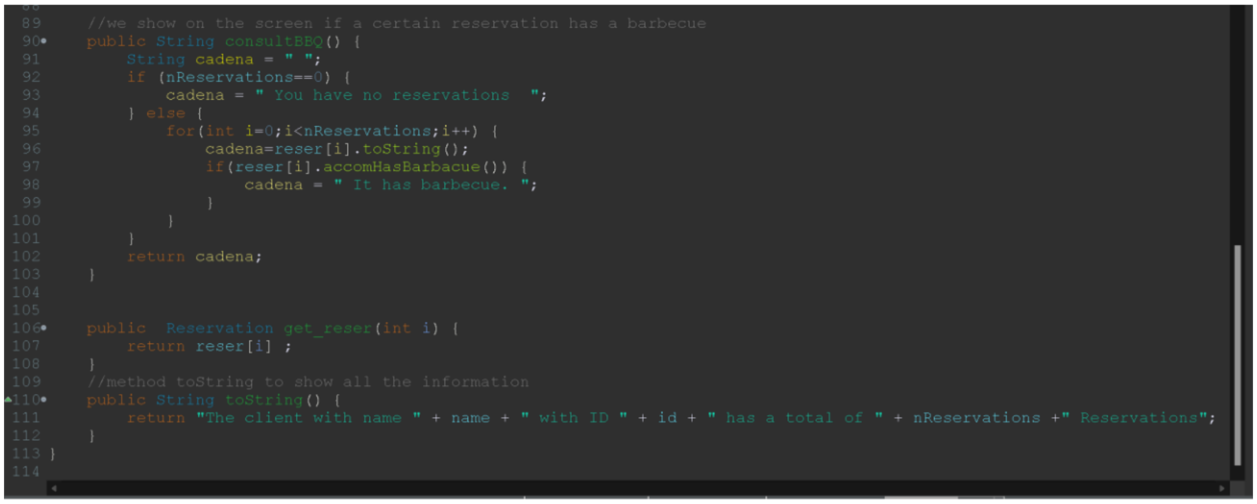
```java
package project;


public class Customers {
    protected String name;
    protected String id;
    protected long creditCard;
    protected int nReservations = 0;
    protected boolean vip ;
    public Reservation[] reser= new Reservation[5];

    //constructor
    public Customers(String nameClient, String idClient, long creditCardClient, boolean Type) { //constructor
        name = nameClient;
        id = idClient;
        creditCard = creditCardClient;
        vip = Type ;
    }

    //getters and setters
    public String getname() {
        return name;
    }

    public void setname(String x) {
        name=x;
    }

    public String getid() {
        return id;
    }

    public void setid(String x) {
        id=x;
    }

    public long getcreditcard() {
        return creditCard;
```

Attributes, getters and setters



```java
    //we add a reservation if the number of reservations is lower than 5
    public void addReserv(int numDays, Accomodation acco, Customers idClient) {
        if (idClient.nReservations<5) {
        reser[idClient.nReservations]= new Reservation( numDays, acco) ;
        nReservations++ ;
        } else {
            System.out.println("you cant have more reservations") ;
        }

    }

    //method to check if we can apply a discount
    public int calculateDisc( ) {

        //we see if the number of reservations is equal or lower than 1 to know if we can apply the discount
        if (nReservations>=1 && vip == false) {
            return 5 ;
        }else if(nReservations>=1 && vip == true) {
            return 10 ;
        }
        else return 0 ;
    }

    //we show the reservations a certain client has
    public String consultReservation(String id) {
        String cadena = " ";
        if (nReservations==0) {
            cadena = "  ";
        } else {
            for(int i=0;i<nReservations;i++) {
                cadena+=reser[i].toString();
            }
        }
        return cadena;
    }
```

