



Vysoká škola  
polytechnická  
Jihlava

# VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Katedra technických studií  
Obor Aplikovaná informatika

## P l á n o v a č ú k o l ů a u d á l o s t í

dokumentace k semestrální práci  
Programování desktopových aplikací

Jméno a příjmení: Jiří Klusáček  
E-mail: klusacek@student.vspj.cz  
Datum: 22. června 2017

# Obsah

1	Zadání a požadavky na aplikaci .....	3
2	Návrh řešení .....	3
2.1	Úkol.....	3
2.2	Událost .....	3
2.3	Ukládání dat v aplikaci.....	4
2.4	Další formuláře .....	4
3	Způsob a popis implementace .....	4
3.1	Úkoly a události .....	4
3.2	Hlavní formulář.....	5
3.3	Přidávání a editace úkolů a událostí.....	6
3.4	Přehled všech úkolů a událostí.....	7
3.5	Správa druhů .....	7
4	Zhodnocení a závěr.....	7

# 1 Zadání a požadavky na aplikaci

Plánovač úkolů a událostí bude uchovávat úkoly a události. Úkoly se liší od událostí pouze přidáním času. Jako úkol se představuje například odevzdání semestrální práce na elearning v určité datum a čas. Jako událost se představuje například narozeniny člena rodiny (není nezbytné uvádět čas).

Další požadavky na aplikaci:

- Dědičnost v aplikaci
- Využití více oken aplikace (například přidávání a úprava položek seznamu)
- Práce se soubory (načítání položek ze souboru XML)
- Řazení (aplikace bude umět řadit položky)
- Atraktivní a živý design
- Menu aplikace

## 2 Návrh řešení

Pro správný chod aplikace bude potřeba vytvořit objekty, do kterých budeme ukládat data. Dle zadání vychází dva objekty.

- Úkol
- Událost

### 2.1 Úkol

Úkol představuje činnost, kde je kromě data důležitý i čas. Takovým jednoduchým případem je typická zkouška na vysoké škole. V objektu úkol evidujeme tyto informace:

- Název úkolu
- Druh
- Datum
- Čas
- Podrobnosti

Z logiky věci vyplývá, že název úkolu bude povinně zadávaný. Dalšími povinnými atributy budou datum a čas. Druh je tzv. nepovinný, stejně jako podrobnosti.

### 2.2 Událost

Událost je činnost nebo pouze připomínka, kde je jenom důležitý datum. Asi nejběžnějším případem jsou narozeniny. V tomto objektu proto evidujeme tyto informace:

- Název události
- Druh

- Datum
- Podrobnosti

Podobně jako u úkolu jsou povinnými položkami název události a datum. Nepovinnými položkami budou druh a podrobnosti.

## 2.3 Ukládání dat v aplikaci

Možným řešením by bylo evidovat objekty v poli nebo v arraylistu. Pole má však jednu podstatnou nevýhodu, a to pevná velikost pole. Pevnou velikost pole však nemusí mít arraylist, ale toto pole by se nedalo dobře použít při práci s XML. Asi nejlepším řešením je využití seznamu s typem úkol nebo událost. Tento seznam je výhodný při práci s XML a v této aplikaci je to velmi podstatná věc. Tyto seznamy budou vždy seřazené vzestupně podle data, aby se v dalším průběhu programu nemuselo řazení řešit.

Aby se nevyskytovalo mnoho podobných názvu druhů například jeden s mezerou nebo s pomlčkou budeme evidovat i druhy v seznamu.

## 2.4 Další formuláře

Pro přehlednost a intuitivní ovládání aplikace bude aplikace obsahovat i další formuláře a dialogová okna:

- Formulář pro přidávání a editaci úkolů a událostí.
- Formulář pro přehled všech úkolů a událostí, které bude možno filtrovat podle druhů.
- Formulář pro správu druhů, kde bude možno vymazat již nepoužívané druhy.
- Dialogová okna pro importování a exportování dat do souborů typu XML.

# 3 Způsob a popis implementace

## 3.1 Úkoly a události

Pro úkoly a události jsem vytvořil zvláštní soubor UkolyAUdalosti.cs, kde je v jedné třídě implementována Událost. Událost má parametry:

- Datum – parametr typu DateTime pro uchování data
- Název – parametr typu string pro uchovávání názvu události
- Druh – parametr typu string pro uchovávání druhu události
- Popis – parametr typu string pro uchovávání případných podrobností o události

Dále třída obsahuje parametrický konstruktorem pro vytvoření instance tohoto objektu a bezparametrický konstruktorem pro práci se soubory typu XML. Kromě toho obsahuje i metodu pro správný výpis data v dlouhém formátu.

