

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Student	: Pavel Čápek
Vedoucí bakalářské práce	: doc. Ing. Alena Buchalcevová, Ph.D.
Oponent bakalářské práce	: RNDr. Vladimír Tichý

Téma bakalářské práce:

NetBeans GUI Builder

ROK: 2010

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal.

V Praze dne 30. 6. 2010

.....

podpis

Poděkování

Rád bych tímto poděkoval vedoucí své práce doc. Ing. Aleně Buchalceové, Ph.D. za čas a pomoc, kterou mi věnovala.

Abstrakt

Cílem této bakalářské práce je prohloubit čtenářovy znalosti nástroje NetBeans GUI Builder a rozšířit tak jeho možnosti při používání nástroje NetBeans GUI Builder během vytváření grafického uživatelského rozhraní v jazyku Java. Čtenář je postupně seznamován s různými třídami knihovny Swing a na základních příkladech je mu předváděna funkcionality a vlastnosti těchto tříd. Autor čerpá ze svých vlastních zkušeností s nástrojem NetBeans GUI Builderu a formou popisu-návodu společně s množstvím obrázků ukazuje použití jednotlivých tříd. Po přečtení této práce bude mít čtenář znalosti, které mu pomohou využít přednosti nástroje NetBeans GUI Builder.

Annotation

The goal of the bachelor thesis is to improve reader's knowledge in NetBeans GUI Builder and expand its possibilities of using NetBeans GUI Builder for creating graphical user interface in Java programming language. The readers will be acquainted with some Swing classes and will be shown how these classes work and their properties. Author uses his own experience from NetBeans GUI Builder and form description – instruction with a lot of pictures shows a using of every single class. In the end the reader will be better in using NetBeans GUI Builder.

Obsah

1. Úvod.....	8
1.1. Vymezení tématu práce a důvod výběru tématu	8
1.2. Cíle práce	8
1.3. Metoda dosažení cíle	8
1.4. Předpoklady a omezení práce	8
1.5. Vývojové prostředí NetBeans a jeho porovnání s ostatními IDE.....	9
1.6. Návrh uživatelského rozhraní	9
2. Správce rozložení	11
2.1. Free design – volný režim návrhu.....	12
2.2. AbsoluteLayout.....	12
2.3. BorderLayout.....	13
2.4. BoxLayout	13
2.5. CardLayout.....	14
2.6. FlowLayout.....	14
2.7. GridBagLayout	14
2.8. NullLayout.....	15
2.9. GridLayout	15
3. Nástroje pro ovládání aplikace - Swing Menus, Swing Containers	16
3.1. TabbedPane.....	16
3.2. MenuBar	19
3.2.1. Přidání klávesových zkratk.....	20
3.3. Nástrojová lišta.....	21
3.3.1. vlastnosti JToolBaru	21
4. Textová pole, Textové oblasti	22
4.1. JTextField.....	22
4.2. JPasswordField	22
4.3. JFormattedField	22
4.4. JTextArea	23
4.5. JEditorPane	24
4.6. JTextPane.....	24
5. Složitější Swing komponenty	25
5.1. JList komponenta.....	25
5.1.1. Předávání vybrané položky JListu	25

5.1.2. Přidání položek do JListu.....	27
5.1.3. Import položek do JListu z databáze.....	30
5.1.4. JList – některé důležité vlastnosti	32
5.2. JTable.....	33
5.2.1. Předávání dat z tabulky dalším třídám	35
5.2.2. Používání třídy JTable bez propojení s databází	36
5.2.3. Propojení třídy JTable s databází	36
5.2.4. Propojení tabulky s databází a úprava záznamů v databázi	38
5.2.5. Selection model u třídy JTable.....	40
6. Java Persistence, NetBeans IDE 6.9	41
7. Závěr	44
8. Seznam literatury a zdrojů	45
9. Terminologický slovník.....	47

1.Úvod

1.1. Vymezení tématu práce a důvod výběru tématu

Téma bakalářské práce NetBeans GUI Builder jsem si vybral z důvodu přesvědčení, že toto téma přinese čtenáři užitečné informace, které mu pomohou zvládnout tento poměrně nový nástroj (v době svého uvedení znám jako Matisse GUI Builder), který se ve velmi oblíbeném open source vývojovém prostředí pro Javu NetBeans objevil od verze NetBeans 5.0 (vydána roku 2006)[9]. Od té doby je tento GUI Builder pevnou součástí NetBeans IDE. Dalším důvodem ke zpracování toho tématu byla poměrně malý výskyt zdrojů, jak tištěných, tak elektronických, které by se zabývaly tématem vytváření GUI pomocí nástroje NetBeans GUI Builder.

1.2. Cíle práce

Cílem mé práce je seznámit čtenáře s komponentami, které mu nabízí NetBeans GUI Builder, ukázat jejich nejčastější použití a seznámit čtenáře s jejich vlastnostmi a nastavením. Čtenář je v první části práce seznámen se správcí rozložení, které nabízí NetBeans GUI Builder, dále s jednotlivými komponentami, které se používají při vytváření GUI, jako jsou nástrojové lišty, panely a textové komponenty. Dále je čtenář seznámen se složitějšími komponentami jako např. JList, JTable, jejichž používání v NetBeans GUI Builderu má své limity.

Tato práce ukazuje výhody a limity, které přináší používání nástroje NetBeans GUI Builder a pomáhá čtenáři zrychlit a zefektivnit počáteční práce a seznamování s tímto nástrojem.

1.3. Metoda dosažení cíle

Práce je tvořena formou popisu-návodu, kdy je vybraná třída, popsána a poté, pokud je to potřeba, následuje ukázka použití této třídy.

1.4. Předpoklady a omezení práce

Předpokladem k využití této práce je, že čtenář má alespoň základní znalosti programovacího jazyku Java a fungování jeho principů. Další předpoklad je, že čtenář má

alespoň základní zkušenosti s prací ve vývojovém prostředí NetBeans, případně jiném, které používá nástroj GUI Builder. Pokud čtenář tyto zkušenosti nemá, mohu mu doporučit literaturu [3], [4]. Tato práce na ně volně navazuje a věnuje se problémům, které v nich nejsou popsány buď z důvodu rozsáhlosti práce, nebo jde o nové třídy a nástroje, které nebyly obsaženy v předchozích verzích IDE NetBeans.

1.5. Vývojové prostředí NetBeans a jeho porovnání s ostatními vývojovými nástroji

NetBeans společně s Eclipse představují špičky mezi open source vývojovými prostředími. Obě vývojová prostředí nabízejí možnost jak vytvářet grafické uživatelské rozhraní pomocí nástroje GUI editor [15]. Není mým cílem srovnávání těchto dvou vývojových prostředí, ale dle mého názoru v oblasti vývoje GUI pomocí nástrojů GUI editorů má prostředí NetBeans velký náskok nad svým konkurentem. Zejména díky faktu, že nástroj NetBeans GUI Builder je standardní součástí prostředí NetBeans IDE, zatímco v prostředí Eclipse je potřeba doinstalovat nějaký plug-in z nabídky (například Jigloo Swing GUI Builder, Eclipse GUI Builder, a jiné – některé tyto doplňky jsou placené)[5]. Ale ani tyto plug-iny nedosahují takového komfortu a úrovně jako nástroj NetBeans GUI Builder.

Kromě těchto dvou vývojových prostředí a jejich nástrojů, které slouží k snadnému vytváření GUI, existují i další, zejména komerční nástroje, které také nabízejí možnost pohodlného vytváření GUI pomocí GUI editorů, jsou to například IntelliJ IDEA od společnosti JetBrains, JBuilder od firmy Borland, nebo JDeveloper firmy ORACLE. S těmito produkty nemám bohužel zkušenosti, takže nemůžu zhodnotit jejich nástroje pro vytváření GUI.

Při zpracování této bakalářské práce bylo použito vývojové prostředí NetBeans 6.8

1.6. Návrh uživatelského rozhraní

Uživatelské rozhraní představuje nástroje, kterými uživatel ovládá chování počítačového programu. Uživatel ovlivňuje počítačový program pomocí vstupů a získává

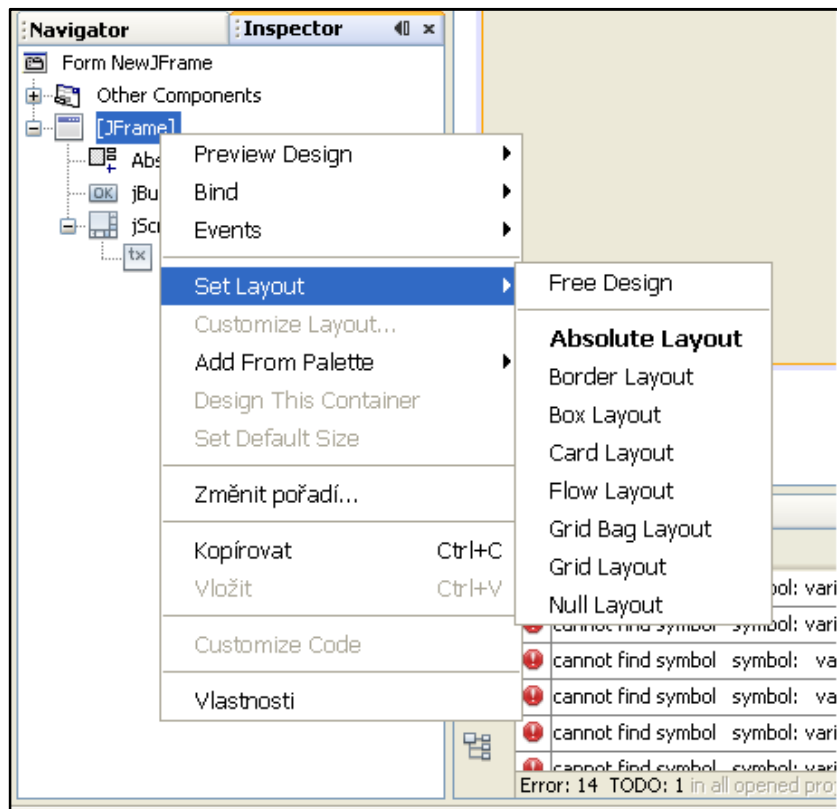
od něj výstupy. Současné nejpoužívanější uživatelské rozhraní je grafické uživatelské rozhraní - GUI. Předcházela mu textové uživatelské rozhraní a ještě před tím uživatelské rozhraní pomocí příkazové řádky. Uživatelské rozhraní je klíčové pro práci s programem a z pohledu uživatele je uživatelsky vstřícné a intuitivní GUI jeden ze základních faktorů hodnocení kvality programu. Grafické uživatelské rozhraní má několik psaných i nepsaných pravidel, které by měly být brány v potaz při vývoji aplikací, patří mezi ně např. přítomnost horní nástrojové lišty (Menu Bar), která obsahuje hlavní nastavení a možnosti ovládání aplikace, pod ní bývá umístěna lišta Tool bar, která se většinou skládá z grafických ikon, které reprezentují často používané akce v aplikaci a jejich používání má uživateli zjednodušit a urychlit práci.

Nástroj NetBeans GUI Builder představuje výrazné zjednodušení při vytváření uživatelského rozhraní v aplikacích založených na platformě Java. Umožňuje vytvořit rychleji a jednodušeji kvalitní GUI i začínajícím programátorům, kteří by dříve vytvářeli GUI jen pomocí psaní zdrojového kódu. NetBeans GUI Builder poskytuje nástroje, díky kterým je možné vytvořit GUI, které bude vyhovovat současným nárokům a potřebám.

2. Správce rozložení

Prostředí NetBeans GUI Builder nabízí uživateli několik rozložení, kromě rozložení Free design, které je standardně nastaveno při vytvoření nového formuláře (JFrame), nebo panelu tu jsou k dispozici i další rozložení z knihovny Swing jako např. Absolute Layout, Border Layout, Box Layout, Card Layout, Grid Layout, Grid Bag Layout, Grid Layout, Null Layout.