method for adding reservation and discount applying

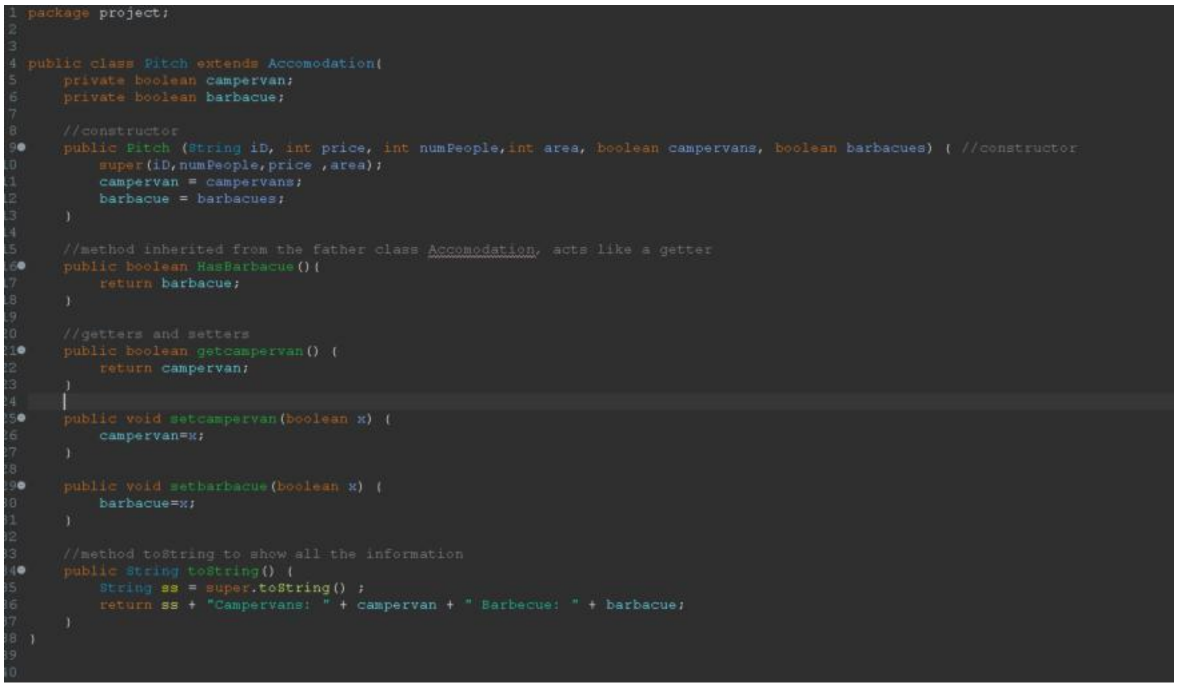


```java
    //we show on the screen if a certain reservation has a barbecue
    public String consultBBQ() {
        String cadena = " ";
        if (nReservations==0) {
            cadena = " You have no reservations  ";
        } else {
            for(int i=0;i<nReservations;i++) {
                cadena=reser[i].toString();
                if(reser[i].accomHasBarbacue()) {
                    cadena = " It has barbecue. ";
                }
            }
        }
        return cadena;
    }


    public  Reservation get_reser(int i) {
        return reser[i] ;
    }
    //method toString to show all the information
    public String toString() {
        return "The client with name " + name + " with ID " + id + " has a total of " + nReservations +" Reservations";
    }
}
```

Additional methods

f) Pitch class

Pitch class also inherits class Accommodation. It has 2 private attributes, constructor method, setters and getters methods, inherited method and toString() method.

