

Programowanie Obiektowe: “Program symulujący pracę bankomatu z funkcją wpłatomatu”

Jan Niedziółka

Spis treści:

1. Specyfikacja problemu. Przyjęte założenia i ograniczenia
2. Opis algorytmów oraz zmiennych
3. Diagram klas
4. Kod programu
5. Instrukcja dla użytkownika
6. Przykładowe dane i wyniki

Specyfikacja problemu. Przyjęte założenia i ograniczenia

Program ma za zadanie symulować działanie bankomatu, jego podstawowe funkcjonalności to:

- dodawanie kont nowych klientów z poziomu Serwisanta;
- wyświetlanie zawartości (ilości poszczególnych banknotów) i uzupełnianie ich z poziomu Serwisanta;
- wypłata i wpłata środków na konto bankowe z poziomu Klienta, dodatkowo doładowywanie telefonu środkami z konta;
- zapisywanie i odczytywanie danych do/z plików, wyświetlanie potwierdzeń transakcji, realizacja podstawowego interfejsu graficznego (SWING);
- obsługa zdarzeń (obsługa przycisków, pól tekstowych, wyskakujących okienek itd.) oraz wyjątków.

Z powodu całkowitej wirtualności zagadnienia przyjęto pewne uproszczenia względem rzeczywistych bankomatów co wpłynęło na komfort poruszania się po

programie (nie trzeba chociażby wpisywać za każdym razem kombinacji 16 cyfr karty bankomatowej i 4 cyfr jej numeru PIN), dosyć drugorzędnie potraktowano też kwestię GUI ograniczając się do najbardziej podstawowych funkcjonalności.

Do reprezentacji banknotów użyto HashMap – opierająca się na parze obiektów Key:Value umożliwia ich bardzo obrazowe i łatwe do zrozumienia dla programisty przedstawienie jako Nominał:Ilość (odpowiednio String i Integer). Poza tym HashMap ma dużą przewagę optymalizacyjną w stosunku do np. tablicy dwuwymiarowej, gdyż pozwala na operowanie w wygodny, szybki i czytelny sposób na takich wartościach, jakie akurat są potrzebne, co ilustruje poniższy przykład:

```
public void set100(int sto) {  
    this.banknoty.put("100pln", sto);  
}
```

Rys.1

Widać, że przekazując jako argument int sto można od razu ustawić ile banknotów o nominale 100 złotych jest przechowywanych w Bankomacie. Innym rozpatrywanym rozwiązaniem były po prostu zmienne typu int (np. int100 = 47), lecz na ostatecznej implementacji zaważyły użyteczne metody klasy HashMap.

Profil Klienta ma dostęp do trzech funkcjonalności w różny sposób wpływających na jego dane jak i na bankomat: Wpłata (zmniejsza się pula środków dostępnych na koncie Klienta ORAZ w Bankomacie), Wpłata (zwiększają się obie sumy), Doładowanie Telefonu (zmniejsza się pula środków Klienta ale nie zmienia się suma pieniędzy dostępnych w Bankomacie). Klient może mieć więcej gotówki niż jest dostępne w Bankomacie ale nie może wypłacić nadwyżki, nie może też wypłacić sumy większej niż stan jego konta. Chociaż nie da się wpłacić bezpośrednio do Bankomatu kwoty niepodzielnej przez 10 (ani też takiej wypłacić), to Klient może mieć na swoim koncie dowolne środki, nawet wyrażane jako liczba zmiennoprzecinkowa (z poziomu Serwisanta można też uzupełnić konto Klienta dowolną kwotą, to uzupełnianie jest traktowane jako „przelew”, nie jako wpłacenie banknotów do wpłatomatu). Wymienione tu założenia zostały albo wymuszone poprzez zgłaszanie wyjątków, albo wynikają z samej struktury aplikacji. Dla ułatwienia korzystania i zwiększenia przejrzystości niektóre komunikaty wypisywane są w konsoli. Z tego samego powodu nie są też ukryte dane zalogowanego użytkownika. Z uwagi na różnorodne sposoby przekazywania zmiennych (z pola tekstowego

„parsowane” na Integer, wpisywane z klawiatury fizycznej i ekranowej oraz jako obliczana, zwracana zmienna) nie udało się uniknąć powtarzania pewnych segmentów kodu lub funkcji z lekkimi tylko zmianami. Należy podkreślić, że zostało to zaniedbane świadomie, ponieważ zunifikowanie funkcji aby działały dla wszystkich przypadków wymagałoby więcej kodu, optymalizacji i obsługi wyjątków niż kilkukrotne przepisanie określonych jego fragmentów.

Opis algorytmów oraz zmiennych

```
}  
public Bankomat(int dziesiec, int dwadziescia, int piecdziesiat, int sto, int dwiescie, int piecset) {  
    this.banknoty.put("10pln", dziesiec);  
    this.banknoty.put("20pln", dwadziescia);  
    this.banknoty.put("50pln", piecdziesiat);  
    this.banknoty.put("100pln", sto);  
    this.banknoty.put("200pln", dwiescie);  
    this.banknoty.put("500pln", piecset);  
    int suma = ((10*this.banknoty.get("10pln")) + (20*this.banknoty.get("20pln")) + (50*this.banknoty.get("50pln"))  
        + (100*this.banknoty.get("100pln")) + (200*this.banknoty.get("200pln")) + (500*this.banknoty.get("500pln")));  
    this.banknoty.put("suma", suma);  
}
```

Całkowita kwota przechowywana na koncie lub w Bankomacie jest obliczana poprzez pomnożenie ilości banknotów o danym nominalne (ilość jest zwracana metodą Value = .get(Key) lub oznaczana jako i10, i20, i50 itd.) przez jego wartość.

Bardzo ciekawym algorytmem jest algorytm do wypłacania pieniędzy z Bankomatu:

```

@Override
public void IWyplata(Klient cus, int k){
    int i500 = 0, i200 = 0, i100 = 0, i50 = 0, i20 = 0, i10 = 0;
    float saldo = cus.getSrodki();
    int confirm = k;
    if(k >= 500) {
        i500 = (int)k/500;
        if(banknoty.get("500pln") >= i500)
            k = (k - (i500 * 500));
        else {
            i500 = banknoty.get("500pln");
            k = (k - (i500 * 500));
        }
    }
    if(k >= 200) {
        i200 = (int)k/200 ;
        if(banknoty.get("200pln") >= i200)
            k = (k - (i200 * 200));
        else {
            i200 = banknoty.get("pln200");
            k = (k - (i200 * 200));
        }
    }
    ,

```

Do metody dostarczana jest wypłacana kwota (int k). Jeśli jest większa niż największy nominał, to sprawdzane jest ile banknotów o tym nominale „pomieści” wypłacana suma. Jeśli banknotów o największym możliwym nominale wystarczy na wypłacenie żądanej kwoty to wypłacana jest jak największa jej część jak największym nominałem, a to co zostanie rozmienia się na mniejsze banknoty. Np. wypłacając 560 pln w pierwszej kolejności wypłacony zostanie banknot 500pln, zostanie 60 a więc ominie 200pln i 100pln, wypłacony zostanie banknot 50pln, ominięty 20pln i wypłacony 10pln (w ten sposób zadana na początku kwota 560pln po przejściu przez wszystkie nominały została rozłożona na części i zrównana do 0).

```

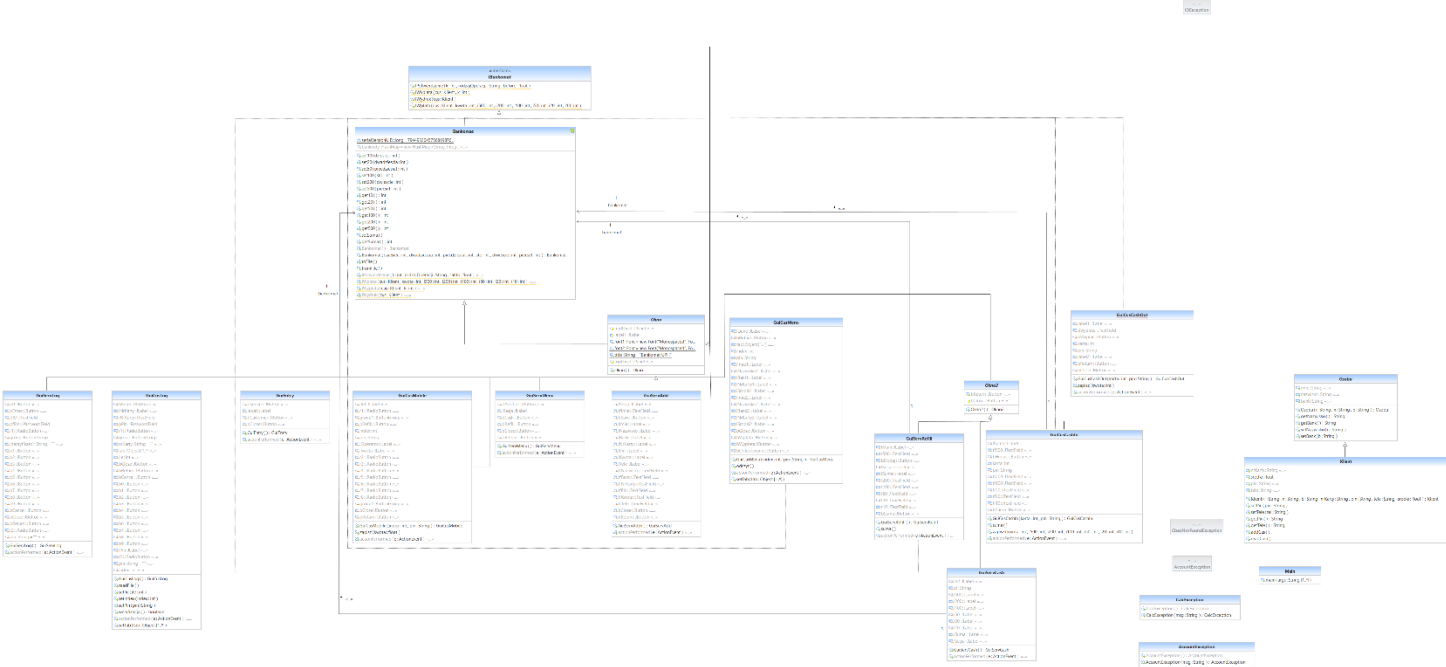
    if (k == 0) {
        try {
            this.set500(get500() - i500);
            this.set200(get200() - i200);
            this.set100(get100() - i100);
            this.set50(get50() - i50);
            this.set20(get20() - i20);
            this.set10(get10() - i10);
            this.setSuma();
            thisToFile();
        } catch (IOException exception) {
            ...
        }
    }

```

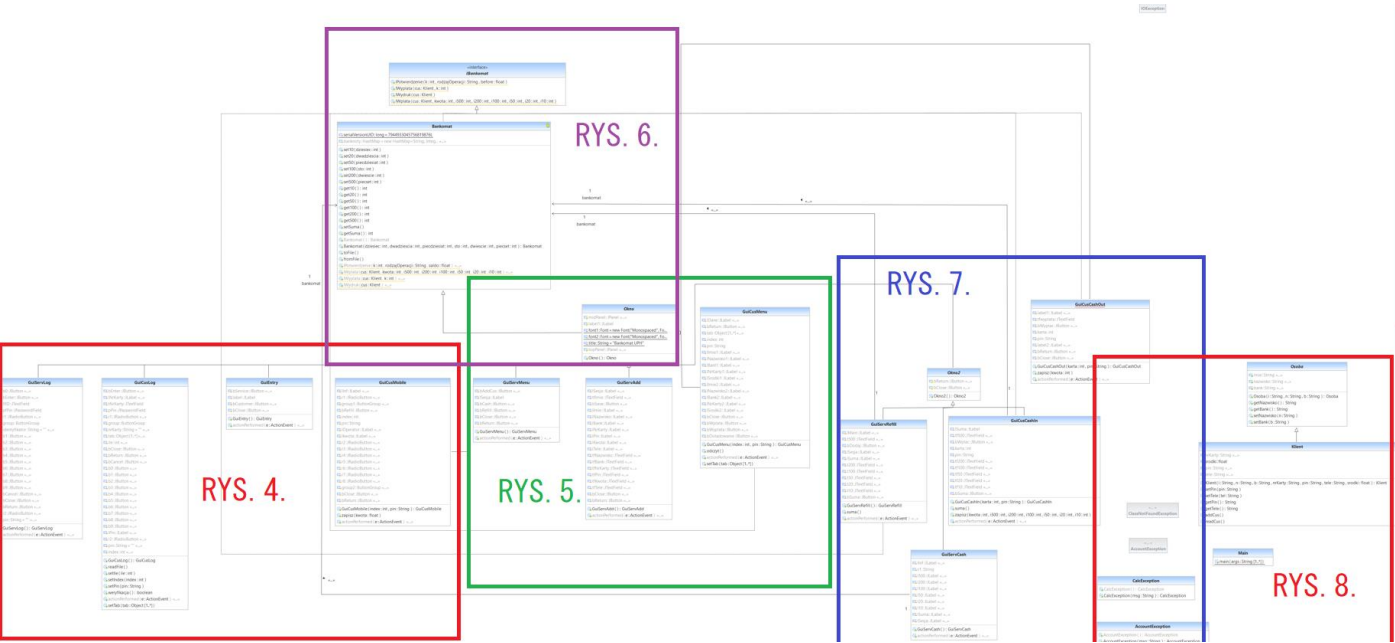
Ilości wypłacanych banknotów (pod zmiennymi i500, i200 itd.) są odejmowane z

puli banknotów przechowywanych w Bankomacie. Jeśli na końcu k będzie większe od 0 to znaczy, że w Bankomacie nie ma wystarczającej ilości środków.

