



**VYSOKÁ ŠKOLA
CHEMICKO-TECHNOLOGICKÁ
V PRAZE**

Automatické zpracování plateb

Semestrální práce

Autor: Vladislav Aulich

Kruh: 159

Studijní obor: Fyzikální a výpočetní chemie

Fakulta: Fakulta chemicko-inženýrská

Akademický rok: 2021/2022

Předmět: Úvod do programování a algoritmů

Vedoucí práce: Ing. Jan Kohout

Praha, 2022

Obsah

1	Teoretická část	3
1.1	Zadání úlohy	3
1.2	Úvod	3
2	Implementace	4
2.1	Schéma řešení	4
2.2	Třídy v projektu	4
2.2.1	Osoba	4
2.2.2	Banka	6
2.2.3	Databáze	6
2.3	Hlavní část programu	12
2.4	GUI	20
2.5	Použitý software	21
2.6	Ukázka programu	22
3	Diskuze	26
	Závěr	27
	Seznam použité literatury	28
	Seznam obrázků	29
	Seznam zdrojových kódů	30

1. Teoretická část

1.1 Zadání úlohy

Vytvořte program, který pomocí formátu csv automaticky přiřadí platby z výpisu účtu členům oddílu.

1.2 Úvod

V mnoha volnočasových útvarech vzniká potřeba třídit a zpracovávat velké množství plateb za příspěvky. Jejich zpracování však zabírá velkou část času rutinní prací která je automatizovatelná. V této práci se zabýváme řešením tohoto problému pomocí souborů ve formátu csv.

Formát csv, z anglického comma-separated values, využívá pro ukládání informací řádky ve kterých jsou jednotlivé položky odděleny oddělovačem, nejčastěji středníkem. Tento formát je hojně využíváný a jednoduchý na obsluhu, a proto jsme ho vybrali pro tento projekt. I vzhledem k faktu, že ve většině bankovních klientů je možnost exportu plateb právě do tohoto formátu.

2. Implementace

2.1 Schéma řešení

Program bude po spuštění od uživatele vyžadovat dvě vstupní tabulky, databáze členů a výpis z účtu banky. Struktura tabulky databáze členů je vyobrazena na Obrázku: 2.1, formát výpisu z banky na Obrázku: 2.2.

	A	B	C	D
1	<Jméno>	<Příjmení>	<email>	<ID>
2	Michal	Novotný	novotny@mail.cz	1202

Obrázek 2.1: Vzor struktury databáze osob

	A	B	C
1	<ID>	<DD.MM.RRRR>	<částka>
2	1202	05.12.2021	2500

Obrázek 2.2: Vzor struktury tabulky s přehledem plateb

Aplikace následně pomocí ID vyhledá, jestli se ve výpisu z banky nachází platba dané osoby. Pokud vše proběhne úspěšně přiřadí osobě datum platby a částku pro zpětné ověření. Uživateli se nabídne export výsledné tabulky. Struktura exportované tabulky je naznačena na Obrázku: 2.3 Dále bude dostupné filtrování osob které zaplatili, nezaplatili nebo zaplatili chybnou částku.

	A	B	C	D	E	F	G
1	<Jméno>	<Příjmení>	<email>	<ID>	<Zaplaceno>	<Datum platby>	<Částka>
2	Michal	Novotný	novotny@mail.cz	1202	1	25.12.2021	2000

Obrázek 2.3: Vzor struktury exportovaných dat

Součástí aplikace nebude rozesílání informace a pokynů k platbě, to bude zajišťovat externí aplikace. Program bude pracovat pouze z databází členů a tabulkou plateb.

2.2 Třídy v projektu

2.2.1 Osoba

První třídou je Osoba. Pro každou osobu je zaznamenáváno jméno, příjmení, email a ID. Identifikátor je pro každou osobu unikátní. Identifikátor zároveň uchovává informaci o cenové kategorii. V našem oddílu máme dvě cenové kategorie o kterých rozhoduje sudost/-lichost identifikátoru. Toto řešení je zároveň i dobře lidsky čitelné.

Po kontrole plateb je každé osobě přiřazen příznak platby (proměnná Zaplaceno), kde 0 - nezaplaceno, 1 - zaplaceno, 2 - chybná částka. Pokud je platba nalezena je ještě přiřazeno datum platby a částka.

Třída obsahuje dva konstruktory. První slouží k nahrání údajů z uživatelského vstupu. Druhý potom k přiřazení kompletních údajů osobě včetně data platby a částky. Pracuje s metodou ToString, která vrací jméno a příjmení osoby.

```
1  class Osoba
2  {
3      public string Jmeno { get; private set; }
4      public string Prijmeni { get; private set; }
5      public string Email { get; private set; }
6      public uint ID { get; private set; }
7      //0 = nezaplaceno; 1 = zaplaceno; 2 = spatna castka
8      public int Zaplaceno { get; set; }
9      public DateTime Datum { get; set; }
10     public decimal Castka { get; set; }
11
12     public Osoba(string jmeno, string prijmeni, string email, uint id)
13     {
14         Jmeno = jmeno;
15         Prijmeni = prijmeni;
16         Email = email;
17         ID = id;
18         Zaplaceno = 0;
19         Datum = default;
20         Castka = 0;
21     }
22     public Osoba(string jmeno, string prijmeni, string email, uint id, int
        zaplaceno, DateTime datum, decimal castka)
23     {
24         Jmeno = jmeno;
25         Prijmeni = prijmeni;
26         Email = email;
27         ID = id;
28         Zaplaceno = zaplaceno;
29         Datum = datum;
30         Castka = castka;
31     }
32     public override string ToString()
33     {
34         return Prijmeni + " " + Jmeno;
35     }
36 }
```

Kód 2.1: Zdrojový kód třídy Osoba

2.2.2 Banka

Obdobně je navržena struktura třídy Banka, jejíž instancí je záznam v tabulce výpisu z banky. Výpis obsahuje ID člena (zadané při platbě, např. variabilní symbol), datum platby a částku. Třída obsahuje metodu ToString, která vrátí ID a datum platby.

```
1  class Banka
2  {
3      public uint ID { get; private set; }
4      public DateTime Datum { get; private set; }
5      public decimal Castka { get; private set; }
6      public Banka(uint id, DateTime platba, decimal castka)
7      {
8          ID = id;
9          Datum = platba;
10         Castka = castka;
11     }
12     public override string ToString()
13     {
14         return ID + " " + Datum;
15     }
16 }
```

