

**Jihočeská univerzita v Českých Budějovicích  
Ekonomická fakulta**

**BAKALÁŘSKÁ PRÁCE**

**Vypracoval: Michael Gyepes**



# **Zadání diplomové práce**

Analýza potřeb domácnosti a tvorba programu pro domácí finance.

The Analysis of the Household Needs and the Creation of the Programme  
of Household Finance.

## Anotace

Úkolem bakalářské práce je analýza potřeb domácnosti a tvorba programu pro domácí finance. První část práce je zaměřena na průzkum trhu v oblasti informačních systémů. Jsou zde prozkoumány vlastnosti dostupných programů, možnosti jejich využití a je provedeno vyhodnocení. Dále je provedena analýza potřeb domácnosti. Na základě získaných dat je vytvořena koncepce, struktura a logika tvořeného programu pro domácí finance. Také je zvolena technologie programu. Následně je vytvořen vlastní program pro domácí finance. Jsou zde popsána některá dílčí řešení. Text je doplněn obrázky, popisy a algoritmy. Závěrem je provedeno porovnání nově vytvořeného programu se stávajícími dostupnými programy a jsou popsány jeho výhody.

The task of the thesis is the analysis of the household needs and developing a programme for households finance. The first part focuses on the market research in the field of information systems. There are analysed the characteristics of available programmes, the possibility of their use and their evaluation. Furthermore, there is an analysis of households needs. On the basis of the data, the concept, structure and logic of the creating programme of household finance is stated. Also, the technology of the programme is chosen. Subsequently, the own programme of household finance is developed. There are described some partial solutions as well. The text is supplemented with pictures, descriptions, and algorithms. In conclusion, there is a comparison of the newly created programme with available existing programmes and its advantages are described.

# PROHLÁŠENÍ

Prohlašuji, že jsem bakalářskou práci na téma Analýza potřeb domácnost a tvorba programu pro domácí finance vypracoval samostatně na základě vlastních zjištění a materiálů, které uvádím v seznamu použité literatury.

V Českých Budějovicích dne 27. 4. 2011

Michael Gyepes



# Obsah

1. Úvod.....	6
2. Přehled situace na trhu.....	6
Přehled dostupných aplikací: .....	6
2.1 Profesionální ekonomicko-informační software K2 .....	7
Komu je určen.....	8
Vyhodnocení.....	8
2.2 Program Domácí účetnictví firmy KASTNER software s.r.o. ....	9
Modul správa .....	12
Vyhodnocení.....	12
2.3. Účetní program Money S3.....	13
Vyhodnocení.....	14
3. Analýza potřeb domácnosti .....	15
3.1. Diference osobních a firemních financí.....	16
4. Praktická část .....	17
4.1. Výběr databáze .....	17
4.2. Výběr programovacího jazyka.....	17
4.3. Databáze .....	18
Připojení k databázi .....	19
4.4. Hlavní obrazovka programu .....	21
4.5. Moduly .....	22
4.6. Obecná funkčnost .....	23
4.7. Modul Deník.....	24
Popis .....	24
Tabulka .....	24
UML .....	25
Modelový příklad.....	25
Obrázek modulu .....	26
Funkčnost modulu .....	26
Nový záznam (Insert) .....	26
Zobrazení záznamů .....	27



Splněno/Nesplněno .....	28
grdSeznam .....	28
4.8. Modul Doklady přijaté.....	29
Popis .....	29
Tabulka .....	29
UML .....	29
Modelový příklad.....	29
Obrázek modulu .....	30
Funkčnost modulu .....	30
Nový záznam (Insert) .....	30
Součet faktur přijatých a celkový dluh.....	31
Hotovostní platba .....	31
4.9. Modul Doklady vydané .....	33
Popis .....	33
Tabulka .....	33
UML .....	33
Modelový příklad.....	33
Obrázek modulu .....	34
Funkčnost modulu .....	34
Nový záznam (Insert) .....	34
Součet faktur přijatých a celkový dluh.....	35
Hotovostní platba .....	35
4.10. Modul Hotovost .....	37
Popis .....	37
Tabulka .....	37
UML .....	37
Modelový příklad.....	38
Obrázek modulu .....	38
Funkčnost modulu .....	38
Nový záznam (Insert) - výdej.....	38
Nový záznam (Insert) – příjem.....	39