Správce rozložení pro vybranou komponentu lze nastavit v levém dolním panelu Inspector (viz. Obr. 2.1)



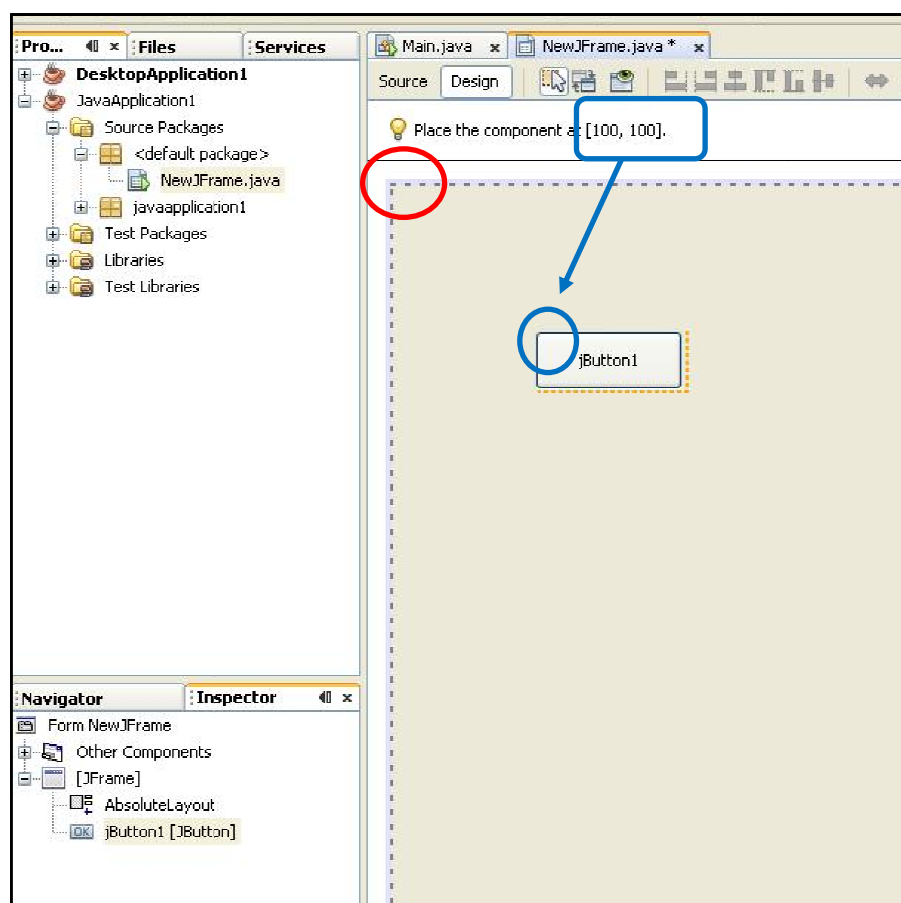
Obr. 2.1: Výběr rozložení [autor]

2.1. Free design – Volný režim návrhu

Free design je zřejmě nejoblíbenější a nepoužívanější rozložení v NetBeans GUI Builder. Toto rozložení používá k umístění prvku vzájemnou polohu vůči ostatním prvkům na formuláři, nebo k samotnému formuláři pomocí vodících čar a ukotvení. Práce s tímto rozvržením byla už popsána např. v pracích [3],[4].

2.2. AbsoluteLayout

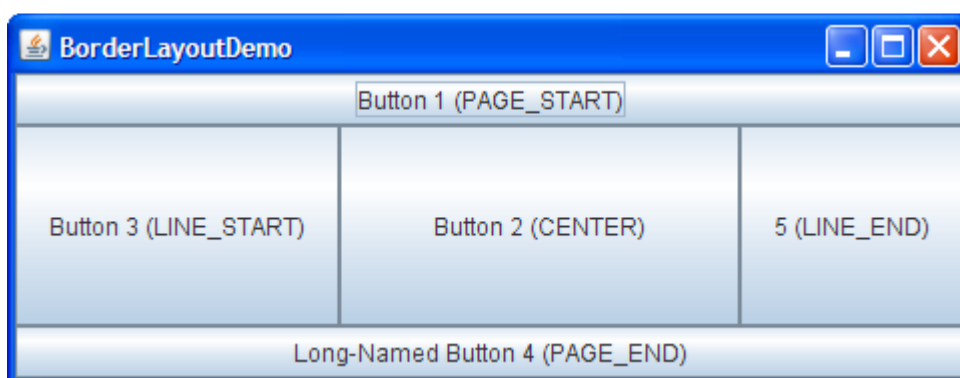
Při práci s tímto rozložením se při umísťování komponent nepoužívají vodící čáry a kotvy uchycení jako v rozložení Free design, ale umístění se řídí pouze soustavou souřadnic. Při pohybu grafickým prvkem na rozložení je vidět jeho aktuální poloha (poloha levého horního rohu - modrá barva) vůči počátku soustavy souřadnic (červená barva), (viz. Obr. 2.1). V tomto rozložení, stejně tak jako v předešlém je možné měnit velikost grafických komponent.



Obr. 2.2: Rozložení Absolute Layout [autor]

2.3. BorderLayout

Jedno z klasických rozložení knihovny Swing, které je známé už z dřívějších, kdy se GUI vytvářelo hlavně pomocí psaní zdrojového kódu. Toto rozložení je rozděleno do čtyř hraničních a jedné středové oblasti (North, South, West, East, Center), (viz. Obr. 2.3). V NetBeans GUI Builder se při tomto rozložení nedá měnit velikost grafických prvků, lze pouze měnit jejich pozice na rozložení. (Měnit velikost grafického prvku lze zprostředkovaně – např. umístit na vybranou pozici JPanel a na něm nastavit nové rozložení, které umožňuje měnit velikost prvků).



Obr.2.3: rozložení Border Layout převzato z [12]

2.4. BoxLayout

Toto rozložení vkládá jednotlivé grafické prvky do formuláře postupně za sebou a je možné měnit pouze jejich pořadí. Ve vlastnostech rozložení je možné určit orientaci, podle které se mají grafické prvky řadit – svisle, nebo vodorovně (viz. Obr. 2.4). BoxLayout také umožňuje nastavit rozteč mezi jednotlivými grafickými prvky.



Obr.2. 4: rozložení BoxLayout [autor]

2.5. CardLayout

Rozložení CardLayout zachází s grafickými prvky jako s kartami, přičemž je viditelný pouze poslední přidáný grafický prvek, ostatní předchozí prvky jsou naskládány pod ním. Toto rozložení se používá zejména, pokud uživatel chce, aby dvě nebo více komponent sdílelo stejný prostor a jejich zobrazení, by bylo podmíněno určitou akcí uživatele například výběrem možnosti z combo boxu, nebo při tvorbě průvodce, instalátorů ve stylu Windows. Další způsob, jak dosáhnout podobného výsledku, bez použití CardLayout je možnost použít záložky panelu – tabbed pane.

2.6. FlowLayout

Toto rozložení nechává prvky na formulář „volně padat“. Ve vlastnostech rozložení je možné určit, zda se budou grafické prvky řadit zleva, zprava, nebo od středu. Prvky se ukládají, dokud není zaplněna oblast podél horního okraje formuláře (viz. Obr. 2.5). Potom se vkládané prvky přesunou na další řádek a celý proces pokračuje znovu. Při vkládání zůstává grafickým prvkům jejich „původní“ velikost.



Obr. 2.5: rozložení FlowLayout převzato z [12]

2.7. GridBagLayout

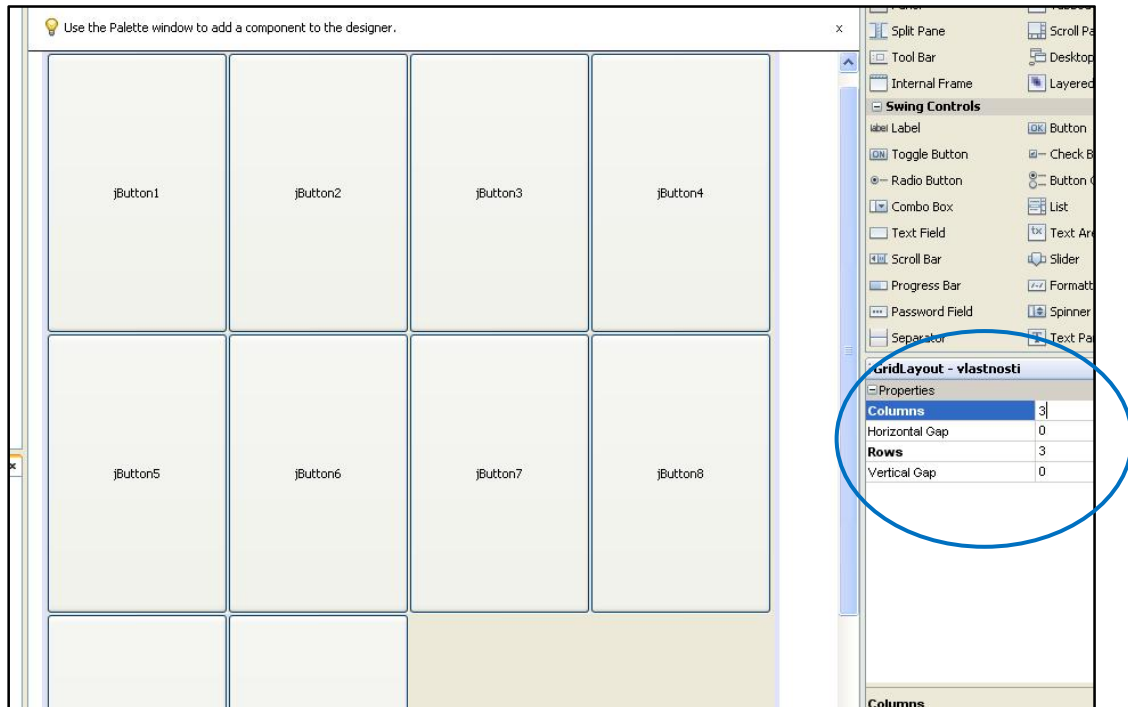
Toto rozložení se použije spíše při vytváření grafického uživatelského rozhraní pomocí psaní kódu. Při použití NetBeans GUI Builderu jsem nenašel žádné uplatnění pro toto rozložení, ve vlastnostech tohoto rozložení není žádná nabídka. Vložené grafické prvky pouze zarovnávají na střed formuláře.

2.8. NullLayout

Toto rozložení se velmi podobné jako rozložení AbsoluteLayout – pozice grafického prvku se také řídí podle souřadnic a jejich velikost lze na formuláři měnit. Drobné rozdíly jsou v těchto rozloženích při práci v režimu psaní kódu – např. přidávání komponent v rozvržení NullLayout se musí každá komponenta přidat pomocí souřadnic x, y a parametrů šířka, výška komponenty, zatímco v rozvržení AbsoluteLayout může uživatel přidat už vytvořenou komponentu. Pokud uživatel pracuje s těmito rozloženími v NetBeans GUI Builderu, může je považovat za shodné.

2.9. GridLayout

Rozložení GridLayout umožňuje vytvářet tabulku komponent. Velikost grafických prvků je po vložení přizpůsobena tomu, aby odpovídala, ve vlastnostech zvolenému počtu řádků a sloupců k velikosti formuláře. Ve vlastnostech rozložení je také možné nastavit, jak velké budou mezery mezi jednotlivými prvky (viz. Obr. 2.6).



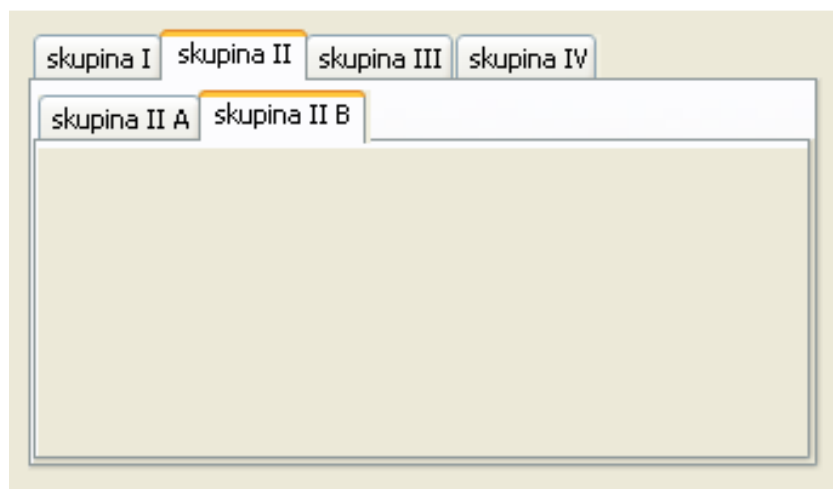
Obr. 2.6: rozložení GridLayout [autor]

3. Nástroje pro ovládání aplikace - Swing Menus, Swing Containers

Mezi nástroje pro ovládání aplikace patří především nástroje pro vytváření horní nástrojové lišty a komponenta tool bar,

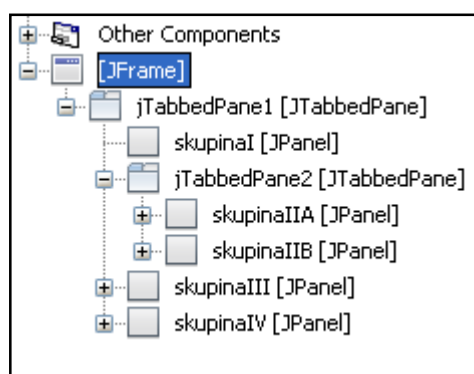
3.1. Tabbed Pane

Tato komponenta z knihovny Swing poskytuje možnost několika komponentám sdílet stejný prostor a pomocí výběru zobrazit pouze jednu z nich. Podrobněji popsat používání této komponenty jsem se rozhodl z důvodů komplikací, které jsem s ní měl během zpracování bakalářské práce. Komponenta Tabbed Pane umožňuje nastavit několik záložek vedle sebe a dále i pak záložky vnořovat pod sebe. Výsledná struktura může vypadat například takto: (viz. Obr. 3.1)



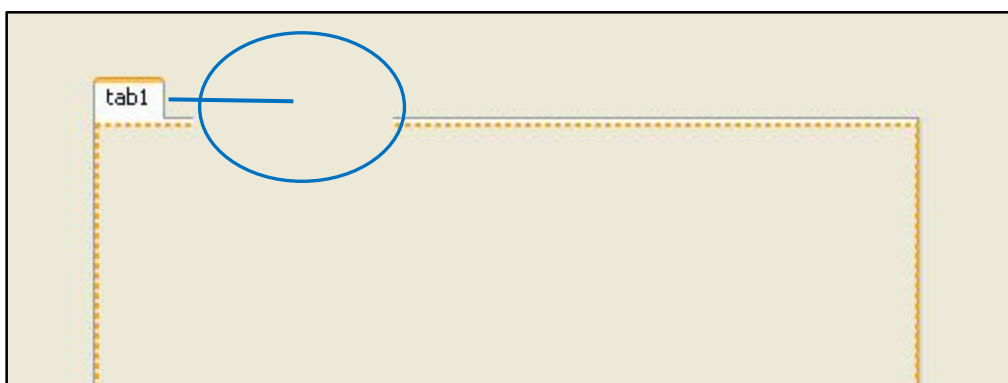
Obr. 3.1: Tabbed Pane [autor]

Při práci s komponentou Tabbed Pane se mi nejlépe osvědčilo postupovat podle následujících pravidel: počet úrovní záložek = počet komponent Tabbed Pane na formuláři (viz. Obr. 3.2). Takto vypadá schéma Tabbed Pane z obrázku 3.1 v oknu Inspector. Záložka první úrovně (skupina I, skupina II, ...) je jeden prvek TabbedPane, který v sobě obsahuje komponenty JPanel, teprve pokud chci vytvořit nižší úroveň záložek, vložím další komponentu JTabbedPane.



