

# **Nástroj pro tvorbu dialogových oken pomocí wxWidgets**

Tool for creating dialog windows by the help of wxWidgets

Josef Trbušek

---

Bakalářská práce  
2009



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav aplikované informatiky  
akademický rok: 2008/2009

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Josef TRBUŠEK**  
Studijní program: **B 3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**

Téma práce: **Nástroj pro tvorbu dialogových oken pomocí wxWidgets**

Zásady pro vypracování:

1. Vypracujte literární rešerši na dané téma.
2. Seznamte se s prostředím wxWidgets.
3. Popište použitá programová prostředí wxWidgets a C++.
4. Popište používaný formát souborů XRC.
5. Vytvořte program pro tvorbu a editaci dialogových oken, která budou použita pro strojírenské výpočty.

Rozsah práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. SMART, Julian, HOCK, Kevin, CSOMOR, Stefan. Cross-Platform GUI Programming with wxWidgets. [s.l.] : [s.n.], 2005. 700 s.
2. WxWidgets [online]. c1997 , 2009-01-06 [cit. 2009-01-19]. Dostupný z WWW: <http://www.wxwidgets.org/>
3. WxWidgets Czech Translate [online]. 2008 , 22. 12. 2008 [cit. 2009-01-19]. Dostupný z WWW: [http://www.wxwidgets.hustej.net/wiki/index.php/Hlavn%C3%AD\\_strana\\_C++](http://www.wxwidgets.hustej.net/wiki/index.php/Hlavn%C3%AD_strana_C++)
4. Builder [online]. c1997-2003 [cit. 2009-01-19]. Dostupný z WWW: <http://www.builder.cz/>
5. XML-based resource system overview [online]. [1997] [cit. 2009-01-19]. Dostupný z WWW: [http://docs.wxwidgets.org/stable/wx\\_xrcoverview.html](http://docs.wxwidgets.org/stable/wx_xrcoverview.html)

Vedoucí bakalářské práce:

**Ing. Pavel Pokorný, Ph.D.**

Ústav aplikované informatiky

Datum zadání bakalářské práce:

**20. února 2009**

Termín odevzdání bakalářské práce:

**1. června 2009**

Ve Zlíně dne 13. února 2009

prof. Ing. Vladimír Vašek, CSc.

*děkan*



doc. Ing. Ivan Zelinka, Ph.D.

*ředitel ústavu*

## ABSTRAKT

Tato bakalářská práce je zaměřena na tvorbu editoru dialogových oken pomocí knihovny wxWidgets. Popisuje tuto knihovnu, její historii a způsob použití při programování pomocí jazyku C++. Dále popisuje formáty použité při realizaci praktické části. Praktická část se zabývá použitím vytvořeného programu a jeho nasazením v praxi a dále popisem programu z pohledu uživatele a programátora.

Cílem práce je vytvořit program, který bude sloužit k tvorbě dialogových oken, jeho popis a způsob využití.

Klíčová slova: wxWidgets, dialogová okna.

## ABSTRACT

This bachelor thesis aims at creating a dialog windows editor by using a wxWidgets library. The thesis describes this library, its history and a way it is used in programming when C++ language is used. Further, it describes formats used in the practical part. The practical part deals with the usage of the created program and its application in practice work. It also deals with a description of the program from the user's and programmer's points of view.

The goal is to create a program which will be used to create dialog windows, its description and a way of usage.

Keywords: wxWidgets, dialog windows, XRC, XML

Úvodem bych rád poděkoval svému vedoucímu práce, panovi Ing. Pavlovi Pokornému, Ph.D. za pomoc a rady při tvorbě bakalářské práce. Dále bych rád poděkoval všem, kteří jakkoliv přispěli k vytvoření této práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval.

V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>10</b>
<b>1 wxWidgets</b> .....	<b>11</b>
1.1 Popis.....	11
1.2 Historie.....	12
1.3 Architektura.....	13
1.4 Programy na základě wxWidgets.....	15
<b>2 Jazyk C/C++</b> .....	<b>16</b>
2.1 Stručná historie jazyka C/C++.....	16
2.2 Výhody objektově orientovaného programování.....	17
2.3 Použití wxWidgets v C++.....	18
<b>3 Formát XRC</b> .....	<b>19</b>
3.1 Seznámení s XRC.....	19
3.2 Popis souboru XRC.....	20
3.3 Povolené prvky a elementy v XRC.....	20
<b>II PRAKTICKÁ ČÁST</b> .....	<b>27</b>
<b>4 Popis programu</b> .....	<b>28</b>
<b>5 Program z pohledu uživatele</b> .....	<b>29</b>
5.1 Hlavní menu.....	30
5.1.1 Menu Soubor.....	30
5.1.2 Menu Nástroje.....	30
5.1.3 Menu Nápověda.....	31
5.2 Boční panel.....	31
5.2.1 Záložka Formulář.....	31
5.2.2 Záložka Možnosti.....	31
5.2.3 Záložka Položky.....	32
5.3 Manipulace s oknem a prvky okna.....	34
<b>6 Program z pohledu programátora</b> .....	<b>35</b>
6.1 Základní datové typy.....	35
6.2 Statické funkce.....	37
6.3 Soubor pro uložení konfigurace programu.....	37
6.4 Vytvoření hlavního okna programu.....	38
6.5 Funkce pro ovládání z menu.....	39

6.5.1	Vytvoření nového okna.....	39
6.5.2	Načtení již uloženého projektu.....	39
6.5.3	Uložení stávajícího projektu.....	40
6.5.4	Funkce menu Nástroje a Nápověda.....	40
6.6	Funkce bočního panelu.....	41
6.6.1	Vytvoření okna.....	41
6.6.2	Vkládání prvků.....	41
6.6.3	Nastavení parametrů prvků.....	42
6.7	Manipulace s oknem a prvky.....	43
6.7.1	Změna velikosti okna.....	43
6.7.2	Manipulace s prvky návrhového okna.....	43
<b>7</b>	<b>Kompatibilita s jinými programy.....</b>	<b>45</b>
	<b>ZÁVĚR.....</b>	<b>46</b>
	<b>ZÁVĚR V ANGLIČTINĚ.....</b>	<b>47</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>48</b>
	<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>49</b>
	<b>SEZNAM OBRÁZKŮ.....</b>	<b>50</b>
	<b>SEZNAM TABULEK.....</b>	<b>51</b>
	<b>SEZNAM PŘÍLOH.....</b>	<b>52</b>



## ÚVOD

Bakalářská práce je zaměřena na návrh programu sloužícímu pro tvorbu a editaci dialogových oken pomocí knihovny wxWidgets. WxWidgets je knihovna obsahující základní prvky pro tvorbu grafického rozhraní (GUI).

Grafické rozhraní ať už pro jakoukoliv aplikaci se skládá z několika základních prvků, kterými mohou být samotné okno, tlačítka, texty, menu apod. A je to právě grafické rozhraní, které poskytuje uživateli jednoduché ovládání programu, který může vykonávat velice složité úkony.

Pro tvorbu grafického rozhraní je možno využít několika způsobů. Je možné jej vytvořit pomocí jazyka C a znalosti Win32API. Nebo je možné použít pokročilejších programovacích jazyků využívajících platformu Microsoft .NET Framework, jako je Visual Basic, C# aj.

Dalším možným způsobem je použití knihoven jako je právě wxWidgets, který byl zvolen pro tuto práci. Hlavní výhodou je, že je tato knihovna multiplatformní, tj. že je použitelná pod různými platformami zahrnujícími systémy jako je Windows, Linux, Macintosh, OS/2 a další. Tato knihovna využívá nativity, tj. že primárně využívá prvky platformy, a až když tyto nejsou k dispozici, pak použije vlastní. Další neméně velkou výhodou je použití pro různé programovací jazyky. Základním je C++, ale lze tuto knihovnu používat pro Perl, Ruby, Java, C#, Python a mnoho dalších. V neposlední řadě je výhodou, že tato knihovna je open source, takže její distribuování je zdarma.

Jako nevýhoda této knihovny se jeví především její zprovoznění pro daný programovací jazyk.

## I. TEORETICKÁ ČÁST

## 1 WxWidgets

### 1.1 Popis

Jak již bylo zmíněno v úvodu, wxWidgets je knihovna sloužící pro tvorbu grafického rozhraní aplikací. To ovšem není zdaleka vše. Kromě toho, že wxWidgets umožňuje reagovat na události jako stisk tlačítka myši nebo klávesy, lze pomocí wxWidgets jednoduše pracovat se soubory, s proudy, s více vlákny a dalšími.

WxWidgets jsou open source, tj. jsou šířeny zdarma a mají otevřený programový kód. [10]

Kromě všech těchto vlastností je tato knihovna i multiplatformní a nativní. To prakticky znamená, že aplikační programové rozhraní je u wxWidgets stejné nebo velice podobné pod různými platformami, které jsou v tomto případě podporovány. Například tedy pokud vytvoříme aplikaci pod Microsoft Windows a s maximálně s malými úpravami ji spustit pod systémem Linux. Navíc, to že wxWidgets dělají aplikaci nativní, přináší další výhodu. V tomto případě jde o to, že aplikace obsahuje ovládací prvky dané nativně (tj. operačním systémem a jeho nastavením). Jen v případě, že by některý prvek nebyl k dispozici, využívá wxWidgets svých vlastních objektů. Toto je výhodné především z hlediska uživatele, kdy si nemusí pod svým operačním systémem zvykat na pro něj nezvyklé zobrazení jednotlivých prvků. [11]

Kód této knihovny je psán pomocí jazyka C++ a stejný jazyk je často i využíván pro vývoj aplikací, především z důvodu rychlosti a přirozenějšího výsledku. Taktéž je u jazyku C++ lepší přístup k nízkoúrovňovým funkcím a není problémem integrace již existujícího programového kódu psaného v jazyku C nebo C++. To ovšem neznamená, že aplikaci nelze vyvíjet pod jiným programovacím jazykem. Naopak wxWidgets umožňují vývoj aplikací i pod jazyky jako je Java, C#, Perl, Ruby, Python atd. Většina z těchto jazyků již však sama o sobě umožňuje vývoj aplikací s grafickým rozhraním, proto využití wxWidgets je u nich především pro jejich multiplatformnost a nativitu.

Celkově toto vše znamená, že tento způsob tvorby aplikace je vhodný jak pro programátora, protože jeho práce bude značně zjednodušena, tak pro uživatele, neboť výsledná aplikace je pro něj přívětivá.

## 1.2 Historie

Historie wxWidgets se datuje do roku 1992. V té době se ovšem ještě nenazývala wxWidgets, ale wxWindows. Hlavním tvůrcem je Julian Smart. Ten v roce 1992 byl zaměstnancem Univerzity v Edinburku a v této době se zabýval tvorbou nástroje pro vytváření digramů s názvem Hardy. Myšlenka vyvinout vlastní multiplatformní knihovnu vzešla z toho, že se nechtěl rozhodovat mezi jednotlivými platformami této doby, pro kterou bude tento projekt určen. [11] [9]

První verze wxWindows 1.0 vychází v září roku 1992. Název je odvozen od „w“-Windows a „x“- X Window system. Jelikož však podporuje jen XView a MFC 1.0, na což si stěžovali především uživatelé překladače C++ firmy Borland, je kód přepsán tak aby využíval Win32 API. Později je také přidána podpora pro Motif. (Motif a XView byly dvě hlavní soupeřící GUI pro X Window system v této době).

Vzhledem k faktu, že již první verze wxWindows byla zveřejněna na FTP serveru a byla tudíž zdarma veřejně dostupná, a k tomu že uživatelé zjistili, že wxWindows dokáže nabídnout lepší řešení než jiné komerční projekty v té době, začala se kolem wxWindows vytvářet rozsáhlá komunita. Z této komunity je vhodné jmenovat několik tvůrců:

**Marcus Holzem** – vytvořil Xt port a v roce 1997 pomohl navrhnout API pro novou verzi wxWindows, v roce 1998 však odešel od verze 2.