Kód 2.2: Zdrojový kód třídy Banka

2.2.3 Databáze

Třída Databáze slouží k vytvoření databáze osob a její správy. Databáze osob bude tvořena instancí třídy List a bude privátní. Pomocí veřejných metod bude umožňovat přidávat a odebírat osoby, načítat vstupní csv soubor, přiřazovat informace o platbách, vyhledávat osoby a exportovat data.

```
1  class Databaze
2  {
3      private List<Osoba> zaznam_osoba;
4
5      public Databaze()
6      {
7          zaznam_osoba = new List<Osoba>();
8      }
9      /// <summary>
10     /// Prida novy kompletne zaznam osoby do databaze pro znamy stav platby
11     /// </summary>
12     /// <param name="jmeno">Jmeno osoby</param>
13     /// <param name="prijmeni">Prijmeni osoby</param>
14     /// <param name="email">Email osoby</param>
15     /// <param name="id">Identifikator osoby</param>
16     /// <param name="zaplaceno">0-znacuje status platby->0-ne;1-ano;2-spatna
17     ///   castka</param>
18     /// <param name="datum">Datum platby</param>
19     /// <param name="castka">Castka platby</param>
20     public void PridejOsobu(string jmeno, string prijmeni, string email, uint
21     id, int zaplaceno, DateTime datum, decimal castka)
22     {
23         Osoba o~ = new Osoba(jmeno, prijmeni, email, id, zaplaceno, datum,
24         castka);
25         zaznam_osoba.Add(o);
26     }
27     /// <summary>
28     /// Prida novy zaznam osoby do databaze pro neznamy stav platby
29     /// </summary>
30     /// <param name="jmeno">Jmeno osoby</param>
31     /// <param name="prijmeni">Prijmeni osoby</param>
32     /// <param name="email">Email osoby</param>
33     /// <param name="id">Identifikator osoby</param>
34     public void PridejOsobu(string jmeno, string prijmeni, string email, uint
35     id)
36     {
37         Osoba o~ = new Osoba(jmeno, prijmeni, email, id);
38         zaznam_osoba.Add(o);
39     }
40     /// <summary>
41     /// Odebere zaznam osoby z~databaze
42     /// </summary>
43     /// <param name="index">index osoby</param>
44     public void OdeberOsobu(int index)
45     {
46         zaznam_osoba.RemoveAt(index);
47     }
48     /// <summary>
```

```
45     /// Zkontroluje jestli odpovídají parametry platby a přiřadí platbu dané
        osobe
46     /// </summary>
47     /// <param name="index">Index osoby</param>
48     /// <param name="id">Identifikátor osoby</param>
49     /// <param name="castka">Castka platby</param>
50     /// <param name="datum">Datum platby</param>
51     /// <param name="castka_pripavka">Castka príspevku pro prípravku</param>
52     /// <param name="castka_druzstva">Castka príspevku pro družstva</param>
53     public void Zaplatil(int index, uint id, decimal castka, DateTime datum,
        decimal castka_pripavka, decimal castka_druzstva)
54     {
55         //Podle id rozhodne družstvo (lícha prípravka, suda družstva)
56         //Test prípravka
57         if (id % 2 != 0)
58         {
59             //Zkontroluje, jestli odpovídá castka
60             if (castka == castka_pripavka)
61             {
62                 //Nastaví příznak na zapláceno
63                 zaznam_osoba[index].Zapláceno = 1;
64                 zaznam_osoba[index].Datum = datum;
65                 zaznam_osoba[index].Castka = castka;
66             }
67             else
68             {
69                 //Nastaví na chybnou castku
70                 zaznam_osoba[index].Zapláceno = 2;
71                 zaznam_osoba[index].Datum = datum;
72                 zaznam_osoba[index].Castka = castka;
73             }
74         }
75         //Test družstva
76         else if (id % 2 == 0)
77         {
78             //Zkontroluje, jestli odpovídá castka
79             if (castka == castka_druzstva)
80             {
81                 //Nastaví příznak na zapláceno
82                 zaznam_osoba[index].Zapláceno = 1;
83                 zaznam_osoba[index].Datum = datum;
84                 zaznam_osoba[index].Castka = castka;
85             }
86             else
87             {
88                 //Nastaví na chybnou castku
89                 zaznam_osoba[index].Zapláceno = 2;
90                 zaznam_osoba[index].Datum = datum;
91                 zaznam_osoba[index].Castka = castka;
```



```
92         }
93     }
94 }
95 /// <summary>
96 /// Hleda identifikator osoby (id) v~zaznamu osob
97 /// </summary>
98 /// <param name="id">Hledane identifikator osoby (id)</param>
99 /// <returns>Index osoby</returns>
100 public int Najdi(uint id)
101 {
102     int index = zaznam_osoba.FindIndex(os => os.ID == id);
103     return index;
104 }
105 /// <summary>
106 /// Vratí osoby z~databaze ve vybrane skupine
107 /// </summary>
108 /// <param name="zaplaceno">Oznacuje status platby->0-ne;1-ano;2-spatna
109     castka</param>
110 /// <returns>Pole osob</returns>
111 public Osoba[] VratVybrane(int zaplaceno)
112 {
113     List<Osoba> vybrani = new List<Osoba>();
114     vybrani = zaznam_osoba.FindAll(os => os.Zaplaceno == zaplaceno);
115     return vybrani.ToArray();
116 }
117 /// <summary>
118 /// Vratí vsechny osoby v~databazi
119 /// </summary>
120 /// <returns>Pole osob</returns>
121 public Osoba[] VratVsechny()
122 {
123     return zaznam_osoba.ToArray();
124 }
125 /// <summary>
126 /// Vratí emaily vybrane skupiny
127 /// </summary>
128 /// <param name="zaplaceno"></param>
129 /// <returns>Seznam emailu oddeleny ;</returns>
130 public string[] VratEmail(int zaplaceno)
131 {
132     List<Osoba> vybrani = new List<Osoba>();
133     vybrani = zaznam_osoba.FindAll(os => os.Zaplaceno == zaplaceno);
134     string[] email = new string[vybrani.Count];
135     //spoji maily do jednoho retezce s~vyuzitim stredniku
136     for (int i = 0; i < vybrani.Count; i++)
137     {
138         email[i] = vybrani[i].Email;
139     }
140     return email;
141 }
```

```
140     }
141     /// <summary>
142     /// Importuje osobu z~csv souboru do databaze
143     /// </summary>
144     /// <param name="oddelovac">Oddelovac csv souboru</param>
145     /// <param name="path">Cesta souboru</param>
146     public void Import(char oddelevac, string path)
147     {
148         zaznam_osoba.Clear();
149         // Otevre soubor pro cteni
150         using (StreamReader sr = new StreamReader(path))
151         {
152             string s;
153             // cte radek po radku
154             while ((s~ sr.ReadLine()) != null)
155             {
156                 // rozdeli string radku podle stredniku
157                 string[] sloupec = s.Split(oddelevac);
158                 string jmeno = sloupec[0];
159                 string prijmeni = sloupec[1];
160                 string email = sloupec[2];
161                 uint id = uint.Parse(sloupec[3]);
162                 // prida uzivatele s~danymi hodnotami
163                 PridejOsobu(jmeno, prijmeni, email, id);
164             }
165         }
166     }
167     /// <summary>
168     /// Exportuje vybranou skupinu osob ve formatu csv dle parametru zaplaceno
169     /// (0-ne;1-ano;2-chybna castka)
170     /// </summary>
171     /// <param name="oddelevac">Oddelovac csv</param>
172     /// <param name="cesta">Cesta souboru</param>
173     /// <param name="zaplaceno">Oznacuje status platby->0-ne;1-ano;2-spatna
174     /// castka</param>
175     public void Export(char oddelevac, string cesta, int zaplaceno)
176     {
177         // otevreni souboru pro zapis
178         using (StreamWriter sw = new StreamWriter(cesta, false))
179         {
180             //Projde vsechny osoby, ktere zaplatili
181             foreach (Osoba o~in VratVybrane(zaplaceo))
182             {
183                 // vytvoreni pole hodnot + ochrana, pokud je zadan oddelevac
184                 string[] hodnoty = { o.Jmeno.Replace(oddelevac, ' '),
185                                     o.Prijmeni.Replace(oddelevac, ' '),
186                                     o.Email.Replace(oddelevac, ' '),
187                                     o.ID.ToString(),
188                                     o.Zaplaceo.ToString(),
```

```
187         o.Datum.ToShortDateString(),
188         o.Castka.ToString() };
189         // vytvoreni radku
190         string radek = String.Join(oddelovac.ToString(), hodnoty);
191         // zapis radku
192         sw.WriteLine(radek);
193     }
194     // vyprazdneni bufferu
195     sw.Flush();
196 }
197 }
198 /// <summary>
199 /// Exportuje vsechny zaznamy osob ve formatu .csv
200 /// </summary>
201 /// <param name="oddelovac">Oddelovac csv</param>
202 /// <param name="cesta">Cesta souboru</param>
203 public void Export(char oddelovac, string cesta)
204 {
205     // otevreni souboru pro zapis
206     using (StreamWriter sw = new StreamWriter(cesta, false))
207     {
208         //Projde vsechny osoby, ktere zaplatili
209         foreach (Osoba o~in VratVsechny())
210         {
211             // vytvoreni pole hodnot + ochrana, pokud je zadan oddelovac
212             string[] hodnoty = { o.Jmeno.Replace(oddelovac, ' '),
213                                 o.Prijmeni.Replace(oddelovac, ' '),
214                                 o.Email.Replace(oddelovac, ' '),
215                                 o.ID.ToString(),
216                                 o.Zaplaceno.ToString(),
217                                 o.Datum.ToShortDateString(),
218                                 o.Castka.ToString() };
219             // vytvoreni radku
220             string radek = String.Join(oddelovac.ToString(), hodnoty);
221             // zapis radku
222             sw.WriteLine(radek);
223         }
224         // vyprazdneni bufferu
225         sw.Flush();
226     }
227 }
228 }
```