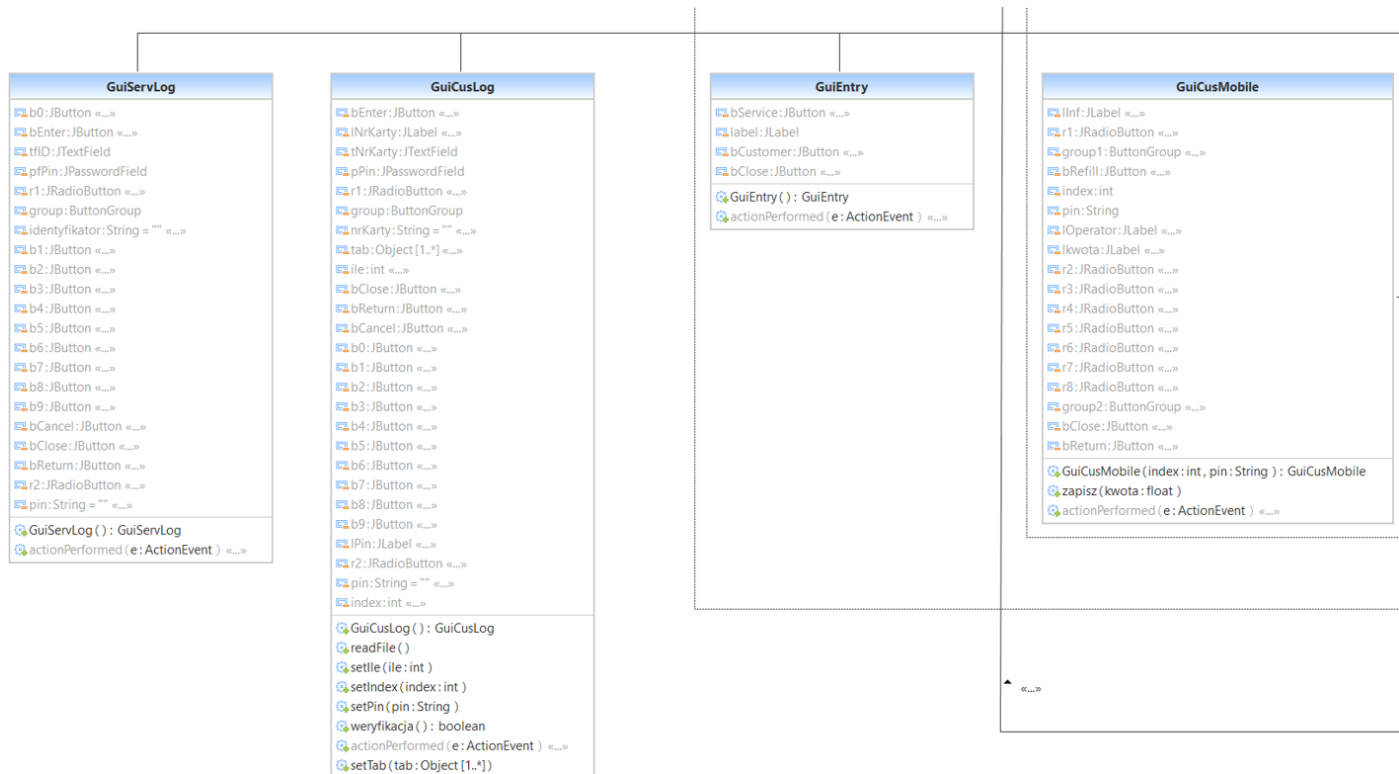
Diagram klas



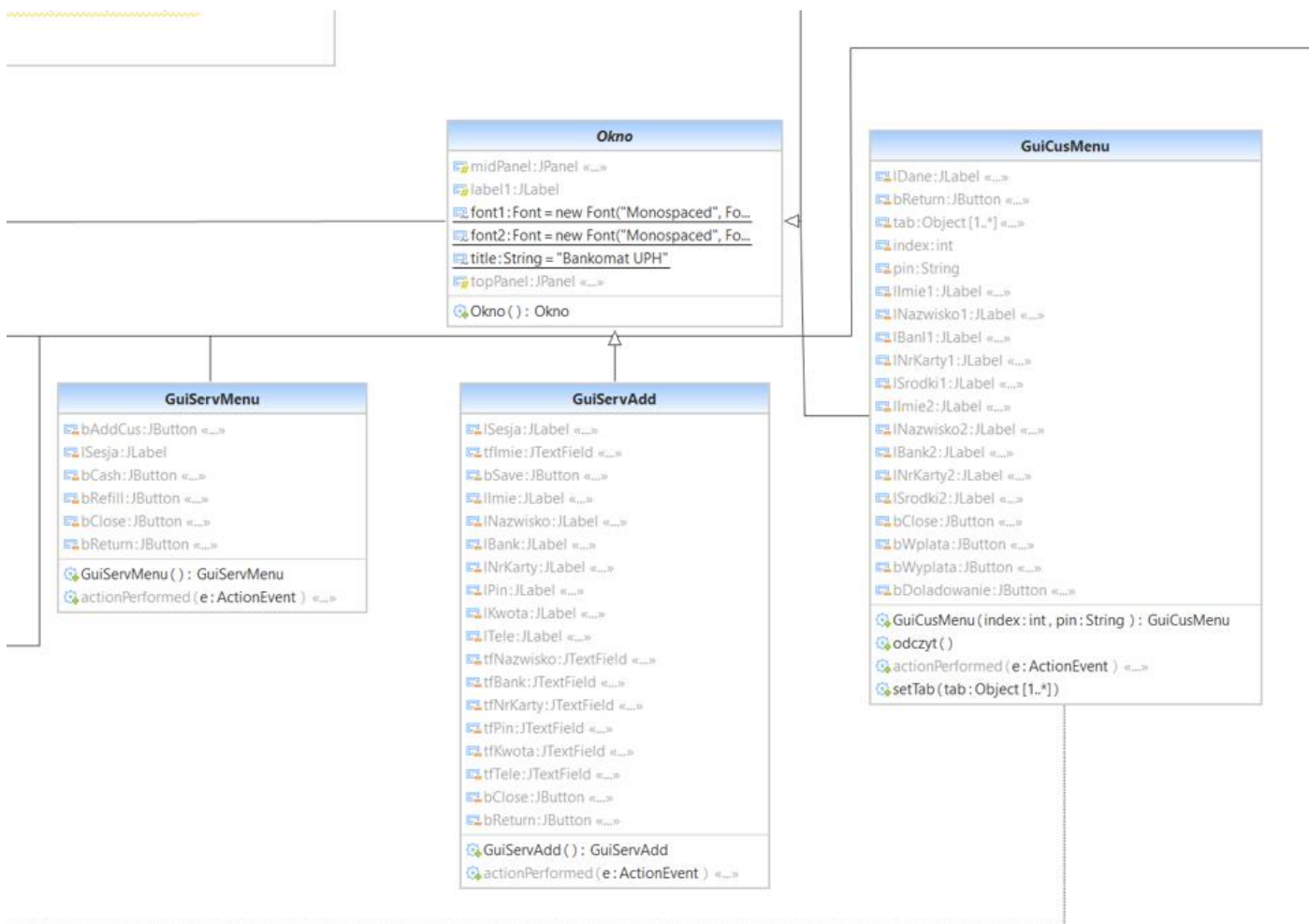
Rys.2.



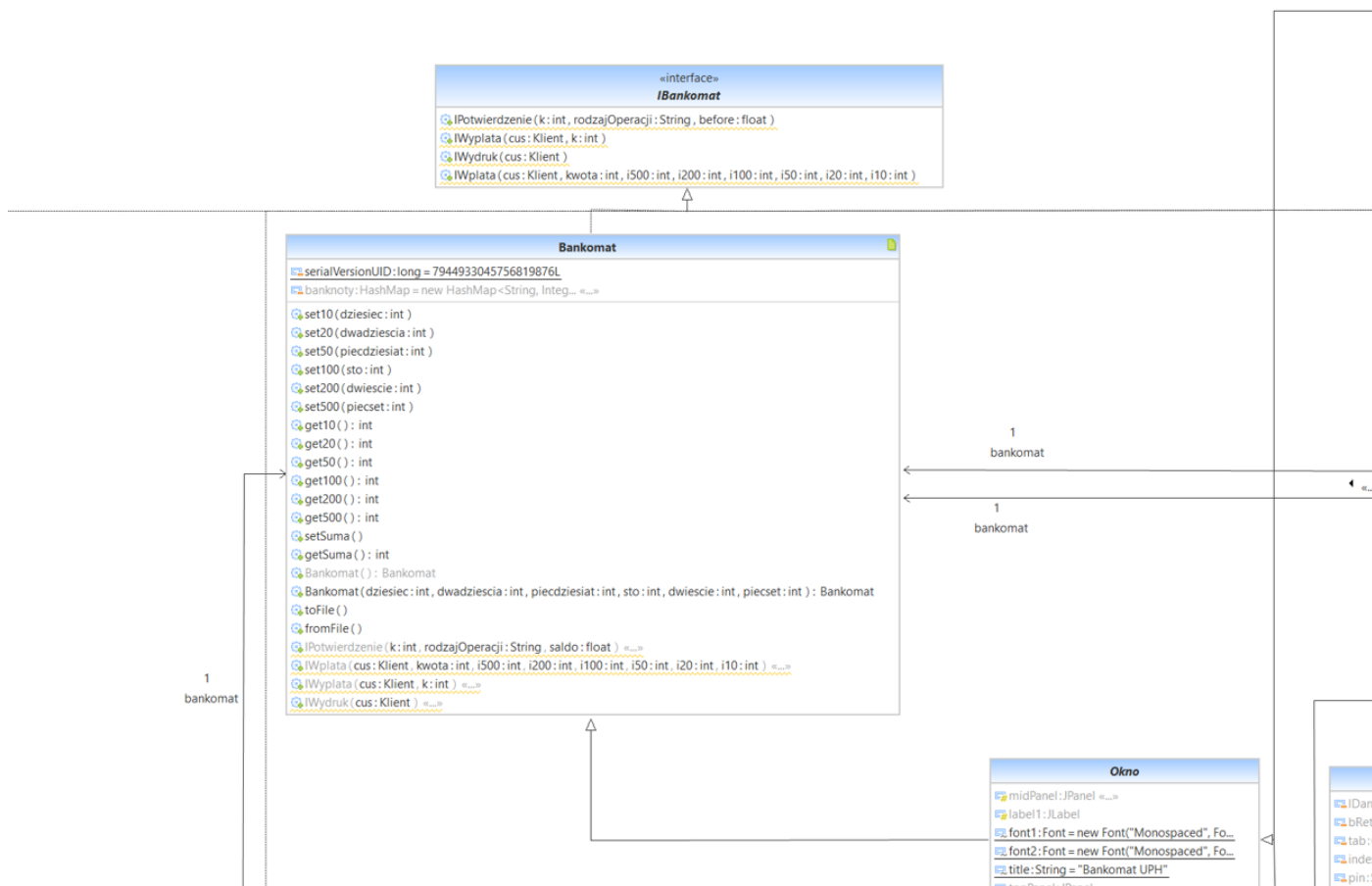
Rys.3.



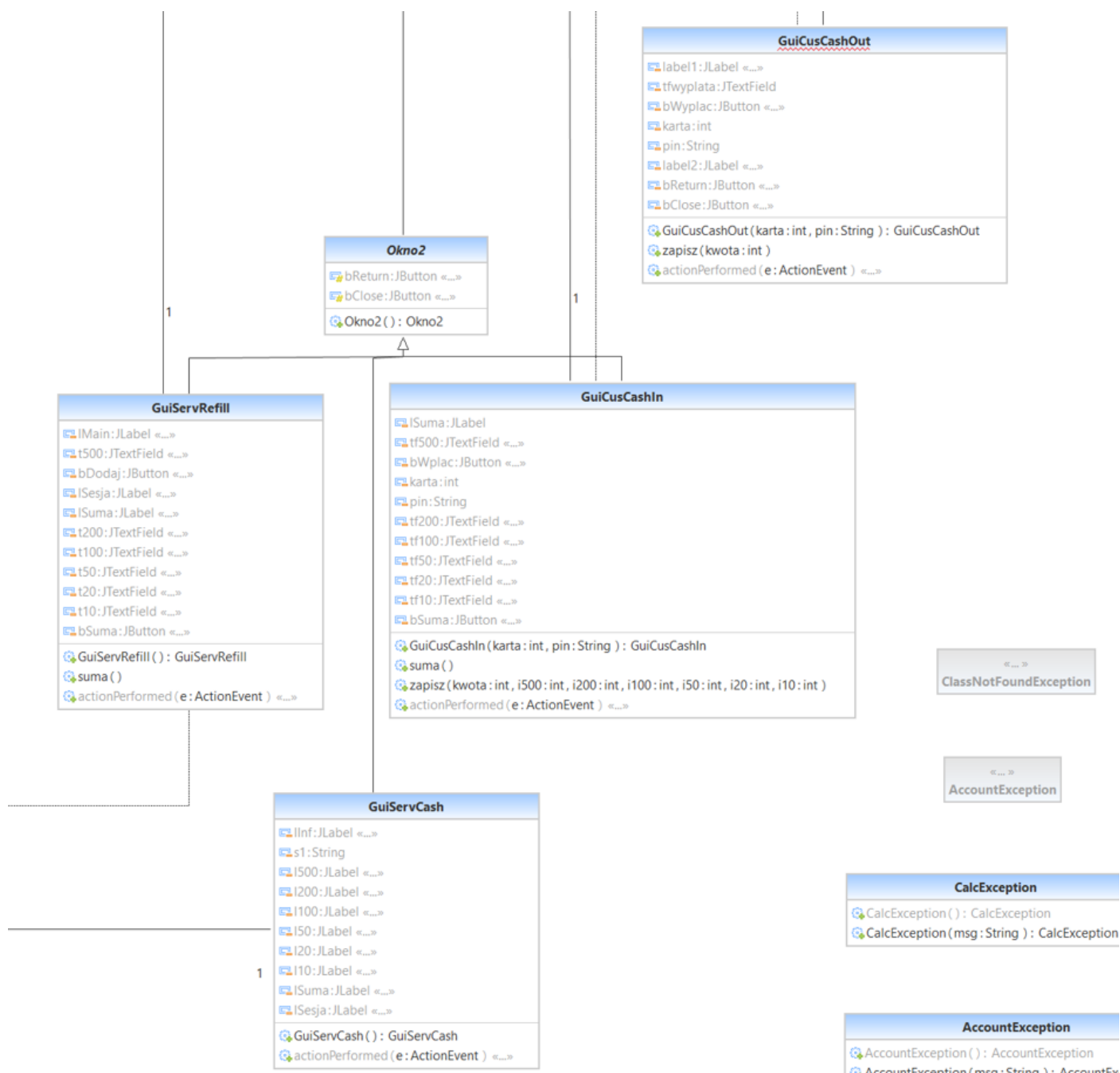
Rys.4.



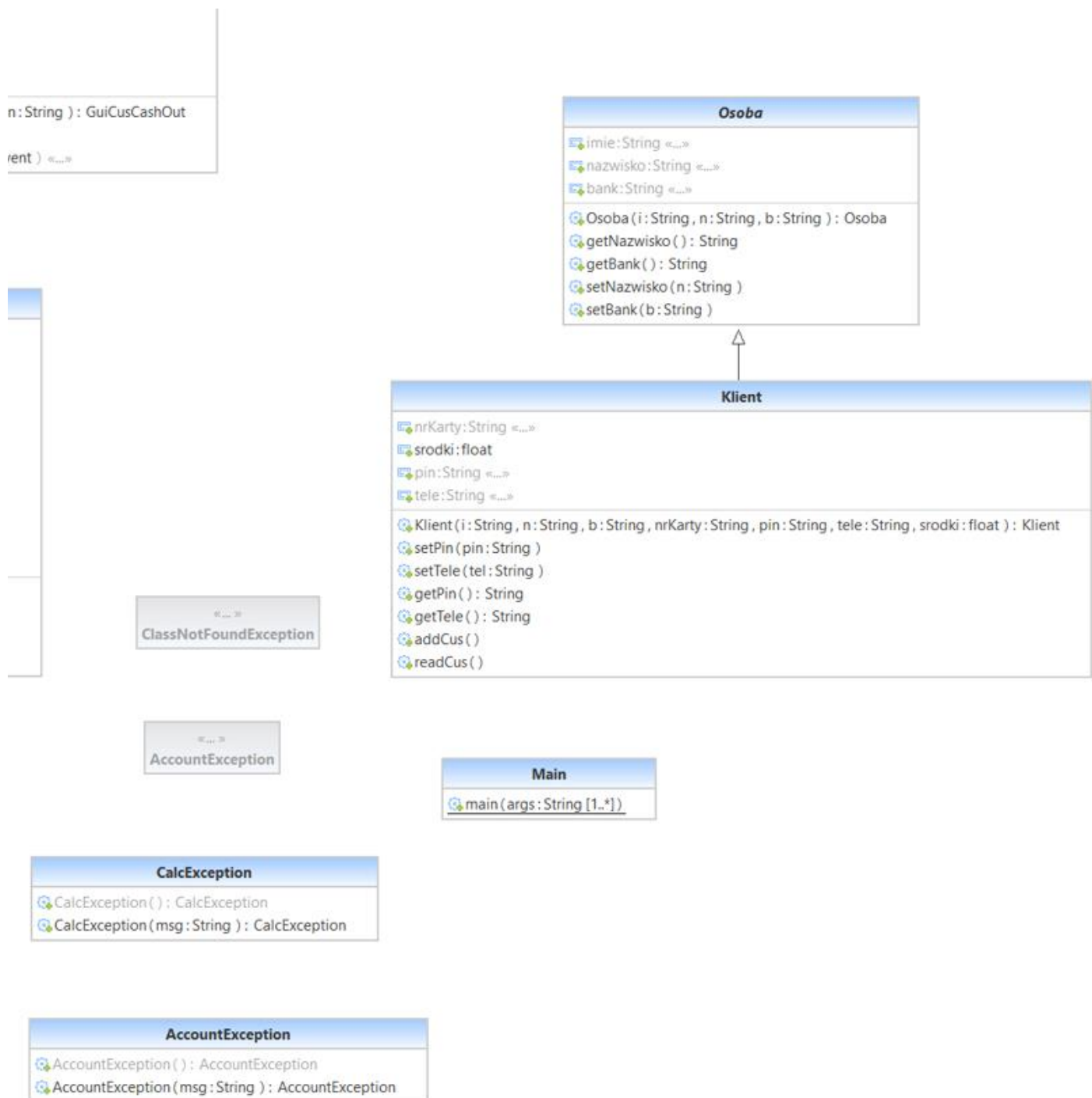
Rys.5.