**Wolfram Gloger** – navrhnul portování na GTK+, neboť GTK+ mělo hlavní problém ohledně toho, že bylo založeno pouze na jazyku C

**Robert Roebing** – hlavní vývojář portu wxGTK, který je dodnes hlavním portem pro systémy Unix/Linux

**Vadim Zeitlin** – přispěl značně k projektu návrhy a kódy

**Stefan Csomor** – v roce 1998 práce na portu pro Mac OS

**Václav Slavík** – tvůrce třídy wxHTML a prohlížeče nápovědy ve formátu HTML (verze 2.1.11)

V roce 2000 začíná firma SciTech, Inc. sponzorovat vývoj wxUniversal, která má být jakýmsi souborem widgetů (tj. prvků uživatelského rozhraní) pro platformy bez svých vlastních. Tento soubor použila první sama firma SciTech pro MGL (nízkoúrovňová grafická knihovna této firmy). Později v roce 2002 Julian Smart a Robert Roebing vyvinuli port wxX11 sloužící pro Unix a X11.

V roce 2003 je již podporován systém Windows CE pro mobilní zařízení.

V roce 2004 dochází kvůli stížnostem společnosti Microsoft k přejmenování z wxWindows na wxWidgets. V téže roce wxMac předělán pro Mac OS X a dále je taky přidána podpora pro Palm OS 6.

V dubnu roku 2005 vychází verze 2.6, která vylepšuje všechny porty. Toto vylepšování pokračuje až do současnosti, kdy poslední verzí je verze 2.8.9.

### 1.3 Architektura

Následující tabulka demonstruje 4 základní vrstvy. Jsou to API (application programming interface – rozhraní pro programování aplikací) wxWidgets, jednotlivé porty následované vrstvou API platform, jež je využívají a nakonec samotný operační systém.

<b>wxWidgets API</b>								
<b>wxWidgets Port</b>								
wxMSW	wxGTK	wxX11	wxMotif	wxMac	wxCocoa	wxOS2	wxPalmOS	wxMGL
<b>Platform API</b>								
Win32	GTK+	Xlib	Motif/ Lesstif	Carbon	Cocoa	PM	Palm OS Protein APIs	MGL
<b>Operating System</b>								
Windows/ Windows CE	Unix/Linux			Mac OS 9/ Mac OS X	Mac OS X	OS/2	Palm OS	Unix/ DOS

Obr. 1: Základní vrstvy API wxWidgets [3]

Hlavními porty tedy jsou: [10]

**wxMSW** – port pro všechny operační systémy Microsoft Windows v jejich 32bitové nebo 64bitové verzi, tj. od Windows 95 dále. Taktéž je podporován systém Windows CE pro mobilní aplikace (včetně Palm OS a Smartphone), což je řešeno pomocí wxWinCE, kterou tento port zahrnuje, ale která obsahuje drobné odlišnosti pro ovládání. Pop patřičném nakonfigurování lze využívat i wxUniversal místo nativních prvků. Pro Linux lze tento port taktéž využít, ale je třeba překladač pomocí nástroje Winelib.

**wxGTK** – využívá ovládacích, jak GTK+ verze 1.x, tak verze 2.x, a je určen pro unixové systémy, které podporují X11 a GTK+ (Linux, Solaris a další).

**wxX11** – používá prvky z wxUniversal (tudíž nepoužívá nativních prvků). Proto je vhodný na embedded systémy (vestavné, jednoúčelové systémy), nebo pro unixové systémy podporující X11, kde není žádoucí kompilace pomocí GTK+, které je ovšem pokročilejší.

**wxMotif** – používá knihovny Motif, OpenMotif nebo Lessmotif. Využitelný je pro unixové systémy. Avšak své uplatnění ztrácí, díky soudobému využívání GNOME nebo GTK+.

**wxMac** – tento port podporuje systémy Mac OS, konkrétněji jejich verze Mac OS 9 a Mac OS X. Pro překlad je zde však třeba využít nástrojů třetích stran. Pro toto je často využíváno nástrojů CodeWarriors a Apple Tools.

**wxCocoa** – vytvořeno pro systém Mac OS X, avšak díky použití rozhraní GNUStep má potenciál stát se využívaným i na jiných platformách. Cocoa je jedním ze čtyř nejužívanějších API pro Mac OS X. Dalšími jsou Carbon, POSIX a Java.

**wxPalmOS** – port pro mobilní platformu Palm OS 6. Je prakticky nejmladším portem, proto jeho vývoj není zdaleka na takovém stupni jako portů ostatních. Přesto lze pro tuto platformu vytvořit zajímavé aplikace.

**wxOS2** – port pro platformu OS/2 nebo eComStation, což je operační systém založený na OS/2.

**wxMGL** – používá knihovnu MGL (nízkoúrovňová grafická knihovna firmy SciTech Software, Inc.) a sadu prvků obsažených ve wxUniversal (knihovna pro systémy bez vlastních prvků)

Vnitřně se celá knihovna dělí do 6 vrstev. Jsou to tyto:

**Společný kód** – kód, který obsahuje všechny základní třídy, jako například *wxBase*, které jsou společné pro všechny porty bez ohledu na jejich určení a použití.

**Obecný kód** – obsahuje prvky, jako je například *wxCalendarCtrl*, které jsou pro danou platformu neznámé a nestandardní. Pro tuto platformu je emuluje a umožňuje tím jejich funkčnost.

**wxUniversal** – jak již bylo zmíněno, implementuje prvky pro platformy, jež nemají svoje vlastní ovládací prvky.

**Platformě závislý kód** – je implementován pomocí nástrojů konkrétní platformy, tj. pro každý port je jedinečný.

**Rozšiřující kód** – spadají sem třídy, které jsou určitým vylepšením základních ovládacích prvků, ale tvoří samostatný prvek. Umístěny jsou v adresáři *contrib*.

**Kód třetích stran** – sem spadají knihovny, které byly vytvořeny mimo wxWidgets, avšak pro správnou funkci vlastností wxWidgets. Například může jít o JPEG, PNG a mnoho jiných.

## 1.4 Programy na základě wxWidgets

Zde bych rád uvedl několik známějších programů, jejichž základem byla právě knihovna wxWidgets. Jsou to: [9]

*VibeStation* – první program, který slouží pro tvorbu a editaci prostorového zvuku. Tento program obsahuje i grafické modelování prostředí včetně materiálů, pohybů atd.

*Audacity* – digitální audio editor

*wxDesigner* – komerční aplikace určená k tvorbě dialogů, tj. obdoba této práce

*poEdit* – překladač zpráv určený zejména vývojářům

*Dataton WATCHOUT* – produkční a prezentační systém, spravující, obrázky, videa, zvuky a další

*Kephra* – textový editor pro programátory, vytvořený v Perlu

*KICAD* – aplikace pro tvorbu elektronických schémat a práce s nimi.

*Anthemion StoryLines* – program vhodný pro spisovatele a scénáristy. Jde o jakýsi editor linie příběhu

Toto nejsou zdaleka všechny programy, které jsou postaveny na základě wxWidgets. Ovšem lze na nich demonstrovat širokou škálu uplatnění knihovny wxWidgets v různých odvětvích.

## 2 Jazyk C/C++

### 2.1 Stručná historie jazyka C/C++

Počátky jazyka C jsou datovány do období mezi lety 1969 a 1973 do Bellových laboratoří AT&T. Hlavními tvůrci jsou Denis Ritchie, Brian Keninghan a Ken Thompson. Název C byl odvozen podle staršího jazyka nazývaného B. [4] [5]

V roce 1973 je jazyk C natolik stabilní, že je do něj přepsán zdrojový kód jádra Unixu, který se tím stává prvním systémem, který je napsán v jiném než strojovém jazyce nebo Assembleru.

V roce 1978 je vydána známá kniha *The C Programming Language*, která dlouhou dobu slouží jako neformální specifikace jazyka.

Během 80. let, kdy již probíhalo rozšíření jazyka o objektovou část na C++, probíhá normování jazyka C. První norma jazyka známá jako ANSI C je vydána v roce 1989. O rok později je vydána i ISO norma. Poté bylo již úsilí soustředěno na jazyk C++. Avšak v roce 1999 byla vydána nová ISO norma známá jako C99 která byla o rok později přijata i jako ANSI. Tato norma rozšiřuje jazyk C o vlastnosti jako inline funkce, definice proměnných kdekoliv (ne nutně na začátku funkce) a další.

V roce 1998 byla přijata první norma jazyka C++, ačkoliv práce na něm probíhaly již od roku 1986. Další pak byla přijata v roce 2003. Aktualizace potom v letech 2006 a 2007. Název jazyka byl odvozen od jazyka C a „++“, což je iterace. Takže lze říci že C++ je jakýmsi navýšením jazyka C.

Jazyk byl vyvíjen tak, aby byl plně kompatibilní s jazykem C. Dá se tedy říci, že jazyk C je podmnožinou jazyka C++. Hlavním vylepšením jazyka C++ je, že jde o objektově orientovaný jazyk. Což přináší třídy a možnosti s nimi spojené, jako je dědičnost apod. Dále jazyk C++ zavádí knihovnu STL (Standard Template Library), což je knihovna šablon, mezi něž patří například vector, map, iterator a jiné. [6] [4]

Vývoj se však nezastavil, a protože je kladen důraz na komponentní programování a podporu vazeb na Internet, vznikají nové jazyky jako C# nebo Java. Obojí vycházejí z jazyka C, avšak jsou samostatnými jazyky bez kompatibility s jazykem C.



## 2.2 Výhody objektově orientovaného programování

Objektově orientované programování ať už v jazyce C++ nebo v jiném jazyce využívajícím objektů má 3 základní principy, které jsou pro programování velkými výhodami. Těmito principy jsou: [4]

**Zapouzdření** (encapsulation) – spojení dat a metod pro manipulaci s těmito daty do nové struktury – třídy. Třída je pak tedy popisem objektů se společnými vlastnostmi, objekt je instancí třídy a s členskými daty třídy lze manipulovat pouze pomocí členských metod třídy. Využití lze vidět v tom, že není třeba psát funkce pro stejné objekty několikrát, ale stačí, aby byly tyto objekty instancí stejné třídy.

**Dědičnost** (inheritance) – umožňuje vytvářet z již existujících tříd (předek, rodičovská třída) třídy nové (odvozená třída, potomek), které obsahují data a metody třídy rodičovské, které mohou upřesňovat anebo mohou přidávat svoje vlastní. Tím se vytváří hierarchie tříd. Čím víc tedy pak jdeme do rodičovských tříd, tím se setkáváme s větším zobecněním a naopak čím jdeme více to potomků, tím se setkáváme s větší konkretizací. Možné je i vícenásobné dědění, což prakticky znamená, že potomek vzniká ze dvou i více předků a obsahuje data a metody od všech těchto předků, což může být nebezpečné z hlediska možných konfliktů. Pro programátora pak praktické využití spočívá v tom, že není třeba jednu stejnou třídu psát několikrát, ale stačí ji zdědit.

**Mnohotvárnost** (polymorphism) – umožňuje používat více verzí stejné funkce. Výběr mezi tím, která z daných funkcí bude použita je možný pomocí včasné nebo pozdní vazby. Včasná vazba je prováděna při překladu programu, kdežto pozdní vazba je prováděna během běhu programu. Pozdní vazba je také spojována často s pojmy abstraktní třída a virtuální metoda. Abstraktní třída je společná třída pro třídy využívající jejich virtuálních metod, které jsou překladačem načteny při překladu do tabulky virtuálních metod. Prakticky to znamená, že virtuální metody jsou v abstraktní třídě pouze deklarovány, ale definovány jsou až v třídě daného potomka. Tato třída pak může být například společným typem pro geometrické prvky vložené do pole, na nichž mohou být prováděny metody jako je rotace, vypočet obsahu apod. V dnešní době a u především novějších jazyků (např. Java) je pod mnohotvárností rozuměna pozdní vazba.

### 2.3 Použití wxWidgets v C++

V této kapitole je popsán způsob, jakým lze zprovoznit knihovnu wxWidgets pro programování v jazyku C/C++, protože nejde zrovna o triviální záležitost.

Základní způsob je vždy stejný. Z webových stránek lze stáhnout zdrojové soubory knihovny a ty přeložit. Zde je způsob, který byl použit při realizaci této práce. Tento způsob byl použit při současném použití Microsoft Visual Studia, ale použitelný by měl být i pod jinými prostředími:

- 1) Stažení zdrojového souboru wxMSW z webu [www.wxwidgets.org](http://www.wxwidgets.org)
- 2) Instalace tohoto souboru
- 3) V adresáři kam byl wxWidgets nainstalován `\build\msw` otevření souboru `wx.dsw` pomocí Microsoft Visual Studia
- 4) Přeložení projektu jak pro Debug, tak pro Release

Toto je však jen jeden možný způsob, který však je obdobný pro všechny platformy. Další možné varianty zprovoznění se odvíjejí podle platformy, pod kterou budou wxWidgets provozovány.