Kód 2.3: Zdrojový kód třídy Databáze

2.3 Hlavní část programu

Obsluha GUI a klíčová část programu je tvořena třídou Form1.

```
1      using System;
2      using System.Collections.Generic;
3      using System.IO;
4      using System.Windows.Forms;
5
6      namespace semestralka_windows_forms
7      {
8          public partial class Form1 : Form
9          {
10             private Database database;
11             private List<Banka> zaznam_banky;
12             decimal castka_pripavka;
13             decimal castka_druzstva;
14             char oddelovac;
15
16
17             public Form1()
18             {
19                 //vytvori list s~platbami z~banky
20                 zaznam_banky = new List<Banka>();
21                 // vytvoreni databaze
22                 database = new Database();
23
24                 InitializeComponent();
25
26                 //Uvodni nastaveni prvku v~GUI
27                 comboBox_oddlovac.SelectedIndex = 0;
28                 numericUpDown_pripavka.Value = 2000;
29                 numericUpDown_druzstva.Value = 2500;
30                 castka_pripavka = numericUpDown_pripavka.Value;
31                 castka_druzstva = numericUpDown_druzstva.Value;
32                 oddlovac = Char.Parse(comboBox_oddlovac.Text);
33                 lbl_platba_datum_vsichni.Text = "";
34                 lbl_import_osob_cesta.Text = "";
35                 lbl_import_plateb_cesta.Text = "";
36             }
37             //Prvni karta (Instrukce)
38             private void button_import_banky_Click(object sender, EventArgs e)
39             {
40                 OpenFileDialog dialog = new OpenFileDialog();
41                 //Omezeni souboru, ktere mohou byt vybrany
42                 dialog.Filter = "CSV soubory (*.csv)|*.csv";
43                 //Zakaz vyberu vice souboru
44                 dialog.Multiselect = false;
45                 //Pokud uzivatel potvrdi
```

```
46         if (dialog.ShowDialog() == DialogResult.OK)
47         {
48             //Ziska cestu k souboru
49             String path = dialog.FileName;
50             lbl_import_plateb_cesta.Text = path;
51             try
52             {
53                 using (StreamReader sr = new StreamReader(path))
54                 {
55                     string s;
56                     // cte radek po radku
57                     while ((s = sr.ReadLine()) != null)
58                     {
59                         // rozdeli string radku podle separatoru
60                         string[] sloupec = s.Split(oddelovac);
61                         uint id = uint.Parse(sloupec[0]);
62                         DateTime datum = DateTime.Parse(sloupec[1]);
63                         int zaplaceno = int.Parse(sloupec[2]);
64                         // zaznam z vypsů s danými hodnotami
65                         Banka b = new Banka(id, datum, zaplaceno);
66                         zaznam_banky.Add(b);
67                     }
68                 }
69             }
70             catch
71             {
72                 MessageBox.Show("Sobor obsahuje chybu ve formátování.  
Zkontrolujte formát vstupních dat dle instrukcí. (Není  
prázdný sloupec?)", "Chyba", MessageBoxButtons.OK,  
                    MessageBoxIcon.Error);
73                 lbl_import_plateb_cesta.Text = "";
74             }
75         }
76     }
77     //Zkontroluje jestli dane osoby zaplatili a přiřadí jim údaje o platbě
78     private void button_porovnej_Click(object sender, EventArgs e)
79     {
80         int index = -1;
81         try
82         {
83             for (int i = 0; i < zaznam_banky.Count; i++)
84             {
85                 index = database.Najdi(zaznam_banky[i].ID);
86                 //Pokud je nalezena osoba
87                 if (index >= 0)
88                 {
89                     database.Zaplatil(index, zaznam_banky[i].ID,  
                        zaznam_banky[i].Castka, zaznam_banky[i].Datum,  
                        castka_pripravka, castka_druzstva);
```

```
90         }
91     catch
92     {
93         MessageBox.Show("Doslo k~chybe, zkontrolujte format vstupnich
                        dat.", "Chyba", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
94     }
95 }
96 //Druha karta (vsichni)
97 //Nacteni csv s~osobami
98 private void button_nacti_osoby_Click(object sender, EventArgs e)
99 {
100     OpenFileDialog dialog = new OpenFileDialog();
101     //Omezeni souboru, ktere mohou byt vybrany
102     dialog.Filter = "CSV soubory (*.csv)|*.csv";
103     //Zakaz vyberu vice souboru
104     dialog.Multiselect = false;
105     //Pokud uzivatel potvrdi
106     if (dialog.ShowDialog() == DialogResult.OK)
107     {
108         //Ziska cestu k~souboru
109         String path = dialog.FileName;
110         lbl_import_osob_cesta.Text = path;
111         try
112         {
113             databaze.Import(oddelovac, path);
114         }
115         catch
116         {
117             MessageBox.Show("Sobor obsahuje chybu ve formatovani.
                        Zkontrolujte format vstupnich dat dle instrukci. (Neni
                        prazdny sloupec?)", "Chyba", MessageBoxButtons.OK,
                        MessageBoxIcon.Error);
118             lbl_import_osob_cesta.Text = "";
119         }
120     }
121 }
122 //Zobrazeni detailu o~osobe po vybrani v~listboxu
123 private void listBox_vsichni_SelectedIndexChanged(object sender,
        EventArgs e)
124 {
125     if (listBox_vsichni.SelectedItem != null)
126     {
127         //Zobrazeni detailu vybrane osoby
128         Osoba u~ = (Osoba)listBox_vsichni.SelectedItem;
129         lbl_jmeno_vsichni.Text = u.Jmeno;
130         lbl_prijmeni_vsichni.Text = u.Prijmeni;
131         lbl_email_vsichni.Text = u.Email;
132         lbl_id_vsichni.Text = u.ID.ToString();
```

```
133         //Zobrazeni skupiny (rozhodnuti podle id)
134         if (u.ID % 2 == 0)
135             lbl_skupina_vsichni.Text = "Druzstva";
136         else if (u.ID % 2 != 0)
137             lbl_skupina_vsichni.Text = "Pripravka";
138         //Rozhodnuti o~platbe
139         if (u.Zaplaceno == 0)
140         {
141             lbl_platba.Text = "Platba neevidovana";
142         }
143         if (u.Zaplaceno == 1)
144         {
145             lbl_platba.Text = "Zaplaceno";
146             lbl_platba_datum_vsichni.Text =
147                 u.Datum.ToShortDateString();
148         }
149         if (u.Zaplaceno == 2)
150         {
151             lbl_platba.Text = "Zaplacena chybna castka";
152             lbl_platba_datum_vsichni.Text =
153                 u.Datum.ToShortDateString();
154         }
155     }
156     //Rucni vlozeni osob do databaze
157     private void button_pridat_database_Click(object sender, EventArgs e)
158     {
159         string jmeno = textBox_jmeno_vsichni.Text;
160         string prijmeni = textBox_prijmeni_vsichni.Text;
161         string email = textBox_email_vsichni.Text;
162         uint id = (uint)numericUpDown_id_vsichni.Value;
163         int zaplaceno = 0;
164         DateTime datum = DateTime.MinValue;
165         decimal castka = 0;
166         if (radioButton_nezaplaceno.Checked)
167             zaplaceno = 0;
168         if (radioButton_zaplaceno.Checked)
169         {
170             zaplaceno = 1;
171             datum = dateTimePicker.Value;
172             if (id % 2 != 0)
173                 castka = castka_pripravka;
174             if (id % 2 == 0)
175                 castka = castka_druzstva;
176         }
177         database.PridejOsobu(jmeno, prijmeni, email, id, zaplaceno, datum,
178             castka);
179         listBox_vsichni.Items.Add(new Osoba(jmeno, prijmeni, email, id,
```

```
        zaplaceno, datum, castka));
179     }
180     //Odebere vybranou osobu z~database
181     private void button_odebrat_database_Click(object sender, EventArgs e)
182     {
183         if (listBox_vsichni.SelectedIndex >= 0)
184         {
185             database.OdeberOsobu(listBox_vsichni.SelectedIndex);
186             listBox_vsichni.Items.RemoveAt(listBox_vsichni.SelectedIndex);
187         }
188         else
189             MessageBox.Show("Vyberte osobu k~odebrání ze seznamu",
190                             "Chyba", MessageBoxButtons.OK, MessageBoxIcon.Error);
191     }
