VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

**PŮJČOVNA FILMŮ**

Semestrální práce

Tomáš Cink

2016/2017

Aplikovaná informatika

Programování desktopových aplikací

# Slovní zadání

Snaha o vytvoření programu půjčovny filmů. Program bude obsahovat záznamy filmů, režie, žánrů, počtu kopii a ceny za půjčení. U každé výpůjčky bude informace o celkové ceně půjčky a data vrácení. U každého vrácení se bude kontrolovat datum, podle kterého se bude vypočítávat penále v případě opoždění. Budeme moci vyhledávat podle názvu filmu, režie a žánru a vždycky bude možné vybrat film z vyhledávání a zobrazit jej v hlavní části programu pro usnadnění bude také možnost o seřazování položek. Pro úpravu a přidávání nových filmů bude vypracován speciální dialogové okno a bude také možnost odstranit filmy a taktéž ukládání a otevírání do xml souborů.

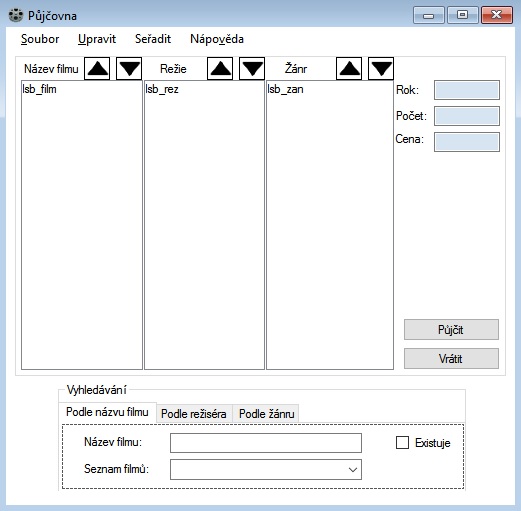
# Zadání

* Dialogová okna na hlavní program, úpravy a půjčky
* Třida pro filmy a půjčky
* Vyhledávání
* Seřazování
* Počítání ceny a penále
* Snaha o čisté a funkční UI

##### **Popis Programu**

V této části si ukážeme dialogová okna popíšeme jejich komponenty a vysvětlíme si jeho funkce

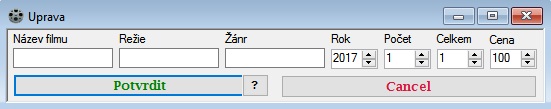
Hlavní okno:



Obsahuje:

* 3x listbox na hlavní položky(Název, Režie, Žánr)
* 3x textbox na zobrazení informací o roku výroby, počtu kusů a ceny k vybraného filmu
* pro Otevírání, Ukládání souborů, úpravy filmů a seřazování(+ zobrazení tlačítek na seřazení)
* Groupbox vyhledávání s TabControlem pro každou položku
* Tlačítka na Půjčku a Vrácení

Dialog úprav:

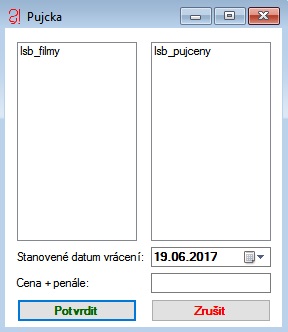


Obsahuje:

* Texboxy na název, režii a žánr
* NumericUpDown komponenty na rok, počet, celkem a cenu
* Tlačítka na potvrzení, ukončení a nápovědy
* Okno upraveno tak, by se nedal zadat prázdný(název, režie, žánr) nebo stejný film. Rok je omezen na určitou množinu čísel. Počet navyšuje celkový počet, když ho přesáhne, aby nedocházelo k situacím, že je větší počet než kolik jich je celkem s minimální hodnotou 0. Cena má nastavenou základní hodnotu na 100, kterou je možné upravovat s minimální hodnotou 1
* Tlačítko potvrdit se aktivuje jenom tehdy kdy jsou slněny všechny požadavky
* Požadavky jsou vypsány po stisknutí ? tlačítka a toto tlačítko zmizí po splnění požadavků

Dialog Půjčka:

Obsahuje:



