



Universiteti i Tiranës  
Fakulteti i Ekonomisë



Departamenti i Statistikës dhe Informatikës së Zbatuar

# Projekt

**Tema:** Të ndërtohet një aplikim GUI në JAVA nëpërmjet të cilit përdoruesit prenotojnë biletat e udhëtimit dhe marrin koston totale të udhëtimit



**Punoi:**

1. Anxhela Lika
2. Era Bilçari
3. Rovenë Dikellari
4. Teodora Plaka

**Grupi:** IE 202

**Pranoi:** Dr. Blerina Vika

## Përmbajtja:

1. Shtrimi i problemit dhe analiza e bërë (në bashkëpunim)
2. Diagrama e Rasteve të përdorimit ~ Teodora Plaka
3. Diagrama UML ~Era Bilçari & Anxhela Lika
4. Strukturimi sipas modelit MVC (Model View Controller) ~ Anxhela Lika
5. Interpretimi i kodimit ~ Rovena Dikellari & Era Bilçari
6. Kodimi i aplikimit (në bashkëpunim)
7. Studimi i mundësive të rezultateve të ekzekutimit ~Teodora Plaka & Rovena Dikellari



## Shtrimi i problemit dhe analiza e bërë:

Ideja e këtij aplikimi është krijimi i një sistemi për t'u dhënë mundësi përdoruesve për të prenotuar biletën e tyre dhe për të llogaritur koston totale të udhëtimit. Nëpërmjet përdorimit të GUI-së përdoruesi plotëson të dhëna personale (emër, mbiemër) si dhe përcakton të dhënat lidhur me udhëtimin si vendi i nisjes, destinacioni, lloji i udhëtimit (vajtje ose vajtje-ardhje) dhe numri i personave.

Rezultati i pritshëm do të jetë afishimi i të dhënave si dhe llogaritja e koston në bazë të çmimit të udhëtimit dhe numrit të personave. Sistemi merr parasysh gabimet e mundshme si pasojë e mosvendosjes së saktë të inputit dhe shfaq mesazh gabimi në mënyrë që përdoruesi të japë të dhëna të vlefshme.

## Diagrama e Rasteve te Përdorimit (Use Case)

Kjo diagramë përdoret për të shprehur funksionalitin e sistemit nga këndvështrimi i përdoruesit.

Fillimisht përcaktojmë aktorët kryesorë dhe me pas rastet e përdorimit(use case):

### AKTORËT:

- **Aktori primar** është përdoruesi, sepse ai do të përdorë aplikacionin dhe do të kryejë veprimet me të
- **Aktori dytësor** (sekondar) është agjensia e udhëtimeve, sepse ajo do të marrë informacione vetëm pasi përdoruesi ka kryer një veprim.  
Nje tjetër aktor dytësor është vetë përdoruesi, pasi ai mer informacione rreth faturës në fund të veprimit

### RASTET E PËRDORIMIT:

Përdoruesi vendos:

- të dhëna personale
- nisjen dhe destinacionin
- tipin e udhëtimit
- numrin e personave
- Ai mund të rezervojë biletën dhe të marrë faturën
- Mund të afishojë detajet
- Mund të boshatisë të dhënat e vendosura dhe të vendosë të dhëna të tjera

\* Për të ndërtuar këtë diagramë është përdorur aplikacioni Lucid (lucid.app).



## Strukturimi sipas modelit MWC

Klasat jane te organizuara sipas modelit MVC.