192     //Export vseh
193     private void button_export_vsichni_Click(object sender, EventArgs e)
194     {
195         //Ulozeni platicu do .csv
196         SaveFileDialog saveFileDialog1 = new SaveFileDialog();
197         saveFileDialog1.Title = "Save text Files";
198         saveFileDialog1.DefaultExt = "csv";
199         saveFileDialog1.Filter = "CSV soubory (*.csv)|*.csv";
200         saveFileDialog1.RestoreDirectory = true;
201         if (saveFileDialog1.ShowDialog() == DialogResult.OK)
202         {
203             try
204             {
205                 database.Export(oddelovac, saveFileDialog1.FileName);
206             }
207             catch
208             {
209                 MessageBox.Show("Vytvorte nový soubor.", "Chyba",
210                                 MessageBoxButtons.OK, MessageBoxIcon.Error);
211             }
212         }
213         //Treti karta (Zaplaceno)
214         //Zobrazení detailu o~osobe po vybrání v~listboxu
215         private void listBox_zaplaceno_SelectedIndexChanged(object sender,
216                                                             EventArgs e)
217         {
218             if (listBox_zaplaceno.SelectedItem != null)
219             {
220                 //Zobrazení detailu vybrane osoby
221                 Osoba u~ = (Osoba)listBox_zaplaceno.SelectedItem;
222                 lbl_jmeno_zaplaceno.Text = u.Jmeno;
223                 lbl_prijmeni_zaplaceno.Text = u.Prijmeni;
224                 lbl_email_zaplaceni.Text = u.Email;
225                 lbl_id_zaplaceno.Text = u.ID.ToString();
226             }
227         }
228     }
```



```
225         //Zobrazeni skupiny (rozhodnuti podle id)
226         if (u.ID % 2 == 0)
227             lbl_skupina_zaplaceno.Text = "Druzstva";
228         else if (u.ID % 2 != 0)
229             lbl_skupina_zaplaceno.Text = "Pripravka";
230         lbl_datum_platba_zaplaceno.Text = u.Datum.ToShortDateString();
231     }
232 }
233 private void button_kopiruj_email_zaplaceno_Click(object sender,
    EventArgs e)
234 {
235     //Zkopirovani emailu do schranky
236     if (!string.IsNullOrEmpty(textBox_zaplaceno.Text))
237         Clipboard.SetText(textBox_zaplaceno.Text);
238 }
239 private void button_export_zaplaceno_Click(object sender, EventArgs e)
240 {
241     //Ulozeni platicu do .csv
242     SaveFileDialog saveFileDialog1 = new SaveFileDialog();
243     saveFileDialog1.Title = "Save text Files";
244     saveFileDialog1.DefaultExt = "csv";
245     saveFileDialog1.Filter = "CSV soubory (*.csv)|*.csv";
246     saveFileDialog1.RestoreDirectory = true;
247     if (saveFileDialog1.ShowDialog() == DialogResult.OK)
248     {
249         try
250         {
251             databaze.Export(oddelovac, saveFileDialog1.FileName, 1);
252         }
253         catch
254         {
255             MessageBox.Show("Vytvorte novy soubor.", "Chyba",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
256         }
257     }
258 }
259 //Ctvrta karta (Nezaplaceno)
260 //Zobrazeni detailu o~osobe po vybrani v~listboxu
261 private void listBox_nezaplaceno_SelectedIndexChanged(object sender,
    EventArgs e)
262 {
263     if (listBox_nezaplaceno.SelectedItem != null)
264     {
265         //Zobrazeni detailu vybrane osoby
266         Osoba u~ = (Osoba)listBox_nezaplaceno.SelectedItem;
267         lbl_jmeno_nezaplaceno.Text = u.Jmeno;
268         lbl_prijmeni_nezaplaceno.Text = u.Prijmeni;
269         lbl_email_nezaplaceni.Text = u.Email;
270         lbl_id_nezaplaceno.Text = u.ID.ToString();
    }
```

```
271         //Zobrazení skupiny (rozhodnutí podle id)
272         if (u.ID % 2 == 0)
273             lbl_skupina_nezaplaceno.Text = "Druzstva";
274         else if (u.ID % 2 != 0)
275             lbl_skupina_nezaplaceno.Text = "Pripravka";
276     }
277 }
278 private void button_zkopiruj_email_nezaplaceno_Click(object sender,
    EventArgs e)
279 {
280     //Zkopirovani emailu do schranky
281     if (!string.IsNullOrEmpty(textBox_nezaplaceno.Text))
282         Clipboard.SetText(textBox_nezaplaceno.Text);
283 }
284 //Export neplaticu
285 private void button_export_nezaplaceno_Click(object sender, EventArgs
    e)
286 {
287     //Ulozeni neplaticu do .csv
288     SaveFileDialog saveFileDialog1 = new SaveFileDialog();
289     saveFileDialog1.Title = "Save text Files";
290     saveFileDialog1.DefaultExt = ".csv";
291     saveFileDialog1.Filter = "CSV soubory (*.csv)|*.csv";
292     saveFileDialog1.FilterIndex = 2;
293     saveFileDialog1.RestoreDirectory = true;
294     if (saveFileDialog1.ShowDialog() == DialogResult.OK)
295     {
296         try
297         {
298             databaze.Export(oddelovac, saveFileDialog1.FileName, 0);
299         }
300         catch
301         {
302             MessageBox.Show("Vytvorte novy soubor.", "Chyba",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
303         }
304     }
305 }
306 //Pata karta (Chybna castka)
307 //Zobrazení detailu o osobě po vybrání v listBoxu
308 private void listBox_chyba_SelectedIndexChanged(object sender,
    EventArgs e)
309 {
310     if (listBox_chyba.SelectedItem != null)
311     {
312         //Zobrazení detailu vybrané osoby
313         Osoba u = (Osoba)listBox_chyba.SelectedItem;
314         lbl_jmeno_chyba.Text = u.Jmeno;
315         lbl_prijmeni_chyba.Text = u.Prijmeni;
```

```
316         lbl_email_chyba.Text = u.Email;
317         lbl_id_chyba.Text = u.ID.ToString();
318         //Zobrazení skupiny (rozhodnutí podle id)
319         if (u.ID % 2 == 0)
320             lbl_skupina_chyba.Text = "Druzstva";
321         else if (u.ID % 2 != 0)
322             lbl_skupina_chyba.Text = "Pripravka";
323         lbl_datum_chyba.Text = u.Datum.ToShortDateString();
324         lbl_castka_chyba.Text = u.Castka.ToString();
325     }
326 }
327 //Exportuje osoby z~chybne zadanou platbou
328 private void button_export_chyba_Click(object sender, EventArgs e)
329 {
330     //Ulození neplatnicu do .csv
331     SaveFileDialog saveFileDialog1 = new SaveFileDialog();
332     saveFileDialog1.Title = "Save text Files";
333     saveFileDialog1.DefaultExt = ".csv";
334     saveFileDialog1.Filter = "CSV soubory (*.csv)|*.csv";
335     saveFileDialog1.FilterIndex = 2;
336     saveFileDialog1.RestoreDirectory = true;
337     if (saveFileDialog1.ShowDialog() == DialogResult.OK)
338     {
339         try
340         {
341             databaze.Export(oddelovac, saveFileDialog1.FileName, 2);
342         }
343         catch
344         {
345             MessageBox.Show("Vytvorte novy soubor.", "Chyba",
346                             MessageBoxButtons.OK, MessageBoxIcon.Error);
347         }
348     }
349 private void button_zkopiruj_email_chyba_Click(object sender,
350     EventArgs e)
351 {
352     //Zkopirovani emailu do schranky
353     if (!string.IsNullOrEmpty(textBox_chybna_castka.Text))
354         Clipboard.SetText(textBox_chybna_castka.Text);
355 }
356 //Aktualizace jednotlivych karet v~GUI
357 private void tabControl1_SelectedIndexchanged(object sender, EventArgs
358     e)
359 {
360     switch ((sender as TabControl).SelectedIndex)
361     {
362         case 0:
363             //Stranka uz je nactena
```

```
362         break;
363     case 1:
364         //Vypln listboxu vsichni
365         listBox_vsichni.Items.Clear();
366         listBox_vsichni.Items.AddRange(database.VratVsechny());
367         break;
368     case 2:
369         //Vypln listboxu zaplaceno + seznam mailu
370         listBox_zaplaceno.Items.Clear();
371         listBox_zaplaceno.Items.AddRange(database.VratVybrane(1));
372         textBox_zaplaceno.Text = string.Join(";",
            database.VratEmail(1));
373         break;
374     case 3:
375         //Vypln listboxu nezaplaceno + seznam mailu
376         listBox_nezaplaceno.Items.Clear();
377         listBox_nezaplaceno.Items.AddRange(database.VratVybrane(0));
378         textBox_nezaplaceno.Text = string.Join(";",
            database.VratEmail(0));
379         break;
380     case 4:
381         //Vypln listboxu chybi + seznam mailu
382         listBox_chyba.Items.Clear();
383         listBox_chyba.Items.AddRange(database.VratVybrane(2));
384         textBox_chybna_castka.Text = string.Join(";",
            database.VratEmail(2));
385         break;
386     }
387 }
388 }
389 }
```