Součty příjmů a výdajů.....	39
4.11.    Modul Banka .....	41
Popis .....	41
Tabulka .....	41
UML .....	41
Modelový příklad.....	42
Obrázek modulu .....	42
Funkčnost modulu .....	43
Nový záznam (Insert) .....	43
Bankovní výdej.....	43
Bankovní příjem .....	44
Platba přijatého ( resp. Vydaného) dokladu .....	44
Obecný výdej .....	45
Nový záznam (Insert) – příjem.....	46
Součty příjmů a výdajů.....	46
4.12.    Modul Provoz vozidel.....	48
Popis .....	48
Tabulka .....	48
UML .....	49
Modelový příklad.....	50
Vozidla.....	50
Jízdy.....	50
Obrázek modulu .....	50
Funkčnost modulu .....	51
Vozidla - nový záznam.....	51
Jízdy - nový záznam.....	51
Přepočet spotřeby .....	51
Kniha dokladů za PHM .....	53
4.13.    Modul Spotřeba .....	54
Popis .....	54
Tabulka .....	54

UML .....	55
Modelový příklad .....	56
Spotřeba .....	56
SpotřebaDetail .....	56
Obrázek modulu .....	56
Funkčnost modulu .....	57
Typ spotřeby – Nový záznam .....	57
Spotřeba detail – Nový záznam .....	57
funkce Přepočítej (btnPrepocitej).....	58
4.14. Modul Pojištění .....	60
Popis .....	60
Tabulka .....	60
UML .....	61
Modelový příklad.....	61
Pojistný předmět.....	61
Pojistné smlouvy .....	61
Obrázek modulu .....	62
Funkčnost modulu .....	62
Tabulka Pojistky – Nový záznam .....	62
Tabulka PojistkyDetail – Nový záznam .....	62
Přepočítej.....	63
4.15. Modul Šablony .....	64
Popis .....	64
Tabulka .....	64
UML .....	65
Modelový příklad.....	65
Obrázek modulu .....	66
Funkčnost modulu .....	66
Kopie záznamu .....	66
4.17. Pomocné číselníky.....	68
4.17.1. Tabulka Skupina .....	68

Popis .....	68
Tabulka .....	68
Data v číselníku.....	68
4.17.2. Tabulka Typ .....	69
Popis .....	69
Tabulka .....	69
Data v číselníku.....	69
4.17.3. Tabulka Druh .....	70
Popis .....	70
Tabulka .....	70
Data v číselníku.....	70
4.17.4. Tabulka Doklad.....	71
Popis .....	71
Tabulka .....	71
Data v číselníku.....	71
5. Zhodnocení nově vytvořeného programu .....	72
6. Závěr.....	74
7. Summary .....	76
8. Přehled použité literatury .....	78
9. Přílohy .....	80

## 1. Úvod

Cílem bakalářské práce je zhodnocení a analýza ekonomických potřeb domácnosti s následným vytvořením informačního softwaru.

Informační software "Domácí účetnictví" bude sloužit k evidenci a správě osobních financí a rodinného hospodaření.

Software umožní přehledně sledovat a plánovat osobní finance nebo hospodaření domácnosti. Umožní tak získat dokonalý přehled o příjmech, výdajích, stavu hotovosti a bankovních účtů. Software rovněž umožní sledovat a plánovat příjmy a výdaje, zaznamenávat spotřeby energií, plynu nebo vody, vést knihu jízd. Domácí účetnictví bude včetně adresáře kontaktů.

Další funkčnosti vyjdou z analýzy potřeb.