1. Klasat Udhetim dhe Rezervim permbajne variablat dhe metodat e nevojshme per te ndertuar funksionimin e nje bilete me te gjitha komponentet e nevojshme. Ato pranojne komandat e klases controller per te gjeneruar outputin, sepse ne vetvete nuk kane mundesi per te dhene rezultat, por vetem per te llogaritur veprimet brenda metodave, prandaj bejne pjese ne modelin e sistemit.
2. Perdorimi I GUI na jep mundesine per te patur nje paraqitje vizuale me te strukturuar dhe te qarte te asaj cka ne duam te bejme. Kijimi i nje frame-i dhe vendosja e komponenteve(paneleve) gjendet te klasa ReservationApp. Aparenca ne teresi se bashku me butonat, labelat, text-fields, combo-boxes etj perbejne pjesen 'View' te programit.
3. Pjesa e trete eshte 'Controller', i cili kontrollon te gjithe funksionimin e sistemit. Ne menyre qe te marrim nje rezultat te suksesshem duhet qe te gjitha te dhenat te vendosen ne menyre te sakte, ashtu si programi e kerkon. Ne te kundert, do te evidentohet gabimi i mundshem cka do te shkaktonte reagim te sistemit me nje mesazh gabimi. Pra, per cdo veprim kemi nje ngjarje pasuese, e cila mund te jete nje mesazh suksesi ose deshtimi. Prandaj, pjesa e dyte e klases se bashku me metoden main perben kontrolluesin e sistemit te krijuar.



## Analiza e kodit

Do të merren në shqyrtim 3 klasat e krijuara për funksionimin dhe paraqitjen e programit:

**1) KlasaUdhëtim** është klasa ku si fillim deklarojmë 3 variablat që na duhen për të bërë programin të cilat janë nisja ,destinacioni dhe çmimi të cilat do të inicializohen nga konstruktori përkatës (rreshti 9).

Në klasën Udhëtim kemi krijuar si dhe **3 metoda** përkatëse për të marrë secilën të dhënë që na duhet , ku në këtë rast, kemi metodat:

- ◆ **getCmimi**
- ◆ **getDestinacion,**
- ◆ **getNisja**

..të cilat i kërkojnë përdoruesit të vendosë inputin përkatës. Kemi përdorur metodën **get...** ,dhe nuk kemi shtuar një metodë **set** për vetëfaktin se të dhënat mbeten statike pas plotësimit të rezervimit me të dhënat fillestare.

**2)KlasaRezervim** është klasa e dytë që përmban të dhënat personale të blerësit si metodat për faturen dhe çmimin. Si fillim kemi inicializuar variablat :

- **emriBlerësit** (deklarohet emri i blerësit si variable private)
- **mbiemriBlerësit**(deklarohet mbiemri i blerësit)
- **nrPersonave** (variabël e tipit int që deklaron numrin e personave që do të rezervojnë udhëtimin)
- **isOneWay** (variable Boolean që by default është vetëm vajtje (pa kthim)
- **udhëtim**(variabli udhëtim pej klasës udhëtim (non-primitive data type) për të thirrur metodat e krijuara tek klasa Udhëtim.

që në këtë rast përdoruesit do t'i duhet ti plotësojë në mënyrë që të bëjë rezervimin e biletës.

Konstruktori përdoret për të inicializuarvariablat e deklaruara më sipër.

Metodat për marrjen e të dhënave nga përdoruesi janë:

- ◆ **metoda getEmriBlerësit** që do na kthejë emrin e përdoruesit,
- ◆ **metoda getMbiemriBlerësit** që do na kthejë mbiemrin e përdoruesit
- ◆ **metoda getNrPersonave** që do na kthejë numrin e personave që do marrin pjesë në udhëtim,
- ◆ **metoda isOneWay** që do na kthejë një boolean, e cila është OneWay by default.

Perveç tyre kemi **metoden Udhëtim** e cila do të aksesojë metodat e klases Udhëtim.

**Metoda getFature** do të afishojë të dhënat e blerësit së bashku me koston e udhëtimit. Në fillim inicializohet si boshe sepse nuk mban asnjë të dhënë para rezervimit.

**Metoda getDetaje** shërben për të dhënë më shumë detaje në lidhje me llojin e udhëtimit si për shembull one way apo round trip.

**Metoda llogaritKoston** shërben për të llogaritur koston në momentin që bëhet rezervimi. Është vendosur kushti sepse ndodhen 2 tipe udhëtimi dhe për këtë arsye kostoja do të llogaritet në mënyrë të ndryshme për secilin tip.