Pro Windows lze například využít instalační balíček wxPack, který kromě knihovny samotné obsahuje i další užitečné věci, jako knihu wxWidgets nebo nástroj wxFormBuilder. [7]

Pro Linux je výhodnou variantou možnost instalace pomocí balíčků. Výhoda především spočívá v jednoduchosti a v aktuálnosti pokud je povolena automatická aktualizace balíčků.

## 3 FORMÁT XRC

### 3.1 Seznámení s XRC

Hlavním účelem formátu XRC se uchovávat data dialogových oken, menu apod. Tento formát byl odvozen od značkovacího jazyka XML (**Extensible Markup Language**). Tvůrcem tohoto formátu je Václav Slavík.

XML jazyk se používá jako jazyk k popisu dat, tzn. že pouze uchovává data a nic jiného s nimi neprovádí. Jak bylo zmíněno, jde o jazyk značkovací, obdobný jako např. HTML, ale na rozdíl od něj u XML nejsou tagy (značky) definovány, tudíž je třeba definovat své vlastní. Pro popis data se využívá buď DTD (Definice typu dokumentu) nebo XML schémat, kdy DTD definuje elementy, které lze pro uchování daného dokumentu použít, kdežto pro XML schéma lze definovat své vlastní elementy. [2] [1]

XML je od roku 1998 normou W3C. XML soubory nacházejí své hlavní uplatnění při tvorbě webových stránek.

XRC je tedy založen na XML. Využívá DTD, kdy pro uložení okna je využito povolených elementů, které jsou popsány dále.

Hlavní výhody XRC jsou: [9]

- Pokud je zdrojový soubor změněn není třeba aplikaci znovu překompilovat a linkovat.
- Je efektivnější mít oddělený zdrojový soubor a kód, kvůli možné obtížné integraci zdroje do kódu.
- Jelikož jde o standard wxWidgets, lze zdrojové soubory použít pod kterýmkoliv programem, který je schopen s tímto standardem pracovat.
- Jednoduchá možnost editace pomocí nástrojů pro editaci souborů XML, které nemusejí být jakkoliv speciální (stačí například poznámkový blok, který je běžnou součástí operačních systémů Windows).

## 3.2 Popis souboru XRC

```
<?xml version="1.0"?>
<resource version="2.3.0.1">
  <object class="wxFrame" name="my_main_frame">
    <size>200,300</size>
    ...
  </object>
</resource>
```

Obr. 2: Základ souboru XRC [8]

Na obrázku výše je základ souboru XRC. Úvodním elementem je verze použitého standardu XML. Hlavním elementem pak je tag `<resource>` který uvozuje prvky uchovávané v souboru. Základním tagem každého prvku je `<object>` definovaný třídou prvku a názvem tohoto prvku. Tento tag je párový a proto může uvozovat jak vlastnosti prvku, jako je velikost, pozice apod., tak i další prvky, které tento prvek může obsahovat (ukázka tedy obsahuje frame s názvem `my_main_frame`, který má velikost 200x300 pixelů). Další příklad souboru XRC je uveden v příloze P IV.

## 3.3 Povolené prvky a elementy v XRC

**wxWindow** – základní třída, kterou dědí ostatní prvky [8]

<code>&lt;exstyle&gt;</code>	- styl okna
<code>&lt;bg&gt;,&lt;fg&gt;</code>	- barva pozadí/popředí
<code>&lt;enabled&gt;</code>	- určuje, zda je prvek ovladatelný či ne
<code>&lt;focused&gt;</code>	- určuje, zda má prvek focus, tj. zda je právě aktivní
<code>&lt;hidden&gt;</code>	- určuje, zda je prvek skrytý
<code>&lt;tooltip&gt;</code>	- nápověda k prvku
<code>&lt;font&gt;</code>	-ohraničuje popis písma použitého pro prvek. Může obsahovat:
<code>&lt;size&gt;</code>	- velikost písma
<code>&lt;style&gt;</code>	- styl písma (italic...)
<code>&lt;weight&gt;</code>	- tloušťka písma (bold...)
<code>&lt;family&gt;</code>	- skupina písma (modern, decorative, ...)
<code>&lt;underlined&gt;</code>	- zda má být podtržení
<code>&lt;encoding&gt;</code>	- kódování
<code>&lt;face&gt;</code>	- písmo pro dané okno

**Sizery** – prvky okna určující rozložení prvků v okně

*sizeritem* – třída pro <object> určující, že prvek náleží danému sizeru, obsahující další <object> který již definuje samotný druh prvku. Sizer vždy musí obsahovat alespoň jednu sizeritem.

<flag> - vlastnosti jako např. zarovnání sizeru

<minsize> - minimální velikost sizeru

<option> - dává určité položce sizeru danou váhu mezi ostatními položkami tohoto sizeru

<border> - udává velikost ohraničení

**wxBoxSizer** – základní sizer, který umísťuje prvky pod sebe (respektive vedle sebe)

<orient> - orientace rozložení prvků (horizontální/vertikální)

**wxStaticBoxSizer** – stejný jako wxBoxSizer s popiskem

<label> - popisek

**wxGridSizer** – sizer, umísťující prvky po řádcích do tabulky, velikost jednoho pole tabulky je dána nejširším a nejvyšším prvkem

<rows> - počet řádků tabulky

<cols> - počet sloupců tabulky

<vgap> - velikost místa mezi prvky ve vertikálním směru

<hgap> - velikost místa mezi prvky v horizontálním směru

**wxFlexGridSizer** – stejný jako wxGridSizer, s rozdílem, že velikost prvku je dána nejvyšším a nejširším prvkem v daném řádku (resp. sloupci)

<growablecols> - seznam indexů sloupců, které mohou měnit svou velikost

<growablerows> - seznam indexů řádků, které mohou měnit svou velikost

**wxButton** – tlačítko, odvozeno od wxWindow

<style> - styl tlačítka (např. popiska vlevo, vpravo atd.)

<label> - popiska tlačítka

<default> - určuje zda bude tlačítko defaultním tlačítkem dialogu

**wxBitmapButton** – tlačítko s obrázkem, odvozené z wxWindow

<bitmap> - cesta k obrázkovému souboru

<selected> - určuje zda je prvek vybraný

<focus> - určuje zda je prvek právě aktivní

<disabled> - určuje zda je prvek neovládatelný

**wxCalendarCtrl** – kalendář, odvozeno od wxWindow

<style> - vlastnosti kalendáře (např. první den týdne je pondělí, zobrazit svátky atd.)

**wxCheckBox** – zaškrťovací políčko, odvozeno od wxWindow

<label> - popiska

<checked> - určuje zda je políčko zatrženo

**wxCheckListBox** – zaškrťovací seznam, založen na wxWindow a wxCheckBox

<content> - uvozuje seznam položek <item>

<item> - položka seznamu s parametry jako wxCheckBox

**wxChoice** – roletové menu pro výběr, dědí z wxWindow

<selected> - index právě zvolené položky

<content> - obdobně jako u wxCheckListBox

<checked> - určuje zda je daná položka vybrána (pro případ možnosti výběru více položek nebo zaškrťovacího roletového menu)

**wxComboBox** – wxChoice s možností vlastního zadávání textu, odvozeno od wxChoice a wxWindow

<style> - styl prvku (např. zobrazit seřazené atd.)

<value> - textová počáteční hodnota

**wxDialog** a **wxFrame** – jde o hlavní prvky, které tvoří okno, rozdíl v nich je pouze ten, že wxFrame je absolutně prázdný, kdežto wxDialog obsahuje už jakoby wxPanel, obojí dědí z wxWindow

<title> - titulek okna

<style> - styl okna (tlačítko pro minimalizaci, maximalizaci, zavření, pevně daná velikost okna, atd.)

<centered> - určuje zda je okno centrováno na střed svého předka (tj. okna které jej vyvolalo)

**wxGauge** – pruh ukazující množství (nejčastěji času), používá se např. při instalaci apod.

<style> - styl (např. horizontální, vertikální atd.)

<range> - maximální hodnota

<value> - počáteční hodnota

<shadow> - šířka stínu

<bezel> - šířka jedné části pruhu

**wxGenericDirCtrl** – prvek, který zobrazuje souborové cesty, obsahuje wxTreeCtrl (strom souborů) a volitelně může obsahovat i wxChoice pro filtrování

- <style> - styl (zobrazit jen složky, atd.)
- <defaultfolder> - cesta k defaultní složce
- <filter> - koncovka souborů které mají být zobrazeny (např. \*.jpg, \*.exe).  
Těchto položek může prvek obsahovat více.
- <defaultfilter> - index filtru, který má být použit jako defaultní.

**wxHtmlWindow** – okno pro zobrazení webových stránek

- <style> - styl (zobrazení postníků apod.)
- <borders> - odsazení zobrazení webové stránky od okraje okna
- <htmlcode> - HTML kód
- <url> - cesta nebo adresa stránky, jež se má zobrazit

**wxListBox** – seznam

- <style> - styl (možnost vybrat jeden prvek nebo více apod.)
- <selection> - index primárně vybraného prvku
- <content> - seznam položek (obdobný jako wxChoice)

**wxListCtrl** – seznam, který může obsahovat více dat o položce (včetně obrázků a pod)

- <style> - styl (způsob zobrazení ikon, výběry apod.)

**wxMenuBar** – lišta s položkami menu

**wxMenu** – menu, obsahuje 1 nebo více položek wxMenuItem

- <style> - styl
- <label> - popiska
- <help> - nápověda k menu

**wxMenuItem** – položka v menu

- <style> - styl
- <label> - popis položky
- <accel> - klávesová zkratka spjatá s položkou
- <help> - nápověda k položce
- <radio> - určuje zda jde o položku typu radio button
- <checkable> - určuje zda jde o zaškrťovací položku
- <enabled> - určuje zda je položka k dispozici
- <checked> - určuje zda je položka zaškrtnuta

<bitmap> - cesta k obrázku, který může být zobrazen u položky

**wxNotebook** – soubor záložek

<style> - zarovnání apod.

<usenotebooksizer> - určuje zda je k manipulaci se záložkami použit sizer

*notebookpage* – třída <object>u, určující záložku, blíže určenou pomocí dalších tagů.

<label> - popiska záložky

<selected> - určuje zda je záložka vybrána

**wxPanel** – prostor, na který jsou umístovány ovládací prvky

<style> - styl panelu

**wxRadioButton** - přepínač

<style> - styl (zda je prvek osamocen nebo v nějaké skupině)

<value> - zda je prvek vybrán či nikoliv

<label> - popiska přepínače

**wxRadioBox** – seznam přepínačů, které jsou vzájemně závislé

<style> - styl (vertikální/horizontální seznam apod.)

<content> - seznam položek

<selection> - index zvolené položky

**wxScrollBar** – posuvník pro rolování okna

<style> - horizontální/vertikální

<value> - aktuální pozice

<range> - maximální hodnota rolování

<thumbsize> - viditelná velikost

<pagesize> - velikost rolované stránky

**wxScrolledWindow** – okno s rolováním

<style> - rolování vertikální/horizontální

**wxSlider** – posuvník pro nastavení hodnoty

<style> - vertikální/horizontální, zobrazení popisků maxima, minima apod.

<value> - aktuální pozice

<min> - minimální pozice

<max> - maximální pozice

<tickfreq> - frekvence znamének

<pagesize> - velikost stránky pro posuvník (posun při klikání)



<linesize> - počet kroků posuvníku při uživatelské posunu

<thumb> - nastavení délky

<selmin> - minimální vybraná pozice

<selmax> - maximální vybraná pozice

**wxSpinButton** – rotační tlačítko

<style> - styl (horizontální/vertikální, ovládání šipkami apod.)

<value> - aktuální hodnota

<min> - minimální hodnota

<max> - maximální hodnota

**wxSpinCtrl** – stejné jako wxSpinButton, s odlišností v tom, že součástí je i textové pole, v němž jsou hodnoty

**wxStaticBitmap** – obrázek

<bitmap> - cesta k obrázku

**wxStaticBox** – prostor ohraničující skupinu prvků

<label> - popiska

**wxStaticLine** – čára

<style> - vertikální/horizontální

**wxStaticText** – text

<style> - zarovnání apod.

<label> - popiska

**wxTextCtrl** – textové pole

<style> - zarovnání apod.

<value> - počáteční hodnota

**wxToolBar** – panel nástrojů

<style> - vertikální/horizontální apod.