Rys.6.



Rys.7.



Rys.8.

Kod programu

Interfejs IBankomat

```
IBankomat.java
1 package atm_final2;
2
3 import java.io.IOException;
4
5
6 public interface IBankomat{
7
8     void IWypłata(Klient cus, int k) throws AccountException;
9
10    void IWydruk(Klient cus);
11
12    void IWpłata(Klient cus, int kwota, int i500, int i200, int i100, int i50, int i20, int i10) throws CalcException;
13
14    void IPotwierdzenie(int k, String rodzajOperacji, float before);
15
16 }
```

Klasa Bankomat

```
Bankomat.java
1 package atm_final2;
2
3 import javax.swing.*;
4
5 /**
6  * @author Jan Niedziolka
7  */
8
9 public class Bankomat extends JFrame implements Serializable, IBankomat {
10
11     /**
12      *
13      */
14     private static final long serialVersionUID = 7944933045756819876L;
15     private HashMap<String, Integer> banknoty = new HashMap<String, Integer>();
16
17     public void set10(int dziesiec) {
18         this.banknoty.put("10pln", dziesiec);
19     }
20
21     public void set20(int dwadziescia) {
22         this.banknoty.put("20pln", dwadziescia);
23     }
24
25     public void set50(int piecdziesiat) {
26         this.banknoty.put("50pln", piecdziesiat);
27     }
28
29     public void set100(int sto) {
30         this.banknoty.put("100pln", sto);
31     }
32
33     public void set200(int dwiescie) {
34         this.banknoty.put("200pln", dwiescie);
35     }
36
37     public void set500(int piecset) {
38         this.banknoty.put("500pln", piecset);
39     }
40
41     public int get10() {
42         if(banknoty.get("10pln") == null) {
43             set10(0);
44             return 0;
45         }
46         else
47             return banknoty.get("10pln");
48     }
49 }
```

Bankomat.java

```

37
38 public int get10() {
39     if(banknoty.get("10pln") == null) {
40         set10(0);
41         return 0;
42     }
43     else
44         return banknoty.get("10pln");
45 }
46 public int get20() {
47     if(banknoty.get("20pln") == null) {
48         set20(0);
49         return 0;
50     }
51     else
52         return banknoty.get("20pln");
53 }
54 public int get50() {
55     if(banknoty.get("50pln") == null) {
56         set50(0);
57         return 0;
58     }
59     else
60         return banknoty.get("50pln");
61 }
62 public int get100() {
63     if(banknoty.get("100pln") == null) {
64         set100(0);
65         return 0;
66     }
67     else
68         return banknoty.get("100pln");
69 }
70 public int get200() {
71     if(banknoty.get("200pln") == null) {
72         set200(0);
73         return 0;
74     }
75     else
76         return banknoty.get("200pln");
77 }

```

Bankomat.java

```

75     else
76         return banknoty.get("200pln");
77 }
78 public int get500() {
79     if(banknoty.get("500pln") == null) {
80         set500(0);
81         return 0;
82     }
83     else
84         return banknoty.get("500pln");
85 }
86
87 public void setSuma() {
88     int suma = ((10*banknoty.get("10pln")) + (20*banknoty.get("20pln")) + (50*banknoty.get("50pln"))
89         + (100*banknoty.get("100pln")) + (200*banknoty.get("200pln")) + (500*banknoty.get("500pln")));
90     this.banknoty.put("suma", suma);
91 }
92 public int getSuma() {
93     if(banknoty.get("suma") == null)
94         return 0;
95     else
96         return banknoty.get("suma");
97 }
98
99
100 public Bankomat() {
101 }
102 public Bankomat(int dziesiec, int dwadziescia, int piecdziesiat, int sto, int dwiescie, int piecset) {
103     this.banknoty.put("10pln", dziesiec);
104     this.banknoty.put("20pln", dwadziescia);
105     this.banknoty.put("50pln", piecdziesiat);
106     this.banknoty.put("100pln", sto);
107     this.banknoty.put("200pln", dwiescie);
108     this.banknoty.put("500pln", piecset);
109     int suma = ((10*banknoty.get("10pln")) + (20*banknoty.get("20pln")) + (50*banknoty.get("50pln"))
110         + (100*banknoty.get("100pln")) + (200*banknoty.get("200pln")) + (500*banknoty.get("500pln")));
111     this.banknoty.put("suma", suma);
112 }
113
114 public void toFile() throws IOException, ClassNotFoundException {
115     ObjectOutputStream outputStream = null;

```

```

Bankomat.java
114 public void toFile() throws IOException, ClassNotFoundException {
115     ObjectOutputStream outputStream = null;
116     try {
117         outputStream = new ObjectOutputStream(new FileOutputStream("bankomat.txt"));
118         outputStream.writeObject(this);
119         outputStream.flush();
120     } finally {
121         if(outputStream != null)
122             outputStream.close();
123     }
124 }
125
126 public void fromFile() throws IOException, ClassNotFoundException {
127     ObjectInputStream inputStream = null;
128     Bankomat atm = null;
129     try {
130         inputStream = new ObjectInputStream(new FileInputStream("bankomat.txt"));
131         while(true) {
132             atm = (Bankomat) inputStream.readObject();
133             this.set500(atm.get500());
134             this.set200(atm.get200());
135             this.set100(atm.get100());
136             this.set50(atm.get50());
137             this.set20(atm.get20());
138             this.set10(atm.get10());
139             this.setSuma();
140         }
141     } catch (EOFException ex) {
142     } finally {
143         if (inputStream != null)
144             inputStream.close();
145     }
146 }
147
148 @Override
149 public void IWpłata(Klient cus, int kwota, int i500, int i200, int i100, int i50, int i20, int i10){
150     int confirm = kwota;
151     try {
152         this.set500(get500() + i500);
153         this.set200(get200() + i200);
154         this.set100(get100() + i100);

```

```

Bankomat.java
151     try {
152         this.set500(get500() + i500);
153         this.set200(get200() + i200);
154         this.set100(get100() + i100);
155         this.set50(get50() + i50);
156         this.set20(get20() + i20);
157         this.set10(get10() + i10);
158         this.setSuma();
159     } catch (NumberFormatException exception) {
160         JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
161     }
162     IPotwierdzenie(confirm, "wpłata środków", cus.getSrodki());
163 }
164
165 @Override
166 public void IWypłata(Klient cus, int k){
167     int i500 = 0, i200 = 0, i100 = 0, i50 = 0, i20 = 0, i10 = 0;
168     float saldo = cus.getSrodki();
169     int confirm = k;
170     if(k >= 500) {
171         i500 = (int)k/500;
172         if(banknoty.get("500pln") >= i500)
173             k = (k - (i500 * 500));
174         else {
175             i500 = banknoty.get("500pln");
176             k = (k - (i500 * 500));
177         }
178     }
179     if(k >= 200) {
180         i200 = (int)k/200;
181         if(banknoty.get("200pln") >= i200)
182             k = (k - (i200 * 200));
183         else {
184             i200 = banknoty.get("pln200");
185             k = (k - (i200 * 200));
186         }
187     }
188     if(k >= 100) {
189         i100 = (int)k/100;
190         if(banknoty.get("100pln") >= i100)
191             k = (k - (i100 * 100));

```

Bankomat.java

```

188         if(k >= 100) {
189             i100 = (int)k/100 ;
190             if(banknoty.get("100pln") >= i100)
191                 k = (k - (i100 * 100));
192             else {
193                 i100 = banknoty.get("100pln");
194                 k = (k - (i100 * 100));
195             }
196         }
197         if(k >= 50) {
198             i50 = (int)k/50 ;
199             if(banknoty.get("50pln") >= i50)
200                 k = (k - (i50 * 50));
201             else {
202                 i50 = banknoty.get("50pln");
203                 k = (k - (i50 * 50));
204             }
205         }
206         if(k >= 20) {
207             i20 = (int)k/20 ;
208             if(banknoty.get("20pln") >= i20)
209                 k = (k - (i20 * 20));
210             else {
211                 i20 = banknoty.get("20pln");
212                 k = (k - (i20 * 20));
213             }
214         }
215         if(k >= 10) {
216             i10 = (int)k/10 ;
217             if(banknoty.get("10pln") >= i10)
218                 k = (k - (i10 * 10));
219             else {
220                 i10 = banknoty.get("10pln");
221                 k = (k - (i10 * 10));
222             }
223         }
224
225         if (k == 0) {
226             try {
227                 this.set500(get500() - i500);
228                 this.set200(get200() - i200);

```

Bankomat.java

```

225         if (k == 0) {
226             try {
227                 this.set500(get500() - i500);
228                 this.set200(get200() - i200);
229                 this.set100(get100() - i100);
230                 this.set50(get50() - i50);
231                 this.set20(get20() - i20);
232                 this.set10(get10() - i10);
233                 this.setSuma();
234                 this.toFile();
235             } catch (IOException exception) {
236                 exception.printStackTrace();
237             } catch (ClassNotFoundException exception) {
238                 JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Błąd", JOptionPane.INFORMATION_MESSAGE);
239                 exception.printStackTrace();
240             } catch (NumberFormatException exception) {
241                 JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
242             }
243
244             IPotwierdzenie(confirm, "wypłata środków", saldo);
245         } else {
246             JOptionPane.showMessageDialog(null, "Brak srodkow w bankomacie", "Wynik", JOptionPane.INFORMATION_MESSAGE);
247         }
248     }
249
250     @Override
251     public void IPotwierdzenie(int k, String rodzajOperacji, float saldo) {
252         //data operacji
253         SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd", 'HH:mm:ss");
254         String date = sdf.format(new Date());
255         //rodzaj operacji
256         //kwota operacji
257         //saldo po operacji
258         String inf = "Data wykonania operacji: " + date +
259             "\nRodzaj wykonanej operacji: " + rodzajOperacji +
260             "\nKwota operacji: " + k + " PLN" +
261             "\nSaldo po operacji: " + saldo;
262         JOptionPane.showMessageDialog(null, inf, "Potwierdzenie: ", JOptionPane.INFORMATION_MESSAGE);
263     }
264
265     @Override

```

```

265 @Override
266 public void IWydruk(Klient cus) {
267     SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd", 'HH:mm:ss");
268     String date = sdf.format(new Date());
269     String wydruk = ("Stan konta w " + date + ": " + cus.getSrodki());
270     JOptionPane.showMessageDialog(null, wydruk, "Drukowanie stanu konta: ", JOptionPane.INFORMATION_MESSAGE);
271 }
272 }
273

```

Klasa Osoba

```

Osoba.java
1 package atm_final2;
2
3 import java.io.Serializable;
4
5 public abstract class Osoba implements Serializable {
6     private String imie, nazwisko, bank;
7
8     public Osoba(String i, String n, String b) {
9         this.imie = i;
10        this.nazwisko = n;
11        this.bank = b;
12    }
13
14    public String getImie() {
15        return imie;
16    }
17    public String getNazwisko() {
18        return nazwisko;
19    }
20    public String getBank() {
21        return bank;
22    }
23
24    public void setImie(String i) {
25        this.imie = i;
26    }
27    public void setNazwisko(String n) {
28        this.nazwisko = n;
29    }
30    public void setBank(String b) {
31        this.bank = b;
32    }
33 }
34

```

Klasa Klient

```
Klient.java
1 package atm_final2;
2
3 import java.io.EOFException;
4
5
6
7
8
9
10 public class Klient extends Osoba{
11     private String nrKarty, pin, tele;
12     private float srodki;
13
14     public Klient(String i, String n, String b, String nrKarty, String pin, String tele, float srodki) {
15         super(i, n, b);
16         this.nrKarty = nrKarty;
17         this.pin = pin;
18         this.tele = tele;
19         this.srodki = srodki;
20     }
21
22     public void setNrKarty(String nK) {
23         this.nrKarty = nK;
24     }
25     public void setPin(String pin) {
26         this.pin = pin;
27     }
28     public void setTele(String tel) {
29         this.tele = tel;
30     }
31     public void setSrodki(float srodki) {
32         this.srodki = srodki;
33     }
34
35     public String getNrKarty() {
36         return nrKarty;
37     }
38     public String getPin() {
39         return pin;
40     }
41     public String getTele() {
42         return tele;
43     }
44     public float getSrodki() {
45         return srodki;
46     }
47 }
```

```
Klient.java
44     public float getSrodki() {
45         return srodki;
46     }
47
48     public void addCus() throws IOException, ClassNotFoundException{
49         Klient[] arr = new Klient[50];
50         ObjectInputStream update = null;
51         int count = 0;
52
53         try {
54             update = new ObjectInputStream(new FileInputStream("klient.txt"));
55             while(true) {
56                 arr[count] = (Klient)update.readObject();
57                 count++;
58             }
59         } catch(EOFException e) {
60             if(update != null)
61                 update.close();
62
63             ObjectOutputStream write = null;
64             try {
65                 write = new ObjectOutputStream(new FileOutputStream("klient.txt"));
66                 for(int i = 0; arr[i] != null; i++)
67                     write.writeObject(arr[i]);
68
69                 write.writeObject(this);
70                 count++;
71                 write.flush();
72             } finally {
73                 if(write != null)
74                     write.close();
75             }
76         } catch(IOException e) {
77             System.out.println("Klient//addCus//IOException");
78         }
79     }
80
81     public void readCus() throws IOException, ClassNotFoundException{
82         Object[][] arr = new Object[50][7];
83         ObjectInputStream read = null;
84         Klient cus = null;
```

```

Klient.java
72     } finally {
73         if(write != null)
74             write.close();
75     }
76 } catch(IOException e) {
77     System.out.println("Klient//addCus//IOException");
78 }
79 }
80
81 public void readCus() throws IOException, ClassNotFoundException{
82     Object[][] arr = new Object[50][7];
83     ObjectInputStream read = null;
84     Klient cus = null;
85     int count = 0;
86
87     try {
88         read = new ObjectInputStream(new FileInputStream("klient.txt"));
89         while(true) {
90             cus = (Klient)read.readObject();
91             arr[count][0] = cus.getImie();
92             arr[count][1] = cus.getNazwisko();
93             arr[count][2] = cus.getBank();
94             arr[count][3] = cus.getNrKarty();
95             arr[count][4] = cus.getPin();
96             arr[count][5] = cus.getTele();
97             arr[count][6] = cus.getSrodki();
98             count++;
99         }
100     } catch EOFException e) {
101     } finally {
102         if(read != null)
103             read.close();
104         /*for(int i = 0; i < count; i++) {
105             for(int j = 0; j < 7; j++) {
106                 System.out.print(arr[i][j] + " ");
107             }
108         }*/
109     }
110 }
111 }