## 2. Přehled situace na trhu

Na trhu je velké množství programů, ve kterých je možné vést účetnictví, včetně domácího. Drtivá většina těchto produktů vychází z programů pro vedení účetnictví malých, středních a větších firem. Tyto programy jsou zaměřeny většinou na činnosti specifické pro firmy. Hlavním obsahem je například nákup a prodej zboží, příjem a výdej faktur, evidence zakázek a také podvojně účetnictví. V běžné domácnosti činnosti zcela nepoužitelné, či jenom částečně použitelné. Vybral jsem dva programy, u kterých jsem prověřil vhodnost pro domácí použití.

### Přehled dostupných aplikací:

Název	Moduly	Stručný popis
K2 [11]	Prodej, nákup, celnice, sklad, prodej	Software umožňuje vést domácí finance, je velmi modulární, pro vedení domácích financí složitý
Domácí účetnictví firmy KASTNER [12]	Účetnictví, kancelář	Software je určený pro domácnost. Výhodou je jeho rozsáhlost
Money S3 [13]	podvojně účetnictví, daňová evidence, adresář, fakturace, sklady, objednávky, mzdy, homebanking, internetový obchod a další aplikace.	Software je určený pro malé a střední firmy. V Money S3 lze vést i domácí finance. Předpokladem je základní znalost podvojně nebo jednoduchého účetnictví.
Money S4 [14]	Software pro střední firmy, nevhodný pro domácí užití	
Money S5 [15]	Software pro střední a velké firmy, nevhodný pro domácí užití	
Abra G1 [16]	Software pro střední firmy, nevhodný pro domácí užití	
Abra G2 [17]	Software pro střední firmy, nevhodný pro domácí užití	

Abra G3 [18]	Software pro střední a velké firmy, nevhodný pro domácí užití
Abra G4 [19]	Software pro střední a velké firmy, nevhodný pro domácí užití
Vision 32 [20]	Software pro střední a velké firmy, nevhodný pro domácí užití
Helios [21]	Software pro střední a velké firmy, nevhodný pro domácí užití
SAP [22]	Software pro střední a velké firmy, nevhodný pro domácí užití
Altus Vario [23]	Software pro střední a velké firmy, nevhodný pro domácí užití

Tabulka 1 - přehled dostupných aplikací

Na trhu je mnoho dalších komerčních programů, které nejsou vhodné pro domácí využití. Pro domácí využití programy téměř neexistují.

## 2.1 Profesionální ekonomicko-informační software K2

Profesionální ekonomicko informační software K2, výrobce K2 Atmitec, Ostrava [11], obsahuje tyto moduly:

PRODEJ - slouží k řízení a realizaci obchodních případů zákazníků.

NÁKUP - slouží k řízení vstupu zboží, surovin a služeb od dodavatelů.

CELNICE - slouží k evidenci pohybu zboží na celním skladě a vystavování průvodních dokladů – JSD.

SKLAD - slouží k zaznamenání toku a popisu parametrů veškerých firemních pohybů.