<bitmapsizesize> - velikost obrázku tlačítka panelu nástrojů

<margin> - rozpětí tlačítka panelu nástrojů

<packing> - velikost prostoru mezi tlačítky

<separation> - velikost oddělovače

*tool* – typ třídy <object>u, pro položku panelu nástrojů, jež může mít tagy:

<bitmap> - cesta k hlavnímu obrázku tlačítka

<bitmap2> - cesta k obrázku tlačítka, který se může zobrazit při stisku

<toggle> - nastavení zda se obrázek při stisku má měnit

<tooltip> - jednoduchá nápověda pro prvek

<longhelp> - obsáhlejší nápověda k prvku

*separator* – třída pro <object>, určující oddělovač mezi skupinami tlačítek panelu nástrojů. Tato třída neobsahuje žádné tagy.

**wxTreeCtrl** – seznam uspořádaný do stromové hierarchie

<style> - styl (zda mají být použita tlačítka či nikoliv apod.)

## **II. PRAKTICKÁ ČÁST**

## 4 POPIS PROGRAMU

Program vytvořený jako součást této práce slouží pro návrh a editaci dialogových oken. Jako prvky lze použít následující:

- *box* – ohraničení oblasti s prvky
- *button* – tlačítko
- *choice* – roletové menu
- *checkBox* – zaškrtačací pole
- *editText* – pole pro zadání textu
- *line – horizontal* – horizontální čára
- *line – vertical* – vertikální čára
- *listBox* – seznam
- *picture* – obrázek
- *radioButton* – přepínací tlačítko
- *staticText* – text

Nejsou sice použity všechny možné prvky, ale přesto tyto postačují pro navržení většiny potřebných záležitostí pro dialogová okna.

Pro dané prvky lze nastavovat základní parametry jako je pozice, velikost, barva popředí a pozadí, popiska apod.

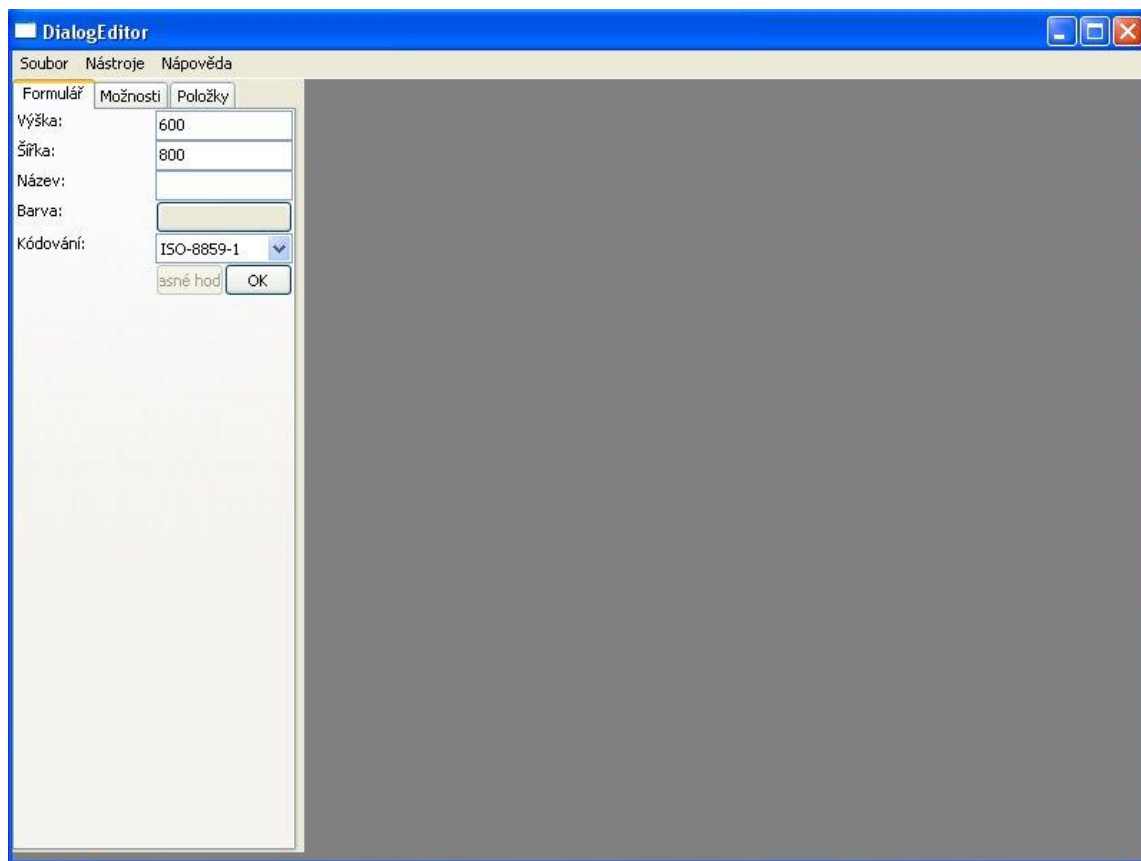
Dále program umožňuje uložení okna do standardního souboru typu XRC, jeho načtení, obnovení posledního stavu okna se souborem, který byl uložen a možnost náhledu vytvořeného okna.

Pochopitelně součástí programu je i možnost manipulace s velikostí a pozicí hlavního okna programu.

Použití programu kromě využití pro běžná dialogová okna je i pro další programování, jako mohou být i výpočtová okna pro strojní výpočty, kdy lze jednotlivým použitým prvkům okna přiřadit proměnné a podle obsahu těchto prvků vypočítávat jiné hodnoty.

## 5 PROGRAM Z POHLEDU UŽIVATELE

Zde je uveden popis programu z uživatelského pohledu. Jde o jakýsi návod pro používání programu.

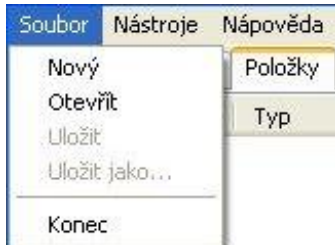


Obr. 3: Základní zobrazení okna programu

V horní části pod hlavní lištou jsou jednotlivá menu pro nastavení a ovládání programu. V levé části pak boční panel sloužící pro tvorbu a manipulaci jednotlivých prvků vytvářeného okna. Zbylou část vyplňuje návrhové okno, do nějž lze vkládat prvky a manipulovat s nimi.

## 5.1 Hlavní menu

### 5.1.1 Menu Soubor



Obr. 4: Menu Soubor

*Nový* – vytvoření nového okna, pokud je rozpracováno jiné okno pak je zobrazena výzva pro uložení

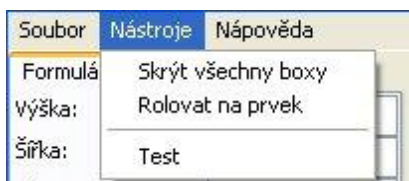
*Otevřít* – otevření již uložené souboru s vytvořeným oknem

*Uložit* – uložit stávající okno do stávajícího souboru. Pokud okno nebylo doposud vytvořeno pak je zobrazena nabídka pro zadání názvu souboru.

*Uložit jako...* - umožňuje uložit soubor pod jiným názvem, než byl doposud uložen, s tím že původní soubor zůstává zachován.

*Konec* – ukončení programu. Pokud změny v rozpracovaném okně nebyly doposud uloženy, pak je zobrazena výzva k možnému uložení.

### 5.1.2 Menu Nástroje



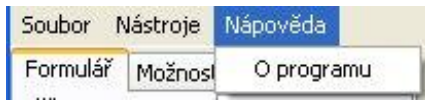
Obr. 5: Menu Nástroje

*Skrýt všechny boxy* – zaškrtnutá položka. Hromadně skryje v návrhovém okně všechny prvky typu box. Vhodné zejména v případě, kdy boxy v pořadí všech prvků překrývají svým prostorem jiné prvky.

*Rolovat na prvek* – zaškrtnutý prvek. Při povolení této položky při kliku na položku v seznamu položek bude v návrhovém okně rolováno na tuto položku.

*Test* – zobrazí náhled okna, tak jak bude vypadat při běžném použití.

### 5.1.3 Menu Nápověda



Obr. 6: Menu Nápověda

*O programu* – stručné informace o programu

## 5.2 Boční panel

### 5.2.1 Záložka Formulář

Tato záložka slouží pro nastavení základních vlastností vytvářeného okna



Obr. 7: Záložka Formulář

*Výška* – výška vytvářeného okna

*Šířka* – šířka vytvářeného okna

*Název* – titulek vytvářeného okna (tj. text v hlavní liště)

*Barva* – barva vytvářeného okna

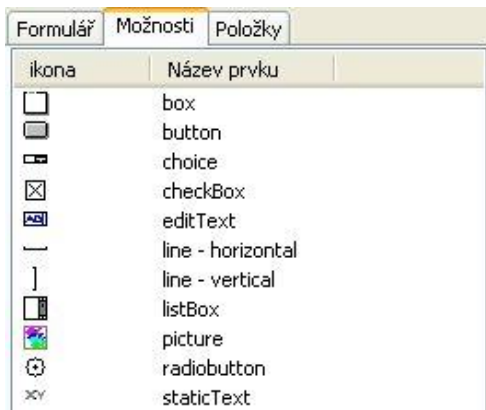
*Kódování* – určuje, v jakém kódování budou uloženy texty okna

*Aktualizovat* – obnoví hodnoty na této záložce dle aktuálních pro vytvářené okno

*OK* – potvrzení pro vytvoření okna s danými hodnotami

### 5.2.2 Záložka Možnosti

Obsahuje seznam všech prvků, které je možné použít pro návrh dialogového okna. Ovladatelnost této záložky je zpřístupněna až po vytvoření základu dialogového okna.



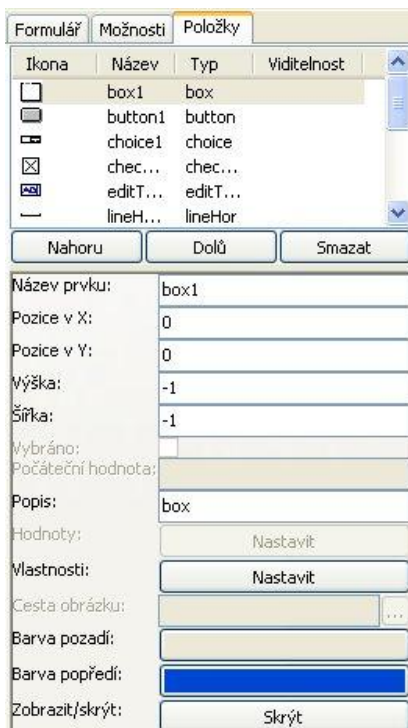
Obr. 8: Záložka Možnosti

Prvek lze do okna umístit dvěma způsoby:

- 1) Dvojklik levým tlačítkem myši na požadovaném prvku – prvek je umístěn do levého horního rohu okna
- 2) Přetažení prvku ze seznamu přímo na návrhové okno – prvek je přidán přímo na místo kam byl přetažen

### 5.2.3 Záložka Položky

V této záložce je obsažen jednak seznam všech položek, které již byly přidány do vytvářeného okna a jednak nastavení pro jednotlivé položky.



Obr. 9: Záložka Položky



*Kliknutí na položku v seznamu prvků* – označení prvku v návrhovém okně, případné narolování na označený prvek (možnost v menu *Nástroje*). Pokud je položka zobrazena červeně, pak toto označení znamená, že položka je mimo vymezené rozmezí navrhovaného okna.

*Nahoru* – posun položky v seznamu směrem nahoru

*Dolů* – posun položky v seznamu směrem dolů

*Smazat* – smazání označené položky

*Název prvku* – název prvku, pod kterým bude uložen a zobrazen v seznamu vložených prvků. Tento název musí být jedinečný. Primárně je název odvozen od typu prvku a jeho počtu vložení.

*Pozice v X (Y)* – pozice prvku v okně

*Výška (šířka)* – velikost prvku

*Vybráno* – určuje, zda bude prvek primárně vybrán nebo ne. Dostupné pouze pro prvky typu **checkBox** a **radioButton**

*Počáteční hodnota* – určuje primární hodnotu pro daný prvek. Dostupné pro **choice** (určení primárně vybraného prvku), **listBox** (určení primárně označeného prvku)

*Popis* – popiska daného prvku. Není přístupná pro **choice**, **lineHor**, **lineVer**, **listBox** a **picture**

*Hodnoty* – nastavení položek pro **choice** a **listBox**. V dialogovém okně, které se zobrazí po kliknutí na tlačítko *Nastavit*, lze hodnoty pak nastavit způsobem, kdy 1 položka je 1 řádkem v textovém poli dialogu.