* DateTimePicker pro kontrolu vrácení a informaci, kdy se mají filmy vrátit při půjčce
* 2x Textbox na cenu a penále na stejných pozicích, zobrazí se vždy jen to, které je potřeba
* 2x tlačítko na potvrzení volby a zrušení okna

**Třídy:**

* V této části probereme základ tohoto programu a to třídu Pujcen

public class Film

{

string nazev;

string rezie;

string zanr;

int rok;

[XmlElement("Nazev")]

public string Nazev { get { return nazev; } set { nazev = value; } }

[XmlElement("Rezie")]

public string Rezie { get { return rezie; } set { rezie = value; } }

[XmlElement("Zanr")]

public string Zanr { get { return zanr; } set { zanr = value; } }

[XmlElement("Rok")]

public int Rok { get { return rok; } set { rok = value; } }

public Film() { }

public Film(string n, string r, string z,int y)

{

this.nazev = n;

this.rezie = r;

this.zanr = z;

this.rok = y;

}

}

public class Pujcen : Film

{

public Pujcen(string n, string r, string z, int y, int p, decimal c,int celk) : base(n, r, z, y)

{

this.pocet = p;

this.cena = c;

this.celkem = celk;

}

public Pujcen() { }

int pocet;

decimal cena;

int celkem;

[XmlElement("Celkem")]

public int Celkem { get { return celkem; } set { celkem = value; } }

[XmlElement("Pocet")]

public int Pocet { get { return pocet; } set { pocet = value; } }

public void PlusPocet() { pocet++; }

public void MinusPocet() { pocet--; }

[XmlElement("Cena")]

public decimal Cena { get { return cena; } set { cena = value; } }

}

* Třída Pujcen je zalozena na parenské třídě Filmy v této třídě jsou položky Název, Režie, Žánr a Rok ke každému je vytvořen XmlElement na pojmenování položek v XML souboru a je zde také vytvořen základní konstruktor Film ze kterého vychází konstruktor pro třídu Pujcen
* Obě třídy mají funkce pro vnější nastavování a čtení všech parametrů (get;set)
* Třída Pujcen obsahuje navíc počet, celkem a cenu a navíc funkce pro přídávání a snižování počtu

Hlavní list: List<Pujcen> data = new List<Pujcen>();

Refresh Listboxu:

private void refresh()

{

mazani();

foreach (Pujcen n in data)

{

lsb\_film.Items.Add(n.Nazev);

if (lsb\_rez.Items.Contains(n.Rezie) != true)

lsb\_rez.Items.Add(n.Rezie);

if (lsb\_zan.Items.Contains(n.Zanr) != true)

lsb\_zan.Items.Add(n.Zanr);

}

}

private void mazani()

{

lsb\_film.DataSource = null;

lsb\_film.Items.Clear();

lsb\_rez.DataSource = null;

lsb\_rez.Items.Clear();

lsb\_zan.DataSource = null;

lsb\_zan.Items.Clear();

tb\_f.Text = "";

tb\_zf.Text = "";

tb\_jmenoFind.Text = "";

}

* Slouží k obnově dat v listboxu
* Na začátku se všechna data vymažou a poté znovu nahrají

**Položky menu**

Soubor

Nový:

* Odstraní všechny data z listu data

Otevřít

* Obsahuje 2 možnosti:
  + Zvolit… - otevře soubor xml
  + Backup file – načte předdefinované filmy

Ukládání a Otevírání souboru XML

* V této části si ukážeme jak je řešeno ukládání a načítání dat
* Většina dat je uložena do Listu data, který je také využit pro ukládání i otevírání dat
* Využívájí se třídy System.Xml.Linq a System.Xml.Serialization

Ukládání:

private void uložitToolStripMenuItem\_Click(object sender, EventArgs e)

{

saveFileDialog.Title = "Uložit jako ... ";

saveFileDialog.AddExtension = true;

saveFileDialog.Filter = "Textové soubory (\*.xml)|\*.xml";

if (saveFileDialog.ShowDialog() == DialogResult.OK)

{

try

{

var ser = new XmlSerializer(typeof(List<Pujcen>));

TextWriter writer = new StreamWriter(saveFileDialog.FileName);

ser.Serialize(writer, data);

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

else

Close();