Obr. 3.2 : struktura JTabbedPane v oknu Inspector [autor]

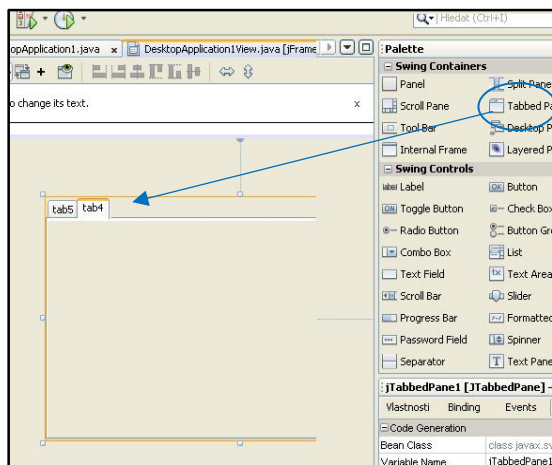
Pokud vytvářím první úroveň záložek, vložím na formulář prvek TabbedPane, upravím jeho pozici a rozměry podle potřeby, poté do něj vložím komponentu JPanel (případně jinou, ale tato komponenta bude vyplňovat celou velikost záložky), následně mohu s touto záložkou pracovat jako s komponentou JPanel. Pokud chci vložit další záložku, která bude na stejné úrovni jako první, použiju k tomu komponentu JPanel (viz. Obr. 3.2).



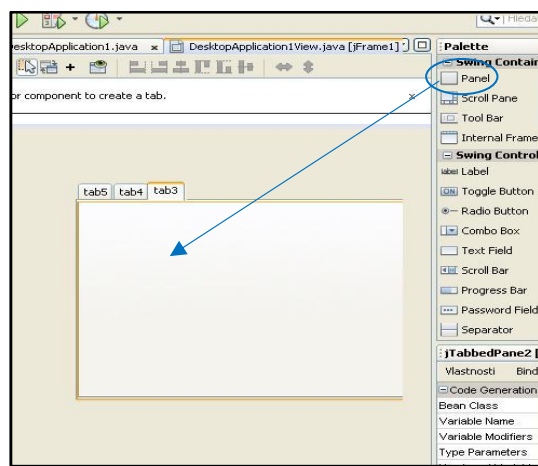
Obr. 3.3: Vkládání další záložky [autor]

Pokud vkládám další záložku(JPanelu) přibližně do míst označenou modrou barvou (důležité je aby kurzor myši odpovídal rovině předchozí záložky - objeví se charakteristická oranžová přerušovaná čára) a tím mohu vytvořit novou záložku už s vlastnostmi prvku

JPanel, nebo jinými, záleží na vloženém prvku, který byl zvolen (viz. Obr. 3.3). Takto lze dále pokračovat ve vytváření záložek na stejné úrovni. Pokud ale potřebuji záložku o úroveň nižší, vloží další kontejner TabbedPane (viz. Obr. 3.4)



Obr.3.4: Vložení záložky nižší úrovně(1) [autor]



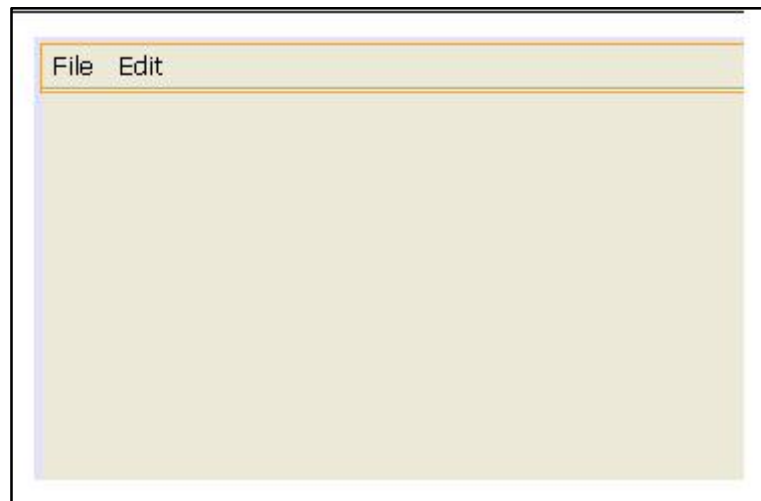
Obr.3.5: – Vložení záložky nižší úrovně(2) [autor]

A pak do prostoru záložky vloží jiný grafický prvek (viz. Obr. 3.5), tím vznikne záložka nižší úrovně.

3.2. Menu Bar

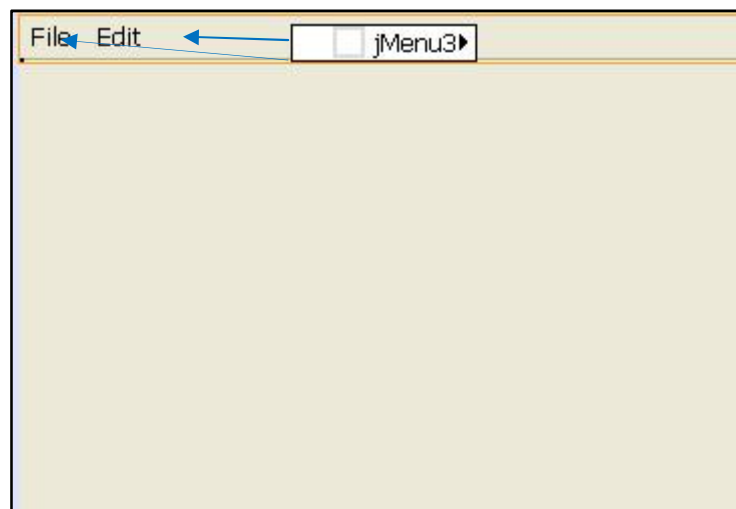
NetBeans GUI Builder umožňuje pomocí nástrojů Menu Bar, Menu, Menu Item, Menu Item / RadioButton, Menu Item / CheckBox rychle a pohodlně vytvořit horní nástrojovou lištu, která je dnes nedílnou součástí téměř všech aplikací a umožnit tím rychlý přístup k základním funkcím a nastavením aplikace.

Po vložení základní komponenty Menu Bar na formulář se v horní části JFrame okna objeví lišta, která bude vypadat takto: (viz. Obr. 3.6)



Obr. 3.6: Vložení MenuBar [autor]

Dále pak už jsou používány pouze položky typu Menu (slouží pouze jako kontejnery pro prvky typu Menu Item, nebo pro další položky typu Menu) a Menu Item.

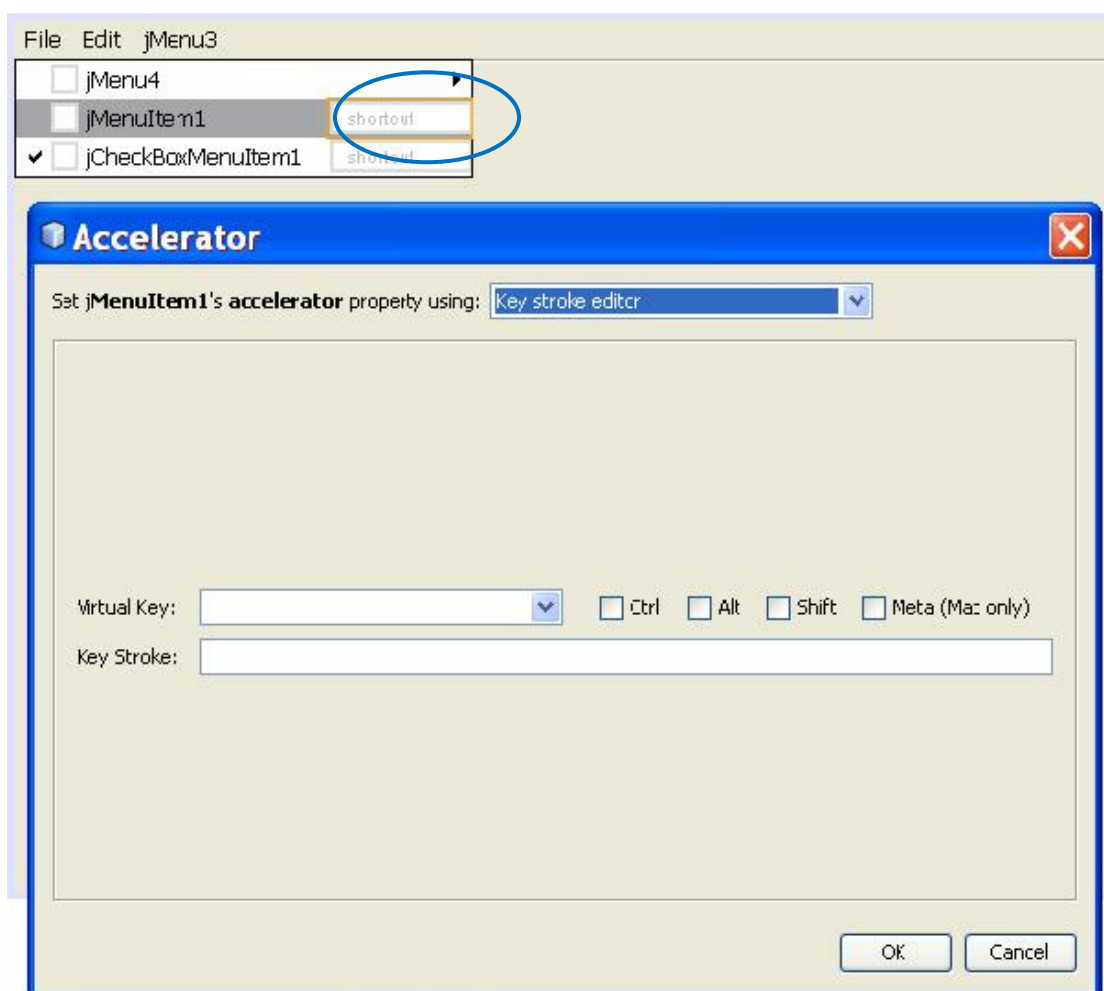


Obr.3.7: Přidání dalších položek do Menu [autor]

Prvek Menu může být přidán buď do horní nástrojové lišty – tím vznikne další prvek v horní liště (viz. Obr. 3.7), nebo může být přidán do již existující položky Menu.

3.2.1. Přidání klávesových zkratk

Klávesové zkratky představují kombinaci několika kláves, po jejichž stisknutí se provede stejná funkce, která je nastavená u dané položky JMenuItem. Klávesové zkratky mohou být použity pro rychlejší práci s aplikací, nebo pokud uživatel pracuje bez myši, klávesové zkratky mu poskytnou přístup k položkám JMenu. Vytvoření klávesové zkratky k položkám typu JMenuItem je snadné, stačí kliknout na nabídku **shortcut** a poté se objeví okno pro nastavení klávesové zkratky (viz. Obr. 3.8).

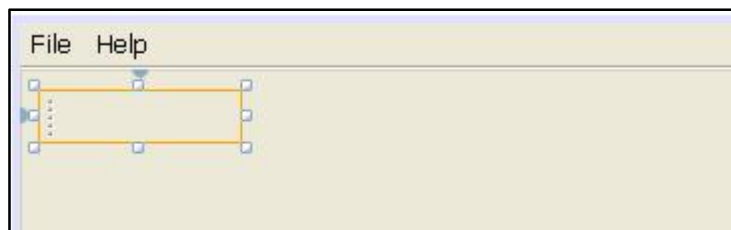


Obr. 3.8: přidání klávesových zkratk k položce JMenuItem [autor]

Práce s tímto nástrojem je poměrně snadná a pohodlná a nepředstavuje pro uživatele velký problém.

3.3. Nástrojová lišta

Nástrojová lišta neboli toolbar se používá pro často používané operace a jejich použití nám má přinést úsporu času. Nástrojová lišta bývá nejčastěji umístěna v horní části okna pod menu, případně na pravé, nebo levé části aplikace. Nástrojová lišta v prostředí NetBeans GUI Builder se nachází ve skupině Swing Containers. Po jejím vložení na formulář vypadá komponenta JToolBar takto (viz. Obr. 3.9):



Obr. 3.9: JToolBar [autor]

Poté už uživatel může do nástrojové lišty vkládat další komponenty, nejčastěji JButton s obrázkem, ale můžou to být i jiné např. JRadioButton, JComboBox a další.

3.3.1. Vlastnosti JToolBaru

Zde jsou uvedeny některé důležité vlastnosti komponenty JToolBar.