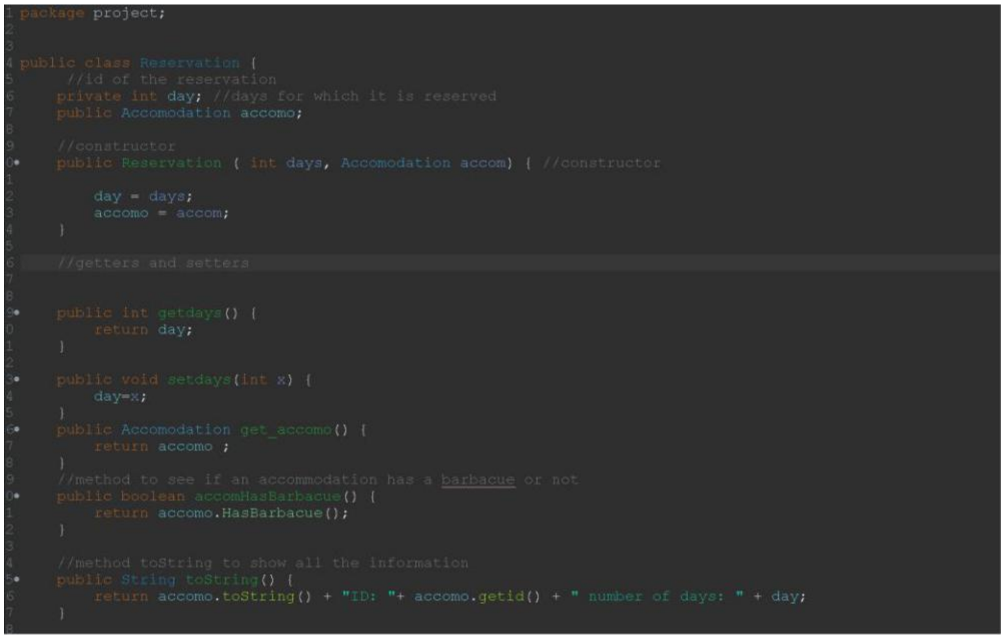


```java
package project;


public class Pitch extends Accomodation{
    private boolean campervan;
    private boolean barbacue;

    //constructor
    public Pitch (String iD, int price, int numPeople,int area, boolean campervans, boolean barbacues) { //constructor
        super(iD,numPeople,price ,area);
        campervan = campervans;
        barbacue = barbacues;
    }

    //method inherited from the father class Accomodation, acts like a getter
    public boolean HasBarbacue(){
        return barbacue;
    }

    //getters and setters
    public boolean getcampervan() {
        return campervan;
    }

    public void setcampervan(boolean x) {
        campervan=x;
    }

    public void setbarbacue(boolean x) {
        barbacue=x;
    }

    //method toString to show all the information
    public String toString() {
        String ss = super.toString() ;
        return ss + "Campervans: " + campervan + " Barbecue: " + barbacue;
    }
}
```

Pitch class codes

g) Reservation class

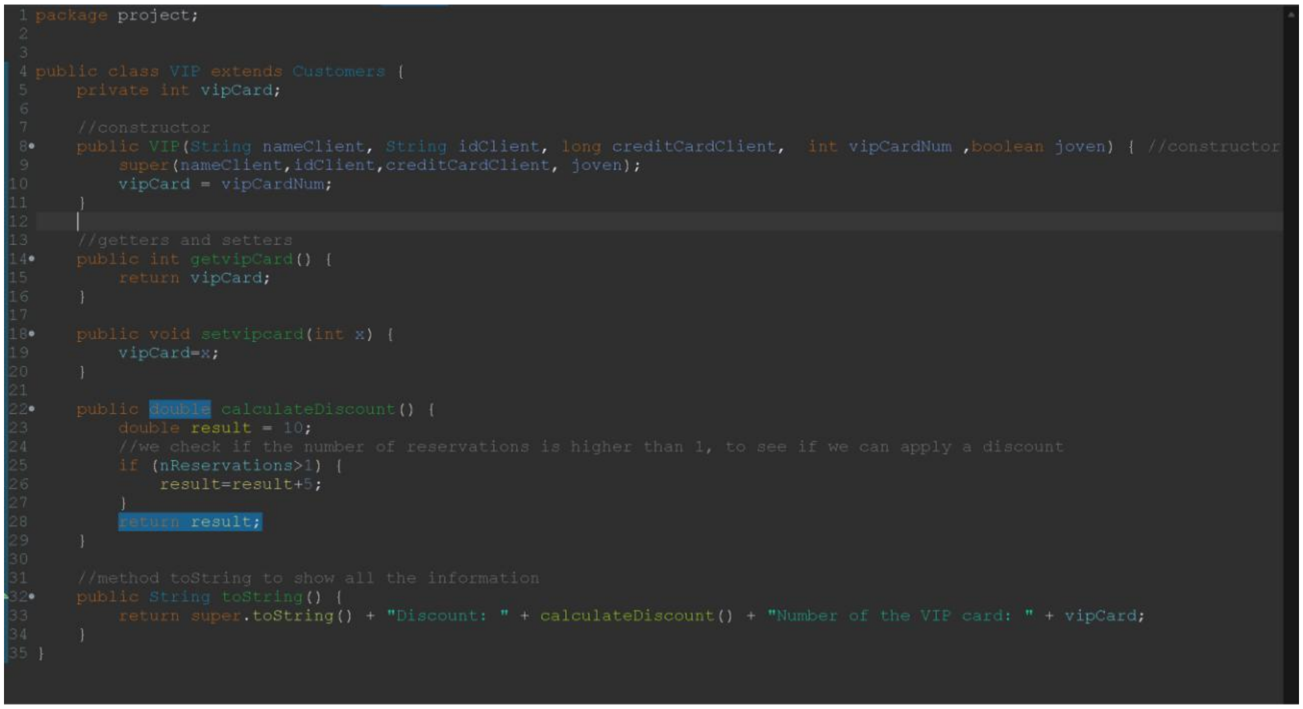
This has association relationship with classes Accommodation and Customer.
It has 1 private,1 public attributes,constructor method,get and set methods,inherited method and toString method.



```java
package project;


public class Reservation {
     //id of the reservation
    private int day; //days for which it is reserved
    public Accomodation accomo;

    //constructor
    public Reservation ( int days, Accomodation accom) { //constructor

        day = days;
        accomo = accom;
    }

    //getters and setters


    public int getdays() {
        return day;
    }

    public void setdays(int x) {
        day=x;
    }
    public Accomodation get_accomo() {
        return accomo ;
    }
    //method to see if an accommodation has a barbacue or not
    public boolean accomHasBarbacue() {
        return accomo.HasBarbacue();
    }

    //method toString to show all the information
    public String toString() {
        return accomo.toString() + "ID: "+ accomo.getid() + " number of days: " + day;
    }
```