}

* Pracujeme s ukládacím dialogem saveFileDialog, kterému po stisku tlačítka uložit se změní název dialogu na „Uložit jako … “, přidá se při ukládání koncovka souboru a nastaví se filtr pouze pro XML, aby se předešlo ukládání jiných typů souborů.
* Když je vyhodnocen dialog stisknutím tlačítka Uložit se spustí Xmlserializer typu List
* TextWriter otevře stream pro zápis do vybraného souboru
* Serializer spustí zápis Listu data do zvoleného souboru
* Pakliže nastane chyba ukáže se chybová hláška

Otevírání:

private void zvolitToolStripMenuItem\_Click(object sender, EventArgs e)

{

openFileDialog.Title = "Zdrojový soubor";

openFileDialog.Filter = "Textové soubory (\*.xml)|\*.xml";

openFileDialog.InitialDirectory = Application.StartupPath; DialogResult rslt = openFileDialog.ShowDialog(this);

if (rslt == DialogResult.OK)

{

if (File.Exists(openFileDialog.FileName))

{

try

{

mazani(); data.Clear();

var maps = from c in XElement.Load(openFileDialog.FileName).Elements("Pujcen") select c;

foreach (var item in maps)

{

data.Add(new Pujcen()

{

Nazev = item.Element("Nazev").Value,

Rezie = item.Element("Rezie").Value,

Zanr = item.Element("Zanr").Value,

Rok = Convert.ToInt32(item.Element("Rok").Value),

Pocet = Convert.ToInt32(item.Element("Pocet").Value),

Cena = Convert.ToDecimal(item.Element("Cena").Value),

Celkem = Convert.ToInt32(item.Element("Celkem").Value)

});

}

refresh();

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

}

refresh();

}

* Pracujeme s dialogem pro otevírání souboru OpenFileDialog
* Změníme název okna, natavíme filtr koncovek pouze na xml a startovací složku na místo sputěného souboru
* Když je dialog potvrzen stiknutím tlačítka Otevřít, ověří se, že soubor existuje a když ano tak se pokusí načíst do Listu data, pakliže je soubor v jiném formátu nebo data neodpovídají vyskočí chybové okno

Konec

private void konecToolStripMenuItem\_Click(object sender, EventArgs e)

{

if(MessageBox.Show(Properties.Resources.EXIT\_MESSAGE, Properties.Resources.EXIT\_TITLE,

MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

Application.Exit();

}

* Po stisknutí možnosti konec se ukáže okno, které se dotáže na ukončení a upozorní na možnou ztrátu dat
* Pakliže se potvrdí aplikace se uzavře, když se zruší aplikace pokračuje

Upravit

* Další část v menu obsahující:
  + Upravit
  + Přidat
  + Odebrat
* Pracují s dialogem Uprava

Upravit

private void upravitToolStripMenuItem1\_Click(object sender, EventArgs e)

{

if (lsb\_film.SelectedIndex >= 0)

{

Uprava uprava = new Uprava();

uprava.Text = "Upravit film";

lisboxpresun();

uprava.X = c;

foreach (Pujcen n in data)

{

if ((object)lsb\_film.SelectedItem == (object)(n.Nazev))

{

uprava.Nazev = n.Nazev;

uprava.Rezie = n.Rezie;

uprava.Zanr = n.Zanr;

uprava.Rok = n.Rok;

uprava.Pocet = n.Pocet;

uprava.Cena = n.Cena;

uprava.Celkem = n.Celkem;

}

}

if (uprava.ShowDialog() == DialogResult.OK)

{

data.RemoveAll(Name => Name.Nazev == (string)lsb\_film.SelectedItem);

data.Add(new Pujcen(

uprava.Nazev,

uprava.Rezie,

uprava.Zanr,

uprava.Rok,

uprava.Pocet,

uprava.Cena,

uprava.Celkem));

refresh();

}

}

}

* Funguje pouze když je vybraná položka v lisboxu filmy
* Spouští se dialog Uprava, který přejmenujeme a poté do něj skopírujeme požadovaný film + list c ve kterém není obsažen tento film kvůli kontrole duplikace
* Po potvrzení dialogu se původní film z listu data smaže a přidá se nový

Přidat

private void přidatToolStripMenuItem\_Click(object sender, EventArgs e)

{

Uprava pridat = new Uprava();

pridat.Text = "Přidat film";

pridat.X = data;