- *Floatable*
- *Orientation*
- *toolTipText*

Vlastnost *Floatable* dovoluje při spuštění aplikace přesouvat nástrojovou lištu. Vlastnost *Orientation* rozhoduje, zda budou prvky v JToolBaru řazeny horizontálně, nebo vertikálně. *ToolTipText* označuje popis, který se zobrazí, pokud je kurzor umístěn nad ikonu JToolBar .

4. Textová pole, textové oblasti

Do skupiny vstupní textová pole se řadí jednak jednoduché grafické prvky, které umožňují uživateli vložit krátký, nejčastěji jednořádkový text. Pro vkládání více řádkových textů je lepší použít nástroje pro textové oblasti jako `JTextArea`, `JTextPane`, `JEditorPane`.

4.1. Komponenta `JTextField`

`JTextField` představuje nejjednodušší textové pole, které má ale i tak velmi široké použití. Nově přidáný `JTextField` bude mít délku, jako jeho název (`JTextField1`, `JTextField2`, atd.), podle rozložení, které je nastaveno na formuláři, bude záležet, jestli bude moct být upravována šířka a délka `JTextFieldu`. Většina základních nastavení pro třídu `JTextField` je rychle a pohodlně přístupná přes záložku Vlastnosti – např. barva a font písma, počáteční text barva pozadí atd. Práce se třídou `JTextField` v prostředí NetBeans GUI Builder je rychlejší a pohodlnější než při klasickém psaní zdrojového kódu.

4.2. Komponenta `JPasswordField`

Tato třída je velmi podobná jako předchozí třída `JTextField` (je jejím potomkem). `JTextField` složí k zadávání hesel – místo zadávaných znaků od uživatele se zobrazují tečky. Naprostá většina vlastností je stejná jako u `JTextFieldu`.

4.3. Komponenta `JFormattedTextField`

`JFormattedField` umožňuje na rozdíl od obyčejného `JTextFieldu` filtrovat vkládané znaky od uživatele. `JFormattedField` umožňuje nastavit omezení, které zabrání vkládání nechtěných znaků. Ve vlastnostech `JFormattedTextField` je navíc položka `formattedFactory` a po jejím rozkliknutí se zobrazí okno s možností upravit a nastavit pravidla pro vstupní text (viz. Obr. 4.1), zde jsou uvedeny některé zástupné znaky a jejich význam:

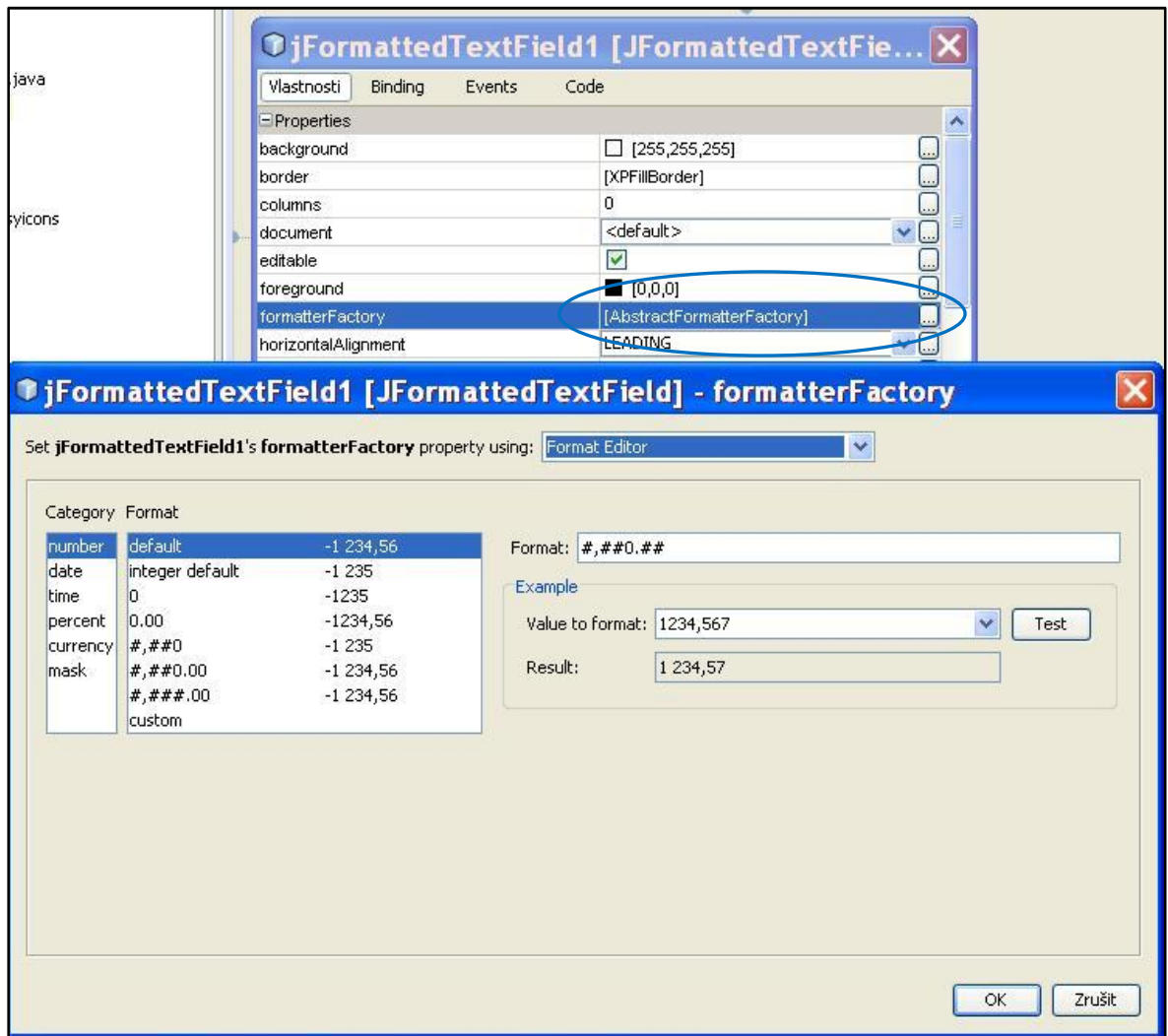
- libovolná číslice

? – libované písmeno

L – libovolné písmeno, velká písmena jsou převedena na malá

U – libovolné písmeno, malá jsou převedena na velká

* -cokoliv



Obr. 4.1: JFormattedTextField [autor]

4.4. Komponenta JTextArea

Jeden ze základních nástrojů pro práci s více řádkovým textem v knihovně Swing. Práce s touto komponentou je v prostředí NetBeans GUI Builderu poměrně snadná a jednoduchá, uživatel především ocení možnost rychlého výběru umístění a změny velikosti komponenty oproti dřívějšímu zdlouhavému psaní zdrojového kódu.

4.5. Komponenta JEditorPane

Swing komponenta, která umožňuje upravovat a měnit styl částí textů, které obsahuje. Další její výhodou je, že JEditorPane podporuje zobrazení ve formátu HTML

4.6. Komponenta JTextPane

Podtřída JEditorPane navíc umožňuje vkládat mezi text obrázky.

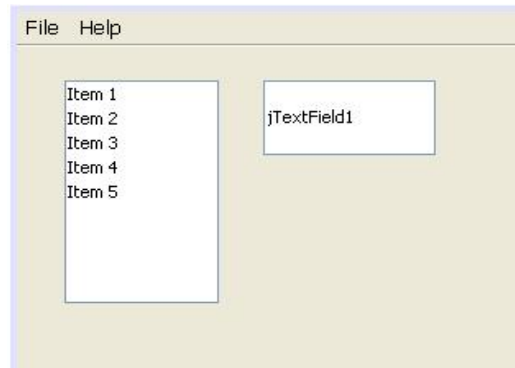
5. Složitější Swing komponenty

5.1. JList komponenta

Tato komponenta umožňuje zobrazit seznam položek typu String a jednu, nebo více jich vybrat. Mezi nejběžnější operace s komponentou JList patří načtení položek do JListu, přidávání a odebírání položek a předání vybrané položky jako parametr k dalšímu použití.

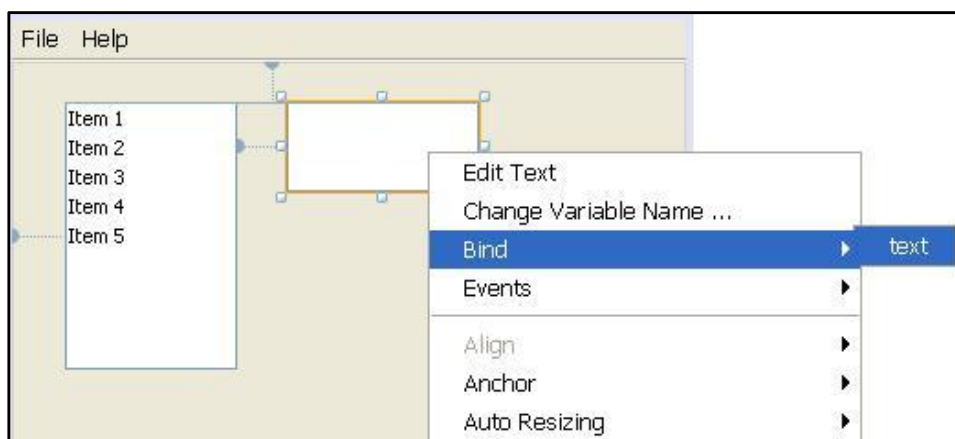
5.1.1. Předání vybrané položky z Jlistu.

Tato operace je řešena v NetBeans GUI Builderu poměrně lehce a je možné jí zvládnout bez psaní kódu, stačí pouze u komponenty, které má být předána položka ze seznamu, použít vlastnost Bind. To je možné ukázat na krátkém a zjednodušeném příkladě. (viz. Obr. 5.1)



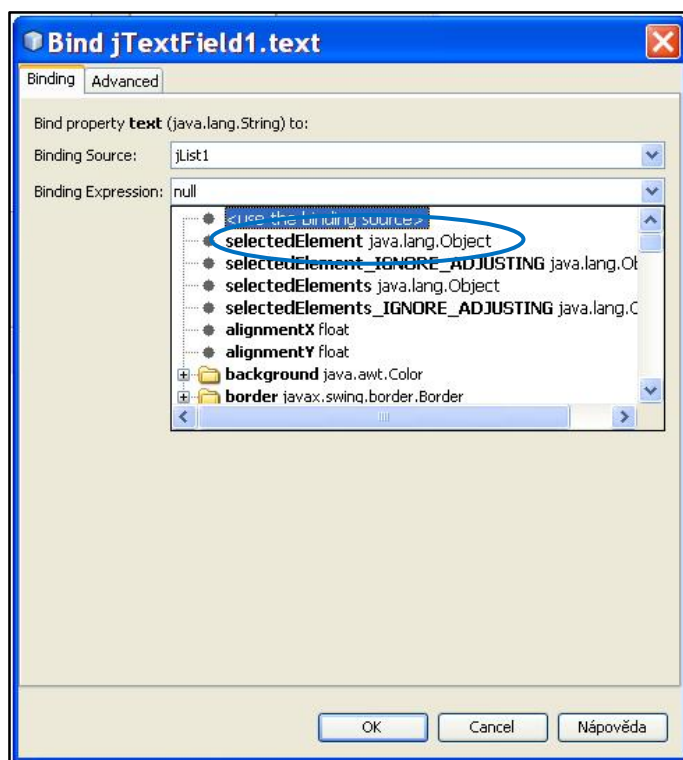
Obr. 5.1: JList- příklad [autor]

Na obrázku vlevo je komponenta JList a vpravo JTextField a je potřeba, aby v textovém poli byla zobrazena aktuálně vybraná položka z JListu. K tomu postačí, pokud u komponenty JTextField uživatel rozklikne možnost Bind -> Text (viz. Obr. 5.2)



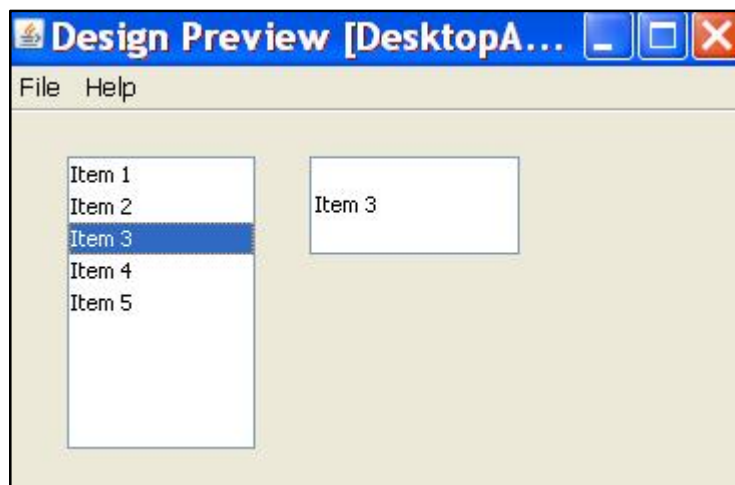
Obr. 5.2: JList – Bind text [autor]

A na závěr už jenom vybere hodnoty *Binding Source* - JList a *Binding Expression* zvolí *selectedElement* (případně *selectedElements*, pokud je ve vlastnostech JListu povolen multiplicitní výběr) (viz. Obr. 5.3).