Další třídou je rozšiřující třída Úkol, která implementuje všechny parametry ze třídy Událost. Třída Úkol má oproti předchozí třídě navíc tyto parametry:

- Hodina – parametr typu integer pro uchovávání hodiny
- Minuta – parametr typu integer pro uchovávání minuty

Pozn.: Samozřejmě, že by šlo využít pro datum i čas pouze jeden parametr, ale práce ve formuláři mi přišla příliš nepřehledná.

Třída Úkol má také konstruktor, který rozšiřuje konstruktor předchozí třídy. Pro práci se soubory typu XML také využíváme bezparametrický konstruktor. Jedinou metodou v této třídě je metoda pro správný výpis času ve tvaru HH:MM.

## 3.2 Hlavní formulář

V aplikaci uchováváme seznam událostí, seznam úkolů a seznam druhů. Při spuštění aplikace se do těchto seznamů automaticky načtou data ze souborů XML. To samé platí i pro uložené druhy. Ty se načtou z textového souboru do pole, které se transformuje do seznamu.

Při načtení formuláře se pomocí funkcí vygeneruje datum a pozdrav a provede se nastavení data v kalendáři na dnešní datum.

Při změně data na kalendáři se aktualizuje i ListView se zobrazenými úkoly a událostmi. Navíc ke každé položce se přiřadí index (pořadí v seznamu) pro snadnější vyhledávání položek při úpravě a odstraňování položky. Pokud na tento den není naplánovaná žádná událost nebo činnost zobrazí se nápis, že nejsou data k dispozici. To samé platí i pro tlačítka. Pokud není vybrána nějaká událost nebo úkol tak jsou tlačítka pro úpravu a odstranění deaktivována a lze pouze jen přidávat. Při výběru nějaké události nebo úkolu se zobrazí podrobnosti, pokud jsou k dispozici.

Po kliknutí na tlačítko přidat se zobrazí dialogové okno pro přidání události nebo úkolu, dle toho, na jaké kartě se uživatel nachází (popis tohoto okna bude následovat). Stejné dialogové okno se zobrazí i při úpravě, ale má již nastavené parametry upravovaného úkolu. Při úpravě se odstraní již zadaný úkol nebo událost a vytvoří se nová událost nebo úkol. Při jakékoliv úpravě a přidávání se seznamy znovu seřadí a aktualizuje se i ListView. Dále se mohou zobrazit informace metodou toString o vybrané položce formou MessageBoxu.

V horním menu lze zobrazit přehledy všech úkolů a událostí. Pak lze spravovat druhy, se kterými se pracuje. Popisy těchto oken budou následovat. Dále pomocí SaveFileDialogu lze exportovat úkoly a události do XML. Také lze pomocí OpenFileDialogu importovat úkoly a události ze souborů XML, avšak hrozí, že neuložená data budou přepsána daty novými. Poslední položkou menu je položka O aplikaci, kde je přístupná tato dokumentace ve formátu pdf.

Při ukončení práce s formuláři se data ze seznamů událostí a úkolů uloží do XML souboru. To samé platí i pro druhy, které se uloží do textového souboru.

### 3.3 Přidávání a editace úkolů a událostí

Tento formulář se využívá při přidávání a editaci úkolů a událostí.

Dialogové okno políčko pro název, který nesmí být prázdný. Dále obsahuje radioButtons, pomocí kterých určíme druh. Druh může být prázdný nebo z již předdefinovaných druhů nebo lze vytvořit nový druh. Další podstatnou částí je výběr data a času. Datum vybíráme pomocí datePickeru, který zabezpečuje to, aby nebylo vybráno datum, které neexistuje, a kde lze vybrat datum přímo v kalendáři. Čas je tvořen dvěma vlastními prvky založené na numericUpDown. Pouze je zabezpečeno, aby šla zadat v případě hodin hodnota od 0 do 23 a v případě minut od 0 do 59 a také to, aby se hodnoty hodin a času zobrazovaly dvěma číslicemi. Poslední komponentou je místo pro zadávání podrobností, ovšem tyto podrobnosti nejsou povinné.