h) VIP customer class

This class is in dependency relationship with customers class.It has 1 private attribute.It has constructor method,set and get mothods,inherited method from customers class and toString method.

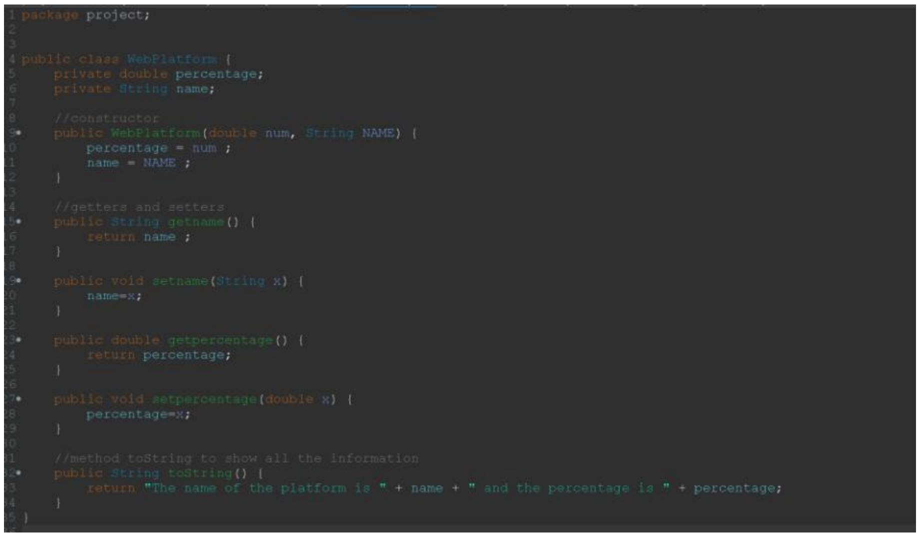


```java
package project;


public class VIP extends Customers {
    private int vipCard;

    //constructor
    public VIP(String nameClient, String idClient, long creditCardClient,  int vipCardNum ,boolean joven) { //constructor
        super(nameClient,idClient,creditCardClient, joven);
        vipCard = vipCardNum;
    }

    //getters and setters
    public int getvipCard() {
        return vipCard;
    }

    public void setvipcard(int x) {
        vipCard=x;
    }

    public double calculateDiscount() {
        double result = 10;
        //we check if the number of reservations is higher than 1, to see if we can apply a discount
        if (nReservations>1) {
            result=result+5;
        }
        return result;
    }

    //method toString to show all the information
    public String toString() {
        return super.toString() + "Discount: " + calculateDiscount() + "Number of the VIP card: " + vipCard;
    }
}
```