Kód 2.4: Zdrojový kód třídy Form1

2.4 GUI

Grafické prostředí využívá jako hlavní prvek `tabControl`. Jednotlivé karty provází uživatele nastavením a zobrazují výstupy programu. Tento prvek byl vybrán kvůli přehlednosti. Realizace GUI je naznačena v sekci Ukázka programu.

2.5 Použitý software

Pro tvorbu dokumentace bylo využito prostředí online \LaTeX editoru Overleaf. Jako podklad byla využita šablona k maturitní práci Gymnázia Jana Keplera, která byla graficky upravena.

Programování samotné aplikace probíhalo ve vývojovém prostředí Visual Studio 2019 od společnosti Microsoft. Změny na projektu byly průběžně zaznamenávány verzovacím systémem Git a následně vkládány na server Github. Odkaz na repozitář: https://github.com/aulichv/semestralka_windows_forms.git.

2.6 Ukázka programu

Správce plateb

Instrukce Všichni Zaplaceno Nezaplaceno Chybná částka

1) Vyberte soubor (.csv) s údaji o členech ve formátu uvedeném na obrázku.

	A	B	C	D
1	<Jméno>	<Příjmení>	<email>	<ID>
2	Michal	Novotný	novotny@mail.cz	1202

<Jméno>;<Příjmení>;<email>;<ID>
Michal;Novotný;novotny@mail.cz;1202

2) Vyberte soubor (.csv) s údaji o platbách ve formátu uvedeném na obrázku.