```

Klasa Okno (gui)

```

Okno.java
1 package atm_final2;
2
3 import javax.swing.*;
4
5
6 public abstract class Okno extends Bankomat {
7     protected JPanel midPanel, topPanel;
8     protected JLabel label1;
9     static final Font font1 = new Font("Monospaced", Font.BOLD, 35);
10    static final Font font2 = new Font("Monospaced", Font.PLAIN, 25);
11    static final String title = "Bankomat UPH";
12
13    public Okno() {
14        setSize(860, 840);
15        this.setDefaultCloseOperation(EXIT_ON_CLOSE);
16
17        topPanel = new JPanel();
18        topPanel.setBounds(0, 0, 860, 840);
19        topPanel.setBackground(Color.GRAY);
20        topPanel.setLayout(new BorderLayout());
21        topPanel.setLayout(null);
22        label1 = new JLabel(title);
23        label1.setBounds(300, 50, 400, 40);
24        label1.setFont(font1);
25        topPanel.add(label1);
26
27        midPanel = new JPanel();
28        midPanel.setLayout(new BorderLayout());
29        midPanel.setLayout(null);
30        midPanel.setBounds(100, 100, 600, 600);
31        midPanel.setBackground(Color.LIGHT_GRAY);
32
33        topPanel.add(midPanel);
34        add(topPanel);
35    }
36 }
37
38

```


Klasa Okno2 (gui)

```
Okno2.java
1 package atm_final2;
2
3 import javax.swing.JButton;
4
5
6
7 public abstract class Okno2 extends Okno {
8     protected JButton bReturn, bClose;
9
10    public Okno2()
11    {
12        JLabel label500 = new JLabel("500 PLN: " );
13        label500.setBounds(120, 100, 250, 50);
14        label500.setFont(font1);
15        midPanel.add(label500);
16
17        JLabel label200 = new JLabel("200 PLN: " );
18        label200.setBounds(120, 150, 250, 50);
19        label200.setFont(font1);
20        midPanel.add(label200);
21
22        JLabel label100 = new JLabel("100 PLN: " );
23        label100.setBounds(120, 200, 250, 50);
24        label100.setFont(font1);
25        midPanel.add(label100);
26
27        JLabel label50 = new JLabel("50 PLN: " );
28        label50.setBounds(120, 250, 250, 50);
29        label50.setFont(font1);
30        midPanel.add(label50);
31
32        JLabel label20 = new JLabel("20 PLN: " );
33        label20.setBounds(120, 300, 250, 50);
34        label20.setFont(font1);
35        midPanel.add(label20);
36
37        JLabel label10 = new JLabel("10 PLN: " );
38        label10.setBounds(120, 350, 250, 50);
39        label10.setFont(font1);
40        midPanel.add(label10);
41
42        JLabel labelSuma = new JLabel("Laczna kwota: " );
43
44
45        JLabel labelSuma = new JLabel("Laczna kwota: " );
46        labelSuma.setBounds(80, 450, 400, 50);
47        labelSuma.setFont(font1);
48        midPanel.add(labelSuma);
49
50        bReturn = new JButton("Powrót");
51        bReturn.setBounds(0, 550, 100, 50);
52        midPanel.add(bReturn);
53
54        bClose = new JButton("Zamknij");
55        bClose.setBounds(500, 550, 100, 50);
56        midPanel.add(bClose);
57    }
58 }
```

Klasa Main

```
Mainjava
1 package atm_final2;
2
3 import javax.swing.*;
4
5
6 public class Main {
7     public static void main(String[] args) throws IOException, ClassNotFoundException {
8         GuiEntry entry = new GuiEntry();
9         entry.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10        entry.setVisible(true);
11    }
12 }
13
```

Klasa GuiEntry

```
GuiEntry.java
1 package atm_final2;
2
3 import javax.swing.*;
4
5
6 public class GuiEntry extends Okno implements ActionListener {
7     private JButton bService, bCustomer, bClose; //bReturn;
8     private JLabel label;
9
10    public GuiEntry() {
11        label = new JLabel("Wybór użytkownika");
12        label.setBounds(175, 100, 300, 50);
13        label.setFont(font2);
14        midPanel.add(label);
15
16        bService = new JButton("Pracownik");
17        bService.setBounds(175, 300, 300, 50);
18        midPanel.add(bService);
19
20        bCustomer = new JButton("Klient");
21        bCustomer.setBounds(175, 400, 300, 50);
22        midPanel.add(bCustomer);
23
24        bClose = new JButton("Zamknij");
25        bClose.setBounds(500, 550, 100, 50);
26        midPanel.add(bClose);
27
28        /*bReturn = new JButton("Powrót");
29        bReturn.setBounds(0, 550, 100, 50);*/
30
31        bService.addActionListener(this);
32        bCustomer.addActionListener(this);
33        bClose.addActionListener(this);
34        //bReturn.addActionListener(this);
35    }
36
37    @Override
38    public void actionPerformed(ActionEvent e) {
39        Object src = e.getSource();
40        if (src == bService) {
41            dispose();
42            GuiServLog serviceLogin = new GuiServLog();
43        }
44    }
45}
```

```

35         bClose.addActionListener(this);
36         //bReturn.addActionListener(this);
37     }
38
39     @Override
40     public void actionPerformed(ActionEvent e) {
41         Object src = e.getSource();
42         if (src == bService) {
43             dispose();
44             GuiServLog serviceLogin = new GuiServLog();
45             serviceLogin.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
46             serviceLogin.setVisible(true);
47         } else if (src == bCustomer) {
48             dispose();
49             GuiCusLog customerLogin = new GuiCusLog();
50             customerLogin.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
51             customerLogin.setVisible(true);
52         } else if (src == bClose) {
53             dispose();
54         }
55     }
56 }

```

Klasa GuiCusLog

```

GuiCusLog.java
1 package atm_final2;
2
3 import javax.swing.*;
4
11
12 public class GuiCusLog extends Okno implements ActionListener {
13     private JButton bEnter, bClose, bReturn, bCancel, b0, b1, b2, b3, b4, b5, b6, b7, b8, b9;
14     private JLabel lNrKarty, lPin;
15     private JTextField tNrKarty;
16     private JPasswordField pPin;
17     private JRadioButton r1, r2;
18     private ButtonGroup group;
19     private String nrKarty = "", pin = "";
20     private Object[][] tab;
21     private int ile, index;
22
23     public GuiCusLog() {
24         setTitle("Logowanie");
25
26         try {
27             readFile();
28         } catch (IOException exception) {
29             exception.printStackTrace();
30         } catch (ClassNotFoundException exception) {
31             JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Klient logowanie", JOptionPane.INFORMATION_MESSAGE);
32             exception.printStackTrace();
33         } catch (NumberFormatException exception) {
34             JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Klient logowanie3", JOptionPane.INFORMATION_MESSAGE);
35         }
36
37         lNrKarty = new JLabel("Numer karty: ");
38         lNrKarty.setBounds(50, 100, 200, 40);
39         lNrKarty.setFont(font2);
40         midPanel.add(lNrKarty);
41
42         tNrKarty = new JTextField("");
43         tNrKarty.setBounds(250, 100, 200, 40);
44         midPanel.add(tNrKarty);
45
46         lPin = new JLabel("Pin: ");
47         lPin.setBounds(50, 200, 100, 40);
48         lPin.setFont(font2);

```

```
46     lPin = new JLabel("Pin: ");
47     lPin.setBounds(50, 200, 100, 40);
48     lPin.setFont(font2);
49     midPanel.add(lPin);
50
51     pPin = new JPasswordField("");
52     pPin.setBounds(250, 200, 100, 40);
53     midPanel.add(pPin);
54
55     b1 = new JButton("1");
56     b1.setBounds(225, 350, 50, 50);
57     midPanel.add(b1);
58     b2 = new JButton("2");
59     b2.setBounds(275, 350, 50, 50);
60     midPanel.add(b2);
61     b3 = new JButton("3");
62     b3.setBounds(325, 350, 50, 50);
63     midPanel.add(b3);
64     b4 = new JButton("4");
65     b4.setBounds(225, 400, 50, 50);
66     midPanel.add(b4);
67     b5 = new JButton("5");
68     b5.setBounds(275, 400, 50, 50);
69     midPanel.add(b5);
70     b6 = new JButton("6");
71     b6.setBounds(325, 400, 50, 50);
72     midPanel.add(b6);
73     b7 = new JButton("7");
74     b7.setBounds(225, 450, 50, 50);
75     midPanel.add(b7);
76     b8 = new JButton("8");
77     b8.setBounds(275, 450, 50, 50);
78     midPanel.add(b8);
79     b9 = new JButton("9");
80     b9.setBounds(325, 450, 50, 50);
81     midPanel.add(b9);
82     b0 = new JButton("0");
83     b0.setBounds(275, 500, 50, 50);
84     midPanel.add(b0);
85     bEnter = new JButton("Enter");
86     bEnter.setBounds(225, 500, 100, 50);
```

```

GuiCusLog.java
80     b9.setBounds(325, 500, 100, 50);
81     midPanel.add(b9);
82     b0 = new JButton("0");
83     b0.setBounds(275, 500, 50, 50);
84     midPanel.add(b0);
85     bEnter = new JButton("Enter");
86     bEnter.setBounds(325, 500, 100, 50);
87     midPanel.add(bEnter);
88     bCancel = new JButton("Cofnij");
89     bCancel.setBounds(175, 500, 100, 50);
90     midPanel.add(bCancel);
91     bClose = new JButton("Zamknij");
92     bClose.setBounds(500, 550, 100, 50);
93     midPanel.add(bClose);
94     bReturn = new JButton("Powrót");
95     bReturn.setBounds(0, 550, 100, 50);
96     midPanel.add(bReturn);
97     r1 = new JRadioButton("nr karty");
98     r2 = new JRadioButton("pin");
99     r1.setBounds(450, 350, 80, 20);
100    r2.setBounds(450, 400, 80, 20);
101    r1.setBackground(Color.LIGHT_GRAY);
102    r2.setBackground(Color.LIGHT_GRAY);
103    group = new ButtonGroup();
104    group.add(r1);
105    group.add(r2);
106    midPanel.add(r1);
107    midPanel.add(r2);
108
109    b1.addActionListener(this);
110    b2.addActionListener(this);
111    b3.addActionListener(this);
112    b4.addActionListener(this);
113    b5.addActionListener(this);
114    b6.addActionListener(this);
115    b7.addActionListener(this);
116    b8.addActionListener(this);
117    b9.addActionListener(this);
118    b0.addActionListener(this);
119    bEnter.addActionListener(this);
120    bCancel.addActionListener(this);
121    bClose.addActionListener(this);

```

```

GuiCusLog.java
118    b0.addActionListener(this);
119    bEnter.addActionListener(this);
120    bCancel.addActionListener(this);
121    bClose.addActionListener(this);
122    bReturn.addActionListener(this);
123    r1.addActionListener(this);
124    r2.addActionListener(this);
125    }
126
127    @Override
128    public void actionPerformed(ActionEvent e) {
129        Object source = e.getSource();
130        if (source == b1) {
131            if (r1.isSelected()) {
132                nrKarty = nrKarty + "1";
133                tNrKarty.setText(nrKarty);
134            } else if (r2.isSelected()) {
135                pin = pin + "1";
136                pPin.setText(pin);
137            }
138        } else if (source == b2) {
139            if (r1.isSelected()) {
140                nrKarty = nrKarty + "2";
141                tNrKarty.setText(nrKarty);
142            } else if (r2.isSelected()) {
143                pin = pin + "2";
144                pPin.setText(pin);
145            }
146        } else if (source == b3) {
147            if (r1.isSelected()) {
148                nrKarty = nrKarty + "3";
149                tNrKarty.setText(nrKarty);
150            } else if (r2.isSelected()) {
151                pin = pin + "3";
152                pPin.setText(pin);
153            }
154        } else if (source == b4) {
155            if (r1.isSelected()) {
156                nrKarty = nrKarty + "4";
157                tNrKarty.setText(nrKarty);
158            } else if (r2.isSelected()) {

```

```

GuiCusLog.java
157         tNrKarty.setText(nrKarty);
158     } else if (r2.isSelected()) {
159         pin = pin + "4";
160         pPin.setText(pin);
161     }
162 } else if (source == b5) {
163     if (r1.isSelected()) {
164         nrKarty = nrKarty + "5";
165         tNrKarty.setText(nrKarty);
166     } else if (r2.isSelected()) {
167         pin = pin + "5";
168         pPin.setText(pin);
169     }
170 } else if (source == b6) {
171     if (r1.isSelected()) {
172         nrKarty = nrKarty + "6";
173         tNrKarty.setText(nrKarty);
174     } else if (r2.isSelected()) {
175         pin = pin + "6";
176         pPin.setText(pin);
177     }
178 } else if (source == b7) {
179     if (r1.isSelected()) {
180         nrKarty = nrKarty + "7";
181         tNrKarty.setText(nrKarty);
182     } else if (r2.isSelected()) {
183         pin = pin + "7";
184         pPin.setText(pin);
185     }
186 } else if (source == b8) {
187     if (r1.isSelected()) {
188         nrKarty = nrKarty + "8";
189         tNrKarty.setText(nrKarty);
190     } else if (r2.isSelected()) {
191         pin = pin + "8";
192         pPin.setText(pin);
193     }
194 } else if (source == b9) {
195     if (r1.isSelected()) {
196         nrKarty = nrKarty + "9";
197         tNrKarty.setText(nrKarty);
198     } else if (r2.isSelected()) {
199         pin = pin + "9";
200         pPin.setText(pin);
201     }
202 } else if (source == b0) {
203     if (r1.isSelected()) {
204         nrKarty = nrKarty + "0";
205         tNrKarty.setText(nrKarty);
206     } else if (r2.isSelected()) {
207         pin = pin + "0";
208         pPin.setText(pin);
209     }
210 } else if (source == bCancel) {
211     if (r1.isSelected()) {
212         if (nrKarty.length() > 0) {
213             String pom = "";
214             for (int i = 0; i < nrKarty.length() - 1; i++) {
215                 pom = pom + nrKarty.charAt(i);
216             }
217             nrKarty = pom;
218             tNrKarty.setText(nrKarty);
219         }
220     } else if (r2.isSelected()) {
221         if (pin.length() > 0) {
222             String pom = "";
223             for (int i = 0; i < pin.length() - 1; i++) {
224                 pom = pom + pin.charAt(i);
225             }
226             pin = pom;
227             pPin.setText(pin);
228         }
229     }
230 } else if (source == bEnter) {
231     try {
232         readFile();
233     } catch (IOException exception) {
234         exception.printStackTrace();
235     }

```

```

GuiCusLog.java
194     } else if (source == b9) {
195         if (r1.isSelected()) {
196             nrKarty = nrKarty + "9";
197             tNrKarty.setText(nrKarty);
198         } else if (r2.isSelected()) {
199             pin = pin + "9";
200             pPin.setText(pin);
201         }
202     } else if (source == b0) {
203         if (r1.isSelected()) {
204             nrKarty = nrKarty + "0";
205             tNrKarty.setText(nrKarty);
206         } else if (r2.isSelected()) {
207             pin = pin + "0";
208             pPin.setText(pin);
209         }
210     } else if (source == bCancel) {
211         if (r1.isSelected()) {
212             if (nrKarty.length() > 0) {
213                 String pom = "";
214                 for (int i = 0; i < nrKarty.length() - 1; i++) {
215                     pom = pom + nrKarty.charAt(i);
216                 }
217                 nrKarty = pom;
218                 tNrKarty.setText(nrKarty);
219             }
220         } else if (r2.isSelected()) {
221             if (pin.length() > 0) {
222                 String pom = "";
223                 for (int i = 0; i < pin.length() - 1; i++) {
224                     pom = pom + pin.charAt(i);
225                 }
226                 pin = pom;
227                 pPin.setText(pin);
228             }
229         }
230     } else if (source == bEnter) {
231         try {
232             readFile();
233         } catch (IOException exception) {
234             exception.printStackTrace();
235         }