### 3) KlasaBileta\_main

Fillimisht importojmë paketat për GUI, për eventet dhe për listën e udhëtimeve.

Vendosim mundësinë për të shfaqur mesazhe gabimi në rast se inputi nuk do të jetë i saktë:

**Import static javax.swing.JOptionPane.ERROR\_MESSAGE;**

Klasa e krijuar trashëgon nga klasa JFrame metodat që do të duhen për krijimin e kornizës prandaj e bëjme 'extends'. Te kjo klasë krijojmë variablin nga klasa Rezervim në mënyrë që të aksesojmë metodat e saj së bashku me metodat e klases Udhetim (të aksesuara paraprakisht në klasën Rezervim).

Krijohet lista me udhëtimet e lejuara që përmbajnë nisjen, destinacionin dhe çmimin. Më pas përmes krijimit të konstruktorit vendosim metodat për ndërtimin e kornizës duke përdorur JFrame dhe panelin me JPanel.

**inicializoFormen();** është metode fillimisht e deklaruar në konstruktor dhe më pas edhe zhvillimi i saj. Ketu do të ndërtohen labelat, combo-boxet dhe butonat. Zgjedhim Grid Layout si dhe krijojmë etiketat me Label duke vendosur emrin për secilën prej tyre. Për emrin dhe mbiemrin përdorim TextField në mënyrë që përdoruesi të vendosë të dhënat e tij personale, ndërsa për nisjen, destinacionin dhe numrin e personave përdorim Combo Box për të zgjedhur nga listat e dhëna.

Për tipin e udhëtimit nëse është vajtj-ardhje ose vetëm ardhje apo vetëm vajtje përdorim Button Group duke qënë se duhet të zgjidhet vetëm njëra prej tyre. Krijohen butonat për Rezervo, Afisho Detajet dhe Clear. Duke përdorur ActionListener dhe metodat e tij kontrollojmë se çfarë ndodh kur klikojmë butonin Rezervo.

Në rastet kur përdoruesi nuk vendos emrin ose mbiemrin, shfaqet një mesazh gabimi në lidhje me vendosjen e pasaktë të të dhënave. Më pas zgjidhen të dhënat e tjera për nisjen, destinacionin, numrin e personave dhe tipin e udhëtimit. Dhe përsëri kontrollohet nëse udhëtimi mund të kryhet, në rast të kundërt shfaqet mesazh gabimi. Kur të gjitha të dhënat janë të plotësuara saktë afishohet mesazh suksesi dhe llogaritet fatura.

Duke përdorur ActionListener kontrollojmë ngjarjet për dy butonat e tjerë Afisho Detajet dhe Clear. Më pas ato shtohen në panelin e krijuar. Kontrollohet nëse udhëtimi nisje –destinacion është i lejuar apo jo dhe nëse po kthehet udhëtimi, nëse jo nuk kthehet gjë. Përdorim