*Vlastnosti* – vlastnosti daného prvku. Tato položka je dostupná pro všechny typy prvků. Po kliknutí na tlačítko *Nastavit*, je zobrazeno dialogové okno s možností zaškrtnout vlastnosti přípustné pro vybraný prvek. Zároveň není možné vybrat vlastnosti, které si vzájemně odporují.

*Cesta obrázku* – nastavuje cestu pro obrázek. Pro nastavení je možno cestu zadat napsáním cesty do textového pole nebo kliknutím na tlačítko „...“, které zobrazí klasický dialog pro výběr souboru. Pokud je soubor s obrázkem neexistuje, pak je zobrazen prázdný obrázek. Tato položka je dostupná pouze pro typ **picture**.

*Barva pozadí (popředí)* – barva pozadí (popředí) prvku. Není dostupná pro **lineHor** a **lineVer**.

*Zobrazit/skryt* – umožňuje zobrazení nebo skrytí vybraného prvku v návrhovém okně podle toho v jakém stavu se prvek nachází, což je naznačeno textem příslušného tlačítka. Dostupné pro všechny typy prvků.

### 5.3 Manipulace s oknem a prvky okna

S hlavním oknem programu lze manipulovat pomocí klasických ovládacích prvků jako je tlačítko pro minimalizaci, maximalizaci a zavření okna. Stejně tak lze okno zvětšovat a zmenšovat pomocí tažení myši za okraj okna a posunovat pomocí tažení myši za hlavní lištu.

Velikost bočního panelu lze zvětšovat či zmenšovat tažením za jeho pravý okraj. Stejně tak lze měnit poměr velikosti seznamu položek a nastavení parametrů na záložce *Položky*.

V návrhovém okně lze pak manipulovat s jednotlivými položkami, přesněji s jejich pozicí a velikostí. Dále pokud je navrhované okno větší než prostor pro zobrazení, lze se po prostoru okna pohybovat pomocí scrollbarů. Měnit velikost a pozici lze pomocí stisku klávesy CTRL a kliknutí a tažení myši. Tyto změny lze učinit při zobrazení patřičného kurzoru, jehož příklady jsou uvedeny níže. Tento kurzor se stejně jako mnoho jiných věcí může mírně lišit podle použitého operačního systému a jeho nastavení.



Obr. 10: Příklad kurzoru pro změnu velikosti (vlevo) a změnu pozice

## 6 PROGRAM Z POHLEDU PROGRAMÁTORA

V této části je uveden popis principu fungování programu, konstrukce, na nichž je program postaven apod.

### 6.1 Základní datové typy

| Název třídy         | Obsah třídy  |
|---------------------|--|
| CalcEdFrame         | Funkce a prvky hlavního okna                             |
| cStaticFunctions    | Statické funkce  |
| cStart              | Spouštěcí třída  |
| cTestWin            | Funkce návrhového okna                                   |
| cWinDataSave        | Funkce pro uložení stávajícího zobrazení okna            |
| DnDObject           | Objekt pro přetažení prvku ze seznamu do návrhového okna |
| PreviewWin          | Okno pro zobrazení náhledového okna                      |
| propertyDlg         | Dialog a funkce pro nastavení vlastností prvku           |
| userItemDescription | Popis jednoho zobrazeného prvku                          |
|                     |  |
| uTextCtrl           | Prvek textového pole s akcí při opuštění tohoto prvku    |
| valuesDlg           | Dialog a funkce pro nastavení hodnot prvku               |

Tab. 1: Tabulka tříd a jejich obsahu

Základní datové typy, mimo již standardních z wxWidgets a C++, jsou obsaženy v souboru *dialogTypes.h*. Definice funkcí potřebných pro tyto typy, případně třídy jsou pak obsaženy v souboru *dialogTypes.cpp*. Soubor *dialogTypes.h* kromě již uvedeného taktéž obsahuje definici konstant pro projekt. Mezi tyto konstanty patří například cesta k souboru, který obsahuje poslední konfiguraci souboru, výčtový typ obsahující konstanty pro ID jednotlivých prvku hlavního okna (menu, tlačítka atd.) a v neposlední řadě výčtový typ obsahující konstanty pro typy prvků, které lze vložit (box, button, atd.)

Dále tento soubor obsahuje definici stěžejního třídy a typu pro celý tento projekt. Touto třídou je třída *userItemDescription*, jejíž struktura včetně popisu je znázorněna na obrázku níže.

```

class userItemDescription
{
public:
    wxString name;           //název prvku
    userItemsType type;     //typ prvku (button, box...)
    wxWindowID parentID;   //vlastník
    wxSize size;           //velikost
    wxPoint position;      //pozice
    wxWindowID id;         //id hodnota
    long properties;       //vlastnosti ve formě čísla typu long
    wxString strProp;      //vlastnosti jako typ string
    size_t primStatus;     //primární stav (výbrání, vybraná položka)
    wxString imagePath;   //cesta k obrázku
    wxWindow* panelItem;   //ukazatel na prvek v návrhovém okně
    wxColour bgColour;     //barva pozadí
    wxColour fgColour;     //barva popředí
    wxString langLabels;  //popisky prvku

    wxString ConvertUserItemsType2Str(userItemsType type);
    //převod typu prvku na string
    wxXmlNode* GetXmlNode(); //získání prvku v xrc formátu
};

```

Obr. 11: Datový typ *userItemDescription*

Jeden prvek této třídy obsahuje veškeré potřebné informace pro vytvoření, uložení a editaci vloženého prvku. Pro všechny prvky okna je pak vytvořen vektor (*vectorUserItemDescription*) a jeho iterátor (*itvectorUserItemDescription*), jejichž základem je právě tato třída, což umožňuje jednoduše procházet všechny tyto prvky vložené do návrhového okna.

Další třídou v tomto souboru je třída *uTextCtrl*, která dědí ze třídy *wxTextCtrl*, která je standardní součástí knihovny *wxWidgets*. Oproti této třídě však třída *uTextCtrl* obsahuje funkci, která reaguje na událost, kdy prvek ztratí focus (tj. přestane být aktivním). Kvůli funkčnosti této funkce je třeba, aby tato třída taktéž obsahovala ukazatel na hlavní okno programu. Použití této třídy bude stejně princip její funkce osvětleno dále.

Poslední třídou deklarovanou v tomto souboru je třída *DnDObject*. Tato třída slouží pro chycení a přetažení objektu myši. Své využití v tomto programu má při přetahování objektu ze seznamu možných prvků do návrhového okna. Dědí ze třídy *wxTextDropTarget*, neboť postačuje, aby přetahovaná data byla textová. Jednoduše řečeno, přetažen je název typu prvku a podle tohoto se vykreslí daný prvek.

## 6.2 Statické funkce

Jde o funkce, které lze volat, aniž by byla použita instance třídy, jejíž metodou funkce je. Podmínkou ovšem je že statická funkce musí používat proměnné a metody, které jsou taktéž statickými.

V programu jsou všechny statické funkce obsaženy v souborech *staticFunctions.h* a *staticFunctions.cpp*.

Patří sem obecné funkce, které jsou využívány v celém programu, a proto bylo vhodné je vytvořit jako statické. Jsou zde zejména funkce pro konverzi mezi datovými typy, nastavení některých elementů a kontrola jejich stavů.

## 6.3 Soubor pro uložení konfigurace programu

Jde o konfigurační textový soubor. Funkce pro jeho vytvoření a načtení jsou obsaženy v souborech *cWinDataSave.h* a *cWinDataSave.cpp*. Konstanta cesta, kde se tento soubor nachází je, jak již bylo řečeno v souboru *dialogTypes.h*.

Tento soubor obsahuje informace o veškerém nastavení hlavního okna programu a dialogových oken, které mohou být vyvolány. Taktéž je zde zapsána informace o naposledy použitém uloženém souboru.

Jelikož *wxWidgets* obsahuje třídu *wxIniConfig*, je vytvoření těchto funkcí velice zjednodušeno. Jako jeden parametr konstrukturu třídy je použit soubor, ze kterého je čteno (respektive je do něj ukládáno). Ve třídě pak jsou funkce pro zápis a čtení dat různých typů. V tomto případě jde především o typy *long* a *string*. Pro čtení a zápis postačují dva parametry a těmi jsou cesta v souboru, kam mají být data zapsána (respektive odkud mají být čtena) a zdroj (respektive cíl) těchto dat. Cestu v souboru si lze pak představit jako klasickou adresářovou strukturu, kdy data jsou zapisována do nějaké proměnné, která je součástí nějaké nadřazené skupiny.

Čtení souboru je voláno v konstruktorech oken a zápis je prováděn při uzavření okna v jeho destrukturu.

Zde je malý příklad toho jak mohou data zapsaná do konfiguračního souboru vypadat. V tomto konkrétním případě jde o zapsání pozice a velikosti hlavního okna programu.

```
[DialogEditor]
positionX=174
positionY=61
SizeW=1225
SizeH=748
```

Obr. 12: Ukázka konfiguračního souboru

## 6.4 Vytvoření hlavního okna programu

Vytvoření celého hlavního okna je součástí konstruktoru třídy *CalcEdFrame*. Jelikož celá tato třída dědí ze třídy *wxFrame*, je nejprve vytvořen tento samotný frame (prázdné okno), přičemž je současně nastavena jeho pozice a velikost.

Posléze je vytvořen menubar (pruh s jednotlivými menu v horní části okna) s jednotlivými menu a jejich prvky.

Na to je zbývající část framu rozdělena na dvě části pomocí *wxSplitterWindow*, což je prvek, který může obsahovat dva prvky třídy *wxWindow* nebo prvky od této třídy odvozené. Pomocí děliče těchto prvků (*sash*), lze měnit poměr velikostí mezi těmito prvky. Tedy levá část okna programu obsahuje prvek *wxNotebook*, který umožňuje přepínat mezi záložkami, které mohou obsahovat různé ovládací prvky. Pravá část je vyhrazena pro vytvoření návrhového okna.

Obdobně je pak použito toto rozdělení na záložce Položky s tím rozdílem, že zde je toto rozdělení horizontální.

Každá záložka levého *wxNotebooku* obsahuje svůj *wxPanel* (prvek pro umístění dalších položek). Ten každý má svůj *sizer* (prvek pro uspořádání položek). Ve všech případech je s výhodou použit *wxFlexGridSizer*, který umožňuje uspořádání do tabulky s flexibilní velikostí jednotlivých polí, což je využito zejména v případě změny poměru velikostí mezi návrhovým oknem a bočním panelem pro současnou změnu prvků pro nastavení.

Pro seznamy použité v bočním panelu je použit prvek *wxListCtrl*, který umožňuje zobrazení potřebné položky, s tím že její data lze zobrazit v několika sloupcích.

Pro ikony prvků je v seznamech použito obrázků ve formátu XPM. Tyto obrázky je jednoduché v programovém kódu použít. Obrázek lze použít coby knihovnu. Stačí tedy pouze na začátku souboru zadat *#include* s názvem souboru obrázku a obrázek lze pak již volat s jeho názvem doplněným o *\_xpm*. Samotný obrázek je pak v souboru vykreslen pomocí symbolů, kdy každý symbol reprezentuje jeden pixel obrázku a barvu tohoto

pixelu. Barva tohoto pixelu je v souboru taktéž definována ve formě RGB. Ukázka souboru XPM je uvedena v příloze této práce (příloha P V).

Takto je vytvořeno základní okno programu. V případě že existuje soubor s uloženou předchozí konfigurací, jsou parametry okna nastaveny podle dat uložených v tomto souboru a je načten soubor s poslední uloženou prací, jehož cesta je obsažena taktéž v konfiguračním souboru. Způsob načtení je popsán později.

## 6.5 Funkce pro ovládání z menu

Základem pro vyvolání funkce z menu je událost *EVT\_MENU*. Každá takováto událost má přiřazeno ID menu, na které má reagovat a funkci, kterou je vyvolání této události ošetřeno.

### 6.5.1 Vytvoření nového okna

Tuto událost ošetřuje funkce *OnNew* třídy *CalcEdFrame*. Nejprve je kontrolováno, zda existuje návrhové okno. Pokud ano pak je vyvolán dialog, jestli chce uživatel stávající práci ztratit nebo ne. Pokud ji ztratit nechce, pak je funkce ukončena. V opačném případě je návrhové okno smazáno včetně prvků, které obsahuje. Zároveň je v záložkách přepnuto na záložku Formulář a funkcí *BlockControls* jsou zneprístupněny ovládací prvky, které nemají smysl bez návrhového okna včetně všech ostatních záložek bočního panelu.