```

```

GuiCusLog.java
230     } else if (source == bEnter) {
231         try {
232             readFile();
233         } catch (IOException exception) {
234             exception.printStackTrace();
235         } catch (ClassNotFoundException exception) {
236             JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
237             exception.printStackTrace();
238         } catch (NumberFormatException exception) {
239             JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
240         }
241         if (weryfikacja()) {
242             System.out.println(pin + " Po weryfikacji");
243             dispose();
244             GuiCusMenu cusMenu = new GuiCusMenu(index, pin);
245             cusMenu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
246             cusMenu.setVisible(true);
247         } else {
248             JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
249         }
250     } else if (source == bReturn) {
251         dispose();
252         GuiEntry entry = new GuiEntry();
253         entry.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
254         entry.setVisible(true);
255     } else if (source == bClose) {
256         dispose();
257     }
258 }
259
260 public void readFile() throws IOException, ClassNotFoundException {
261     Object[][] obj = new Object[50][7];
262     ObjectInputStream ois = null;
263     Klient cus = null;
264     int count = 0;
265     try {
266         ois = new ObjectInputStream(new FileInputStream("klient.txt"));
267         while (true) {
268             cus = (Klient) ois.readObject();
269             obj[count][0] = cus.getImie();
270             obj[count][1] = cus.getNazwisko();
271         }
272     }

```

```

GuiCusLog.java
264     int count = 0;
265     try {
266         ois = new ObjectInputStream(new FileInputStream("klient.txt"));
267         while (true) {
268             cus = (Klient) ois.readObject();
269             obj[count][0] = cus.getImie();
270             obj[count][1] = cus.getNazwisko();
271             obj[count][2] = cus.getBank();
272             obj[count][3] = cus.getNrKarty();
273             obj[count][4] = cus.getPin();
274             obj[count][5] = cus.getSrodki();
275             obj[count][6] = cus.getTele();
276             count++;
277         }
278     } catch (EOFException ex) {
279     } finally {
280         if (ois != null)
281             ois.close();
282     }
283     setTab(obj);
284     setIle(count);
285
286     for (int i = 0; i < count; i++) {
287         for (int j = 0; j < 6; j++) {
288             System.out.print(tab[i][j] + " ");
289         }
290         System.out.println();
291     }
292 }
293
294 public void setTab(Object[][] tab) {
295     this.tab = tab;
296 }
297
298 public void setIle(int ile) {
299     this.ile = ile;
300 }
301
302 public void setIndex(int index) {
303     this.index = index;
304 }
305

```

```

302 public void setIndex(int index) {
303     this.index = index;
304 }
305
306 public void setPin(String pin) {
307     this.pin = pin;
308 }
309
310 public boolean weryfikacja() {
311     String nrkarty = tNrKarty.getText();
312     String pin = pPin.getText();
313     for (int i = 0; i < ile; i++) {
314         if ((tab[i][3].equals(nrkarty)) && (tab[i][4].equals(pin))) {
315             setIndex(i);
316             setPin(pin);
317             return true;
318         }
319     }
320     return false;
321 }
322 }

```

Klasa GuiCusMenu

```

GuiCusMenu.java
1 package atm_final2;
2
3 import javax.swing.*;
4
5 public class GuiCusMenu extends Okno implements ActionListener, IBankomat {
6     private JLabel lDane, lImie1, lNazwisko1, lBan1, lNrKarty1, lSrodki1, lImie2, lNazwisko2, lBank2, lNrKarty2,
7         lSrodki2;
8     private JButton bReturn, bClose, bWplata, bWyplata, bDoladowanie;
9     private Object[][] tab;
10    private int index;
11    private String pin;
12
13    public GuiCusMenu(int index, String pin) {
14        this.index = index;
15        this.pin = pin;
16        setTitle("Menu Klienta");
17
18        try {
19            odczyt();
20        } catch (IOException exception) {
21            exception.printStackTrace();
22        } catch (ClassNotFoundException exception) {
23            JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
24            exception.printStackTrace();
25        } catch (NumberFormatException exception) {
26            JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
27        }
28
29        lDane = new JLabel("Dane właściciela: ");
30        lDane.setBounds(50, 80, 300, 50);
31        lDane.setFont(font2);
32        midPanel.add(lDane);
33
34        lImie1 = new JLabel("Imie: ");
35        lImie1.setBounds(50, 150, 200, 40);
36        lImie1.setFont(font2);
37        midPanel.add(lImie1);
38
39        String s1 = String.valueOf(tab[this.index][0]);
40        lImie2 = new JLabel(s1);
41        lImie2.setBounds(250, 150, 200, 40);
42    }
43
44
45

```


GuiCusMenu.java

```
43 String s1 = String.valueOf(tab[this.index][0]);
44 lImie2 = new JLabel(s1);
45 lImie2.setBounds(250, 150, 200, 40);
46 lImie2.setFont(font2);
47 midPanel.add(lImie2);
48
49 lNazwisko1 = new JLabel("Nazwisko: ");
50 lNazwisko1.setBounds(50, 200, 200, 40);
51 lNazwisko1.setFont(font2);
52 midPanel.add(lNazwisko1);
53
54 s1 = String.valueOf(tab[this.index][1]);
55 lNazwisko2 = new JLabel(s1);
56 lNazwisko2.setBounds(250, 200, 200, 40);
57 lNazwisko2.setFont(font2);
58 midPanel.add(lNazwisko2);
59
60 lBan11 = new JLabel("Bank: ");
61 lBan11.setBounds(50, 250, 200, 40);
62 lBan11.setFont(font2);
63 midPanel.add(lBan11);
64
65 s1 = String.valueOf(tab[this.index][2]);
66 lBank2 = new JLabel(s1);
67 lBank2.setBounds(250, 250, 200, 40);
68 lBank2.setFont(font2);
69 midPanel.add(lBank2);
70
71 lNrKarty1 = new JLabel("Nr karty: ");
72 lNrKarty1.setBounds(50, 300, 200, 40);
73 lNrKarty1.setFont(font2);
74 midPanel.add(lNrKarty1);
75
76 s1 = String.valueOf(tab[this.index][3]);
77 lNrKarty2 = new JLabel(s1);
78 lNrKarty2.setBounds(250, 300, 200, 40);
79 lNrKarty2.setFont(font2);
80 midPanel.add(lNrKarty2);
81
82 lSrodki1 = new JLabel("Środki: ");
83 lSrodki1.setBounds(50, 350, 200, 40);
```

GuiCusMenu.java

```

82     lSrodki1 = new JLabel("Środki: ");
83     lSrodki1.setBounds(50, 350, 200, 40);
84     lSrodki1.setFont(font2);
85     midPanel.add(lSrodki1);
86
87     s1 = String.valueOf(tab[this.index][5]);
88     lSrodki2 = new JLabel(s1);
89     lSrodki2.setBounds(250, 350, 200, 40);
90     lSrodki2.setFont(font2);
91     midPanel.add(lSrodki2);
92
93     bReturn = new JButton("Cofnij");
94     bReturn.setBounds(0, 550, 100, 50);
95     midPanel.add(bReturn);
96
97     bClose = new JButton("Zamknij");
98     bClose.setBounds(500, 550, 100, 50);
99     midPanel.add(bClose);
100
101     bWplata = new JButton("Wpłata gotówki");
102     bWplata.setBounds(20, 400, 280, 50);
103     midPanel.add(bWplata);
104
105     bWyplata = new JButton("Wypłata gotówki");
106     bWyplata.setBounds(20, 460, 280, 50);
107     midPanel.add(bWyplata);
108
109     bDoladowanie = new JButton("Doładowanie telefonu");
110     bDoladowanie.setBounds(310, 460, 280, 50);
111     midPanel.add(bDoladowanie);
112
113     bReturn.addActionListener(this);
114     bWplata.addActionListener(this);
115     bWyplata.addActionListener(this);
116     bClose.addActionListener(this);
117     bDoladowanie.addActionListener(this);
118 }
119
120 @Override
121 public void actionPerformed(ActionEvent e) {
122     Object source = e.getSource();

```

GuiCusMenu.java

```

120 @Override
121 public void actionPerformed(ActionEvent e) {
122     Object source = e.getSource();
123     if (source == bReturn) {
124         dispose();
125         GuiCusLog customerLogin = new GuiCusLog();
126         customerLogin.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
127         customerLogin.setVisible(true);
128     } else if (source == bClose) {
129         dispose();
130     } else if (source == bWplata) {
131         dispose();
132         GuiCusCashIn cashIn = new GuiCusCashIn(index, pin);
133         cashIn.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
134         cashIn.setVisible(true);
135     } else if (source == bWyplata) {
136         dispose();
137         GuiCusCashOut cashOut = new GuiCusCashOut(index, pin);
138         cashOut.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
139         cashOut.setVisible(true);
140
141     }
142     else if (source == bDoladowanie) {
143         GuiCusMobile mobile = new GuiCusMobile(index, pin);
144         mobile.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
145         mobile.setVisible(true);
146     }
147 }
148
149
150 public void odczyt() throws IOException, ClassNotFoundException {
151     Object[][] arr = new Object[50][7];
152     ObjectInputStream ois = null;
153     Klient cus = null;
154     int count = 0;
155     try {
156         ois = new ObjectInputStream(new FileInputStream("klient.txt"));
157         while (true) {
158             cus = (Klient) ois.readObject();
159             arr[count][0] = cus.getImie();
160             arr[count][1] = cus.getNazwisko();

```

```
146     }
147
148 }
149
150 public void odczyt() throws IOException, ClassNotFoundException {
151     Object[][] arr = new Object[50][7];
152     ObjectInputStream ois = null;
153     Klient cus = null;
154     int count = 0;
155     try {
156         ois = new ObjectInputStream(new FileInputStream("klient.txt"));
157         while (true) {
158             cus = (Klient) ois.readObject();
159             arr[count][0] = cus.getImie();
160             arr[count][1] = cus.getNazwisko();
161             arr[count][2] = cus.getBank();
162             arr[count][3] = cus.getNrKarty();
163             arr[count][4] = cus.getPin();
164             arr[count][5] = cus.getSrodki();
165             arr[count][6] = cus.getTele();
166             count++;
167         }
168     } catch (EOFException ex) {
169     } finally {
170         if (ois != null)
171             ois.close();
172     }
173     setTab(arr);
174     for (int i = 0; i < count; i++) {
175         for (int j = 0; j < 7; j++) {
176             System.out.print(tab[i][j] + " ");
177         }
178         System.out.println();
179     }
180 }
181
182 public void setTab(Object[][] tab) {
183     this.tab = tab;
184 }
185
186 }
```

Klasa GuiCusCashIn

```
GuiCusCashIn.java
1 package atm_final2;
2
3 import javax.swing.*;
4
5 public class GuiCusCashIn extends Okno2 implements ActionListener, IBankomat {
6     private JLabel lSuma;
7     private JTextField tf500, tf200, tf100, tf50, tf20, tf10;
8     private JButton bWplac, bSuma; //, bClose, bReturn;
9     private Bankomat bankomat;
10    private int karta;
11    private String pin;
12
13    public GuiCusCashIn(int karta, String pin) {
14        this.pin = pin;
15        this.karta = karta;
16        JLabel lInf = new JLabel("Wprowadzanie banknotów: ");
17        lInf.setBounds(200, 20, 300, 40);
18        lInf.setFont(font2);
19        midPanel.add(lInf);
20
21        tf500 = new JTextField("0");
22        tf500.setBounds(350, 100, 150, 40);
23        midPanel.add(tf500);
24
25        tf200 = new JTextField("0");
26        tf200.setBounds(350, 150, 150, 40);
27        midPanel.add(tf200);
28
29        tf100 = new JTextField("0");
30        tf100.setBounds(350, 200, 150, 40);
31        midPanel.add(tf100);
32
33        tf50 = new JTextField("0");
34        tf50.setBounds(350, 250, 150, 40);
35        midPanel.add(tf50);
36
37        tf20 = new JTextField("0");
38        tf20.setBounds(350, 300, 150, 40);
39        midPanel.add(tf20);
40
41        tf10 = new JTextField("0");
42
43    }
44
45    tf10 = new JTextField("0");
46    tf10.setBounds(350, 350, 150, 40);
47    midPanel.add(tf10);
48
49    lSuma = new JLabel("");
50    lSuma.setBounds(375, 450, 225, 50);
51    lSuma.setFont(font2);
52    midPanel.add(lSuma);
53
54    bSuma = new JButton("Sumuj");
55    bSuma.setBounds(250, 400, 100, 50);
56    midPanel.add(bSuma);
57
58    bWplac = new JButton("Wpłać");
59    bWplac.setBounds(250, 500, 100, 50);
60    midPanel.add(bWplac);
61
62    /*bClose = new JButton("Zamknij");
63    bClose.setBounds(500, 550, 100, 50);
64    midPanel.add(bClose);
65
66    bReturn = new JButton("Powrót");
67    bReturn.setBounds(0, 550, 100, 50);
68    midPanel.add(bReturn);
69 */
70    bClose.addActionListener(this);
71    bReturn.addActionListener(this);
72    bWplac.addActionListener(this);
73    bSuma.addActionListener(this);
74 }
75
76 @Override
77 public void actionPerformed(ActionEvent e) {
78     Object source = e.getSource();
79     if (source == bReturn) {
80         dispose();
81         GuiCusMenu oknoMenuKlienta = new GuiCusMenu(karta, pin);
82         oknoMenuKlienta.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
83         oknoMenuKlienta.setVisible(true);
84     } else if (source == bWplac) {
85         String s500 = tf500.getText();
86     }
87 }
```

```

GuiCusCashIn.java
83 OknoMenuKlienta.setVisible(true);
84 } else if (source == bWpLac) {
85     String s500 = tf500.getText();
86     String s200 = tf200.getText();
87     String s100 = tf100.getText();
88     String s50 = tf50.getText();
89     String s20 = tf20.getText();
90     String s10 = tf10.getText();
91     try {
92         int i500 = Integer.parseInt(s500);
93         int i200 = Integer.parseInt(s200);
94         int i100 = Integer.parseInt(s100);
95         int i50 = Integer.parseInt(s50);
96         int i20 = Integer.parseInt(s20);
97         int i10 = Integer.parseInt(s10);
98
99         int suma = i500 * 500 + i200 * 200 + i100 * 100 + i50 * 50 + i20 * 20 + i10 * 10;
100         int kwota = Integer.parseInt(String.valueOf(suma));
101         zapisz(kwota, i500, i200, i100, i50, i20, i10);
102
103         this.bankomat = new Bankomat();
104         this.bankomat.fromFile();
105         this.bankomat.set500(this.bankomat.get500() + i500);
106         this.bankomat.set200(this.bankomat.get200() + i200);
107         this.bankomat.set100(this.bankomat.get100() + i100);
108         this.bankomat.set50(this.bankomat.get50() + i50);
109         this.bankomat.set20(this.bankomat.get20() + i20);
110         this.bankomat.set10(this.bankomat.get10() + i10);
111         this.bankomat.setSuma();
112         this.bankomat.toFile();
113         JOptionPane.showMessageDialog(null, "Dokonano wpłaty", "Wpłata", JOptionPane.INFORMATION_MESSAGE);
114     } catch (IOException exception) {
115         exception.printStackTrace();
116     } catch (ClassNotFoundException exception) {
117         JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wpłata", JOptionPane.INFORMATION_MESSAGE);
118         exception.printStackTrace();
119     } catch (NumberFormatException exception) {
120         JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wpłata", JOptionPane.INFORMATION_MESSAGE);
121         System.out.println("Nieprawidłowa kwota");
122     }
123 } else if (source == bSuma) {
124     System.out.println("Nieprawidłowa kwota");
125 }
126 } else if (source == bSuma) {
127     suma();
128 } else if (source == bClose) {
129     dispose();
130 }
131 }
132
133 public void suma() {
134     String s500 = tf500.getText();
135     String s200 = tf200.getText();
136     String s100 = tf100.getText();
137     String s50 = tf50.getText();
138     String s20 = tf20.getText();
139     String s10 = tf10.getText();
140     try {
141         int i500 = Integer.parseInt(s500);
142         int i200 = Integer.parseInt(s200);
143         int i100 = Integer.parseInt(s100);
144         int i50 = Integer.parseInt(s50);
145         int i20 = Integer.parseInt(s20);
146         int i10 = Integer.parseInt(s10);
147         int suma = i500 * 500 + i200 * 200 + i100 * 100 + i50 * 50 + i20 * 20 + i10 * 10;
148         lSuma.setText(String.valueOf(suma));
149     } catch (NumberFormatException exception) {
150         JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Sumowanie", JOptionPane.INFORMATION_MESSAGE);
151         System.out.println("Nieprawidłowa kwota");
152     }
153 }
154
155 public void zapisz(int kwota, int i500, int i200, int i100, int i50, int i20, int i10) throws IOException, ClassNotFoundException {
156     Klient[] arr = new Klient[100];
157     ObjectInputStream ois = null;
158     int count = 0;
159     try {
160         ois = new ObjectInputStream(new FileInputStream("klient.txt"));
161         while (true) {
162             arr[count] = (Klient) ois.readObject();
163             count++;
164         }
165     } catch (EOFException ex) {
166         if (ois != null)
167             ois.close();
168     }
169
170     ObjectOutputStream oos = null;
171     arr[this.karta].setSrodki(arr[this.karta].getSrodki() + kwota);
172     Bankomat bankomat = new Bankomat();
173     bankomat.WpLata(arr[this.karta], kwota, i500, i200, i100, i50, i20, i10);
174     try {
175         oos = new ObjectOutputStream(new FileOutputStream("klient.txt"));
176         for (int i = 0; arr[i] != null; i++) {
177             oos.writeObject(arr[i]);
178         }
179         oos.flush();
180     } finally {
181         if (oos != null)
182             oos.close();
183     }
184 } catch (IOException e) {
185     System.out.println("OknoWpLata\\zapisz\\IOException");
186 }
187 }
188 }
189 }
190 }