VIP class codes

I) WebPlatform class

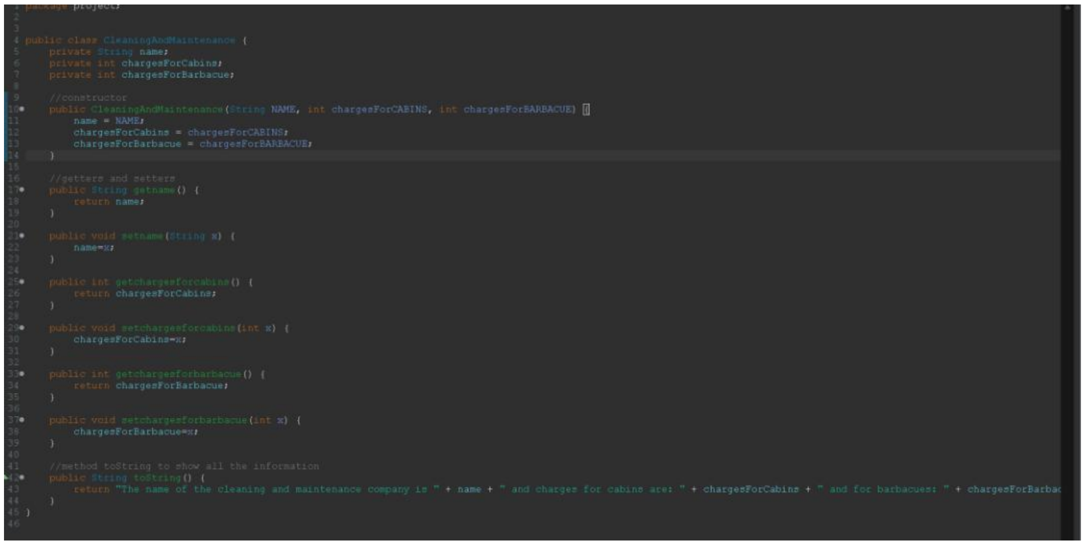
This class is in dependency relationship with Campsite class.It has 2 private attributes,constructor method,get and set methods and toString method



```java
package project;


public class WebPlatform {
    private double percentage;
    private String name;

    //constructor
    public WebPlatform(double num, String NAME) {
        percentage = num ;
        name = NAME ;
    }

    //getters and setters
    public String getname() {
        return name ;
    }

    public void setname(String x) {
        name=x;
    }

    public double getpercentage() {
        return percentage;
    }

    public void setpercentage(double x) {
        percentage=x;
    }

    //method toString to show all the information
    public String toString() {
        return "The name of the platform is " + name + " and the percentage is " + percentage;
    }
}
```

j) Cleaning company class

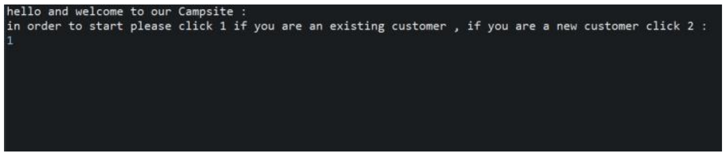
This class is in a dependency relationship with Campsite class. It has 3 private attributes,constructor method,get and set methods and toString method.



```java
package project;


public class CleaningAndMaintenance {
    private String name;
    private int chargesForCabins;
    private int chargesForBarbacue;

    //constructor
    public CleaningAndMaintenance(String NAME, int chargesForCABINS, int chargesForBARBACUE) {
        name = NAME;
        chargesForCabins = chargesForCABINS;
        chargesForBarbacue = chargesForBARBACUE;
    }

    //getters and setters
    public String getname() {
        return name;
    }

    public void setname(String x) {
        name=x;
    }

    public int getchargesforcabins() {
        return chargesForCabins;
    }

    public void setchargesforcabins(int x) {
        chargesForCabins=x;
    }

    public int getchargesforbarbacue() {
        return chargesForBarbacue;
    }

    public void setchargesforbarbacue(int x) {
        chargesForBarbacue=x;
    }

    //method toString to show all the information
    public String toString() {
        return "The name of the cleaning and maintenance company is " + name + " and charges for cabins are: " + chargesForCabins + " and for barbacues: " + chargesForBarbac
    }
}

```

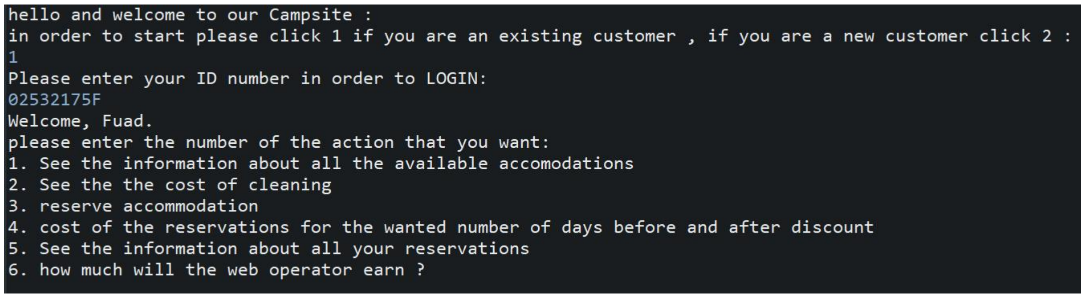
<u>SECTION4</u>: user manual:

On the beginning when the code will start to run, the user will meet a login screen with 2 choosable options, 1 for registered customers (available in customers.txt file) or new customers, who needs to register. When a new customer registers, his/her informations are stored in customers.txt document:



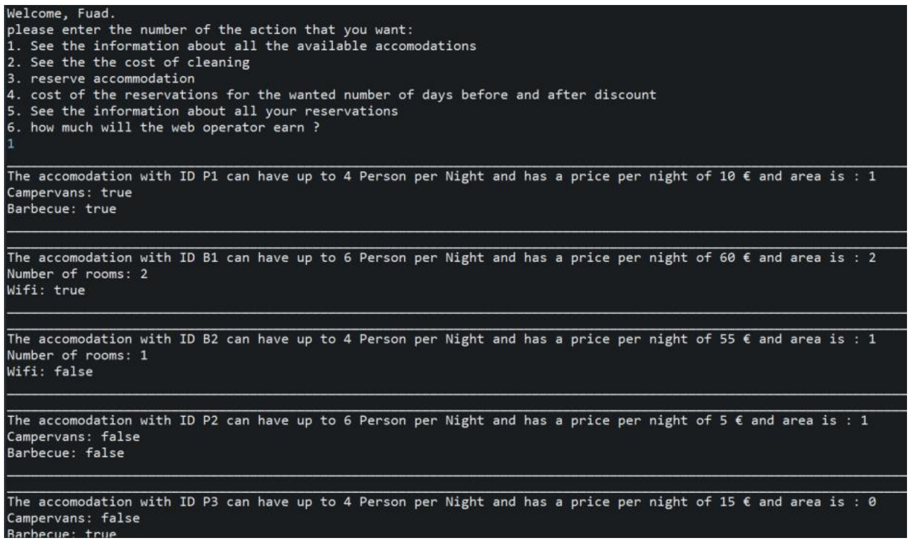
```
hello and welcome to our Campsite :
in order to start please click 1 if you are an existing customer , if you are a new customer click 2 :
1
```

After clicking the appropriate button and logging in, a menu screen will open with 6 choices and the user's name on the top:



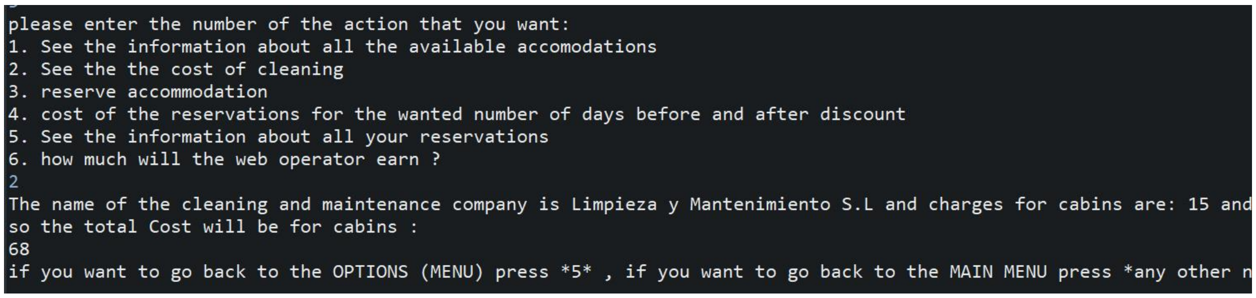
```
hello and welcome to our Campsite :
in order to start please click 1 if you are an existing customer , if you are a new customer click 2 :
1
Please enter your ID number in order to LOGIN:
02532175F
Welcome, Fuad.
please enter the number of the action that you want:
1. See the information about all the available accomodations
2. See the the cost of cleaning
3. reserve accommodation
4. cost of the reservations for the wanted number of days before and after discount
5. See the information about all your reservations
6. how much will the web operator earn ?
```