DialogResult potvrzeni = pridat.ShowDialog();

if (potvrzeni == DialogResult.OK)

{

data.Add(new Pujcen(

pridat.Nazev,

pridat.Rezie,

pridat.Zanr,

pridat.Rok,

pridat.Pocet,

pridat.Cena,

pridat.Celkem));

refresh();

}

}

* Otevře se dialog Uprava
* Přejmenuje se nadpis na Přidat film
* Nahraje se list data na kontrolu duplikací
* Když se vyhodnotí dialog OK přidá se film do listu data a listbox se obnoví

Smazat

private void odebratToolStripMenuItem\_Click(object sender, EventArgs e)

{

data.RemoveAll(Name => Name.Nazev == (string)lsb\_film.SelectedItem);

refresh();

}

* Odstraní se vybraný film

Seřadit

* Obsahuje seřazení položek:
  + A-Z – pro Název, Režie, Žánr
  + Z-A – pro Název, Režie, Žánr
  + Max – pro rok, počet a cena
  + Min – pro rok počet a cena
  + Zobrazit tlačítka – zobrazí nebo schová tlačítka seřazení
* Všechny druhy fungují na stejném kódu jenom se mění parametry využívá se metoda .Sort
* Pro aktualizaci se používá refresh na určitý listbox

A-Z

private void názevToolStripMenuItem1\_Click(object sender, EventArgs e)

{

data.Sort(delegate (Pujcen a, Pujcen b) { return a.Nazev.CompareTo(b.Nazev); });

RefFilm();}

Z-A

private void názevToolStripMenuItem1\_Click(object sender, EventArgs e)

{

data.Sort(delegate (Pujcen a, Pujcen b) { return b.Nazev.CompareTo(a.Nazev); });

RefFilm();}

Nápověda:

* Zobrazí informaci o základním použití programu

**Půjčení a vracení filmů**

* Zpracovává se po stisknutí tlačítka na hlavním programu
* Po stisku se otevře Dialog Pujcka, u kterého se změní název podle toho co vybereme
* Obsahuje 2 listy List a, List b, do kterých se načítá List data, po uzavření se tyto listy se oba složí dohromady a předají do listu data

Půjčit:

private void bt\_pujcit\_Click(object sender, EventArgs e)

{

Pujcka p = new Pujcka();

p.Text = "Půjčení";

p.zamceni();

p.A = data;

p.RefreshA();

DateTime doba = DateTime.Now.AddDays(30);

p.Datum = doba;

if (DialogResult.OK == p.ShowDialog())

{

var res = p.A.Union(p.B).ToList();

if (!p.B.Any())

MessageBox.Show("Nic jste si nevypujcili.", Properties.Resources.INFO\_TITTLE, MessageBoxButtons.OK, MessageBoxIcon.Warning);

else

MessageBox.Show(String.Concat("Vaše vybrané filmy vraťte do: ",

doba.ToString("dd.MM.yyyy"), "\nCelková cena půjčky: ", p.Pujcovne),

Properties.Resources.INFO\_TITTLE, MessageBoxButtons.OK, MessageBoxIcon.Information);

mazani();

data.Clear();

data = res;

refresh();

}

}

* Uzamkneme položku Datum, jelikož ukazuje dobu vrácení (za 30 dnů odedneška)
* Nahrajeme List data do List a
* Při vybrání filmu v listu a se přesune do listu b, ubere se počet a přičte cena
* Pří opačném úkonu se počet zvedne a cena odečte
* Po potvrzení se vytvoří zpráva o dni vrácení filmů a o celkové ceně půjčky a nebo zpráva, že jste si nic nevypůjčili

Vrátit:

private void bt\_vratit\_Click(object sender, EventArgs e)

{

Pujcka v = new Pujcka();

v.Text = "Vrácení";

v.odemc();

v.B = data;

v.RefreshB();

if (DialogResult.OK == v.ShowDialog())

{

var res = v.A.Union(v.B).ToList();

if(!v.A.Any())

MessageBox.Show("Nic jste nevrátili.",Properties.Resources.INFO\_TITTLE, MessageBoxButtons.OK, MessageBoxIcon.Warning);

else

MessageBox.Show(string.Concat("Uspěšně jste vratili filmy. ", "\nVýše penále: ", v.Penale),