Při spuštění formuláře pro **vložení nové položky** se nastaví zobrazení pro úkol nebo událost dle toho, na jaké kartě uživatel stiskne tlačítko. Na tomto formuláři se podle toho nastaví popisky položek a název formuláře. Poté se zatrhne implicitně druh nastaví na „Neurčit druh“. Do comboBoxu se ještě vloží druhy ze seznamu druhů. Datum se nastaví na hodnotu, které je vybrané na kalendáři v hlavním okně. Hodina a minuta se nastaví na aktuální čas.

Po nastavení a stisknutí OK se provede validace názvu a načte se druh z formuláře. Určí se také, jakým způsobem byl zadán druh a popřípadě nový druh projde validací a přidá se do seznamu druhů. Poté se vytvoří položka události nebo úkolu a tato položka se přidá do příslušného seznamu. Tento seznam se seřadí podle data vzestupně a aktualizuje se zobrazený seznam.

Při spuštění formuláře pro **úpravu položky** se okno nastaví opět podle toho, zda se jedná o událost nebo úkol, tak jako se nastavuje v případě nové položky. Dále se podle indexu, který náleží každé položce nastaví název, druh, datum, popřípadě čas a podrobnosti. Všechny položky lze editovat.

Po novém nastavení a stisknutí OK se provede validace dat a načtení dat tak jako u nové položky. Dále se vytvoří objekt. Ještě před samotným vložením dat se provede odstranění starého úkolu a vložení upraveného úkolu. Poté se opět provede řazení a aktualizuje zobrazený seznam.

**Odstranění položky** se neprovádí přes formulář, ale pouze se vyhledá index označené položky. Poté se provede samotné odstranění. Znovu řazení není potřeba provádět, ale pouze se aktualizují zobrazené položky.

### 3.4 Přehled všech úkolů a událostí

Formulář slouží pouze pro zobrazení všech úkolů a událostí, které jsou uloženy v seznamech. Pomocí ovládacích prvků lze vybrat jaké druhy úkolů a událostí se budou zobrazovat. Implicitně je zobrazování nastaveno na netřídění. Pak lze vybrat možnost zobrazit pouze události a úkoly, které nemají určený druh. Poslední možností je vybrat pouze druh, který chce uživatel zobrazit. Panel pro zobrazované úkoly je řešen podobně jako v hlavním okně, takže lze přepínat mezi úkoly a událostmi.

Do přehledu se vždy naimportuje aktuální seznam úkolů a událostí, které se zobrazí. Stejně to platí i pro druhy, které se načtou do comboBoxu.

V tomto přehledu nelze položky přidávat, editovat a odstraňovat.

### 3.5 Správa druhů

Tato aplikace slouží se správě druhů. Je možné, že uživatel bude rozšiřovat druhy při zadávání úkolů a událostí a přestane mít o všech druzích přehled, proto ty druhy, které už nebudou používány lze odstranit.

Formulář obsahuje listBox a dvě tlačítka. Jedno tlačítko pro vymazání a druhé pro ukončení aplikace.

Při označení druhu, který chceme odstranit se provede kontrola, zda opravdu druh nějaká položka neobsahuje. Pokud ano aktivuje se tlačítko pro smazání, v jiném případě toto tlačítko zůstane deaktivované. Při samotném smazání se odebere ze seznamu druhů právě smazaný druh.

## 4 Zhodnocení a závěr

Podle mého názoru je celá aplikace pěkně graficky vyřešená a má jednoduché ovládání v podobě výběru data a třech tlačítek. Dále jsem se snažil, aby všechny komponenty byly jasně pojmenované a bylo jasné, co komponenty dělají.

Znalosti z předmětu jsem aplikoval do této aplikace a rozšířil jsem tuto aplikaci i o další funkcionality programovacího jazyka C#. Využil jsem komponentu ListView a dále také komponenty pro práci s časem a datem. V aplikaci jsem také vytvořil i vlastní komponentu na základě NumericUpDown, která zobrazuje hodiny a minuty formou dvou číslic.

Dále aplikace velmi často pracuje se soubory. Už při samotném spuštění se načtou data ze souborů formátu XML a názvy druhů z textového souboru. Stejně tak se načtou i při práci s ostatními formuláři. Dále se i tato dokumentace dá otevřít z aplikace.