### 6.5.2 Načtení již uloženého projektu

Existující projekt je uložen do souboru formátu XRC, který již byl popsán výše. Celý soubor je tvořen tedy prvky typu *wxXmlNode*, což jsou jednotlivé prvky, kdy jeden takovýto „node“ obsahuje všechny své potomky, které mohou určovat prvky, které obsahuje a jednotlivé vlastnosti.

Ze všeho nejdříve je provedena, zda není rozpracován jiný projekt obdobně jako u vytváření nového projektu a případně tedy vyčištění.

Při načítání jsou pak zpracovány nejdříve vlastnosti navrhovaného okna a poté jsou procházeny „nody“, které značí prvky, jejichž je toto okno vlastníkem.

Je vytvořena proměnná typu *userItemDescription*, která je naplněna na základě potomků které vlastní „node“ a jsou vlastnostmi tohoto prvku. Tato proměnná je posléze přidána do vektoru prvků. Toto načítání prvků je prováděno cyklem do doby, kdy již

potomci hlavního „nodu“ okna neexistují. Po ukončení cyklu je dalším cyklem projit celý vektor a prvky jsou vykresleny do návrhového okna.

### 6.5.3 Uložení stávajícího projektu

Pro uložení existují dvě možnosti. Pokud již projekt byl uložen pak je uložen pod stávajícím názvem souboru. Jestliže uložen doposud nebyl je zobrazeno dialogové okno, které je vytvořeno voláním instance třídy *wxFileDialog*, což je třída knihovny *wxWidgets* zajišťující vytvoření okna pro procházení souborů.

Pro uživatele je potřeba uložení signalizována hvězdičkou na konci titulku hlavního okna programu. Toto je zajištěno pomocí funkce *SetFileChange*, která je funkcí třídy *CalcEdFrame*. Tato funkce nastavuje proměnnou *fileChange*, pomocí které je tento stav také kontrolován. Funkce *SetFileChange* taktéž přidává název souboru do titulku okna, pokud existuje.

Samotné uložení je pak prováděno pomocí tvorby jednotlivých *wxXmlNode* funkcemi *GetXmlNode* (třída *userItemDescription*), která vytvoří „node“ každého jednotlivého prvku navrhovaného okna a *GetXmlSubelement* (statická funkce třídy *cStaticFunctions*). Ta zajišťuje patřičné vytvoření potomka. V kombinaci s funkcí *CreateXmlFile* (třída *CalcEdFrame*) je vytvořen soubor XRC který vyhovuje potřebné normě a je správný z jak syntaktického tak logického hlediska.

### 6.5.4 Funkce menu Nástroje a Nápověda

V tomto případě jde o velice jednoduché funkce, proto jsou shrnuty do jedné kapitoly.

V případě funkce pro skrytí všech boxů je cyklicky procházen vektor prvků. Typ obsažený v tomto prvku je pak porovnáván, zda je typu *box*. Pokud je, pak je prvek, na nějž odkazuje ukazatel, jenž je taktéž součástí porovnávaného prvku a odkazuje na prvek zobrazený v návrhovém okně, nastaven pomocí funkce *Hide* (funkce třídy *wxWindow*, která je základem většiny prvků *wxWidgets*) na skrytý. Pokud je později toto skrytí v menu zrušeno, postupuje se obdobně s tím rozdílem, že je použita funkce *Show*. Součástí skrytí je taktéž nastavení zobrazení textu „Skryta“ v seznamu vložených prvků. Pozice v seznamu je získána na základě pozice ve vektoru.

Při zaškrtnutí položky menu „Rolovat na prvek“ je příslušná proměnná *itemScroll* nastavena na hodnotu *true*, která je porovnávána poté při vykreslování označení zvoleného



prvku ve funkci *OnPaint* třídy *cTestWin*. Zde pokud je tato proměnná kladná, pak je získána pozice prvku a pomocí porovnání a výpočtů je návrhové okno posunuto do pozice, kdy je prvek viditelným v tomto okně. Samozřejmě tato funkce má svůj význam pouze v případě, kdy je návrhové okno větší než je prostor pro jeho zobrazení a kdy tedy posunutí v něm je zajištěno pomocí scrollbarů.

Položka „Test“ v menu Nástroje vyvolává náhled okna, jak bude ve skutečnosti vypadat. Funkce tohoto náhledu jsou ve třídě *PreviewWin*, která je potomkem třídy *wxFrame*. Při volání konstruktoru je tety nejprve inicializován právě frame tvořící tento náhled. Poté je do okna umístěn prvek typu *wxPanel*, na který jsou umísťovány prvky procházením vektoru prvků.

Funkce pro zobrazení informací o programu je pouhým vyvoláním okna typu *wxMessageBox*, což je oknem pro zprávy, různá upozornění apod.

## 6.6 Funkce bočního panelu

### 6.6.1 Vytvoření okna

Jak bylo popsáno v části popisu programu pro uživatele, parametry pro nové okno se nastavují v záložce „Formulář“. Zásadními jsou v tomto případě tlačítka Aktualizovat a OK.

Při tlačítku OK jsou nastavené parametry použity pro konstruktor panelu návrhového okna. Pro tlačítko Aktualizovat, které slouží pro načtení současných parametrů návrhového okna, je proces víceméně opačný. Jsou načteny parametry panelu a ty jsou zapsány do prvků pro nastavení okna.

### 6.6.2 Vkládání prvků

Vložení je možné dvěma způsoby. Dvojklikem na daný prvek v záložce „Možnosti“, kdy je prvek vložen do horního levého rohu návrhového okna nebo přetažením z této záložky do návrhového okna.

Pro vložení dvojklikem je využita událost pro *wxListCtrl*, jež je třídou seznamu prvků, *EVT\_LIST\_ITEM\_ACTIVATED*. Po vyvolání této události je vytvořen prvek typu *userItemDescription*. Tento prvek je inicializován na defaultní hodnoty podle typu vkládaného prvku. Následně je vložen do vektoru prvků a vytvořen v návrhovém okně se zadanými parametry.

Vložení přetažením je obdobné. Rozdíl spočívá v tom, že je využita událost *EVT\_LIST\_BEGIN\_DRAG*. Při této akci je při kliku na prvek, zjištěn typ prvku a ten uložen do instance třídy *wxTextDataObject* jako typ *wxString* (textový řetězec). Přenesen je pomocí třídy *DnDObject*, který při puštění tlačítka myši způsobí to samé jako vložení dvojklikem a navíc načte pozici, na kterou byl objekt přetažen a do té pomocí funkce *Move* prvek posune.

### 6.6.3 Nastavení parametrů prvků

Výběr prvku, jehož parametry budou nastavovány, je uskutečňován v záložce „Položky“ pomocí události *EVT\_LIST\_ITEM\_SELECTED*. Po výběru je aktivována a nastavena tabulka s nastaveními. Parametry jsou získány vybraného prvku, který se nachází ve vektoru prvku. Nalezení prvku v tomto vektoru je zajištěno pomocí proměnné *selItem* jež značí pozici prvku ve vektoru a jejíž hodnota je získána pozicí prvku v seznamu prvků.

Samotné nastavování ve většině případů reaguje na událost *EVT\_TEXT* která reaguje na změnu stavu textového pole. Při této události je přímo měněn parametr prvku a s ním i prvek v návrhovém okně.

Výjimku tvoří změna parametru názvu prvku. Zde je potřeba, aby tento název byl jedinečným, a proto je nutná kontrola na tuto jedinečnost až po potvrzení názvu klávesou *Enter*, neboť jinak by byla kontrolována každá změna názvu, což by mohlo způsobit nepotřebné špatné výsledky této kontroly. V tomto důsledku je kontrola tedy prováděna na událost *EVT\_TEXT\_ENTER*. Navíc je zde ale vytvořena i událost *EVT\_TEXT*, při které je hodnota pole uložena do proměnné. Cílem je, aby při přechodu na změnu jiného parametru bez potvrzení názvu byla hodnota uložena. Jelikož toto pole je typu *uTextCtrl*, pak je hodnota proměnné s názvem při tomto přechodu zapsána po kontrole jako parametr prvku.

Dalšími výjimkami jsou dialogy vyvolané jako reakce na tlačítka pro změnu určitého parametru. Ty reagují na událost *EVT\_BUTTON*. Kromě nastavení cesty obrázku jsou pro tyto dialogy vytvořeny zvláštní třídy, které jsou v konstruktoru plněny případnými parametry prvku. Pokud je ve vyvolaném dialogu stisknuto tlačítko *OK*, pak jsou parametry z třídy získány a zapsány jako parametr prvku.

## 6.7 Manipulace s oknem a prvky

### 6.7.1 Změna velikosti okna

Vyvolání akce je na událost *EVT\_SIZE*. Při této události vztahující se k hlavnímu oknu se vyvolá funkce *OnSize*. Tato vyvolává funkci *SetWindowSize*, ve které dochází k ověření, zda návrhové okno existuje a poté k propočtům pro případné zobrazení scrollbarů. Konstanty v této funkci jsou hodnotami okrajů okna omezujícími prostor návrhového okna.

Funkce *SetWindowSize* je využíváno při jakékoliv možné změně velikosti návrhového okna. Těmito změnami mohou být maximalizace, změna velikosti hlavního okna tažením za jeho okraj nebo změna poměru velikostí mezi návrhovým oknem a bočním panelem.

### 6.7.2 Manipulace s prvky návrhového okna

Pro veškerou manipulaci s prvky návrhového okna se využívá třída *cTestWin*. Zásadní jsou tu funkce *OnPaint*, *OnMouseMove*, *OnLeftMouseDown* a *OnMouseUp*.

Funkce *OnPaint* reaguje na událost *EVT\_PAINT*. Tato událost je vyvolána při každé změně vykreslení prvku, v tomto případě tedy návrhového okna. Využita zde je pro označení právě vybraného prvku. Pro toto je především využito instance třídy *wxPaintDC*. Nastavením jejích parametrů je pak vykreslen červeně lemovaný obdélník, jež prvek kopíruje s přesahem 5 bodů pro každý kraj. Tento přesah je úměrně menší, jestliže se prvek nachází v krajních pozicích okna.

Funkce *OnLeftMouseDown* reaguje na událost *EVT\_LEFT\_DOWN*, neboli stisknutí levého tlačítka myši. Prvotně zachytí pozici myši, kde bylo vykonáno kliknutí a cyklickým procházením vektoru prvků kontroluje, zda se na této pozici vyskytuje prvek okna. Případný prvek je uložen pro další zpracování. V této fázi je zároveň prvek zvýrazněn jako vybraný.

Funkce *OnMouseMove* je akcí na událost *EVT\_MOTION*, tedy pohyb myši. Nejdříve kontroluje, zda je stisknuta klávesa CTRL a podle toho nastavuje, jestli má návrhové okno zachycovat operace myši. Pokud před vyvoláním této akce byla již akce stisknutí tlačítka, pak s prvkem činí manipulaci podle toho, ve které pozici byl stisk učiněn. Pokud by na hranici okraje prvku s tolerancí  $\pm 5$  bodů pak je prvek zvětšován nebo

zmenšován v daném směru tahu myši. Pokud byl stisk uvnitř prvku, pak je prvek posunován. Těmto akcím jsou přiřazeny patřičné kurzory myši.

Při funkci *OnMouseUp* reagující na *EVT\_LEFT\_UP* (uvolnění levého tlačítka myši), jsou veškeré provedené změny zapsány zpět do vektoru prvků a nastavena tabulka s nastaveními prvku.

## 7 KOMPATIBILITA S JINÝMI PROGRAMY

Jedním z hledisek při vytváření programu bylo i to, aby vytvořený soubor okna byl použitelný i pod jinými programy než je tento vytvořený program. Právě z tohoto důvodu byl jako standard pro ukládání volen formát XRC, který lze zpracovat programy, které tento formát podporují.

V tomto případě byl pro ověření použit program *Anthemion DialogBlocks 4.30*, který slouží rovněž pro navrhování dialogových oken. Výsledné zobrazení v tomto programu je uvedeno v příloze této práce P III. Porovnat toto zobrazení lze s přílohou P I, kde je zobrazení ve vytvořeném programu *DialogEditor*.