Obr. 5.3: JList – Bind spojení[autor]

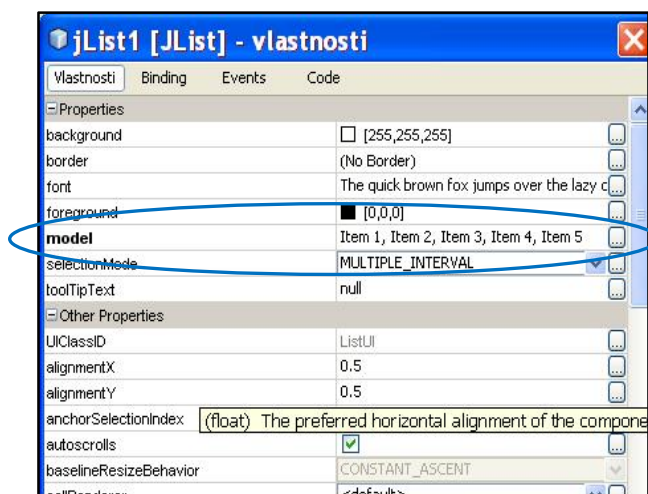
A zde je zobrazen výsledek (viz. Obr. 5.4):



Obr. 5.4: JList- Bind výsledek [autor]

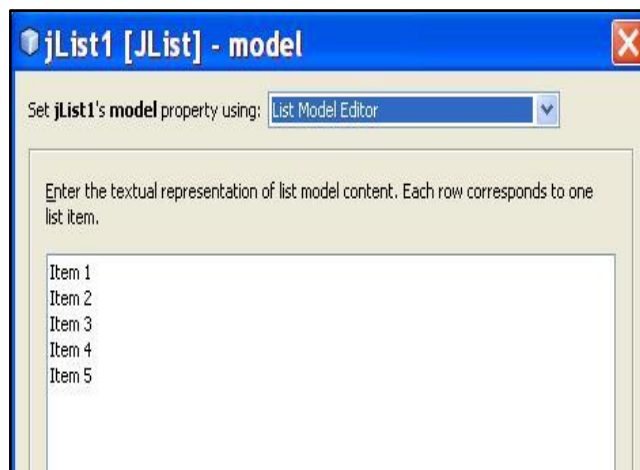
5.1.2. Přidání položek do JListu

Přidání, respektive nastavení položek, které mají být vypsané v JListu je také poměrně snadné, pokud už při vytváření aplikace uživatel zná úplný seznam položek a ty se při běhu aplikace nebudou měnit. Stačí pouze potřebné položky vypsat ve vlastnostech JListu. Po rozkliknutí záložky Vlastnosti uživatel vybere položku model (viz. Obr. 5.5).



Obr.5.5: – JList – model [autor]

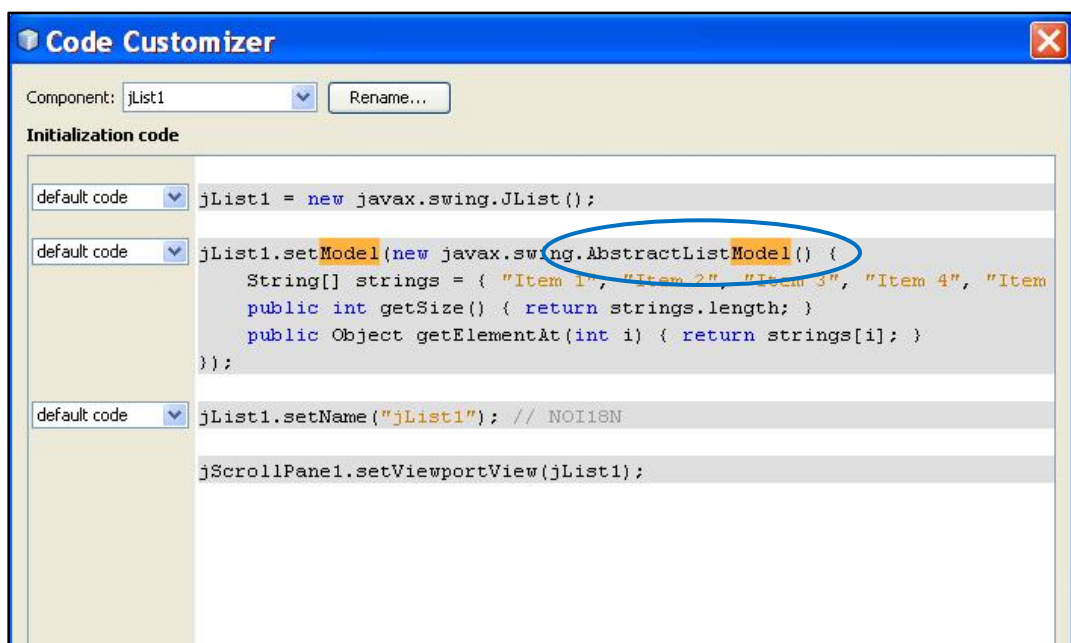
A v *model property* vybere nastavení List Model Editor, poté už jenom zadá potřebné hodnoty, které má JList obsahovat (viz. Obr. 5.6).



Obr. 5.6: JList –ListModel [autor]

Komplikovanější situace nastává, pokud uživatel chce položky JListu upravovat za běhu aplikace. Komponenta JList musí mít při vytvoření nějaký model a NetBeans GUI Builder standardně vytváří model, který umožňuje zobrazovat pevně daný seznam Stringů, který může být měněn pouze v době návrhu. To je způsobeno tím, že knihovna Swing, do které patří komponenta JList, je založena na modelu oddělení zobrazení komponenty, od jejího obsahu a řízení aplikace – koncept MVC.

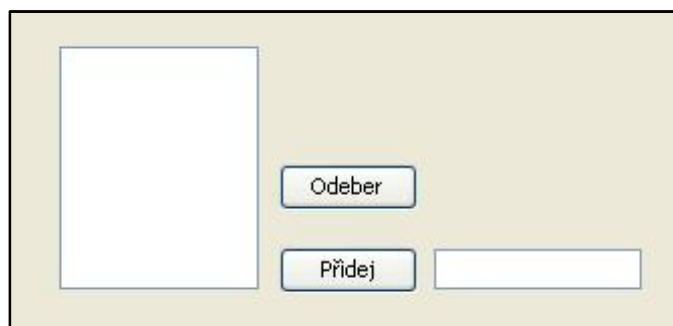
Pokud uživatel umístí komponentu JList na formulář, vypadá vygenerovaný kód takto: (viz. Obr. 3.5)



Obr.5.7: JList- defaultně generovaný kód[autor]

Tady je vidět, že nově vytvořený JList má nastavený model na *AbstractListModel* s pevně přiřazeným polem hodnot. Pokud bude potřeba položky v JListu měnit, má uživatel na výběr dvě možnosti – buď použít jiný model - *DefaultListModel*, nebo vytvořit vlastní třídu, která bude implementovat rozhraní *ListModel*.

Na následujícím příkladě je předvedena možnost použití *DefaultListModelu*. Na formuláři jsou pouze JList, textové pole pro nově přidávané položky JListu a dvě tlačítka na přidávání/odebírání položek (viz. Obr. 5.8).



Obr. 5.8: JList –ukázková propojení [autor]

První věc, kterou je potřeba udělat je, že si ve zdrojovém kódu hlavního formuláře uživatel vytvoří novou proměnou typu *DefaultListModel*(viz. Obr. 5.9).

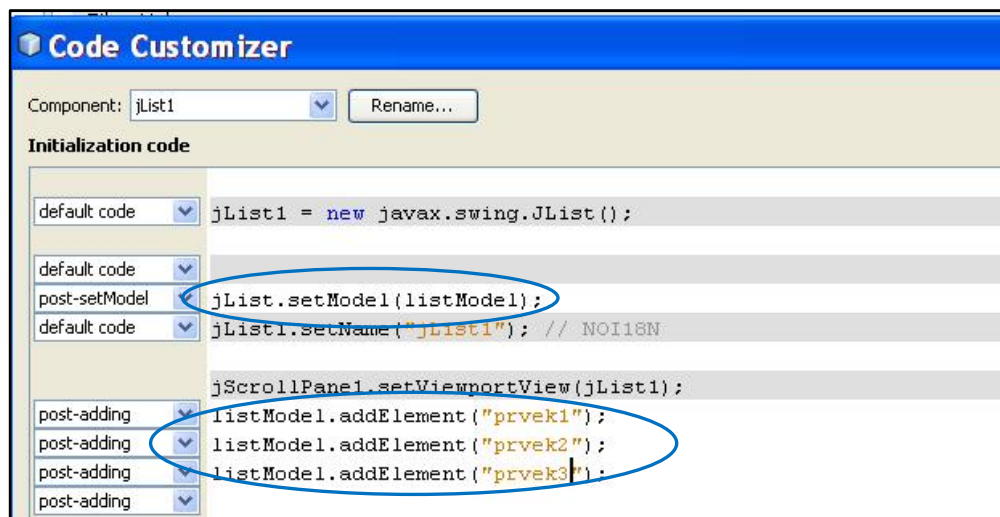
```

24 public class DesktopApplication3View extends JFrameView {
25     private DefaultListModel listModel ;
26
27
28     public DesktopApplication3View(SingleFrameApplication app) {
29         super(app);
30         listModel = new DefaultListModel();
31
32         initComponents();
33     }

```

Obr.5.9: JList – definování nové proměné [autor]

Dále je potřeba změnit nastavení u komponenty JList, ve vlastnostech komponenty na záložce *Vlastnosti* -> *model* -> změnit hodnotu na *Custom Code*. Touto změnou uživatel může upravovat dříve chráněný kód. Poté stačí už jenom nastavit model pomocí funkce *.setModel* a jako parametr předat *DefaultListModel*, který byl vytvořen v hlavním okně aplikace, případně mohou být přidány do JListu i nějaké prvky, pomocí funkce *add* nebo *addElement*, které mají být v JListu při spuštění aplikace(viz. Obr. 5.10).

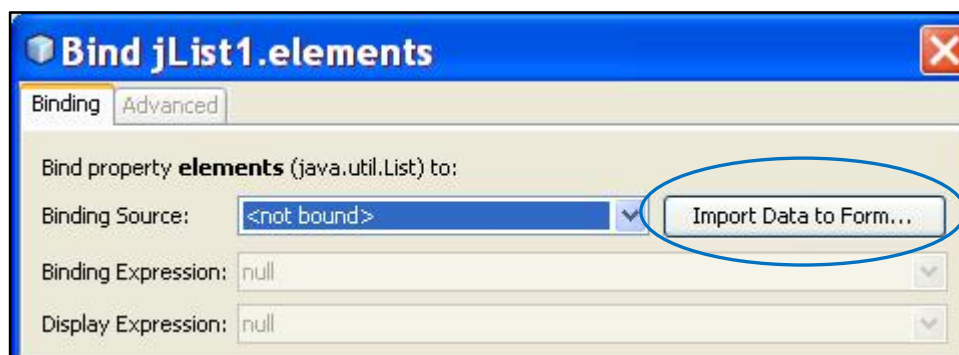


Obr. 5.10: JList – změna kódu [autor]

Zbývá už pouze u obou tlačítek nastavit ve vlastnostech na záložce Events -> actionPerformed kód, který bude text z JTextField přidávat, nebo odebírat prvky z ListModelu. Funkčnost této ukázky bude možné ověřit až při spuštění aplikace, při zobrazení formuláře pomocí Preview Design nebude fungovat

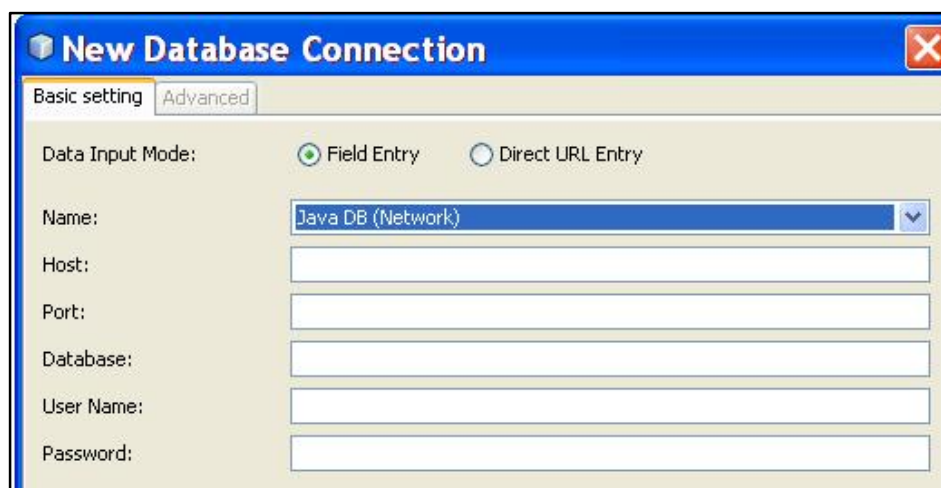
5.1.3. Import položek do JListu z databáze

NetBeans GUI Builder nabízí průvodce, pomocí kterého uživatel může rychle a jednoduše načíst potřebná data z tabulky databáze do komponenty JList. Ve vlastnostech JListu na záložce *Binding* -> rozklikne možnost *elements* a poté se zobrazí tato nabídka (viz. Obr. 5.11)



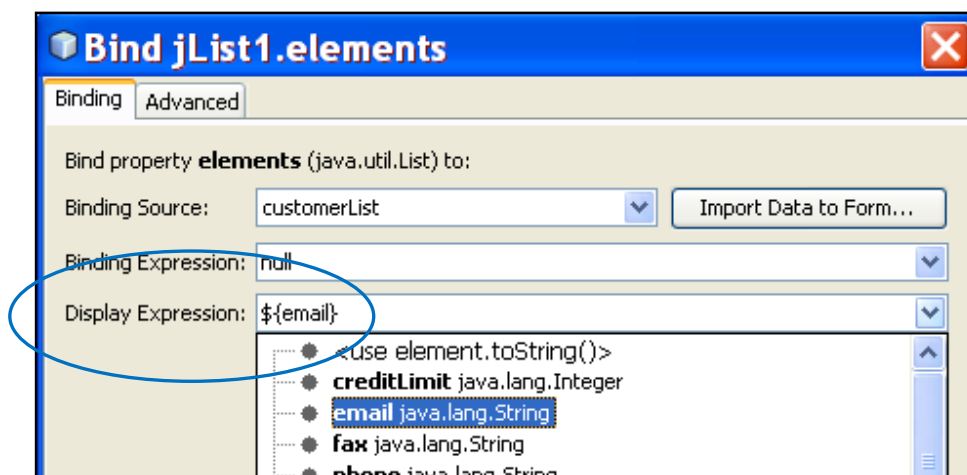
Obr. 5.11: JList načtení seznamu z databáze [autor]

Po zvolení možnosti *Import Data Form..* mohou být vybrány varianty buď již existující databázové spojení, nebo možnost vytvořit nové. Při vytváření nového databázového připojení bude uživatel zadávat klasické údaje (viz. Obr. 5.12)



Obr.5.12: JList – nové připojení do databáze [autor]

Při výběru již existujícího databázového spojení už je jenom potřeba zvolit tabulku, ze které se mají získat údaje a nakonec ještě v nastavení hodnoty *Display Expression* vybrat sloupec, který se má v JListu zobrazit (viz. Obr. 5.13).