	A	B	C
1	<ID>	<DD.MM.RRRR>	<částka>
2	1202	05.12.2021	2500

<ID>;<DD.MM.RRRR>;<částka>
1202;05.12.2021;2500

3) Vyberte použitý oddělovač (výchozí středník).

Oddělovač:

4) Určete výši kontrolovaných částek.

Výše příspěvků družstva

Výše příspěvků příprava

Obrázek 2.4: Karta s instrukcemi

Správce plateb

Instrukce Všichni Zaplaceno Nezaplaceno Chybná částka

1) Vyberte soubor (.csv) s údaji o členech ve formátu uvedeném na obrázku. C:\Users\aulic\Documents\semestr...

	A	B	C	D
1	<Jméno>	<Příjmení>	<email>	<ID>
2	Michal	Novotný	novotny@mail.cz	1202

<Jméno>;<Příjmení>;<email>;<ID>
Michal;Novotný;novotny@mail.cz;1202

2) Vyberte soubor (.csv) s údaji o platbách ve formátu uvedeném na obrázku. C:\Users\aulic\Documents\semestr...

	A	B	C
1	<ID>	<DD.MM.RRRR>	<částka>
2	1202	05.12.2021	2500

<ID>;<DD.MM.RRRR>;<částka>
1202;05.12.2021;2500

3) Vyberte použitý oddělovač (výchozí středník).

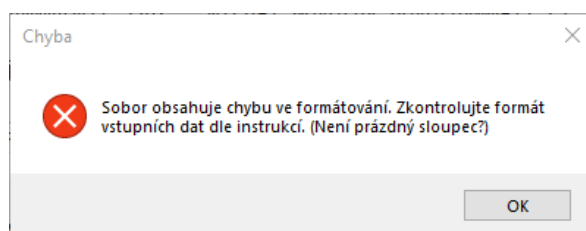
Oddělovač:

4) Určete výši kontrolovaných částek.

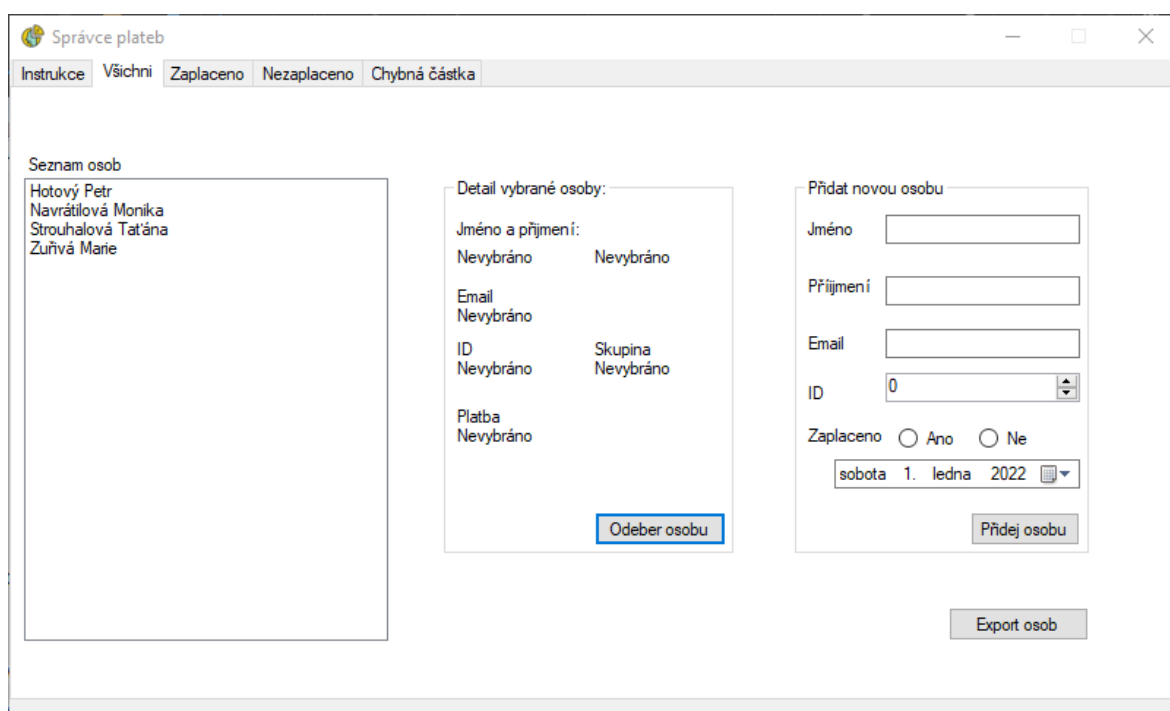
Výše příspěvků družstva

Výše příspěvků příprava

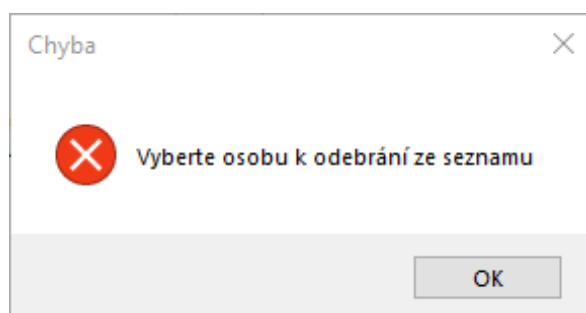
Obrázek 2.5: Karta s instrukcemi po nahrání souborů



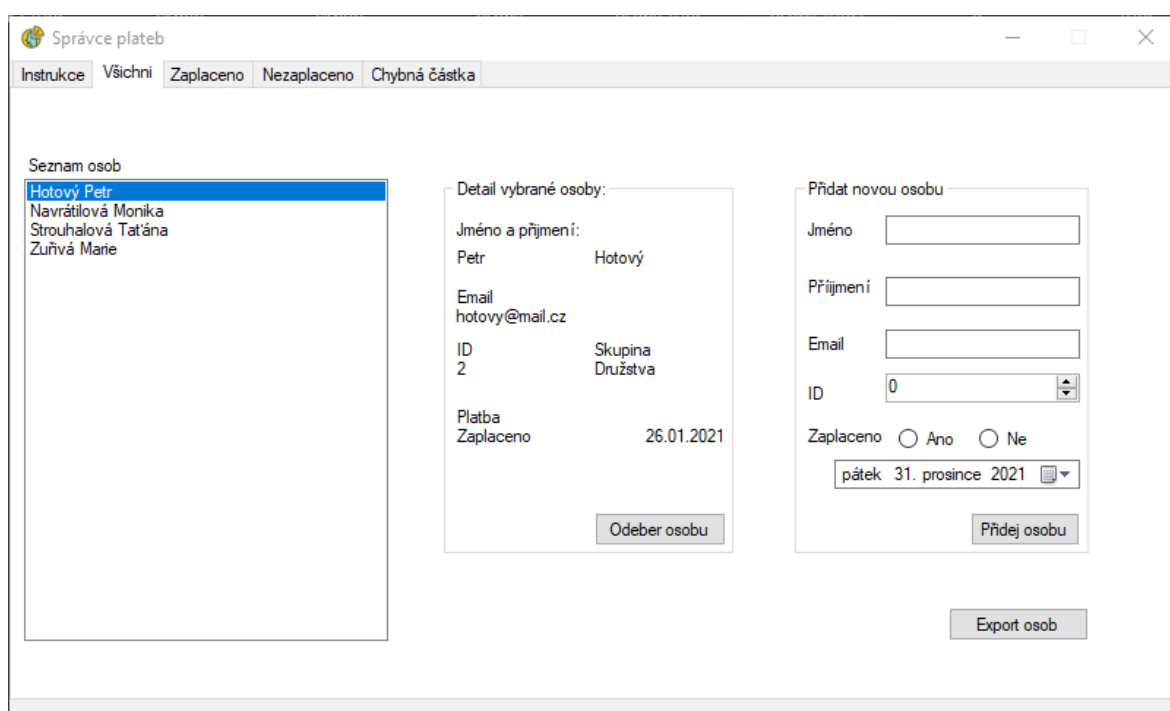
Obrázek 2.6: Chybová hláška informující uživatele o špatném vstupním formátu