Samozřejmě je i to, že okna vytvořená v jiném programu podporujícím XRC lze otevřít ve vytvořeném programu *DialogEditor*. Ovšem podmínkou v tomto případě je to, aby vytvořený soubor obsahoval pouze prvky podporované vytvořeným programem.

## ZÁVĚR

V teoretické části práce byla vypracována literární rešerše týkající se především knihovny wxWidgets a prostředků s ní souvisejících jako je jazyk C++ a formát XRC. Všechny tyto prostředky jsou potřebné pro vytvoření programu jež je součástí praktické části. Teoretická část byla rozdělena na tři menší celky.

První část se zabývala samotnou knihovnou wxWidgets. V této části je obsažen obecný popis knihovny, historie a vývoj této knihovny a v neposlední řadě architektura knihovny. Součástí jsou i příklady programů postavených na základech právě wxWidgets.

Druhá část popisuje stručně programovací jazyk C/C++ a to především jeho historii a výhody jazyku C++ oproti jazyku C (zapouzdření, dědičnost a mnohotvárnost). Nedílnou součástí je i implementace knihovny wxWidgets pro tento programovací jazyk.

Třetí část rozebírá detailně formát XRC, jež je standardním formátem wxWidgets používaným pro ukládání dialogových oken. Hlavním obsahem této části je struktura a přípustné elementy pro tento standard. V popisu je pak popsáno jak formát XRC souvisí s formátem XML, který je dnešní době velice používaným.

Praktická část je především zaměřena na popis programu, který je vytvořen jako součást této práce a jeho použití je jako editor dialogových oken. Tudíž jeho náplní je vytváření, editace, uložení a znovunačtení projektů s dialogovými okny, přičemž jako souboru pro uložení je využito formátu XRC.

Většina praktické části je rozdělena do dvou samostatných celků. Těmito celky jsou popisy vytvořeného programu ze dvou různých pohledů, a to z pohledu uživatele a z pohledu programátora. Zatímco uživatelský popis je víceméně manuálem pro použití programu a seznamuje laika jak co v programu používat, tak programátorský popis se zabývá popisem řešení jednotlivých funkcí, které může uživatel v programu používat a popisem stěžejních prvků celého programu.

V závěru praktické části je pak důkaz o tom, že dialogové okno je použitelným i pod jinými programy než je tento editor.

## ZÁVĚR V ANGLIČTINĚ

In theoretical part of this work, there is a literature search mainly about a wxWidgets library and features included along with it such as C++ language and XRC format. All these features are required to create a program which is included in the practical part.

First part deals with the wxWidgets library as such. A general description of this library, its history and development are all included in this part. Examples of programs which are based in particular on the wxWidgets are also included.

Second part briefly describes C++ programming language, mainly its history and advantages of C++ language over C language (casing, heredity and diversification). An integral part of this work is also an implementation of wxWidgets library in this programming language.

Third part deals with XRC format in detail. It is a standard format of wxWidgets, used to save dialogue boxes. Main subject of this part is a structure and acceptable elements for this standard.

Practical part is mainly aimed at describing the program, which is created and included as a part of this thesis, and its purpose is to serve as a dialog windows editor. Therefore its job is creating, modification, saving and loading of projects with dialog windows while the XRC format is used as a saving file.

Most of the practical part is divided into two separate parts. These parts are descriptions of the program from two different points of view – view of a user and of a programmer aswell. While the user description is more or less a manual for program usage and familiarizes a secular with an actual use of this program, programmer description is concerned with solutions of particular functions which a user can use in this program and also with a description of the fundamental features of the whole program.

On conclusion of the practical part, there is evidence of the possibility that the dialog window is also compatible under other programs and not only under this editor.

**SEZNAM POUŽITÉ LITERATURY**

- [1] BLAŽEK, Michal. *XML pro začátečníky : 1. část* [online]. 2007 , 27.3.2007 [cit. 2009-05-10]. Dostupný z WWW: < <http://programujte.com/index.php?akce=clanek&cl=2007030501-xml-pro-zacatecniky-1-cast>>
- [2] KOSEK, Jiří. *XML* [online]. c1999 , 28.5.1999 [cit. 2009-05-10]. Dostupný z WWW: <<http://www.kosek.cz/clanky/xml/>>
- [3] SMART, Julian, HOCK, Kevin, CSOMOR, Stefan. *Cross-Platform GUI Programming with wxWidgets*. [s.l.] : [s.n.], 2005. 700 s.
- [4] ŠALOUN, Petr. *Programovací jazyk C++ pro zelenáče*. [s.l.] : Neocortex, c2005. 254 s. ISBN 80-86330-18-4
- [5] *C (programovací jazyk) - Wikipedie, otevřená encyklopedie* [online]. [2000] , 3.5.2009 [cit. 2009-05-10]. Dostupný z WWW: <[http://cs.wikipedia.org/wiki/C\\_\(programovac%C3%AD\\_jazyk\)](http://cs.wikipedia.org/wiki/C_(programovac%C3%AD_jazyk))>
- [6] *C++ - Wikipedie, otevřená encyklopedie* [online]. [2000] , 10.3.2009 [cit. 2009-05-10]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/C%2B%2B>>
- [7] *MPP: Instalace a integrace vývojových nástrojů pro wxWidgets* [online]. [2006] , 14. říjen 2008 [cit. 2009-05-10]. Dostupný z WWW: <<http://vyuka.fai.utb.cz/mod/resource/view.php?id=1376>>
- [8] *Using XML Resources with XRC - WxWiki* [online]. [2004] , 26 March 2009 [cit. 2009-05-10]. Dostupný z WWW: <[http://wiki.wxwidgets.org/Using\\_XML\\_Resources\\_with\\_XRC](http://wiki.wxwidgets.org/Using_XML_Resources_with_XRC)>
- [9] *WxWidgets* [online]. [1997] , 8.5.2009 [cit. 2009-05-10]. Dostupný z WWW: <<http://www.wxwidgets.org/>>
- [10] *WxWidgets - Wikipedie, otevřená encyklopedie* [online]. [2000] , 21.3.2009 [cit. 2009-05-10]. Dostupný z WWW: <<http://cs.wikipedia.org/wiki/WxWidgets>>
- [11] *WxWidgets Czech Translate* [online]. 2008 , 10.4.2009 [cit. 2009-05-10]. Dostupný z WWW: <[http://www.wxwidgets.hustej.net/wiki/index.php/Hlavn%C3%AD\\_strana](http://www.wxwidgets.hustej.net/wiki/index.php/Hlavn%C3%AD_strana)>



**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

|             |  |
|-------------|--|
| ANSI        | American National Standards Institute, americká norma  |
| API         | application programming interface, rozhraní pro programování aplikací  |
| Digram      | skupina dvou písmen  |
| DTD         | definice typu dokumentu  |
| GTK+        | Gimp Toolkit, open source knihovna pro vytváření GUI   |
| GUI         | Graphical User Interface, grafické uživatelské rozhraní  |
| ISO         | International Organization for Standardization, mezinárodní norma  |
| MGL         | nízkoúrovňová grafická knihovna firmy SciTech  |
| Open source | program jež má otevřený zdrojový kód a je zdarma přístupný   |
| STL         | Standard Template Library, knihovna šablon   |
| Widget      | prvek uživatelského rozhraní   |
| Win32API    | programovací rozhraní pro tvorbu aplikací s grafickým rozhraním využívající funkce systému Windows                 |
| XML         | Extensible Markup Language   |
| XPM         | X-Picture Map, formát obrázku, kde každé barvě pixelu je přiřazen symbol a pomocí těchto symbolů obrázek vykreslen |
| XRC         | XML Resource   |

**SEZNAM OBRÁZKŮ**

|  |    |
|--|----|
| Obr. 1: Základní vrstvy API wxWidgets.....                               | 13 |
| Obr. 2: Základ souboru XRC.....  | 20 |
| Obr. 3: Základní zobrazení okna programu.....                            | 29 |
| Obr. 4: Menu Soubor.....   | 30 |
| Obr. 5: Menu Nástroje.....   | 30 |
| Obr. 6: Menu Nápověda.....   | 31 |
| Obr. 7: Záložka Formulář.....  | 31 |
| Obr. 8: Záložka Možnosti.....  | 32 |
| Obr. 9: Záložka Položky.....   | 32 |
| Obr. 10: Příklad kurzoru pro změnu velikosti (vlevo) a změnu pozice..... | 34 |
| Obr. 11: Datový typ userItemDescription.....                             | 36 |
| Obr. 12: Ukázka konfiguračního souboru.....                              | 38 |

## SEZNAM TABULEK

|  |    |
|--|----|
| Tab. 1: Seznam tříd a jejich obsahu..... | 35 |
|--|----|

## SEZNAM PŘÍLOH

Příloha P I – Okno aplikace ve Windows XP a Vista

Příloha P II – Vytvořené okno ve Windows XP a Vista

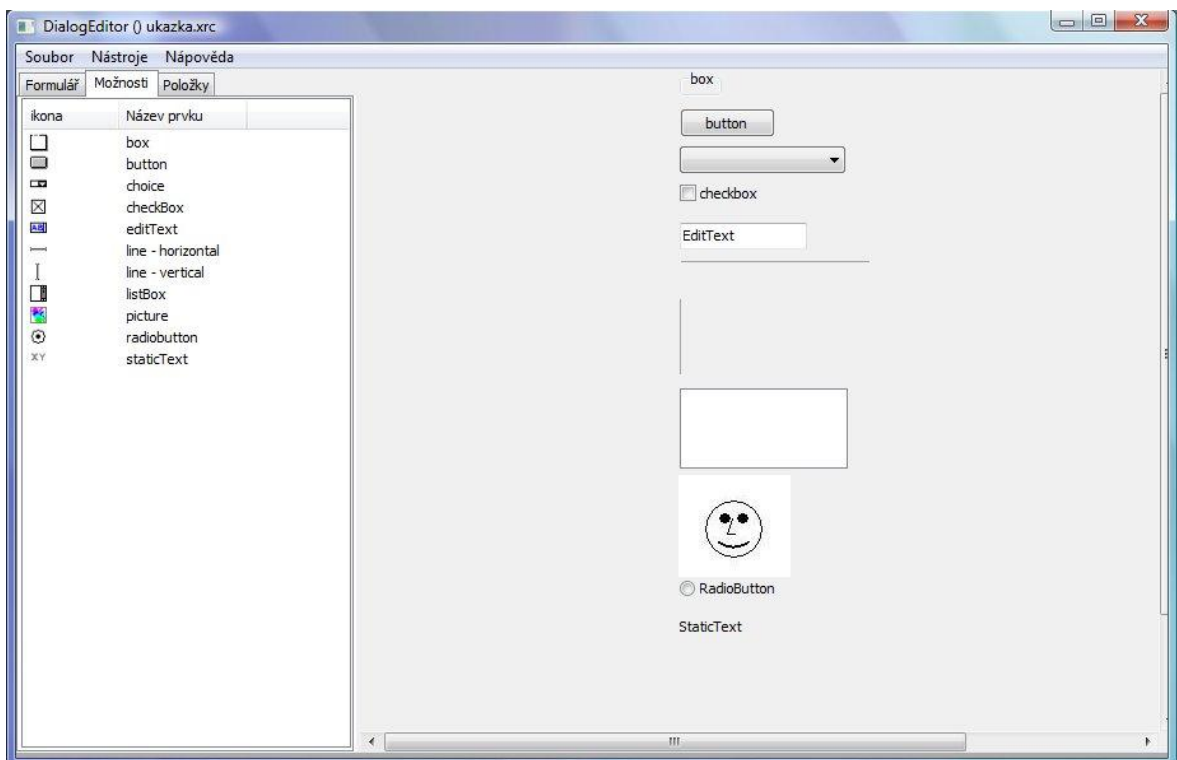
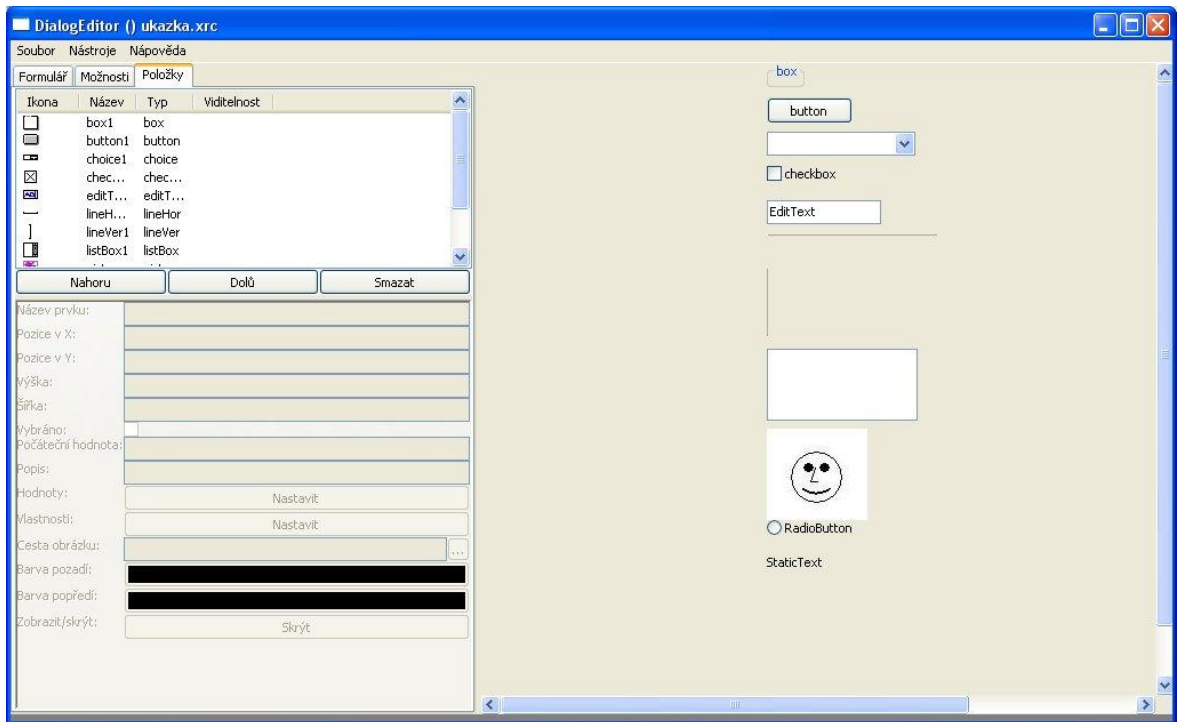
Příloha P III – Zobrazení v programu Anthemion DialogBlocks 4.30