Obr. 5.13: JList – výběr databázového sloupce

5.1.4. JList – některé důležité vlastnosti

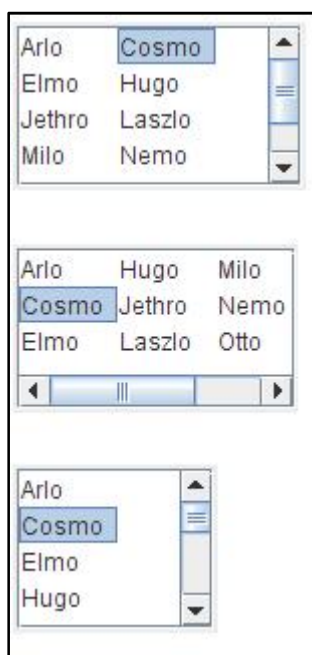
Mezi nejdůležitější vlastnosti komponenty JList patří, **selection mode**.



Komponenta JList nabízí tři režimy pro výběr svých položek. V režimu `SINGLE_SELECTION`, může být vybrána pouze jedna položka. V dalším režimu `SINGLE_INTERVAL_SELECTION` může uživatel vybrat více položek, pokud označí jednu, pomocí klávesy `Shif` může označit i některé položky nad, nebo pod ní. Poslední režim, který může být nastaven u komponenty JList je `MULTIPLE_INTERVAL_SELECTION` jedná se o defaultní nastavení, pomocí kláves `Shift` a `Ctrl` jsou možné všechny kombinace výběru. Ukázky nastavení lze vidět (viz. Obr. 5.14).

Obr. 5.14: JList selection mode [autor]

Další důležitou vlastností komponenty JList je **LayoutOrientation mode**

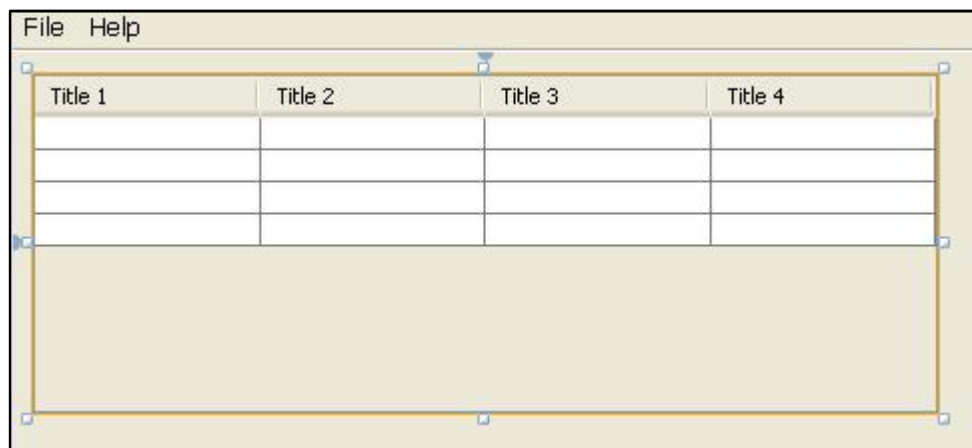


Jde o nastavení, které zajišťuje řazení prvků. První možnost `HORIZONTAL_WRAP` zajistí, že JList zobrazuje své položky zleva doprava, po dosažení konce šířky JListu se další položky umístí na nový řádek. Možnost `VERTICAL_WRAP` vykresluje položky JListu od shora dolů, když bude dosaženo konce JListu, vytvoří se nový sloupec. Poslední možnost `VERTICAL`. Jde o výchozí nastavení, ve kterém jsou položky zobrazeny pouze v jednom sloupci od shora dolů. Ukázky nastavení lze vidět (viz. Obr. 5.14).

Obr. 5.15: orientace položek v JListu (převzato z [13])

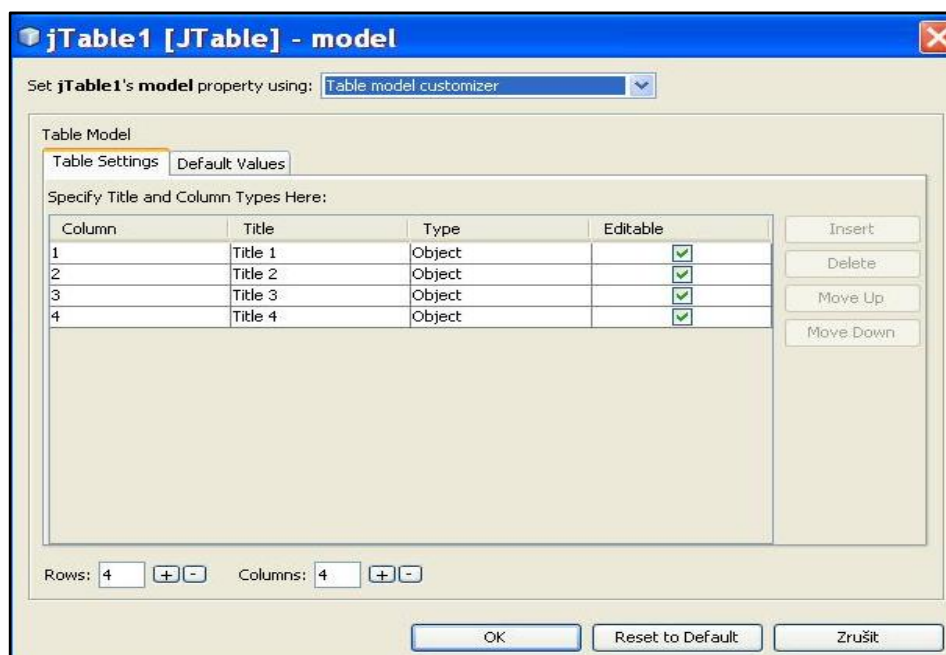
5.2. JTable

JTable je třída, která se používá pro zobrazení a editaci dvourozměrné tabulky. Podobně jako u třídy JList, je u třídy JTable odděleno zobrazení komponenty JTable a jejího obsahu TableModel. Třída JTable je velmi rozsáhlá a poskytuje širokou škálu vlastností a nastavení. Na následujících stránkách bych chtěl shrnout jaká zjednodušení a výhody přináší práce se třídou JTable v prostředí NetBeans GUI Builder. Po vložení tabulky na formulář se objeví takováto tabulka (viz. Obr. 5.16)



Obr. 5.16: JTable po vytvoření [autor]

Po rozkliknutí záložky *Vlastnosti* -> a možnosti *model*, se zobrazí základní výběr pro nastavení sloupců tabulky (viz. Obr. 5.17)



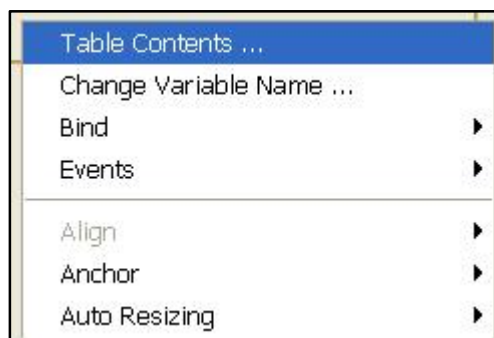
Obr. 5.17: JTable – Table model [autor]

Uživatel má na výběr:

- měnit počet sloupců v tabulce,
- název sloupce, měnit jejich pořadí
- datový typ, který bude ve sloupci obsažen (Object, String, Boolean, Integer, Byte, Short, Long, Double, Float) a
- zda bude daný sloupec editovatelný.

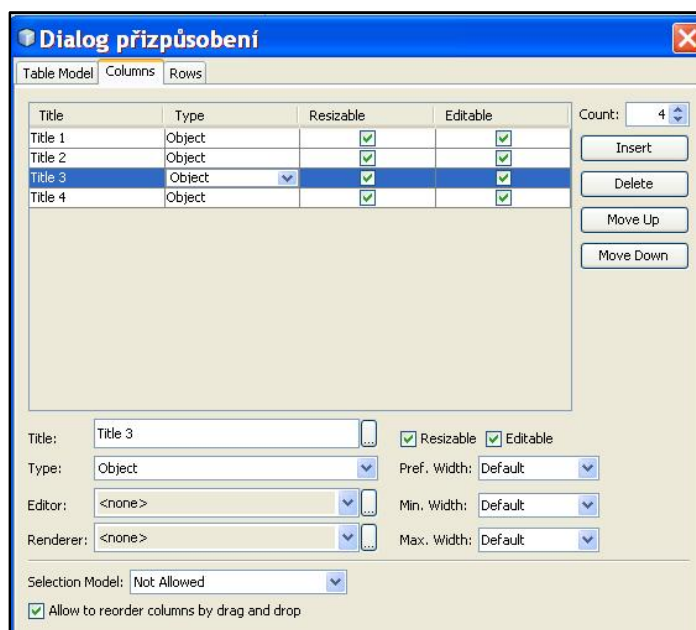
To vše může provádět, pokud se pohybuje v nastavení modelu *Table model customizer* (defaultně nastavený), v tomto nastavení (viz. Obr. 5.17) vytvoří vlastní *TableModel*, který pak bude předán třídě *JTable* k zobrazení, další možnosti nastavení *JTable* modelu jsou: *Value from existing component* – získání modelu z již existující tabulky, nebo možnost *Custom Code* – zadání modelu přímo, pomocí kódu.

NetBeans GUI Builder nabízí další nastavení vlastností tabulky v možnosti *Table Contents* (viz. Obr. 5.18)



Obr. 5.18: *JTable* – nabídka [autor]

Po jejím rozkliknutí a výběru možnosti *User Specified* na záložce *Table Model* se uživatel dostane do této nabídky (viz. Obr. 5.19)



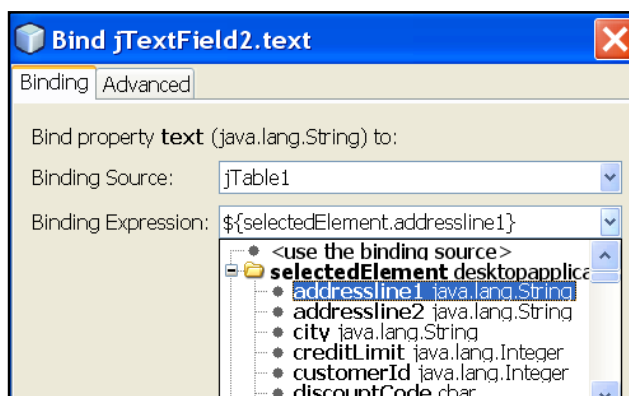
Obr.5.19: JTable – Table Contents [autor]

V této nabídce může uživatel nastavit minimální, maximální a preferovanou šířku sloupců tabulky a také, zda bude možné při běhu aplikace měnit šířku sloupců.