```

Klasa GuiCusCashOut

```
GuiCusCashOut.java
1 package atm_final2;
2
3 import javax.swing.*;
4
5
6
7
8
9 public class GuiCusCashOut extends Okno implements ActionListener, IBankomat{
10     private JLabel label1, label2;
11     private JTextField tfwyplata;
12     private JButton bWypłac, bReturn, bClose;
13     private int karta;
14     private String pin;
15
16     public GuiCusCashOut(int karta, String pin) {
17         this.pin = pin;
18         this.karta = karta;
19         label1 = new JLabel("Wypłata: ");
20         label1.setBounds(250, 50, 250, 40);
21         label1.setFont(font2);
22         midPanel.add(label1);
23
24         label2 = new JLabel("Wprowadź kwotę: ");
25         label2.setBounds(100, 250, 250, 40);
26         label2.setFont(font2);
27         midPanel.add(label2);
28
29         tfwyplata = new JTextField("");
30         tfwyplata.setBounds(350, 250, 150, 40);
31         midPanel.add(tfwyplata);
32
33         bWypłac = new JButton("Wypłać");
34         bWypłac.setBounds(250, 400, 100, 50);
35         midPanel.add(bWypłac);
36
37         bReturn = new JButton("Powrót");
38         bReturn.setBounds(0, 550, 100, 50);
39         midPanel.add(bReturn);
40
41         bClose = new JButton("Zamknij");
42         bClose.setBounds(500, 550, 100, 50);
43
44         bWypłac.addActionListener(this);
45         bReturn.addActionListener(this);
```

```

GuiCusCashOut.java
44     bWyplac.addActionListener(this);
45     bReturn.addActionListener(this);
46     bClose.addActionListener(this);
47 }
48
49 @Override
50 public void actionPerformed(ActionEvent e) {
51     Object source = e.getSource();
52     if (source == bWyplac) {
53         String s1 = tfwyplata.getText();
54         try {
55             int kwotaInt = Integer.parseInt(s1);
56             if (kwotaInt % 10 == 0) {
57                 int kwotaFloat = Integer.parseInt(s1);
58                 zapisz(kwotaFloat);
59             } else {
60                 JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
61             }
62         } catch (NumberFormatException exception) {
63             JOptionPane.showMessageDialog(null, "Nieprawidłowa kwota", "Wynik", JOptionPane.INFORMATION_MESSAGE);
64         } catch (IOException exception) {
65             exception.printStackTrace();
66         } catch (ClassNotFoundException exception) {
67             JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
68             exception.printStackTrace();
69         }
70     } else if (source == bReturn) {
71         dispose();
72         GuiCusMenu oknoMenuKlienta = new GuiCusMenu(karta, pin);
73         oknoMenuKlienta.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
74         oknoMenuKlienta.setVisible(true);
75     } else if (source == bClose) {
76         dispose();
77     }
78 }
79
80 public void zapisz(int kwota) throws IOException, ClassNotFoundException {
81     Klient[] arr = new Klient[100];
82     ObjectInputStream ois = null;
83     int count = 0;
84     try {
85         try {
86             ois = new ObjectInputStream(new FileInputStream("klient.txt"));
87             while (true) {
88                 arr[count] = (Klient) ois.readObject();
89                 count++;
90             }
91         } catch (EOFException ex) {
92             if (ois != null)
93                 ois.close();
94         }
95         try {
96             Bankomat bankomat = new Bankomat();
97             bankomat.fromFile();
98
99             if ((arr[this.karta].getSrodki() >= kwota) && (bankomat.getSuma() >= kwota)) {
100                 //tutaj dodac zmienianie sie liczby banknotow w bankomacie
101                 arr[this.karta].setSrodki(arr[this.karta].getSrodki() - kwota);
102                 bankomat.IWyplata(arr[this.karta], kwota);
103             }
104         } catch (NumberFormatException exception) {
105             JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
106             System.out.println("Nieprawidłowa kwota");
107         }
108
109         ObjectOutputStream oos = null;
110         try {
111             oos = new ObjectOutputStream(new FileOutputStream("klient.txt"));
112             for (int i = 0; arr[i] != null; i++) {
113                 oos.writeObject(arr[i]);
114             }
115             oos.flush();
116         } finally {
117             if (oos != null)
118                 oos.close();
119         }
120     } catch (IOException e) {
121         System.out.println("OknoWyplata//zapisz//IOException");
122     }
123 }
124 }

```

Klasa GuiCusMobile

```
GuiCusMobile.java
1 package atm_final2;
2
3 import javax.swing.*;
4
5
6
7
8
9 public class GuiCusMobile extends Okno implements ActionListener {
10     private JLabel lInf, lOperator, lkwota;
11     private JRadioButton r1, r2, r3, r4, r5, r6, r7, r8;
12     private ButtonGroup group1, group2;
13     private JButton bRefill, bClose, bReturn;
14     private int index;
15     private String pin;
16
17     public GuiCusMobile(int index, String pin) {
18         this.index = index;
19         this.pin = pin;
20         lInf = new JLabel("Dładowanie telefonu: ");
21         lInf.setBounds(200, 20, 300, 40);
22         lInf.setFont(font2);
23         midPanel.add(lInf);
24
25         lOperator = new JLabel("Wybierz operatora: ");
26         lOperator.setBounds(50, 100, 300, 40);
27         lOperator.setFont(font2);
28         midPanel.add(lOperator);
29
30         r1 = new JRadioButton("Orange");
31         r1.setBounds(50, 150, 120, 40);
32         r1.setFont(font2);
33         r1.setBackground(Color.yellow);
34
35         r2 = new JRadioButton("Plus");
36         r2.setBounds(170, 150, 120, 40);
37         r2.setFont(font2);
38         r2.setBackground(Color.yellow);
39
40         r3 = new JRadioButton("T-Mobile");
41         r3.setBounds(290, 150, 120, 40);
42         r3.setFont(font2);
43         r3.setBackground(Color.yellow);
44
45         r4 = new JRadioButton("Play");
```



```

GuiCusMobile.java
45     r4 = new JRadioButton("Play");
46     r4.setBounds(410, 150, 120, 40);
47     r4.setFont(font2);
48     r4.setBackground(Color.yellow);
49
50     group1 = new ButtonGroup();
51     group1.add(r1);
52     group1.add(r2);
53     group1.add(r3);
54     group1.add(r4);
55     midPanel.add(r1);
56     midPanel.add(r2);
57     midPanel.add(r3);
58     midPanel.add(r4);
59
60     lkwota = new JLabel("Wybierz kwotę doładowania: ");
61     lkwota.setFont(font2);
62     lkwota.setBounds(50, 200, 300, 40);
63     midPanel.add(lkwota);
64
65     r5 = new JRadioButton("10 PLN");
66     r5.setBounds(50, 250, 120, 40);
67     r5.setFont(font2);
68     r5.setBackground(Color.yellow);
69
70     r6 = new JRadioButton("25 PLN");
71     r6.setBounds(170, 250, 120, 40);
72     r6.setFont(font2);
73     r6.setBackground(Color.yellow);
74
75     r7 = new JRadioButton("50 PLN");
76     r7.setBounds(290, 250, 120, 40);
77     r7.setFont(font2);
78     r7.setBackground(Color.yellow);
79
80     r8 = new JRadioButton("100 PLN");
81     r8.setBounds(410, 250, 120, 40);
82     r8.setFont(font2);
83     r8.setBackground(Color.yellow);
84
85     group2 = new ButtonGroup();

```

```

GuiCusMobile.java
83     r8.setBackground(Color.yellow);
84
85     group2 = new ButtonGroup();
86     group2.add(r5);
87     group2.add(r6);
88     group2.add(r7);
89     group2.add(r8);
90     midPanel.add(r5);
91     midPanel.add(r6);
92     midPanel.add(r7);
93     midPanel.add(r8);
94
95     bRefill = new JButton("Doładuj");
96     bRefill.setBounds(250, 350, 100, 50);
97     midPanel.add(bRefill);
98
99     bClose = new JButton("Zamknij");
100    bClose.setBounds(0, 550, 100, 50);
101    midPanel.add(bClose);
102
103    bReturn = new JButton("Powrót");
104    bReturn.setBounds(500, 550, 100, 50);
105    midPanel.add(bReturn);
106
107    bClose.addActionListener(this);
108    bRefill.addActionListener(this);
109    bReturn.addActionListener(this);
110 }
111
112 @Override
113 public void actionPerformed(ActionEvent e) {
114     Object source = e.getSource();
115     if (source == bReturn) {
116         dispose();
117         GuiCusMenu cusLogin = new GuiCusMenu(index, pin);
118         cusLogin.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
119         cusLogin.setVisible(true);
120     } else if (source == bRefill) {
121         if ((r1.isSelected() || r2.isSelected() || r3.isSelected() || r4.isSelected())
122             && (r5.isSelected() || r6.isSelected() || r7.isSelected() || r8.isSelected())) {
123             float kwota;

```

```

GuiCusMobile.java
119     cusLogin.setvisioie(true);
120 } else if (source == bRefill) {
121     if ((r1.isSelected() || r2.isSelected() || r3.isSelected() || r4.isSelected())
122         && (r5.isSelected() || r6.isSelected() || r7.isSelected() || r8.isSelected())) {
123         float kwota;
124         if (r5.isSelected())
125             kwota = 10;
126         else if (r6.isSelected())
127             kwota = 25;
128         else if (r7.isSelected())
129             kwota = 50;
130         else
131             kwota = 100;
132
133         try {
134             zapisz(kwota);
135         } catch (IOException exception) {
136             exception.printStackTrace();
137         } catch (ClassNotFoundException exception) {
138             JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Doładowywanie telefonu", JOptionPane.INFORMATION_MESSAGE);
139             exception.printStackTrace();
140         } catch (NumberFormatException exception) {
141             JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Doładowywanie telefonu", JOptionPane.INFORMATION_MESSAGE);
142             System.out.println("Nieprawidłowa kwota");
143         }
144     }
145 } else if (source == bClose) {
146     dispose();
147 }
148 }
149
150 public void zapisz(float kwota) throws IOException, ClassNotFoundException {
151     Klient[] arr = new Klient[100];
152     ObjectInputStream ois = null;
153     int count = 0;
154     try {
155         ois = new ObjectInputStream(new FileInputStream("klient.txt"));
156         while (true) {
157             arr[count] = (Klient) ois.readObject();
158             count++;
159         }
160     } catch (EOFException ex) {
161         if (ois != null)
162             ois.close();
163
164         ObjectOutputStream oos = null;
165         if (arr[index].getSrodki() >= kwota) {
166             arr[this.index].setSrodki(arr[this.index].getSrodki() - kwota);
167             JOptionPane.showMessageDialog(null, "Doładowano telefon", "Wynik", JOptionPane.INFORMATION_MESSAGE);
168         } else
169             JOptionPane.showMessageDialog(null, "Niewystarczająca ilość środków", "Wynik",
170                 JOptionPane.INFORMATION_MESSAGE);
171
172         try {
173             oos = new ObjectOutputStream(new FileOutputStream("klient.txt"));
174             for (int i = 0; arr[i] != null; i++) {
175                 oos.writeObject(arr[i]);
176             }
177             oos.flush();
178         } finally {
179             if (oos != null)
180                 oos.close();
181         }
182     } catch (IOException e) {
183         System.out.println("OknoDoladTele\\zapisz\\IOException");
184     }
185 }
186

```

GuiServLog

```
GuiServLog.java
1 package atm_final2;
2
3 import javax.swing.*;
4
5
6
7
8 public class GuiServLog extends Okno implements ActionListener {
9     private JButton b0, b1, b2, b3, b4, b5, b6, b7, b8, b9;
10    private JButton bEnter, bCancel, bClose, bReturn;
11    private JTextField tfID;
12    private JPasswordField pfPin;
13    private JRadioButton r1, r2;
14    private ButtonGroup group;
15    private String identyfikator = "", pin = "";
16
17    public GuiServLog() {
18        setTitle("Serwis logowanie");
19
20        JLabel lSesja = new JLabel("Sesja pracownicza");
21        lSesja.setForeground(Color.red);
22        lSesja.setBounds(0, 0, 300, 50);
23        lSesja.setFont(font2);
24        topPanel.add(lSesja);
25
26        JLabel lIdentyfikator = new JLabel("Identyfikator: ");
27        lIdentyfikator.setBounds(30, 100, 250, 40);
28        lIdentyfikator.setFont(font2);
29        midPanel.add(lIdentyfikator);
30
31        tfID = new JTextField("");
32        tfID.setBounds(250, 100, 100, 40);
33        midPanel.add(tfID);
34
35        JLabel lPin = new JLabel("Pin: ");
36        lPin.setBounds(180, 200, 100, 40);
37        lPin.setFont(font2);
38        midPanel.add(lPin);
39
40        pfPin = new JPasswordField("");
41        pfPin.setBounds(250, 200, 100, 40);
42        midPanel.add(pfPin);
43
44        b1 = new JButton("1");
```

```
44     b1 = new JButton("1");
45     b1.setBounds(225, 350, 50, 50);
46     midPanel.add(b1);
47     b2 = new JButton("2");
48     b2.setBounds(275, 350, 50, 50);
49     midPanel.add(b2);
50     b3 = new JButton("3");
51     b3.setBounds(325, 350, 50, 50);
52     midPanel.add(b3);
53     b4 = new JButton("4");
54     b4.setBounds(225, 400, 50, 50);
55     midPanel.add(b4);
56     b5 = new JButton("5");
57     b5.setBounds(275, 400, 50, 50);
58     midPanel.add(b5);
59     b6 = new JButton("6");
60     b6.setBounds(325, 400, 50, 50);
61     midPanel.add(b6);
62     b7 = new JButton("7");
63     b7.setBounds(225, 450, 50, 50);
64     midPanel.add(b7);
65     b8 = new JButton("8");
66     b8.setBounds(275, 450, 50, 50);
67     midPanel.add(b8);
68     b9 = new JButton("9");
69     b9.setBounds(325, 450, 50, 50);
70     midPanel.add(b9);
71     b0 = new JButton("0");
72     b0.setBounds(275, 500, 50, 50);
73     midPanel.add(b0);
74     bEnter = new JButton("Enter");
75     bEnter.setBounds(325, 500, 100, 50);
76     midPanel.add(bEnter);
77     bCancel = new JButton("Cofnij");
78     bCancel.setBounds(175, 500, 100, 50);
79     midPanel.add(bCancel);
80     bClose = new JButton("Zamknij");
81     bClose.setBounds(500, 550, 100, 50);
82     midPanel.add(bClose);
83     bReturn = new JButton("Powrót");
84     bReturn.setBounds(0, 550, 100, 50);
```

```

GuiServLog.java
81     bClose.setBounds(500, 550, 100, 50);
82     midPanel.add(bClose);
83     bReturn = new JButton("Powrót");
84     bReturn.setBounds(0, 550, 100, 50);
85     midPanel.add(bReturn);
86     r1 = new JRadioButton("Nr Id");
87     r2 = new JRadioButton("PIN");
88     r1.setBounds(450, 350, 80, 20);
89     r2.setBounds(450, 400, 80, 20);
90     r1.setBackground(Color.GRAY);
91     r2.setBackground(Color.GRAY);
92     group = new ButtonGroup();
93     group.add(r1);
94     group.add(r2);
95     midPanel.add(r1);
96     midPanel.add(r2);
97
98     b1.addActionListener(this);
99     b2.addActionListener(this);
100    b3.addActionListener(this);
101    b4.addActionListener(this);
102    b5.addActionListener(this);
103    b6.addActionListener(this);
104    b7.addActionListener(this);
105    b8.addActionListener(this);
106    b9.addActionListener(this);
107    b0.addActionListener(this);
108    bEnter.addActionListener(this);
109    bCancel.addActionListener(this);
110    bClose.addActionListener(this);
111    bReturn.addActionListener(this);
112    r1.addActionListener(this);
113    r2.addActionListener(this);
114 }
115
116 @Override
117 public void actionPerformed(ActionEvent e) {
118     Object source = e.getSource();
119     if (source == b1) {
120         if (r1.isSelected()) {
121             identyfikator = identyfikator + "1":