funksonet `getNisjet()` dhe `getDestinations()` për t'i aksesuar këto të dhëna respektivisht nga lista e deklaruar në fillim. Në fund shfaqet mesazh suksesi kur udhëtimi është i mundur dhe gabim nëse nuk realizohet. Gjithashtu bëjmë dhe thirrjen në main për programin që u krijua.

```
1 //nisim me klasen udhetim duke percaktuar 3 variablat kyce: nisje, destinacion dhe cmimi
2 public class Udhetim {
3
4     private String nisja;
5
6     private String destinacion;
7
8     private Double cmimi;
9
10    public Udhetim( String nisja, String destinacion, Double cmimi) { //konstruktori i klases
11        this.destinacion = destinacion; //inicializimi i variablave
12        this.nisja = nisja;
13        this.cmimi = cmimi;
14    }
15
16    public String getNisja() {
17        return nisja;
18    }
19
20    public Double getCmimi() { //nuk duhet nie metode set sepse te dhenat mbeten statike pas plotesimit te rezervimit
21        return cmimi;
22    }
23
24    public String getDestinacion() {
25        return destinacion;
26    }
27
28 }
29
30 }
```

```
1 Rezervim.java x Udhetim.java Bileta_main.java
2 public class Rezervim
3 {
4     private String emriBleresit; //deklarohet emri i bleresit te biletes si private
5
6     private String mbiemriBleresit;
7
8     private int nrPersonave;
9
10    private boolean isOneWay; //variabel boolean qe by default eshte vetem vajtje(pa kthim)
11
12    private Udhetim udhetim; //variabli udhetim prej klases Udhetim (non-primitive data type)
13
14    public Rezervim(String emriBleresit, String mbiemriBleresit, int nrPersonave, boolean isOneWay, Udhetim udhetim) {
15        this.emriBleresit = emriBleresit;
16        this.mbiemriBleresit = mbiemriBleresit;
17        this.nrPersonave = nrPersonave;
18        this.udhetim = udhetim;
19        this.isOneWay = isOneWay;
20    }
21
22
23    //metodat per marrjen e te dhenave nga perdoruesi (emri, mbiemri, numri, tipi, dhe udhetim-permban nisjen, destinacionin dhe cmimin)
24    public String getEmriBleresit() {
25        return emriBleresit;
26    }
27
28    public String getMbiemriBleresit() {
29        return mbiemriBleresit;
30    }
31
32    public int getNrPersonave() {
33        return nrPersonave;
34    }
35
36    public boolean isOneWay() {
37        return isOneWay;
38    }
39
40    public Udhetim getUdhetim() {
41        return udhetim;
42    }
43 }
```



```

32 public int getNrPersonave() {
33     return nrPersonave;
34 }
35
36 public boolean isOneWay() {
37     return isOneWay;
38 }
39
40 public Udhetim getUdhetim() {
41     return udhetim;
42 }
43
44 public String getFature() { //fatura e tipit string e cila do te afishojte te dhenat e bleresit se bashku me cmimin
45     String fature = ""; //fillimisht e inicializoime si boshe sepse nuk permaban asnje te dhene(para rezervimit)
46     fature += "Emri i bleresit " + getEmriBleresit() + "\n";
47     fature += "Mbiemri i bleresit " + getMbiemriBleresit() + "\n";
48     fature += "Kostja e udhetimit " + llogaritKoston();
49
50     return fature;
51 }
52
53 public String getDetaje() { //shtojme detaje qe duam ti afishojme
54     return "Tipi i udhetimit= " + (isOneWay() ? "Eshte oneWay" : "Eshte round trip"); //nese tipi i udhetimit eshte vajtje apo vajtje-ardhie
55 }
56
57 private double llogaritKoston() {
58     if (isOneWay()) {
59         return getUdhetim().getCmimi() * getNrPersonave();
60     }
61
62     return getUdhetim().getCmimi() * getNrPersonave() * 2; //nese eshte vajtje ardhje, kostoja do te jete dyfish (*2)
63 }
64

```

```

1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.util.ArrayList;
6 import java.util.Arrays;
7 import java.util.List;
8
9 //KLASA QE IMPLEMENTON METODAR E GUI
10 import static javax.swing.JOptionPane.ERROR_MESSAGE; //Metode e gatshme e librarise javax.swing.JOptionPane kur duam te tregojme mesazh gabimi
11 public class Bileta_main extends JFrame {
12
13     private Rezervim rezervim;
14
15     //lista permaban nisjet, destinacionet dhe cmimet respektive
16     private static final List<Udhetim> UDHETIMET_E_LEJUARA = Arrays.asList(
17         new Udhetim("Milan", "Tirana", 100.99),
18         new Udhetim("Tirana", "Rome", 130.99),
19         new Udhetim("Tirana", "Athens", 100.99),
20         new Udhetim("Athens", "Frankfurt", 200.99),
21         new Udhetim("Paris", "Tirana", 150.99),
22         new Udhetim("Lisbon", "Nice", 106.99)
23     );
24
25     public Bileta main() {
26         super("Bileta_main");
27         setVisible(true);
28         setDefaultCloseOperation(EXIT_ON_CLOSE); //metode per te ndaluar vazhdimin e programit pasi ne e heqim (close)
29         setSize(700, 500);
30         JPanel pane = new JPanel(); //krijojme panelin qe do te permbaje elementet e kerkuara (komponenti)
31         setContentPane(pane);
32
33         initializeForm();
34