5.2.1. Předání dat z tabulky dalším třídám

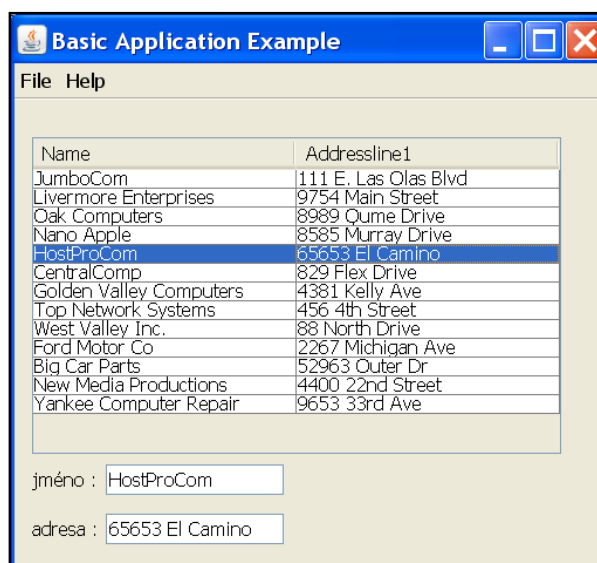
Pokud uživatel chce, aby vybraná data z tabulky byly předávány dalším třídám, je řešení podobné jako u třídy JList. Řešení tohoto problému lze nejlépe ukázat na krátkém příkladě: tabulka, která obsahuje 2 sloupce a je potřeba zobrazit vybrané hodnoty do JTextFieldu. Vytvořím si tabulku se dvěma sloupci (zdroj dat je buď z databáze, nebo zadané údaje), selection mode – single selection viz kapitola 5.2.5.

Přidám na formulář JTextField a nastavím u něj *Vlastnosti -> Binding -> Text* (viz. Obr. 5.20)



Ob. 5.20.: Binding výběr pro JTextField

A zde je zobrazen výsledek propojení (viz. Obr. 5.21):



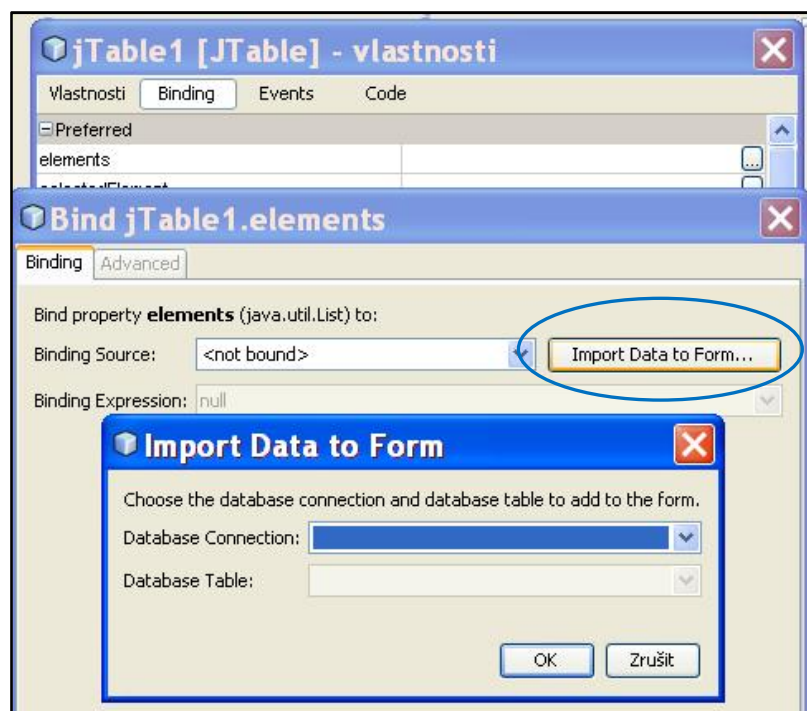
Obr. 5.21: Propojení JTable a JTextField

5.2.2. Používání třídy JTable bez propojení s databází

Pokud uživatel zamýšlí používat třídu JTable ve své aplikaci k něčemu náročnějšímu než pouze statické zobrazení dat, například přidávání a mazání údajů a zdroj dat tabulky nebude načten z databáze, tak už NetBeans GUI Builder nenabízí žádné nástroje ke zjednodušení. Uživatel bude muset vytvořit vlastní třídu, která je potomkem abstraktní třídy *javx.swing.AbstractTableModel* a poté nechat databázi zobrazit. Tento postup je zmíněn např. ve zdrojích [1], [2].

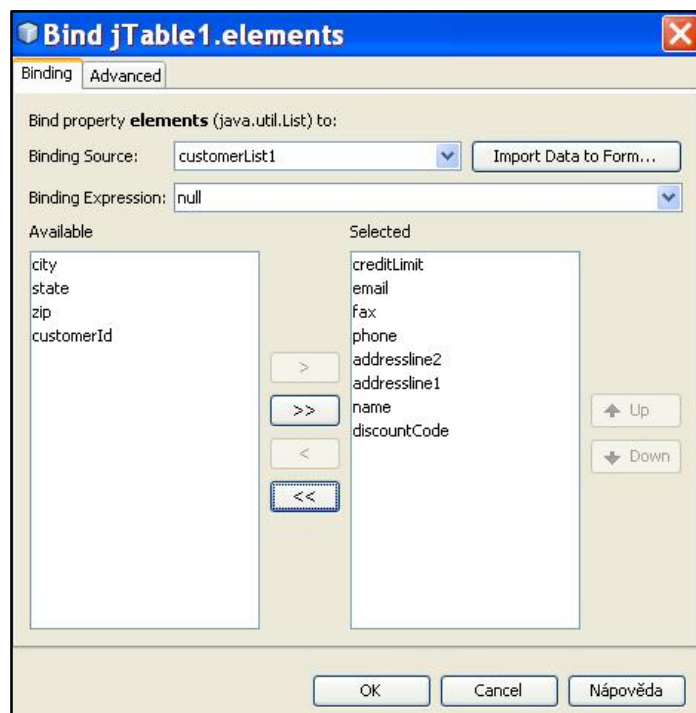
5.2.3. Propojení třídy JTable s databází

Zobrazení databáze pomocí tabulky patří mezi nejčastější použití třídy JTable. V NetBeans GUI Builderu je toto propojení snadné a lze ho zvládnout pomocí jednoduchého průvodce. Ve vlastnostech tabulky vybrat možnost *Bindings -> elements* a možnost *Import Data to Form*



Obr. 5.22: jTable – načtení dat z databáze [autor]

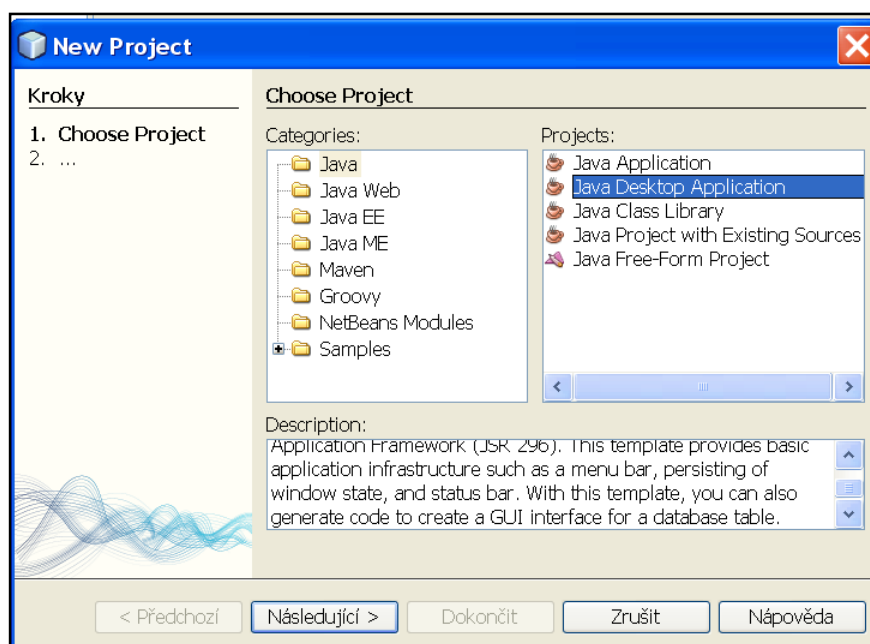
Poté uživatel zvolí, jestli se bude jednat o nové databázové spojení, nebo spojení již vytvořené. (viz. Obr. 5.22) Dále je potřeba zvolit ze které tabulky databáze se mají sloupce tabulky vybírat. A nakonec už jen může vybrat, jestli chce zobrazit celou tabulku, nebo jen některé její sloupce. (viz. Obr. 5.23).



Obr. 5.23: jTable – výběr sloupců podle databáze [autor]

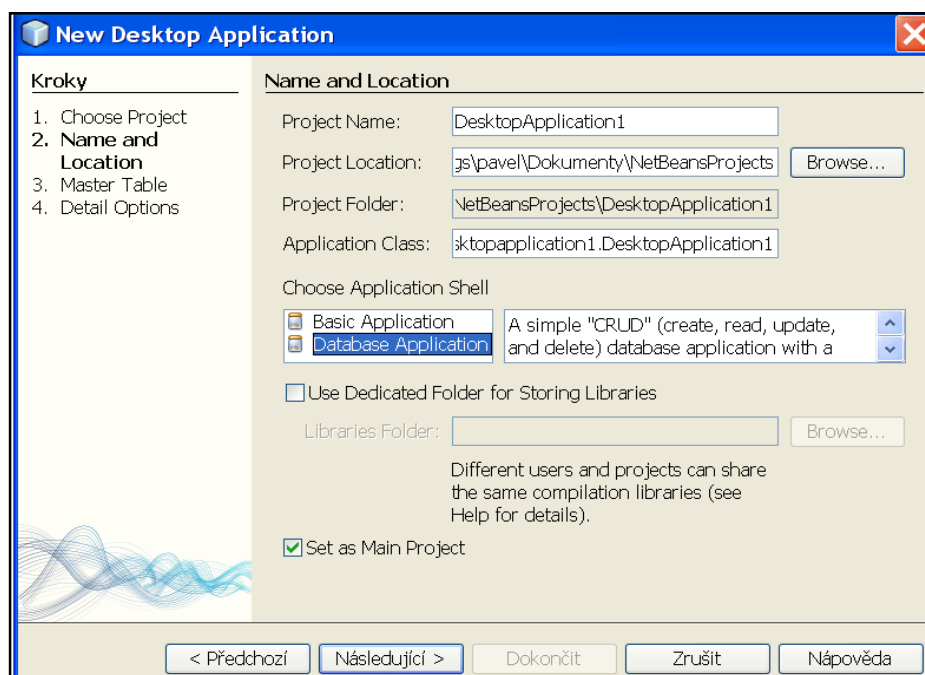
5.2.4. Propojení tabulky s databází a úprava záznamů v databázi

Řešení tohoto problému, kdy uživatel, má tabulku, která je připojena k databázi a je potřeba upravovat, přidávat data není v nějakém nastavení vlastnosti třídy JTable, ale už při vytváření projektu. NetBeans poskytuje zvláštní projekt pro aplikace propojené s databází. Po výběru *Nový projekt -> Java- > Java Desktop Application* (viz. Obr. 5.24)

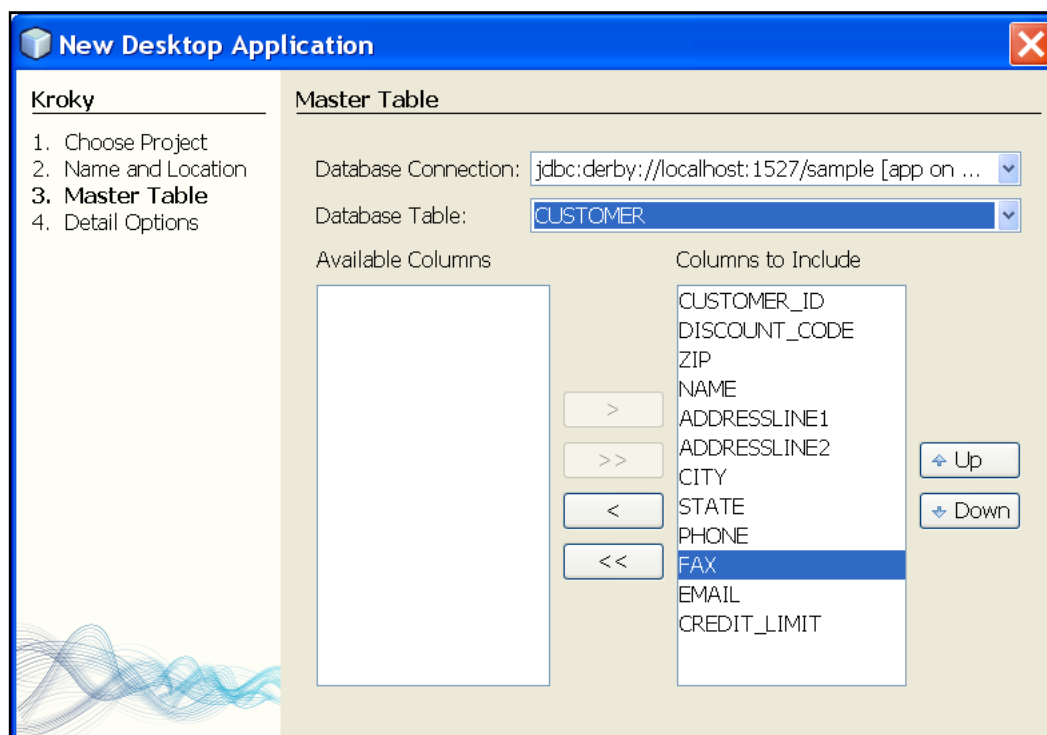


Obr. 5.24.: Vytvoření nového projektu

Na další záložce vybrat možnost *Database Application* (viz Obr. 5.25)



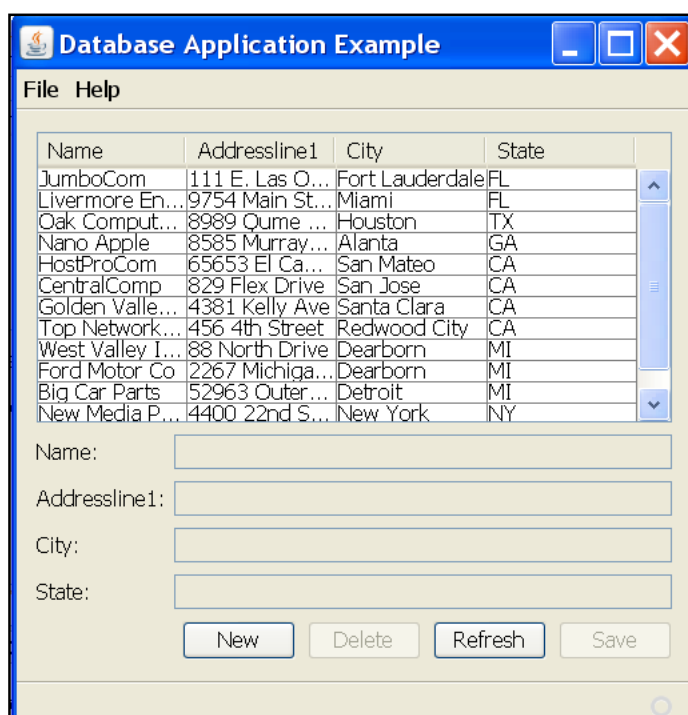
Obr. 5.25: Vytvoření projektu spojeného s databází



Obr. 5.26: Výběr databáze a zobrazovaných sloupečků

poté už jen zvolit tabulku a sloupce, které budou zobrazeny v tabulce (viz Obr. 5.26).