Properties.Resources.INFO\_TITTLE, MessageBoxButtons.OK, MessageBoxIcon.Information);

mazani();

data.Clear();

data = res;

refresh();

}

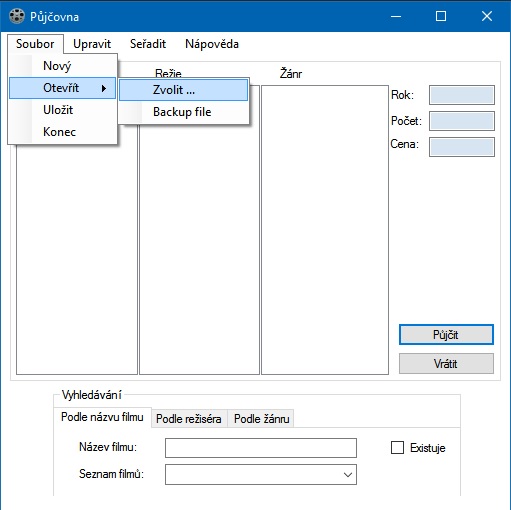
* Obdoba jak při půjčování akorát se list data nahrává na pravou stranu do listu b
* DateTimePicker slouží ke kontrole datumu vrácení
* Místo ceny se počítá penále 20kč\*počet vrácených filmů, když byli vráceny pozdě

**Vyhledávání**

* Vyhodnocuje se automaticky po změně textu v texboxu
* Ignoruje velká a malá písmena: StringComparison.OrdinalIgnoreCase;
* Výsledky jsou zobrazovány v comboboxu, ve kterém lze nalezený film vybrat a ten se posléze zatrhne v hlavní části programu a ukáží se veškeré informace o tomto filmu
* Při každé změně textu se tento combobox vysype a naplní novými daty
* Pro vyhledávání se využívá funkce. IndexOf() do které zadáme požadovaný řetězec k vyhodnocení a informace o ignoraci velikosti písmen

**Ukázka funkce Programu**

* V této části si ukážeme, jak tento program funguje



spustíme program

Soubor Otevřít Zvolit… s

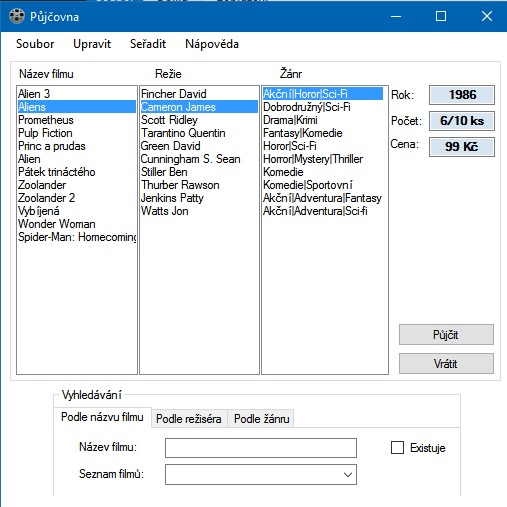
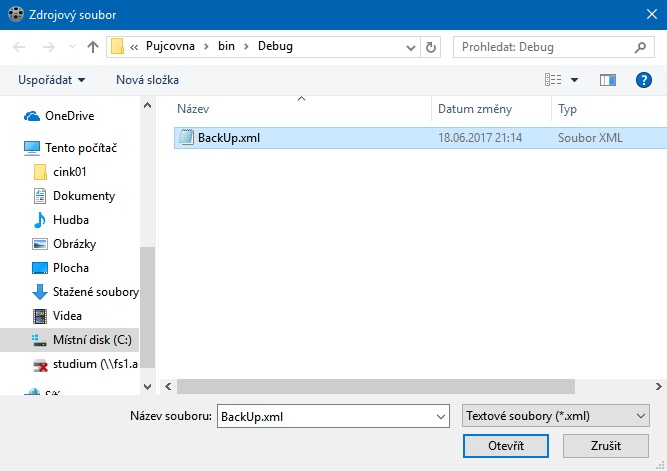
Vybereme soubor

Například:

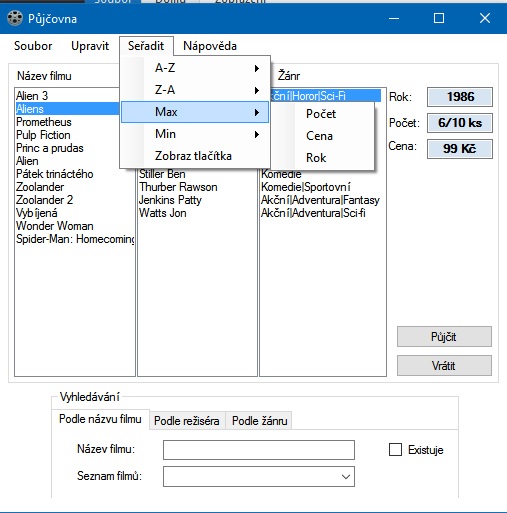
zvolíme BackUp.xml

potvrdíme stisknutím

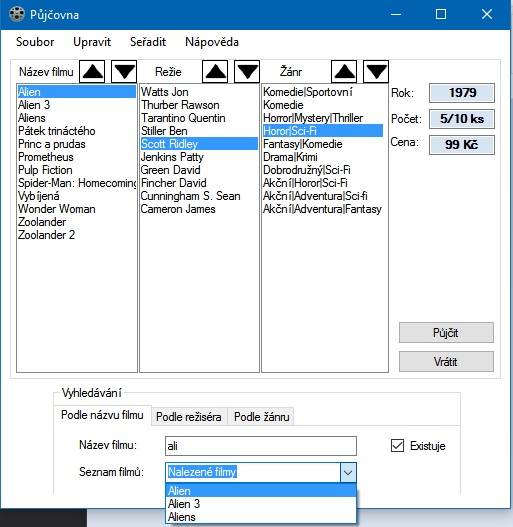
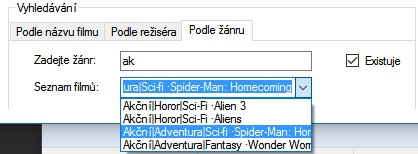
otevřít



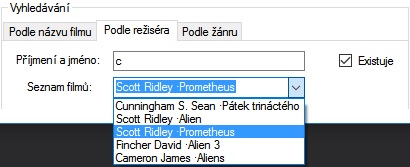
* Stisknutím na název si můžeme zobrazit dodatečné informace o filmu
* Dále se můžeme vrhnout na řazení a vyhledávání