```

```
117 public void actionPerformed(ActionEvent e) {
118     Object source = e.getSource();
119     if (source == b1) {
120         if (r1.isSelected()) {
121             identyfikator = identyfikator + "1";
122             tfID.setText(identyfikator);
123         } else if (r2.isSelected()) {
124             pin = pin + "1";
125             pfPin.setText(pin);
126         }
127     } else if (source == b2) {
128         if (r1.isSelected()) {
129             identyfikator = identyfikator + "2";
130             tfID.setText(identyfikator);
131         } else if (r2.isSelected()) {
132             pin = pin + "2";
133             pfPin.setText(pin);
134         }
135     } else if (source == b3) {
136         if (r1.isSelected()) {
137             identyfikator = identyfikator + "3";
138             tfID.setText(identyfikator);
139         } else if (r2.isSelected()) {
140             pin = pin + "3";
141             pfPin.setText(pin);
142         }
143     } else if (source == b4) {
144         if (r1.isSelected()) {
145             identyfikator = identyfikator + "4";
146             tfID.setText(identyfikator);
147         } else if (r2.isSelected()) {
148             pin = pin + "4";
149             pfPin.setText(pin);
150         }
151     } else if (source == b5) {
152         if (r1.isSelected()) {
153             identyfikator = identyfikator + "5";
154             tfID.setText(identyfikator);
155         } else if (r2.isSelected()) {
156             pin = pin + "5";
157             pfPin.setText(pin);
158         }
159     }
```

```

GuiServLog.java
155         } else if (r2.isSelected()) {
156             pin = pin + "5";
157             pfPin.setText(pin);
158         }
159     } else if (source == b6) {
160         if (r1.isSelected()) {
161             identyfikator = identyfikator + "6";
162             tfID.setText(identyfikator);
163         } else if (r2.isSelected()) {
164             pin = pin + "6";
165             pfPin.setText(pin);
166         }
167     } else if (source == b7) {
168         if (r1.isSelected()) {
169             identyfikator = identyfikator + "7";
170             tfID.setText(identyfikator);
171         } else if (r2.isSelected()) {
172             pin = pin + "7";
173             pfPin.setText(pin);
174         }
175     } else if (source == b8) {
176         if (r1.isSelected()) {
177             identyfikator = identyfikator + "8";
178             tfID.setText(identyfikator);
179         } else if (r2.isSelected()) {
180             pin = pin + "8";
181             pfPin.setText(pin);
182         }
183     } else if (source == b9) {
184         if (r1.isSelected()) {
185             identyfikator = identyfikator + "9";
186             tfID.setText(identyfikator);
187         } else if (r2.isSelected()) {
188             pin = pin + "9";
189             pfPin.setText(pin);
190         }
191     } else if (source == b0) {
192         if (r1.isSelected()) {
193             identyfikator = identyfikator + "0";
194             tfID.setText(identyfikator);
195         } else if (r2.isSelected()) {

```

```
196         pin = pin + "0";
197         pfPin.setText(pin);
198     }
199     } else if (source == bCancel) {
200         if (r1.isSelected()) {
201             if (identyfikator.length() > 0) {
202                 String pom = "";
203                 for (int i = 0; i < identyfikator.length() - 1; i++) {
204                     pom = pom + identyfikator.charAt(i);
205                 }
206                 identyfikator = pom;
207                 tfID.setText(identyfikator);
208             }
209         } else if (r2.isSelected()) {
210             if (pin.length() > 0) {
211                 String pom = "";
212                 for (int i = 0; i < pin.length() - 1; i++) {
213                     pom = pom + pin.charAt(i);
214                 }
215                 pin = pom;
216                 pfPin.setText(pin);
217             }
218         }
219     } else if (source == bClose) {
220         dispose();
221     } else if (source == bReturn) {
222         GuiEntry entry = new GuiEntry();
223         entry.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
224         entry.setVisible(true);
225     } else if (source == bEnter) {
226         String identyfikator = tfID.getText();
227         String pin = String.valueOf(pfPin.getPassword());
228         if (identyfikator.equals("") && pin.equals("")) {
229             dispose();
230             GuiServMenu service = new GuiServMenu();
231             service.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
232             service.setVisible(true);
233         }
234     }
235 }
236 }
```


Klasa GuiServMenu

```
GuiServMenu.java
1 package atm_final2;
2
3 import javax.swing.*;
4
5
6
7
8
9 public class GuiServMenu extends Okno implements ActionListener {
10     private JButton bAddCus, bCash, bRefill, bClose, bReturn;
11     private JLabel lSesja;
12
13     public GuiServMenu() {
14         setTitle("Sesja Pracownicza");
15
16         lSesja = new JLabel("Sesja pracownicza");
17         lSesja.setForeground(Color.red);
18         lSesja.setBounds(0, 0, 300, 50);
19         lSesja.setFont(font2);
20         topPanel.add(lSesja);
21
22         bAddCus = new JButton("Dodaj nowego klienta");
23         bAddCus.setBounds(175, 150, 275, 50);
24         midPanel.add(bAddCus);
25
26         bCash = new JButton("Wyświetl zawartość bankomatu");
27         bCash.setBounds(175, 250, 275, 50);
28         midPanel.add(bCash);
29
30         bRefill = new JButton("Uzupełnij bankomat");
31         bRefill.setBounds(175, 350, 275, 50);
32         midPanel.add(bRefill);
33
34         bClose = new JButton("Zamknij");
35         bClose.setBounds(500, 550, 100, 50);
36         midPanel.add(bClose);
37
38         bReturn = new JButton("Powrót");
39         bReturn.setBounds(0, 550, 100, 50);
40         midPanel.add(bReturn);
41
42         bAddCus.addActionListener(this);
43         bClose.addActionListener(this);
44         bCash.addActionListener(this);
45         bReturn.addActionListener(this);
46
47     }
48
49     @Override
50     public void actionPerformed(ActionEvent e) {
51         Object source = e.getSource();
52         if (source == bAddCus) {
53             dispose();
54             GuiServAdd addCus = new GuiServAdd();
55             addCus.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
56             addCus.setVisible(true);
57         } else if (source == bReturn) {
58             dispose();
59             GuiServLog serviceLogin = new GuiServLog();
60             serviceLogin.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
61             serviceLogin.setVisible(true);
62         } else if (source == bCash) {
63             dispose();
64             GuiServCash checkCash = null;
65             try {
66                 checkCash = new GuiServCash();
67             } catch (IOException exception) {
68                 exception.printStackTrace();
69             } catch (ClassNotFoundException exception) {
70                 exception.printStackTrace();
71             }
72             checkCash.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
73             checkCash.setVisible(true);
74         } else if (source == bRefill) {
75             dispose();
76             GuiServRefill refill = new GuiServRefill();
77             refill.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
78             refill.setVisible(true);
79         } else if (source == bClose) {
80             dispose();
81         }
82     }
83 }
84
```

Klasa GuiServAdd

```
GuiServAdd.java
1 package atm_final2;
2
3 import javax.swing.*;
4
5
6
7
8
9
10 public class GuiServAdd extends Okno implements ActionListener {
11     private JLabel lSesja, lImie, lNazwisko, lBank, lNrKarty, lPin, lKwota, lTele;
12     private JTextField tfImie, tfNazwisko, tfBank, tfNrKarty, tfPin, tfKwota, tfTele;
13     private JButton bSave, bClose, bReturn;
14
15     public GuiServAdd() {
16         lSesja = new JLabel("Sesja pracownicza");
17         lSesja.setForeground(Color.red);
18         lSesja.setBounds(0, 0, 300, 50);
19         lSesja.setFont(font2);
20         topPanel.add(lSesja);
21
22         lImie = new JLabel("Imie:");
23         lImie.setBounds(100, 100, 150, 40);
24         lImie.setFont(font2);
25         midPanel.add(lImie);
26
27         tfImie = new JTextField("");
28         tfImie.setBounds(300, 100, 150, 40);
29         midPanel.add(tfImie);
30
31         lNazwisko = new JLabel("Nazwisko:");
32         lNazwisko.setBounds(100, 150, 150, 40);
33         lNazwisko.setFont(font2);
34         midPanel.add(lNazwisko);
35
36         tfNazwisko = new JTextField("");
37         tfNazwisko.setBounds(300, 150, 150, 40);
38         midPanel.add(tfNazwisko);
39
40         lBank = new JLabel("Bank:");
41         lBank.setBounds(100, 200, 150, 40);
42         lBank.setFont(font2);
43         midPanel.add(lBank);
44
45         tfBank = new JTextField("");
46         tfBank.setBounds(300, 200, 150, 40);
```

```
GuiServAdd.java
45         tfBank = new JTextField("");
46         tfBank.setBounds(300, 200, 150, 40);
47         midPanel.add(tfBank);
48
49         lNrKarty = new JLabel("Nr karty:");
50         lNrKarty.setBounds(100, 250, 150, 40);
51         lNrKarty.setFont(font2);
52         midPanel.add(lNrKarty);
53
54         tfNrKarty = new JTextField("");
55         tfNrKarty.setBounds(300, 250, 150, 40);
56         midPanel.add(tfNrKarty);
57
58         lPin = new JLabel("Pin:");
59         lPin.setBounds(100, 300, 150, 40);
60         lPin.setFont(font2);
61         midPanel.add(lPin);
62
63         tfPin = new JTextField("");
64         tfPin.setBounds(300, 300, 150, 40);
65         midPanel.add(tfPin);
66
67         lTele = new JLabel("Tele:");
68         lTele.setBounds(100, 350, 150, 40);
69         lTele.setFont(font2);
70         midPanel.add(lTele);
71
72         tfTele = new JTextField("");
73         tfTele.setBounds(300, 350, 150, 40);
74         midPanel.add(tfTele);
75
76         lKwota = new JLabel("Kwota:");
77         lKwota.setBounds(100, 400, 150, 40);
78         lKwota.setFont(font2);
79         midPanel.add(lKwota);
80
81         tfKwota = new JTextField("");
82         tfKwota.setBounds(300, 400, 150, 40);
83         midPanel.add(tfKwota);
84
85         bSave = new JButton("Zapisz");
```

```

84
85         bSave = new JButton("Zapisz");
86         bSave.setBounds(225, 450, 150, 50);
87         midPanel.add(bSave);
88
89         bClose = new JButton("Zamknij");
90         bClose.setBounds(500, 550, 100, 50);
91         midPanel.add(bClose);
92
93         bReturn = new JButton("Powrót");
94         bReturn.setBounds(0, 550, 100, 50);
95         midPanel.add(bReturn);
96
97         bSave.addActionListener(this);
98         bClose.addActionListener(this);
99         bReturn.addActionListener(this);
100     }
101
102     @Override
103     public void actionPerformed(ActionEvent e) {
104         Object source = e.getSource();
105         if (source == bSave) {
106             String imie = tfImie.getText();
107             String nazwisko = tfNazwisko.getText();
108             String bank = tfBank.getText();
109             String nrKarty = tfNrKarty.getText();
110             String pin = tfPin.getText();
111             String tele = tfTele.getText();
112             String pom = tfKwota.getText();
113             float kwota = 0;
114             if ((imie.length() > 1) && (nazwisko.length() > 1) && (bank.length() > 1)
115                 && (nrKarty.length() > 1) && (pin.length() == 4) && (tele.length() > 7)) {
116                 try {
117                     kwota = Float.parseFloat(pom);
118                     DecimalFormat df = new DecimalFormat();
119                     df.setMaximumFractionDigits(2);
120                     df.format(kwota);
121                     Klient klient = new Klient(imie, nazwisko, bank, nrKarty, pin, tele, kwota);
122                     klient.addCus();
123                     JOptionPane.showMessageDialog(null, "Zapisano", "Dodawanie klienta", JOptionPane.INFORMATION_MESSAGE);
124                     klient.readCus();

```

```

123         JOptionPane.showMessageDialog(null, "Zapisano", "Dodawanie klienta", JOptionPane.INFORMATION_MESSAGE);
124         klient.readCus();
125     } catch (IOException exception) {
126         exception.printStackTrace();
127     } catch (ClassNotFoundException exception) {
128         JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Dodawanie klienta",
129             JOptionPane.INFORMATION_MESSAGE);
130         exception.printStackTrace();
131     } catch (NumberFormatException exception) {
132         JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Dodawanie klienta",
133             JOptionPane.INFORMATION_MESSAGE);
134         System.out.println("Nieprawidłowa kwota");
135     }
136 } else {
137     JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Dodawanie klienta", JOptionPane.INFORMATION_MESSAGE);
138 }
139 } else if (source == bReturn) {
140     dispose();
141     GuiServMenu service = new GuiServMenu();
142     service.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
143     service.setVisible(true);
144 } else if (source == bClose) {
145     dispose();
146 }
147 }
148 }