The first choice in the menu gives us the information about all the available accomodations. Users can check the availability of the most appropriate accommodation for them from here



```
Welcome, Fuad.
please enter the number of the action that you want:
1. See the information about all the available accomodations
2. See the the cost of cleaning
3. reserve accommodation
4. cost of the reservations for the wanted number of days before and after discount
5. See the information about all your reservations
6. how much will the web operator earn ?
1

The accomodation with ID P1 can have up to 4 Person per Night and has a price per night of 10 € and area is : 1
Campervans: true
Barbecue: true


The accomodation with ID B1 can have up to 6 Person per Night and has a price per night of 60 € and area is : 2
Number of rooms: 2
Wifi: true


The accomodation with ID B2 can have up to 4 Person per Night and has a price per night of 55 € and area is : 1
Number of rooms: 1
Wifi: false


The accomodation with ID P2 can have up to 6 Person per Night and has a price per night of 5 € and area is : 1
Campervans: false
Barbecue: false


The accomodation with ID P3 can have up to 4 Person per Night and has a price per night of 15 € and area is : 0
Campervans: false
Barbecue: true
```

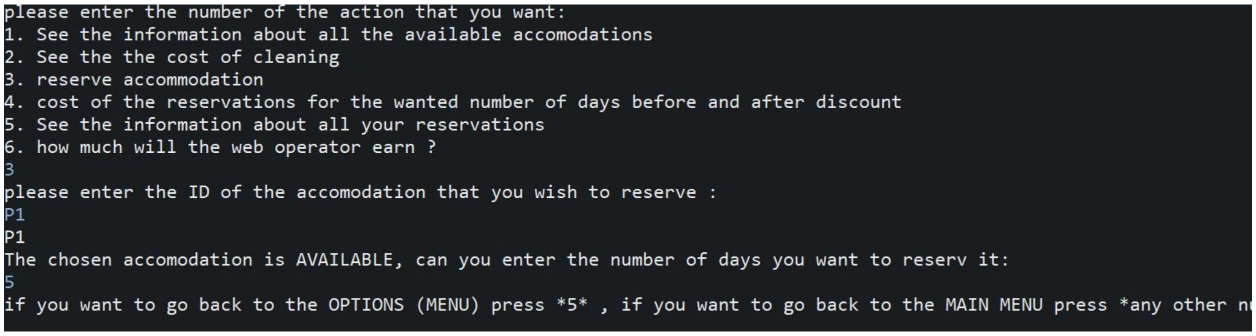
The second option in the menu corresponds to see the cost of cleaning for a reserved accommodation. If the user has not recerved any accommodation yet, it will display a “0”. If an accommodation was chosen, then it will output the cost of cleaning.
*For using this menu option, the client needs to reserve an accommodation first!



```
please enter the number of the action that you want:
1. See the information about all the available accomodations
2. See the the cost of cleaning
3. reserve accommodation
4. cost of the reservations for the wanted number of days before and after discount
5. See the information about all your reservations
6. how much will the web operator earn ?
2
The name of the cleaning and maintenance company is Limpieza y Mantenimiento S.L and charges for cabins are: 15 and
so the total Cost will be for cabins :
68
if you want to go back to the OPTIONS (MENU) press *5* , if you want to go back to the MAIN MENU press *any other n
```

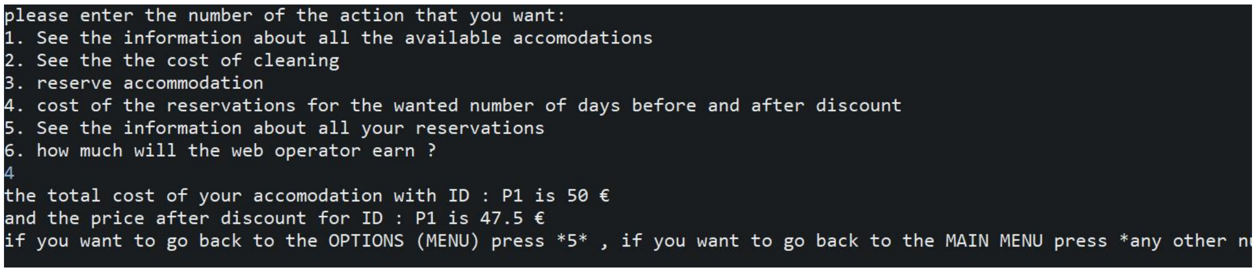
At the end of each output, you can see a sentence referring to numbers in order to return back to the login output or menu output.