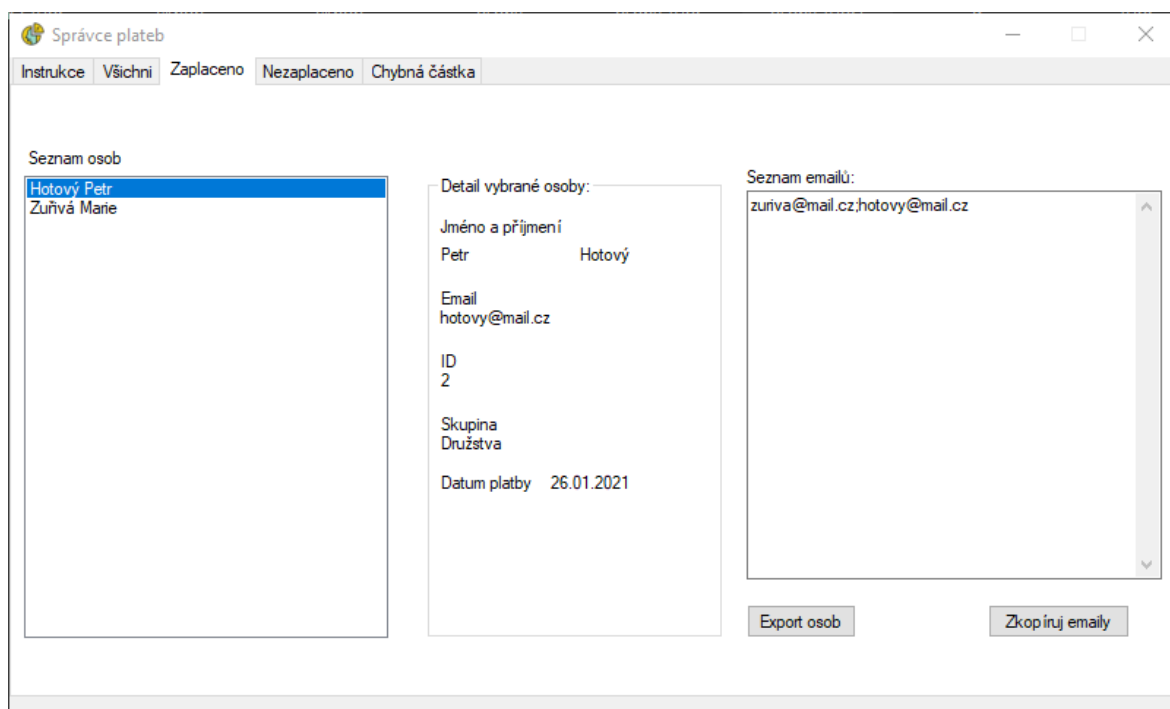
Obrázek 2.7: Karta zobrazující všechny osoby, bez výběru



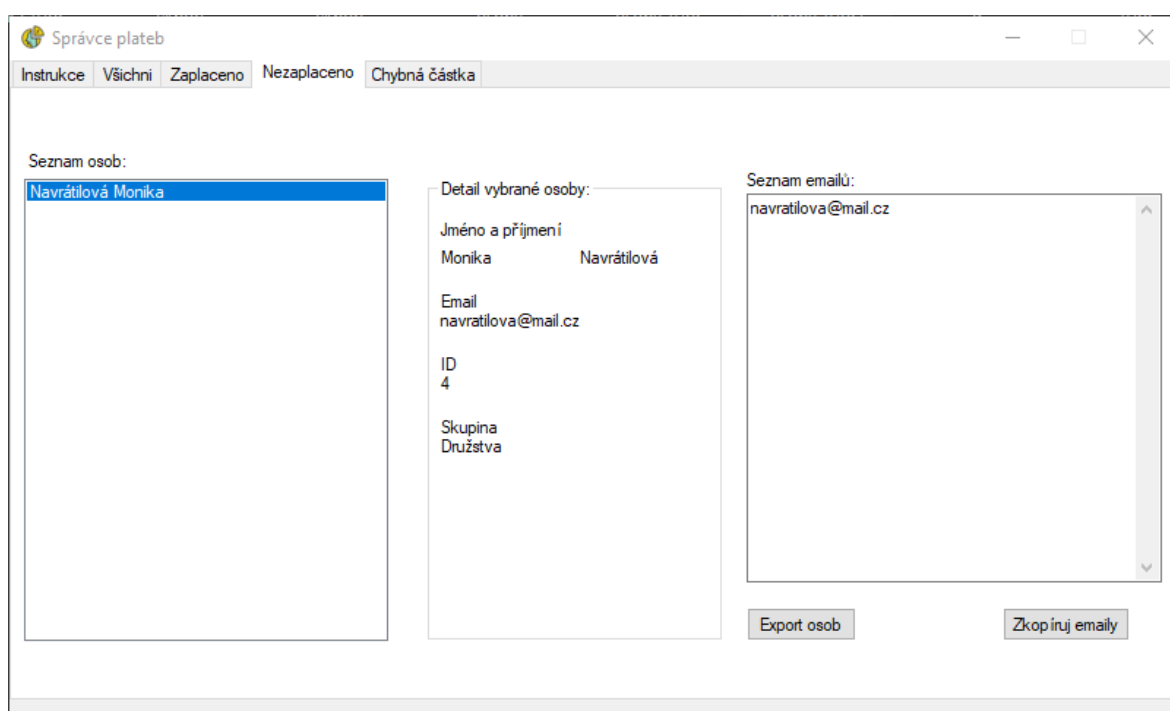
Obrázek 2.8: Chybová hláška, pro odebrání musí být vybraná nějaká osoba