Takto vypadá formulář vytvořený vývojovým prostředím NetBeans pro editaci dat obsažených v tabulce.



Obr. 5.27: automaticky vytvoření formulář po dokončení průvodce nového projektu

5.2.5. Selection model u třídy JTable

Podobně jako u komponenty JList jsou u JTable k dispozici tři možnosti výběrů:

- *Multiple Interval Selection*
- *Single Interval Selection*
- *Single Selection*

První možnost *Multiple Interval Selection* umožňuje při práci s tabulkou jakoukoliv kombinaci výběru – lze vybrat libovolný počet řádků a sloupců. Při druhé možnosti *Single Interval Selection* mohou být vybrány pouze řádky, které jsou vedle sebe. V poslední možnosti *Single Selection* může být vybrán pouze jeden řádek tabulky.

Oproti JListu má třída JTable ještě navíc vlastnost *ColumnSelectionAllowed*, která umožňuje specifikovat výběr na úrovni jednotlivých buněk v tabulce.

6. Java Persistence API, NetBeans IDE 6.9

V další části své bakalářské práce bych se chtěl věnovat nástrojům, které se starají o spolupráci grafických objektů a Java Persistence API (JPA), ve druhé části bych krátce představil novinky ve verzi NetBeans IDE 6.9, které se týkají nástroje GUI Builder.

6.1. Java Persistence API

Java Persistence API (JPA) - framework umožňující objektově relační mapování ORM. Mezi nejznámější implementace JPA patří Hibernate, TopLink atd. . Jde hlavně o tyto tři prvky:

Entity Manager – Rozhraní používané k interakci s persistence kontextem. Entity Manager slouží ke spravování entit. Za entitu může být považován objekt, který reprezentuje data v databázi. Typicky entitní třída reprezentuje tabulku v relační databázi a každá instance této třídy pak koresponduje k jedné řádce tabulky. Základní operace entity manageru s entitou jsou persist (insert), remove (delete), merge (update), find (select). Entity manager pracuje s dotazy podobné SQL dotazům, používá však Java Persistence Query Language, který je nezávislý na technologii databáze.

Query – rozhraní, které slouží k ovládání dotazu. Vznikne po zavolání metody createQuery na Entity Manageru s parametrem String ve formátu Java Persistence query language

Query Result – slouží pro interpretaci výsledku dotazu Query. Představuje třídu typu List, kterou vrátí rozhraní Query po zavolání metody getResultList.

Všechny tyto tři komponenty patří do skupiny neviditelných (nevizuálních) komponent, takže nebudou vidět na hlavním formuláři, ale pouze v levém dolním okně *Inspector*. Pokud uživatel pracuje se třídami JList, JComboBox, nebo JTable a jako zdroj dat použije tabulku z databáze, automaticky se vytvoří prvky Entity Manager, Query

a Query Result, které reprezentují daný databázový dotaz a jeho interpretaci v GUI. Samostatné využití komponent Java Persistence, bez přímého spojení s třídami JList, JComboBox a JTable může mít v NetBeans GUI Builderu využití dotazu do tabulky ve většině případů je ale postačující spojení s databází provést přes nějakou třídu a její vlastnost, jako např. JTable nebo JList.

6.2. NetBeans IDE 6.9

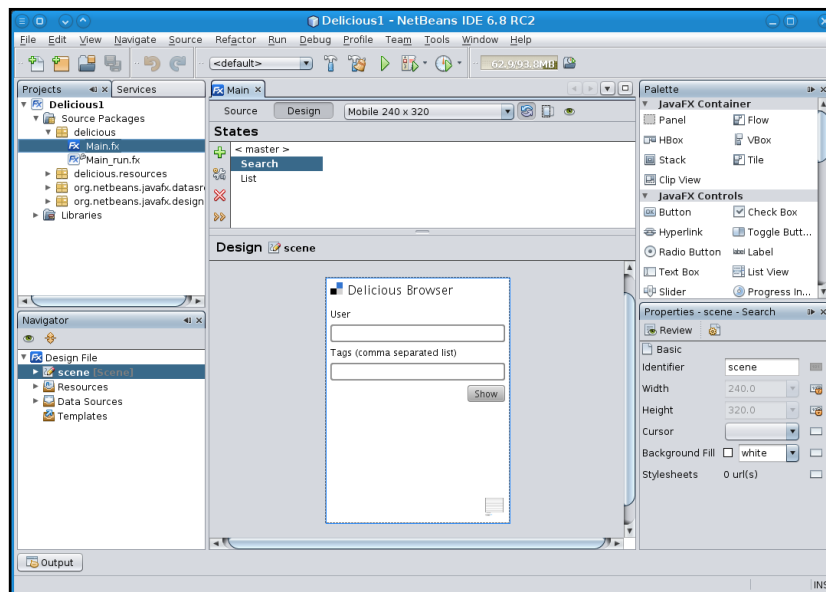
Novinky obsažené v poslední dostupné verzi vývojového prostředí NetBeans, které bylo uvolněno ke dni 15.6.2010, a týkají se vytváření GUI pomocí nástroje NetBeans GUI Builder. Jednou z hlavních novinek je představení nástroje Java FX Composer. Tento nástroj pro práci s platformou Java FX, který byl dříve dostupný pro prostředí NetBeans pouze jako plug-in je nyní plně integrován do vývojového prostředí NetBeans. Jiné novinky týkající se vytváření GUI v nové verzi NetBeans nejsou.

Java FX

Představuje softwarovou platformu z dílny společnosti Sun Microsystems, která je postavená na bázi Javy. Java FX slouží k vývoji RIA aplikací (Rich Internet Applications) – jedná se hlavně o zajištění interaktivity u webových stránek. Java FX webové aplikace se formou plug-inu spouští z prohlížeče. Projekt JavaFX vznikl jako odpověď na rozšíření platforem jako jsou Adobe Flash, nebo Microsoft Silverlight. Java FX je nezávislá na prostředí, lze ji spouštět kdekoli, kde běží JRE, včetně mobilních telefonů.

Java FX Composer

Java FX Composer je vizuální nástroj pro rychlou tvorbu JavaFX GUI aplikací, který je velmi podobný NetBeans GUI builderu. Zde je ukázka, jak vypadá vývojové prostředí FX Composer.



Obr. 8.1 převzato z [11]

7. Závěr

Tato bakalářská práce přináší čtenáři přehled o možnostech, které mu nabízí nástroj NetBeans GUI Builder a základní informace, které mu pomohou při vytváření grafického uživatelského rozhraní, pokud k tomu použije nástroj NetBeans GUI Builder. Věnuje se jednodušším třídám, se kterým je práce v NetBeans GUI Builderu poměrně snadná a intuitivní, tak i složitějším Swing třídám jako JList a JTable, které i přes kvality nástroje NetBeans GUI Builder budou muset být pravděpodobně upravovány vytvářením zdrojového kódu. Většina

Společně s pracemi [3], nebo [4] poskytuje čtenáři komplexní náhled na nástroj NetBeans GUI Builder, po přečtení se bude orientovat ve výhodách, ale i omezeních, které ho mohou potkat při vytváření GUI potkat. Tato práce může být přínosná studentům kurzů 4IT115 a 4IT353, které mají v náplni práci s grafickým uživatelským prostředím v jazyku Java.

8. Seznam literatury a zdrojů:

- [1] BRUCE, Eckel; *Myslíme v jazyku Java knihovna zkušeného programátora*. Praha: Grada, 2001. ISBN 80-247-0027-1
- [2] BUCHALCEVOVÁ, Alena; PAVLÍČKOVÁ, Jarmila; PAVLÍČEK, Luboš. *Základy softwarového inženýrství materiály ke cvičení*. Praha: Oeconomica, 2007. ISBN 978-80-245-1270-9
- [3] BUCHALCEVOVÁ, Alena; PITKA Lukáš. *Vývojové prostředí NetBeans*. Praha: Oeconomica, 2007. ISBN 978-80-245-1206-8
- [4] PUSIANKOVA, Tatsiana. *NetBeans GUI Builder*. Bakalářská práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky. 2010
- [5] *Eclipse Plug-in Central* [online] dostupný z URL:
<<http://www.eclipseplugincentral.com/>>
- [6] *Java™ Platform, Standard Edition 7 API Documentation* [online] dostupný z URL:
<<http://java.sun.com/javase/7/docs/api/>>
- [7] *Javabyexample.wisdomplug.com: select your ide NetBeans vs Eclipse* [online] dostupný z URL: <http://javabyexample.wisdomplug.com/component/content/article/61-select-your-ide-netbeans-vs-eclipse.html>
- [8] *Learning Swing with the NetBeans IDE* [online] dostupný z URL:
<<http://java.sun.com/docs/books/tutorial/uiswing/learn/index.html>>
- [9] *NetBeans Releases* [online] dostupný z URL:
<<http://netbeans.org/community/releases/index.html>>
- [10] *NetBeans Wiki: BestPracticesWithJPAAndBeansBinding* [online] dostupný z URL:
<<http://wiki.netbeans.org/BestPracticesWithJPAAndBeansBinding>>

- [11] *NetBeans Wiki: JavaFXComposer* [online] dostupný z URL:
<<http://wiki.netbeans.org/JavaFXComposer>>
- [12] *The Java Tutorials: A Visual Guide to Layout Managers* [online] dostupný z URL:
<<http://java.sun.com/docs/books/tutorial/uiswing/layout/visual.html>>
- [13] *The Java Tutorials: How to Use Lists* [online] dostupný z URL:
<<http://java.sun.com/docs/books/tutorial/uiswing/components/list.html>>
- [14] *Tutorials Getting Started With the NetBeans IDE Tutorial* [online] dostupný z URL:
<http://java.sun.com/developer/onlineTraining/tools/netbeans_part1/>
- [15] *Wikipedia : Comparison of integrated development environments* [online]
dostupný z URL:
<http://en.wikipedia.org/wiki/Comparison_of_integrated_development_environments>

9. Terminologický slovník:

Termín	Anglický originál	Popis [zdroj]
Grafické uživatelské rozhraní	GUI – graphical user interface	Uživatelské rozhraní, které slouží k ovládání aplikace pomocí interaktivních grafických prvků [http://en.wikipedia.org/wiki/Graphical_user_interface]
GUI editor	GUI editor	Označován také jako GUI Designer – nástroj používaný při vývoji softwaru, který má zjednodušit vytváření GUI. Programátor už nemusí psát zdrojový kód GUI, ale ten je automaticky vytvořen při používání předem připravených komponent. Bývá součástí některých IDE . [http://en.wikipedia.org/wiki/Graphical_user_interface_builder]
JRE	JRE – Java Runtime Environment	Základní instalace Javy, která umožňuje spouštět Java aplikace. [autor]
MVC	MVC – Model View Controller	je návrhový vzor, který rozděluje datový model aplikace, uživatelského rozhraní a řídicí logiku do tří nezávislých komponent tak, že modifikace některé z nich má minimální vliv na ostatní [autor]
plug-in	plug-in	Software, který je vyvíjen jako součást, nebo rozšíření pro nějakou jinou aplikaci, plug-in sám o sobě není funkční [autor]
Rozložení, rozvržení	Layout	Grafické uspořádání tiskové nebo elektronické stránky, případně i jiné plochy. [http://cs.wikipedia.org/wiki/Layout]

Správce rozložení	Layout Manager	Layout manager je objekt, který určuje velikost a umístění komponent v kontejneru. [http://java.sun.com/docs/books/tutorial/uiswing/layout/using.html]
vývojové prostředí	IDE—integrated development environment	Software sloužící k vytváření programů, většinou se jedná o komplexní balík, který obsahuje editor zdrojového kódu, kompilátor, nástroje pro testování atd. [http://cs.wikipedia.org/wiki/V%C3%BDvojov%C3%A9_prost%C5%99ed%C3%AD]