The option 3 is for reserving an accommodation, the client should enter the ID of the accommodation, if it is available the code will provide him with the opportunity to introduce the number of days the client wants to reserve the accommodation for:



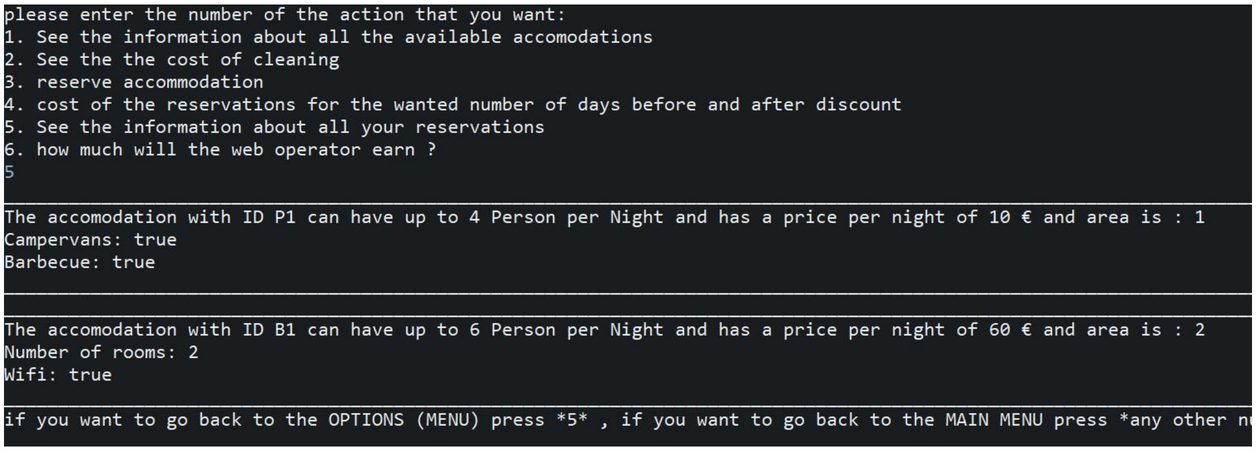
```
please enter the number of the action that you want:
1. See the information about all the available accomodations
2. See the the cost of cleaning
3. reserve accommodation
4. cost of the reservations for the wanted number of days before and after discount
5. See the information about all your reservations
6. how much will the web operator earn ?
3
please enter the ID of the accomodation that you wish to reserve :
P1
P1
The chosen accomodation is AVAILABLE, can you enter the number of days you want to reserv it:
5
if you want to go back to the OPTIONS (MENU) press *5* , if you want to go back to the MAIN MENU press *any other nu
```

Option 4 is for calculating the total cost of the accommodation using the reservation informations.



```
please enter the number of the action that you want:
1. See the information about all the available accomodations
2. See the the cost of cleaning
3. reserve accommodation
4. cost of the reservations for the wanted number of days before and after discount
5. See the information about all your reservations
6. how much will the web operator earn ?
4
the total cost of your accomodation with ID : P1 is 50 €
and the price after discount for ID : P1 is 47.5 €
if you want to go back to the OPTIONS (MENU) press *5* , if you want to go back to the MAIN MENU press *any other nu
```

Option 5 gives all the information about reserved accommodations of the certain user:



```
please enter the number of the action that you want:
1. See the information about all the available accomodations
2. See the the cost of cleaning
3. reserve accommodation
4. cost of the reservations for the wanted number of days before and after discount
5. See the information about all your reservations
6. how much will the web operator earn ?
5
________________________________________________________________________________________
The accomodation with ID P1 can have up to 4 Person per Night and has a price per night of 10 € and area is : 1
Campervans: true
Barbecue: true
________________________________________________________________________________________
________________________________________________________________________________________
The accomodation with ID B1 can have up to 6 Person per Night and has a price per night of 60 € and area is : 2
Number of rooms: 2
Wifi: true
________________________________________________________________________________________
if you want to go back to the OPTIONS (MENU) press *5* , if you want to go back to the MAIN MENU press *any other n
```

The last option, option 6 provides us with the choice to estimate the gonorar of web operator, that is calculated as a percentage fee if a client reserved any accommodation over 7 days. It will display an appropriate number according to the made reservations.