```

Klasa GuiServCash

```
GuiServCash.java
1 package atm_final2;
2
3 import javax.swing.*;
4
5 public class GuiServCash extends Okno2 implements ActionListener {
6     private JLabel lInf, l500, l200, l100, l50, l20, l10, lSuma, lSesja;
7     private Bankomat bankomat;
8     private String s1;
9
10    public GuiServCash() throws IOException, ClassNotFoundException {
11        this.bankomat = new Bankomat();
12        try {
13            bankomat.fromFile();
14        } catch (IOException exception) {
15            exception.printStackTrace();
16        } catch (ClassNotFoundException exception) {
17            JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
18            exception.printStackTrace();
19        } catch (NumberFormatException exception) {
20            JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
21        }
22        lSesja = new JLabel("Sesja pracownicza");
23        lSesja.setForeground(Color.red);
24        lSesja.setBounds(0, 0, 300, 50);
25        lSesja.setFont(font2);
26        topPanel.add(lSesja);
27
28        lInf = new JLabel("Liczba poszczególnych nominałów w bankomacie: ");
29        lInf.setBounds(0, 20, 600, 40);
30        lInf.setFont(font2);
31        midPanel.add(lInf);
32
33        s1 = String.valueOf(bankomat.get500());
34        l500 = new JLabel(s1);
35        l500.setBounds(350, 100, 150, 50);
36        l500.setFont(font2);
37        midPanel.add(l500);
38
39        s1 = String.valueOf(bankomat.get200());
40        l200 = new JLabel(s1);
41        l200.setBounds(350, 150, 150, 50);
42
43        s1 = String.valueOf(bankomat.get20());
44        l20 = new JLabel(s1);
45        l20.setBounds(350, 200, 150, 50);
```

```
GuiServCash.java
43        s1 = String.valueOf(bankomat.get200());
44        l200 = new JLabel(s1);
45        l200.setBounds(350, 150, 150, 50);
46        l200.setFont(font2);
47        midPanel.add(l200);
48
49        s1 = String.valueOf(bankomat.get100());
50        l100 = new JLabel(s1);
51        l100.setBounds(350, 200, 150, 50);
52        l100.setFont(font2);
53        midPanel.add(l100);
54
55        s1 = String.valueOf(bankomat.get50());
56        l50 = new JLabel(s1);
57        l50.setBounds(350, 250, 150, 50);
58        l50.setFont(font2);
59        midPanel.add(l50);
60
61        s1 = String.valueOf(bankomat.get20());
62        l20 = new JLabel(s1);
63        l20.setBounds(350, 300, 150, 50);
64        l20.setFont(font2);
65        midPanel.add(l20);
66
67        s1 = String.valueOf(bankomat.get10());
68        l10 = new JLabel(s1);
69        l10.setBounds(350, 350, 150, 50);
70        l10.setFont(font2);
71        midPanel.add(l10);
72
73        s1 = String.valueOf(bankomat.getSuma());
74        lSuma = new JLabel(s1);
75        lSuma.setBounds(375, 450, 150, 50);
76        lSuma.setFont(font2);
77        midPanel.add(lSuma);
78
79        bClose.addActionListener(this);
80        bReturn.addActionListener(this);
81    }
82
83    @Override
```

```

81     }
82
83     @Override
84     public void actionPerformed(ActionEvent e) {
85         Object zrodlo = e.getSource();
86         if (zrodlo == bReturn) {
87             dispose();
88             GuiServMenu oknoMenuPracownika = new GuiServMenu();
89             oknoMenuPracownika.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
90             oknoMenuPracownika.setVisible(true);
91         } else if (zrodlo == bClose) {
92             dispose();
93         }
94     }
95 }

```

Klasa GuiServRefill

```

GuiServRefill.java
1 package atm_final2;
2
3 import javax.swing.*.*;
4
5
6
7
8
9 public class GuiServRefill extends Okno2 implements ActionListener, IBankomat {
10     private JLabel lMain, lSesja, lSuma;
11     private JTextField t500, t200, t100, t50, t20, t10;
12     private JButton bDodaj, bSuma;
13     private Bankomat bankomat;
14
15     public GuiServRefill() {
16         setTitle("Uzupełnianie środków");
17
18         lSesja = new JLabel("Sesja pracownicza");
19         lSesja.setForeground(Color.red);
20         lSesja.setBounds(0, 0, 300, 50);
21         lSesja.setFont(font2);
22         topPanel.add(lSesja);
23
24         lMain = new JLabel("Uzupełnianie bankomatu: ");
25         lMain.setBounds(100, 20, 500, 40);
26         lMain.setFont(font2);
27         midPanel.add(lMain);
28
29         t500 = new JTextField("0");
30         t500.setBounds(350, 100, 150, 40);
31         midPanel.add(t500);
32
33         t200 = new JTextField("0");
34         t200.setBounds(350, 150, 150, 40);
35         midPanel.add(t200);
36
37         t100 = new JTextField("0");
38         t100.setBounds(350, 200, 150, 40);
39         midPanel.add(t100);
40
41         t50 = new JTextField("0");
42         t50.setBounds(350, 250, 150, 40);
43         midPanel.add(t50);
44
45         t20 = new JTextField("0");

```

GuiServRefill.java

```

45     t20 = new JTextField("0");
46     t20.setBounds(350, 300, 150, 40);
47     midPanel.add(t20);
48
49     t10 = new JTextField("0");
50     t10.setBounds(350, 350, 150, 40);
51     midPanel.add(t10);
52
53     lSuma = new JLabel("");
54     lSuma.setBounds(375, 450, 225, 50);
55     lSuma.setFont(font2);
56     midPanel.add(lSuma);
57
58     bSuma = new JButton("Sumuj");
59     bSuma.setBounds(250, 400, 100, 50);
60     midPanel.add(bSuma);
61
62     bDodaj = new JButton("Dodaj");
63     bDodaj.setBounds(250, 500, 100, 50);
64     midPanel.add(bDodaj);
65
66     bReturn.addActionListener(this);
67     bDodaj.addActionListener(this);
68     bSuma.addActionListener(this);
69     bClose.addActionListener(this);
70 }
71
72 @Override
73 public void actionPerformed(ActionEvent e) {
74     Object source = e.getSource();
75     if (source == bReturn) {
76         dispose();
77         GuiServMenu service = new GuiServMenu();
78         service.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
79         service.setVisible(true);
80     } else if (source == bDodaj) {
81         String s500 = t500.getText();
82         String s200 = t200.getText();
83         String s100 = t100.getText();
84         String s50 = t50.getText();
85         String s20 = t20.getText();
86         String s10 = t10.getText();

```

```

87         try {
88             int i500 = Integer.parseInt(s500);
89             int i200 = Integer.parseInt(s200);
90             int i100 = Integer.parseInt(s100);
91             int i50 = Integer.parseInt(s50);
92             int i20 = Integer.parseInt(s20);
93             int i10 = Integer.parseInt(s10);
94             //int suma = i500 * 500 + i200 * 200 + i100 * 100 + i50 * 50 + i20 * 20 + i10 * 10;
95             this.bankomat = new Bankomat();
96             this.bankomat.fromFile();
97             this.bankomat.set500(bankomat.get500() + i500);
98             this.bankomat.set200(bankomat.get200() + i200);
99             this.bankomat.set100(bankomat.get100() + i100);
100             this.bankomat.set50(bankomat.get50() + i50);
101             this.bankomat.set20(bankomat.get20() + i20);
102             this.bankomat.set10(bankomat.get10() + i10);
103             this.bankomat.setSuma();
104             this.bankomat.toFile();
105
106             JOptionPane.showMessageDialog(null, "Dodano", "Wynik", JOptionPane.INFORMATION_MESSAGE);
107         } catch (IOException exception) {
108             exception.printStackTrace();
109         } catch (ClassNotFoundException exception) {
110             JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
111             exception.printStackTrace();
112         } catch (NumberFormatException exception) {
113             JOptionPane.showMessageDialog(null, "Nieprawidłowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
114         }
115     } else if (source == bSuma) {
116         suma();
117     } else if (source == bClose) {
118         dispose();
119     }
120 }

```

```

121
122 public void suma() {
123     String s500 = t500.getText();
124     String s200 = t200.getText();
125     String s100 = t100.getText();
126     String s50 = t50.getText();
127     String s20 = t20.getText();
128     String s10 = t10.getText();
129     try {
130         int i500 = Integer.parseInt(s500);
131         int i200 = Integer.parseInt(s200);
132         int i100 = Integer.parseInt(s100);
133         int i50 = Integer.parseInt(s50);
134         int i20 = Integer.parseInt(s20);
135         int i10 = Integer.parseInt(s10);
136         int suma = i500 * 500 + i200 * 200 + i100 * 100 + i50 * 50 + i20 * 20 + i10 * 10;
137         lsuma.setText(String.valueOf(suma));
138
139     } catch (NumberFormatException exception) {
140         JOptionPane.showMessageDialog(null, "Nieprawidloowe dane", "Wynik", JOptionPane.INFORMATION_MESSAGE);
141     }
142 }
143 }
144

```

Exceptions:

```

AccountException.java
1 package atm_final2;
2
3 public class AccountException extends Exception{
4     public AccountException() {
5
6     }
7     public AccountException(String msg) {
8         super(msg);
9     }
10 }
11

CalcException.java
1 package atm_final2;
2
3 public class CalcException extends Exception {
4     public CalcException() {
5
6     }
7     public CalcException(String msg) {
8         super(msg);
9     }
10 }

```

Instrukcja dla użytkownika

Do projektu załączone zostały wypełnione danymi pliki (klient.txt i bankomat.txt), pierwszy z nich zawiera dane klientów, ich przykładowe konta bankowe, a drugi informację o przechowywanych przez bankomat banknotach. W przypadku gdyby któryś z tych był pusty, użytkownik powinien zacząć od zalogowania się na profil Serwisanta (pola login i hasło należy pozostawić puste) i dodać kilku klientów według instrukcji wewnątrz aplikacji lub uzupełnić gotówkę w bankomacie (nie jest to konieczne, gotówkę można wpłacać również z poziomu klienta).

Sterując przyciskami Zakończ, Powrót, przyciskami klawiatury ekranowej (można używać również „zwykłej”, fizycznej klawiatury) oraz innymi przyciskami widocznymi na ekranie można przenosić się między profilem Serwisanta a Klienta. Podczas wpłacania pieniędzy na konto czy uzupełniania Bankomatu nie jest konieczne wciskanie przycisku Suma. Po zamknięciu programu wszystkie dane powinny pozostać nienaruszone.

W przypadku korzystania z klawiatury ekranowej, tam gdzie jest to dostępne należy zaznaczyć odpowiedni RadioButton.

Przykładowe dane i wyniki

Uzupełnianie Bankomatu	
Bankomat UPH	
Liczba poszczególnych nominałów w ba...	
500 PLN:	84
200 PLN:	346
100 PLN:	54
50 PLN:	382
20 PLN:	107
10 PLN:	314
Laczna kwota:	140980

Etap 1.

Stan przed uzupełnieniem.

Uzupełnianie bankomatu:

500 PLN:

200 PLN:

100 PLN:

50 PLN:

20 PLN:

10 PLN:

Sumuj

Laczna kwota: 1760

Etap 2.

Dodanie po dwóch banknotów każdego nominału.

Uwaga: nie jest konieczne uzupełnianie każdego pola, można dodać np. same banknoty dwustuzłotowe.

Liczba poszczególnych nominałów w ba..

500 PLN: 86

200 PLN: 348

100 PLN: 56

50 PLN: 384

20 PLN: 109

10 PLN: 316

Laczna kwota: 142740

Etap 3.

Stan Bankomatu po uzupełnieniu.

140980 + 1760 =

142,740

Jak widać, zgadza się nie tylko liczba banknotów, ale również łączna kwota zgromadzona w Bankomacie.

Profil Klienta:

Dane właściciela:

Imie:	adam
Nazwisko:	nowak
Bank:	abc
Nr karty:	12341
Środki:	50202.3

Wpłata gotówki

Wyplata gotówki

Doładowanie telefonu

Cofnij

Zamknij

Etap 1.

Wybór funkcji z menu, należy zwrócić uwagę na odkryte dane. Liczba banknotów w Bankomacie jest taka sama jak w poprzednim przykładzie.

Wprowadzanie ban...

500 PLN:

0

200 PLN:

0

100 PLN:

3

50 PLN:

7

20 PLN:

0

10 PLN:

0

Sumuj

Laczna kwota: 650

Wpłać

Etap 2.

Dodawane są banknoty do konta zalogowanego Klienta.

Potwierdzenie:



Data wykonania operacji: 2021-11-15, 14:22:30

Rodzaj wykonanej operacji: wpłata środków

Kwota operacji: 650 PLN

Saldo po operacji: 50852.3

OK

Etap 3.

Potwierdzenie wpłaty (tak samo przy wypłacie).

Bankomat UPH

Dane właściciela:

Imie: adam

Nazwisko: nowak

Bank: abc

Nr karty: 12341

Środki: 50852.3

Wpłata gotówki

Wypłata gotówki

Doładowanie telefonu

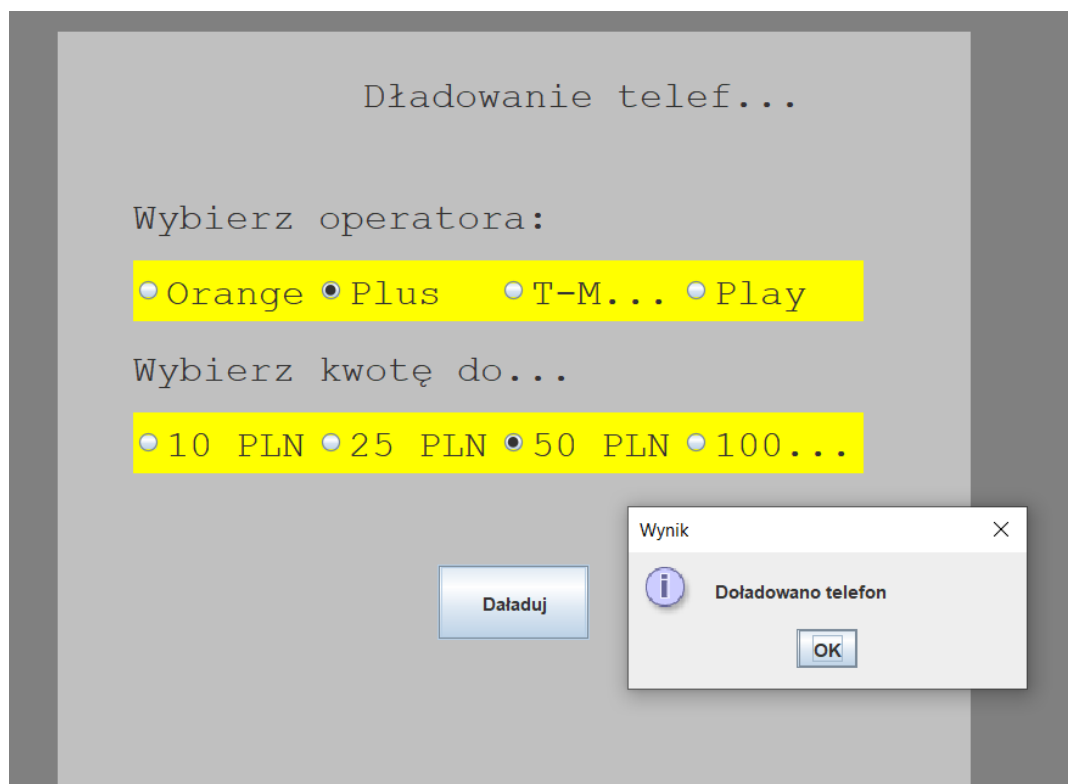
Etap 4.

Jak widać, stan konta powiększył się o 650 pln.

$50852.3 - 650 =$

50,202.3

Różnica pomiędzy etapem 2. a 4. wynosi dokładnie 650 pln.



Etap 5.
Doładowano telefon o 50 pln.

Imie:	adam
Nazwisko:	nowak
Bank:	abc
Nr karty:	12341
Środki:	50802.3

Wpłata gotówki

Wyplata gotówki

Doładowanie telefonu

Etap 6.
Środki po doładowaniu telefonu – dokładnie o 50 pln mniej niż na ilustracji przy Etapie 4.

Liczba poszczególnych nominałów w ba...

500 PLN:	86
200 PLN:	348
100 PLN:	59
50 PLN:	391
20 PLN:	109
10 PLN:	316

Laczna kwota: 143390

Powrót

Zamknij

Etap 7.

Stan Bankomatu po wpłaceniu 650 pln (doładowywanie telefonu nie wpływa na liczbę banknotów w Bankomacie).

Jak widać, stan banknotów zmienił się o wpłacone w Etapie 2. trzy banknoty 100pln i 7 banknotów 50pln.