```

```

33     inicializoFormen();
34 }
35
36
37
38 public void inicializoFormen() {
39     JPanel paneli = new JPanel();
40
41     paneli.setLayout(new GridLayout(5, 4)); // permes GridLayout vendosim sa rreshta dhe kolona kemi
42
43     // Emri
44     paneli.add(new JLabel("Emri")); //Krijohet etiketa e emertuar "emri"
45     JTextField emriInput = new JTextField(); //vendosim me text-field emrin e perditesuar gjate plotesimit te biletës
46     paneli.add(emriInput); //merret inputi per te printuar faturen
47
48     // Mbiemri
49     paneli.add(new JLabel("Mbiemri"));
50     JTextField mbiemriInput = new JTextField();
51     paneli.add(mbiemriInput);
52
53     // Nisja
54     paneli.add(new JLabel("Nisja"));
55     JComboBox<String> nisjaInput = new JComboBox<>(getNisjet()); //perdorim combo-box per perzgjedhjen e vendeve nga do te nisemi
56     paneli.add(nisjaInput);
57
58     // Destinacion
59     paneli.add(new JLabel("Destinacioni"));
60     JComboBox<String> destinacioniInput = new JComboBox<>(getDestinacionet());
61     paneli.add(destinacioniInput);
62
63     // Tipi i udhetimit
64     ButtonGroup grupimiRadioButonave = new ButtonGroup(); // Nevojitet qe vetem nje radio button te selektohet
65     JRadioButton oneWay = new JRadioButton("One way");

```

```

65     JRadioButton oneWay = new JRadioButton("One way");
66     JRadioButton roundTrip = new JRadioButton("RoundTrip");
67     grupimiRadioButonave.add(oneWay);
68     grupimiRadioButonave.add(roundTrip);
69
70     paneli.add(new JLabel("Tipi i udhetimit"));
71     paneli.add(oneWay);
72     paneli.add(roundTrip);
73     paneli.add(new JLabel("")); // Per te mbushur njesine bosh ne menyre qe te strukturohen te dhenat e shfaqura
74
75     // Nr i personave
76     paneli.add(new JLabel("Nr i personave"));
77     JComboBox<Integer> nrIPersonaveInput = new JComboBox<>(new Integer[]{1, 2, 3, 4, 5});
78     paneli.add(nrIPersonaveInput);
79
80     paneli.add(new JLabel(""));
81     paneli.add(new JLabel(""));
82
83     // Butonat
84     JButton rezervo = new JButton("Rezervo"); //butoni per te rezervuar udhetimin pas plotesimit te te dhenave
85     JButton afishoDetaje = new JButton("Afisho Detaje");
86     JButton pastro = new JButton("Clear"); //pastrohen te dhenat per tu vendosur te dhena te reja
87
88     rezervo.addActionListener(new ActionListener() { //Eventi(ngjarja) qe shfaqet kur klikohet butoni rezervo
89
90     public void actionPerformed(ActionEvent actionEvent) { //metoda e ActionListener
91
92         String emri = emriInput.getText();
93
94         if (emri == null || emri.isEmpty()) {
95             afishoErrorMessage("Ju lutem vendosni emrin!");
96             return;