Příloha P IV – Ukázka XRC souboru

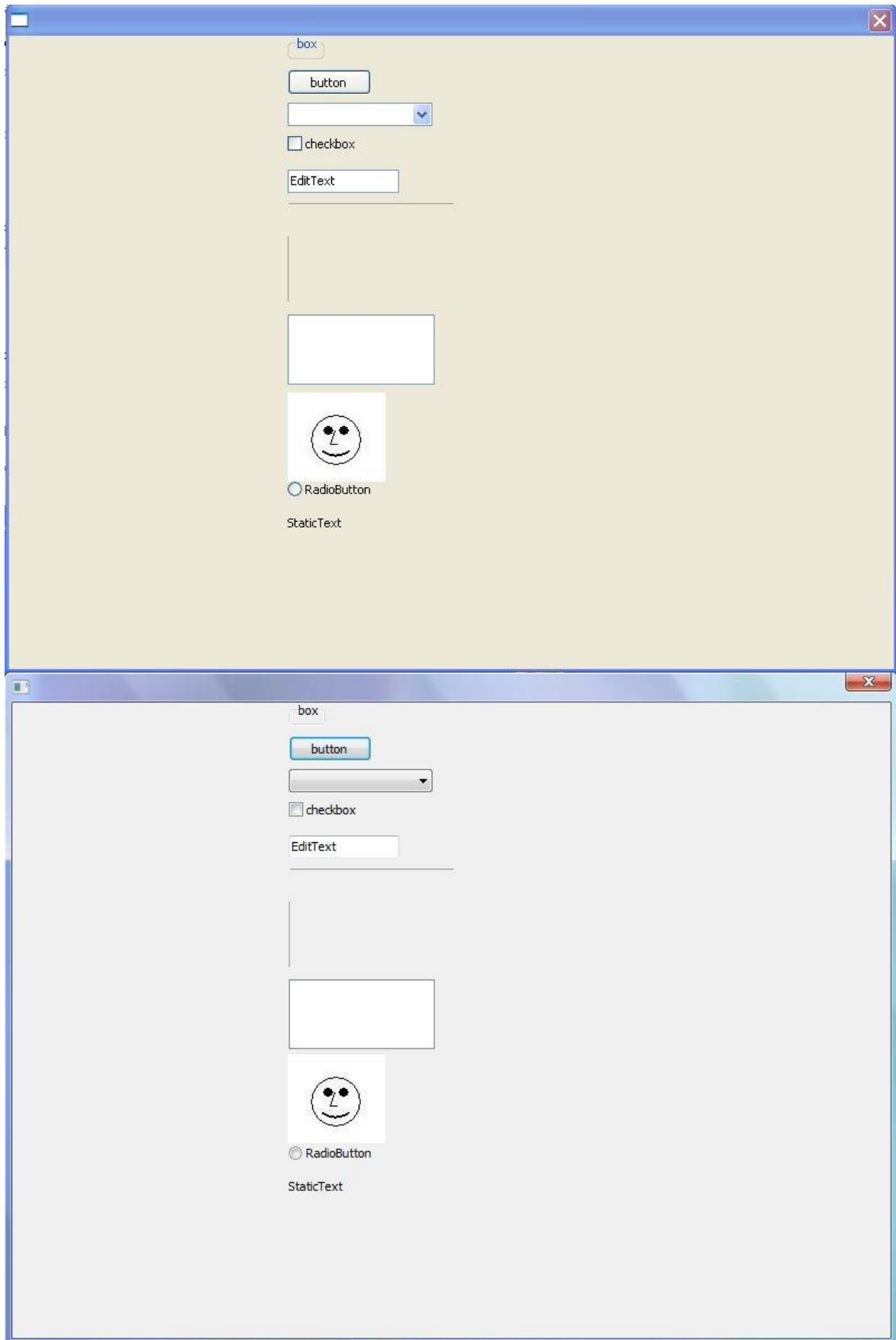
Příloha P V – Ukázka XPM souboru

Příloha P VI – Struktura přiloženého CD

# PŘÍLOHA P I: OKNO APLIKACE VE WINDOWS XP A VISTA

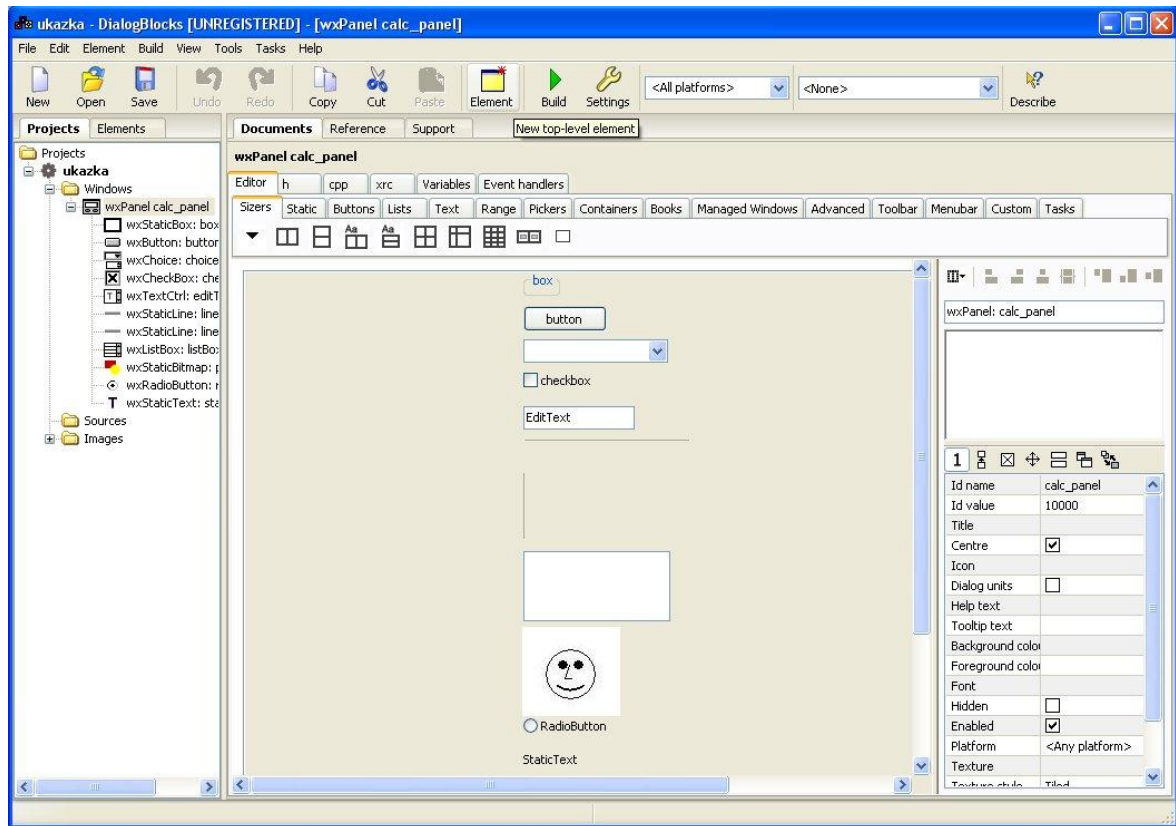


## PŘÍLOHA P II: VYTVOŘENÉ OKNO VE WINDOWS XP A VISTA



# PŘÍLOHA P III: ZOBRAZENÍ V PROGRAMU ANTHEMION

## DIALOGBLOCKS 4.30



## PŘÍLOHA P IV: UKÁZKA XRC SOUBORU

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<resource version="2.3.0.1" xmlns="http://www.wxwidgets.org/wxxrc">
  <object class="wxPanel" name="calc_panel">
    <style>wxTAB_TRAVERSAL</style>
    <exstyle>wxWS_EX_BLOCK_EVENTS</exstyle>
    <size>800,600</size>
    <title></title>
    <centered>1</centered>
    <object name="box1" class="wxStaticBox">
      <label>box</label>
      <pos>257,239</pos>
    </object>
    <object name="button1" class="wxButton">
      <label>button</label>
      <pos>221,52</pos>
    </object>
    <object name="choice1" class="wxChoice">
      <pos>215,118</pos>
      <selection>0</selection>
    </object>
    <object name="checkBox1" class="wxCheckBox">
      <label>checkbox</label>
      <pos>10,131</pos>
      <checked>0</checked>
    </object>
    <object name="editText1" class="wxTextCtrl">
      <value>EditText</value>
      <pos>13,78</pos>
    </object>
    <object name="lineHor1" class="wxStaticLine">
      <style>wxLI_HORIZONTAL</style>
      <size>40,1</size>
      <pos>0,0</pos>
    </object>
    <object name="listBox1" class="wxListBox">
      <style>wxLB_SINGLE</style>
      <pos>0,207</pos>
      <selection>0</selection>
    </object>
    <object name="picture1" class="wxStaticBitmap">
      <pos>140,144</pos>
    </object>
    <object name="radioButton1" class="wxRadioButton">
      <label>RadioButton</label>
      <pos>76,37</pos>
      <value>0</value>
    </object>
    <object name="staticText1" class="wxStaticText">
      <label>StaticText</label>
      <pos>18,288</pos>
    </object>
  </object>
</resource>
```



## PŘÍLOHA P V: UKÁZKA XPM SOUBORU

```
/* XPM */
static char *button_xpm[] = {
/* columns rows colors chars-per-pixel */
"16 16 5 1",
" c None",
". c Black",
"X c #FFFFFF",
"o c #808080",
"O c #C0C0C0",
/* pixels */
"
",
"
",
" ..... ",
" .XXXXXXXXXXXXX.o",
" .XOOOOOOOOOOO.o",
" .XOOOOOOOOOOO.o",
" .XOOOOOOOOOOO.o",
" .XOOOOOOOOOOO.o",
" .XOOOOOOOOOOO.o",
" .XOOOOOOOOOOO.o",
" .XOOOOOOOOOOO.o",
" .XOOOOOOOOOOO.o",
" .....oo",
" oooooooooooooo ",
"
",
"
",
"
",
};
```

## **PŘÍLOHA P VI: STRUKTURA PŘILOŽENÉHO CD**

- /práce

Obsahuje text bakalářské práce ve formátech DOC a PDF

- /programy

Obsahuje program Anthemion DialogBlocks 4.30, na kterém je testována kompatibilita vytvořeného okna a wxWidgets 2.8.10

- /projekt

Obsahuje vytvořený program, spouští se souborem DialogEditor.exe

- /příklad

Obsahuje vzorový projekt pro program