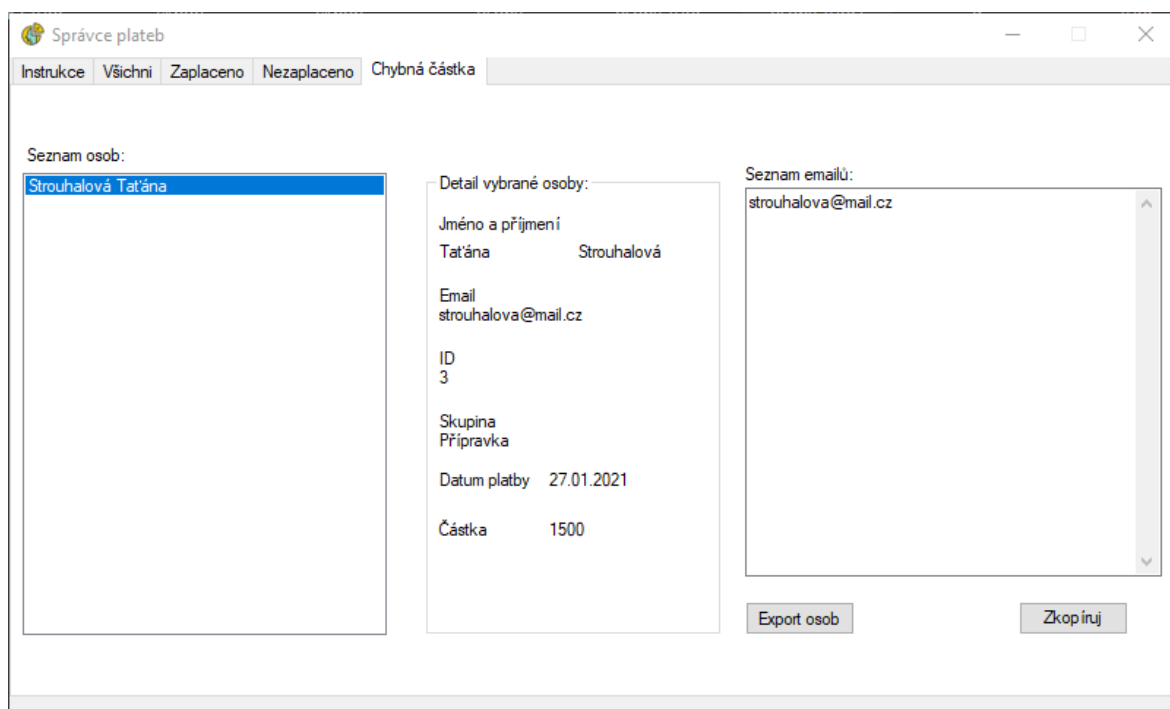
Obrázek 2.9: Karta zobrazující vybranou osobu



Obrázek 2.10: Karta zobrazující osoby, které zaplatili (po kontrole plateb)



Obrázek 2.11: Karta zobrazující osoby, které nezaplatili (po kontrole plateb)



Obrázek 2.12: Karta zobrazující osoby, které zaplatili chybnou částku (po kontrole plateb)

3. Diskuze

Při tvorbě GUI jsem narážel na problém s aktualizací prvku listBox po znovuotevření karty (například pokud je do databáze přidána nová osoba). Toto jsem vyřešil tím, že se po kliknutí na danou kartu listBox vymaže a načte znovu. Toto řešení má ale úskalí v případě většího množství vykreslovaných osob, kde zpomaluje běh programu. Pro rozsah dat na kterém bude program využíván jsem při testování na žádné problémy s rychlostí nebo stabilitou nenarazil.

Celkové řešení pomocí pásu karet, které se z počátku zdálo jako přehledné zpětně hodnotím jako neefektivní. Zejména z důvodu duplikace (jednotlivé karty se od sebe příliš neliší). Proto by možná bylo lepší filtrovat data například pomocí prvku checkBox, kde by si uživatel zaškrtoval jednotlivé vrstvy filtrů. Stejně tak by mohlo fungovat tlačítko pro export, kde by si uživatel v novém okně vybral jaká data chce exportovat.

Nicméně si myslím, že program je uživatelsky přívětivý a plní očekávanou funkci. Práci na projektu považuji za přínosnou a domnívám se, že jsem zadání splnil.

Závěr

V rámci projektu jsme vytvořili program pro automatické zpracování plateb. Program dokáže v uživatelsky přívětivé podobě přiřadit platby z výpisu účtu jednotlivých osobám v databázi.

Seznam použité literatury

- [12] *How to change listing caption?* [online], [cit. 2021-11-13]. 2012. URL: <http://tilia.wikidot.com/citace-internetovych-zdroju-v-bibtexu>.
- [aut21] Tilia Cordata authors. *Citace internetových zdrojů v BibTeXu*. [online], [cit. 2021-11-12]. 2021. URL: <http://tilia.wikidot.com/citace-internetovych-zdroju-v-bibtexu>.
- [Čápa] David Čápka. *Lekce 4 - Uložení objektů do CSV v C#*. [Online; navštíveno 1. 01. 2022]. URL: <https://www.itnetwork.cz/csharp/soubory-a-sit/c-sharp-tutorial-ulozeni-objektu-do-textoveho-souboru-csv>.
- [Čápb] David Čápka. *Lekce 5 - Uložení objektů do CSV v C# část 2*. [Online; navštíveno 1. 01. 2022]. URL: <https://www.itnetwork.cz/csharp/soubory-a-sit/c-sharp-tutorial-ulozeni-objektu-do-textoveho-souboru-csv-dokonzeni/?all-comments#comments>.
- [Mic21] Microsoft. *Dokumentace k jazyku C#*. [online], [cit. 2021-11-12]. 2021. URL: <https://docs.microsoft.com/cs-cz/dotnet/csharp/>.
- [Ove21] Overleaf. *Code listing*. [Online; cit. 12. 11. 2021]. 2021. URL: https://cs.overleaf.com/learn/latex/Code_listing.
- [sta] stackoverflow. *Windows Forms event On Select Tab?* [Online; navštíveno 1. 01. 2022]. URL: <https://stackoverflow.com/questions/2921483/windows-forms-event-on-select-tab/2921522>.
- [Wik20] Wikipedie. *CSV — Wikipedie: Otevřená encyklopedie*. [Online; navštíveno 1. 01. 2022]. 2020. URL: <https://cs.wikipedia.org/w/index.php?title=CSV&oldid=18612073>.

Seznam obrázků

2.1	Vzor struktury databáze osob	4
2.2	Vzor struktury tabulky s přehledem plateb	4
2.3	Vzor struktury exportovaných dat	4
2.4	Karta s instrukcemi	22
2.5	Karta s instrukcemi po nahrání souborů	22
2.6	Chybová hláška informující uživatele o špatném vstupním formátu	23
2.7	Karta zobrazující všechny osoby, bez výběru	23
2.8	Chybová hláška, pro odebrání musí být vybraná nějaká osoba	23
2.9	Karta zobrazující vybranou osobu	24
2.10	Karta zobrazující osoby, které zaplatili (po kontrole plateb)	24
2.11	Karta zobrazující osoby, které nezaplatili (po kontrole plateb)	25
2.12	Karta zobrazující osoby, které zaplatili chybnou částku (po kontrole plateb) . . .	25

Seznam zdrojových kódů

2.1	Zdrojový kód třídy Osoba	5
2.2	Zdrojový kód třídy Banka	6
2.3	Zdrojový kód třídy Databáze	7
2.4	Zdrojový kód třídy Form1	12