```

```

96         return;
97     }
98
99
100     String mbiemri = mbiemriInput.getText();
101     if(mbiemri==null||mbiemri.isEmpty())
102     {
103         afishoErrorMessage("Ju lutem vendosni mbiemrin!");
104         return;
105     }
106     String nisja = (String) nisjaInput.getSelectedItem(); //perzgjedhim nga lista e krijuar me ComboBox
107     String destinacion = (String) destinacioniInput.getSelectedItem();
108     int nrPersonave = (int) nrIPersonaveInput.getSelectedItem();
109
110
111     boolean isOneWay = oneWay.isSelected();
112
113     Udhetim udhetim = merrUdhetimInSipasParametrave(nisja, destinacion);
114
115     if (udhetim == null) { // TODO check other validations
116         afishoErrorMessage("Udhetimi i kerkuar nuk ofrohet per momentin");
117         return;
118     }
119
120     rezervim = new Rezervim(emri, mbiemri, nrPersonave, isOneWay, udhetim); //krijojme nje instance
121     afishoSuccessMessage(rezervim.getFature());
122 }
123 });
124
125
126
127 pastro.addActionListener(new ActionListener() { // Eventi kur klikohet butoni pastro
128

```

```

129     public void actionPerformed(ActionEvent actionEvent) {
130         emriInput.setText("");
131         mbiemriInput.setText("");
132         grupimiRadioButonave.clearSelection();
133     }
134 });
135
136 afishoDetaje.addActionListener(new ActionListener() { // Eventi kur klikohet butoni afishoDetaje
137
138     public void actionPerformed(ActionEvent actionEvent) {
139         if (rezervim == null) {
140             afishoErrorMessage("Nuk ka rezervim te bere");
141             return;
142         }
143         afishoSuccessMessage(rezervim.getDetaje());
144     }
145 });
146
147 paneli.add(rezervo);
148 paneli.add(afishoDetaje);
149 paneli.add(pastro);
150
151 add(paneli); // Shto ne JFrame
152
153 //lllogarisim udhetimin nise-destinacion, nese eshte i lejuar apo jo
154 private Udhetim merrUdhetimInSipasParametrave(String nisja, String destinacion) {
155     for (Udhetim udhetim : UDHETIMET_E_LEJUARA) { //bredhim te gjitha udhetimet e permendura
156         if (udhetim.getNisja().equals(nisja) && udhetim.getDestinacion().equals(destinacion)) { //nese nisja eshte e pranishme ne liste dhe
157                                                     //per kete nise ekziston nje destinacion
158             return udhetim;
159         }
160     }

```

```

159     }
160     }
161     }
162     return null;
163 }
164
165 private String[] getNisjet() {
166     List<String> nisjet = new ArrayList<>();
167     for (Udhetim udhetim : UDHETIMET_E_LEJUARA) {
168         nisjet.add(udhetim.getNisja());
169     }
170     return nisjet.stream().toArray(String[]::new); //konvertojme listen e vendeve nga do te nisemi ne nje array
171 }
172
173 private String[] getDestinacionet() {
174     List<String> destinacionet = new ArrayList<>();
175     for (Udhetim udhetim : UDHETIMET_E_LEJUARA) {
176         destinacionet.add(udhetim.getDestinacion());
177     }
178     return destinacionet.stream().toArray(String[]::new);
179 }
180
181 private void afishoSuccessMessage(String message) { //afishohet mesazh suksemi nese udhetimi eshte i mundur
182     JOptionPane.showMessageDialog(this, message);
183 }
184
185 private void afishoErrorMessage(String message) { //afishohet mesazh gabimi nese udhetimi nuk mund te realizohet
186     JOptionPane.showMessageDialog(this, message, "Mesazhi gabimi", ERROR_MESSAGE);
187 }
188 public static void main(String args[]) { //therrasim metoden main per programin qe ndertuam
189     new Bileta_main();
190 }
191 }