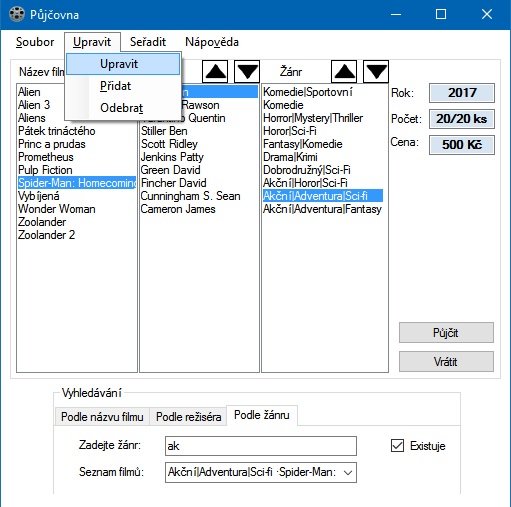


pomocí upravit můžeme libovolně srovnávat položky

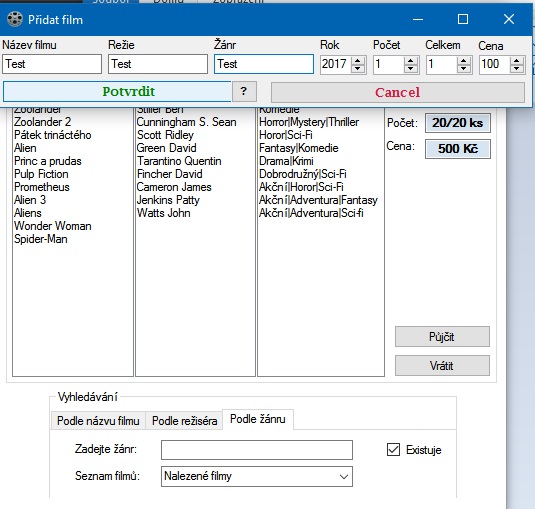
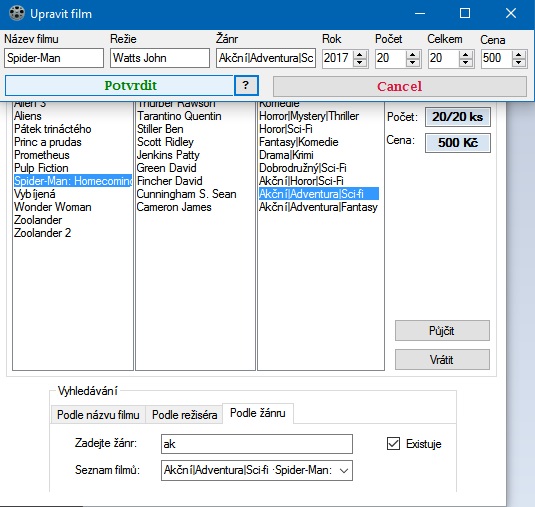


Pomocí vyhledávání můžeme najít požadovaný film, režiséra nebo žánr





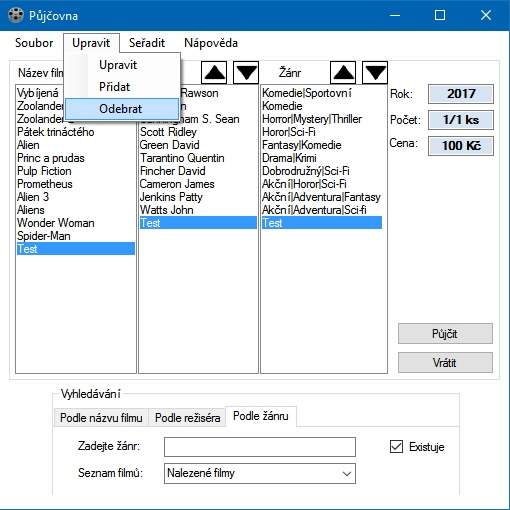
Pomocí upravit můžeme měnit stávající film podle našich představ



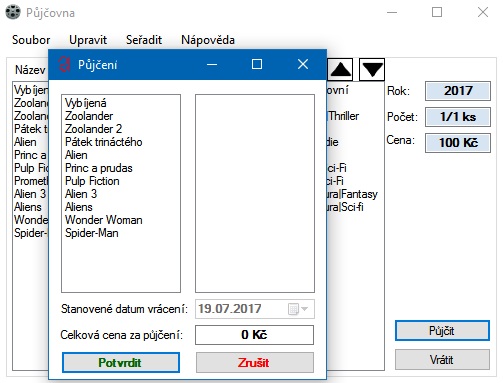
Podobně pak můžeme i přidávat

Nový film

případně odstraňovat



* Po všech úpravách se můžeme vrhnout na půjčky a vracení



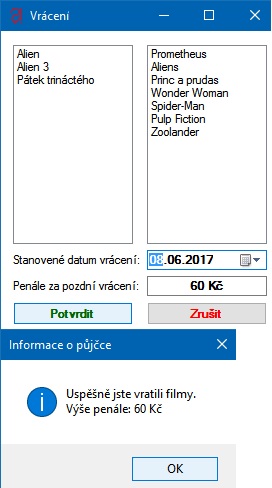
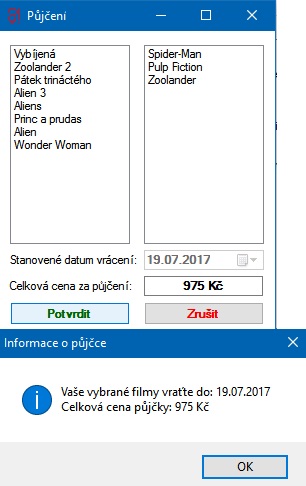
Stiskneme Půjčit

Klikáme na filmy, které si hodláme půjčit

Po vybráni klikneme na potvrdit

Jestli chceme vrátit zvolíme Vrátit

Navolíme položky, které chceme vrátit, zadáme datum na potvrdíme



Obojí nám vrátí zprávu s informací o půjčce/vrácení