```

```

1 public class Rezervim
2 {
3
4     private String emriBleresit; //deklarohet emri i bleresit te biletas si private
5
6     private String mbiemriBleresit;
7
8     private int nrPersonave;
9
10    private boolean isOneWay; //variabel boolean qe by default eshte vetem vajtje(pa kthim)
11
12    private Udhetim udhetim; //variabli udhetim prej klases Udhetim (non-primitive data type)
13
14    public Rezervim(String emriBleresit, String mbiemriBleresit, int nrPersonave, boolean isOneWay, Udhetim udhetim) {
15        this.emriBleresit = emriBleresit;
16        this.mbiemriBleresit = mbiemriBleresit;
17        this.nrPersonave = nrPersonave;
18        this.udhetim = udhetim;
19        this.isOneWay = isOneWay;
20    }
21
22
23    //metodat per marrjen e te dhenave nga perdoruesi (emri, mbiemri, numri, tipi, dhe udhetim-permban nisen, destinacionin dhe cmimin)
24    public String getEmriBleresit() {
25        return emriBleresit;
26    }
27
28    public String getMbiemriBleresit() {
29        return mbiemriBleresit;
30    }
31
32    public int getNrPersonave() {

```

```

32    public int getNrPersonave() {
33        return nrPersonave;
34    }
35
36    public boolean isOneWay() {
37        return isOneWay;
38    }
39
40    public Udhetim getUdhetim() {
41        return udhetim;
42    }
43
44    public String getFature() { //fatura e tipit string e cila do te afishojte te dhenat e bleresit se bashku me cmimin
45        String fature = ""; //fillimisht e inicializojme si boshe sepse nuk permban asnje te dhene(para rezervimit)
46        fature += "Emri i bleresit " + getEmriBleresit() + "\n";
47        fature += "Mbiemri i bleresit " + getMbiemriBleresit() + "\n";
48        fature += "Kostja e udhetimit " + llogaritKoston();
49
50        return fature;
51    }
52
53    public String getDetaje() { //shtojme detaje qe duam ti afishojme
54        return "Tipi i udhetimit= " + (isOneWay() ? "Eshte oneWay" : "Eshte round trip"); //nese tipi i udhetimit eshte vajtje apo vajtje-ardhie
55    }
56
57    private double llogaritKoston() {
58        if (isOneWay()) {
59            return getUdhetim().getCmimi() * getNrPersonave();
60        }
61
62        return getUdhetim().getCmimi() * getNrPersonave() * 2; //nese eshte vajtje ardhje, kostoja do te jete dyfish (*2)
63    }
64 }

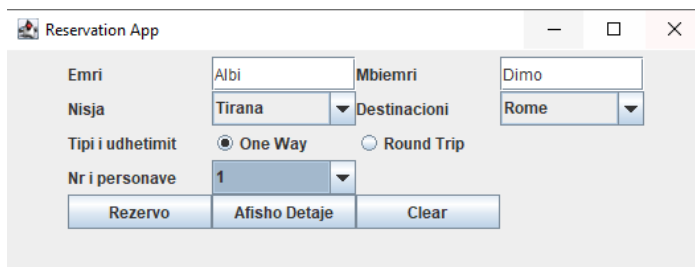
```



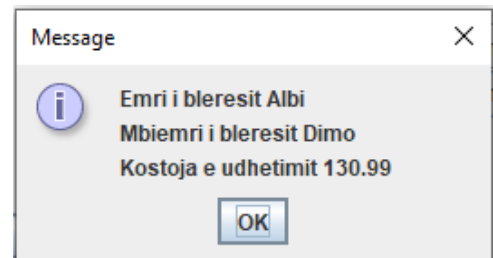
## Studimi i mundesive të rezultateve të ekzekutimit

Për aplikacionin tonë kemi këto rezultate të mundshme:

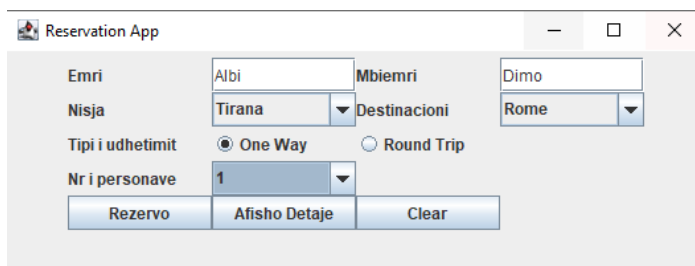
1. Përdoruesi vendos të dhëna të rregullta në të gjitha fushat dhe shtyp butonin “Rezervo”  
Në ekran do të shfaqet një mesazh që do të paraqesë emrin, mbiemrin e përdoruesit dhe koston e udhëtimit



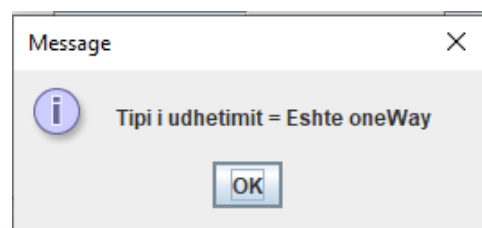
The screenshot shows the 'Reservation App' window. The form fields are filled with valid data: 'Emri' (First Name) is 'Albi', 'Mbiemri' (Last Name) is 'Dimo', 'Nisja' (Origin) is 'Tirana', 'Destinacioni' (Destination) is 'Rome', 'Tipi i udhetimit' (Trip Type) is 'One Way' (selected), and 'Nr i personave' (Number of people) is '1'. At the bottom, there are three buttons: 'Rezervo' (Reserve), 'Afisho Detaje' (Show Details), and 'Clear'.



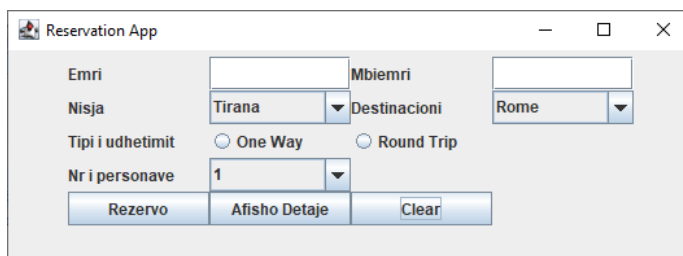
2. Përdoruesi vendos të dhëna të rregullta në të gjitha fushat dhe shtyp butonin “Afisho Detaje”  
Në ekran do të shfaqet një mesazh që do të paraqesë tipin e udhëtimit



The screenshot shows the 'Reservation App' window with the same valid data as in the first screenshot. The 'Rezervo' button is highlighted, indicating it has been clicked.

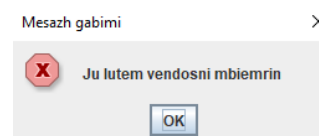
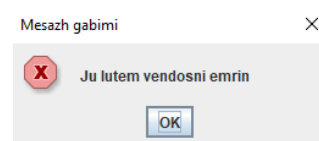


3. Përdoruesi vendos të dhëna të rregullta në të gjitha fushat dhe shtyp butonin “Clear”  
Fushat do të boshatisen të gjitha për t’u plotësuar me të dhëna të reja



The screenshot shows the 'Reservation App' window after the 'Clear' button has been clicked. All input fields are now empty, except for the 'Nisja' (Origin) and 'Destinacioni' (Destination) dropdown menus, which still show 'Tirana' and 'Rome' respectively. The 'Rezervo' button is highlighted.

4. Përdoruesi vendos të dhëna të parregullta . Në këtë rast kemi disa nënraste:
  - a) Përdoruesi harron të vendosë emrin  
Shfaqet një mesazh gabimi
  - b) Përdoruesi harron të vendosë mbiemrin



- c) Përdoruesi vendos një udhëtim me nisje dhe destinacion që nuk ofrohen nga kompania  
Do të afishohet një mesazh gabimi “*Udhëtimi i kërkuar nuk ofrohet për momentin*”.

Reservation App

Emri

Alma

Mbiemri

Bega

Nisja

Paris

Destinacioni

Frankfurt

Tipi i udhëimit

☒ One Way

☐ Round Trip

Nr i personave

5

Rezervo

Afsho Detaje

Clear

Mesazh gabimi

Udhëtimi i kërkuar nuk ofrohet për momentin

OK