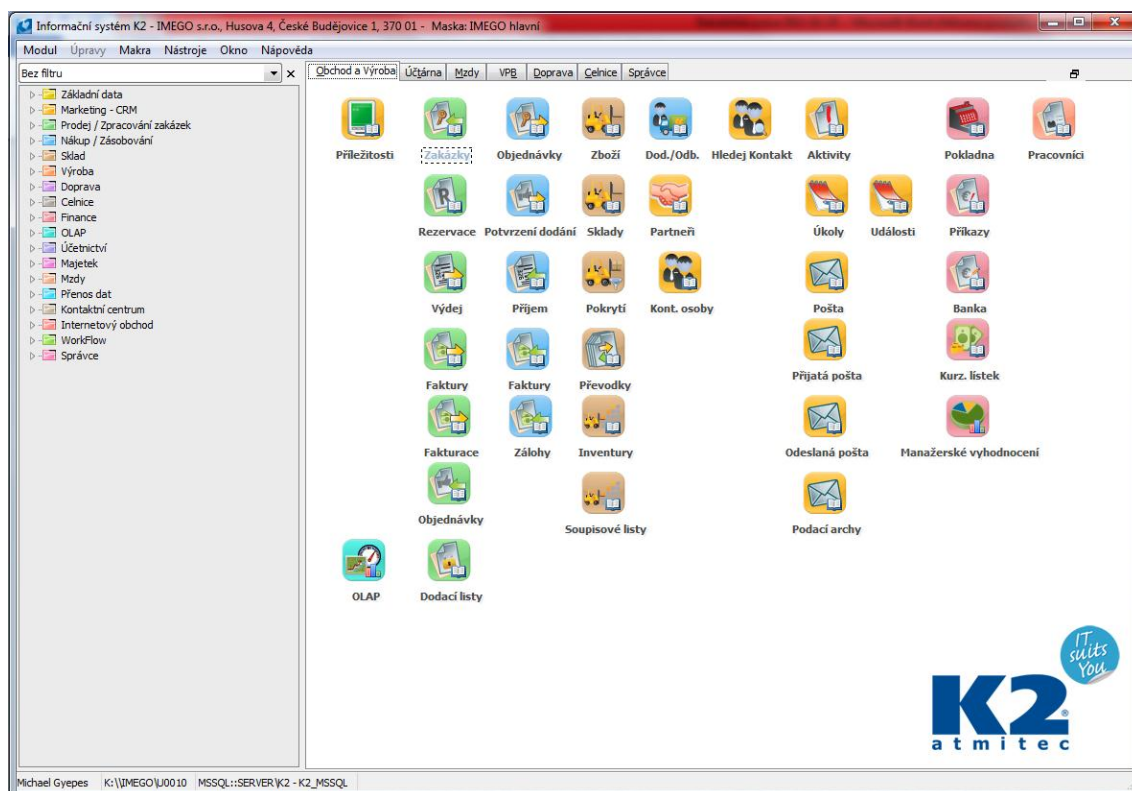
DOPRAVA - Slouží k plánování a řízení rozvozů zboží zákazníkům, svozů zboží od dodavatelů a sledování výnosnosti jednotlivých aut.

VÝROBA - slouží k tvorbě technologických postupů, plánování a sledování výroby.

FINANCE - slouží k on-line přehledu a automatizaci navádění plateb a platebních příkazů.

Modul Pokladna slouží k vedení libovolného množství pokladních knih v libovolných měnách.

MARKETING - slouží k vedení prodejních příležitostí směrem k uzavření obchodu.



Obrázek 1 – hlavní obrazovka EIS K2

### Komu je určen

Produkt EIS K2 [11] je určen především firmám, které vyžadují přizpůsobení informačního systému svým podmínkám a specifikům. Hlavní výhodou je možnost přizpůsobení a automatizace pracovních postupů pomocí specializovaných funkcí.

Firma může vytvářet složité pracovní postupy a procesy. Využitím registrovaných funkcí je správce schopen definovat podmíněné chování systému odvozené od vkládaných hodnot nebo od naplnění podmínek nutných k pokračování v pracovním postupu. Lze tak do programu vkládat logiku, která souvisí s daty a s předmětem podnikání více než s vytvářeným dokladem. EIS K2 [11] umožňuje vytvářet diferencované funkční akce jako přijetí poptávky, objednávky, pokrytí objednávky, rezervace zboží, výdej, fakturace, platba, zaúčtování a vyhodnocení závislé na okolnostech jejich vzniku (uživatel, místo, kniha, středisko, zákazník, výrobek).

### Vyhodnocení

EIS K2 [11] je velký systém určený především pro firmy. Pro jeho fungování je nutná instalace databáze Oracle nebo MS SQL. Instalace databází a EIS K2 [11] jsou náročné na výkon PC a vyžadují značný volný prostor na pevném disku.

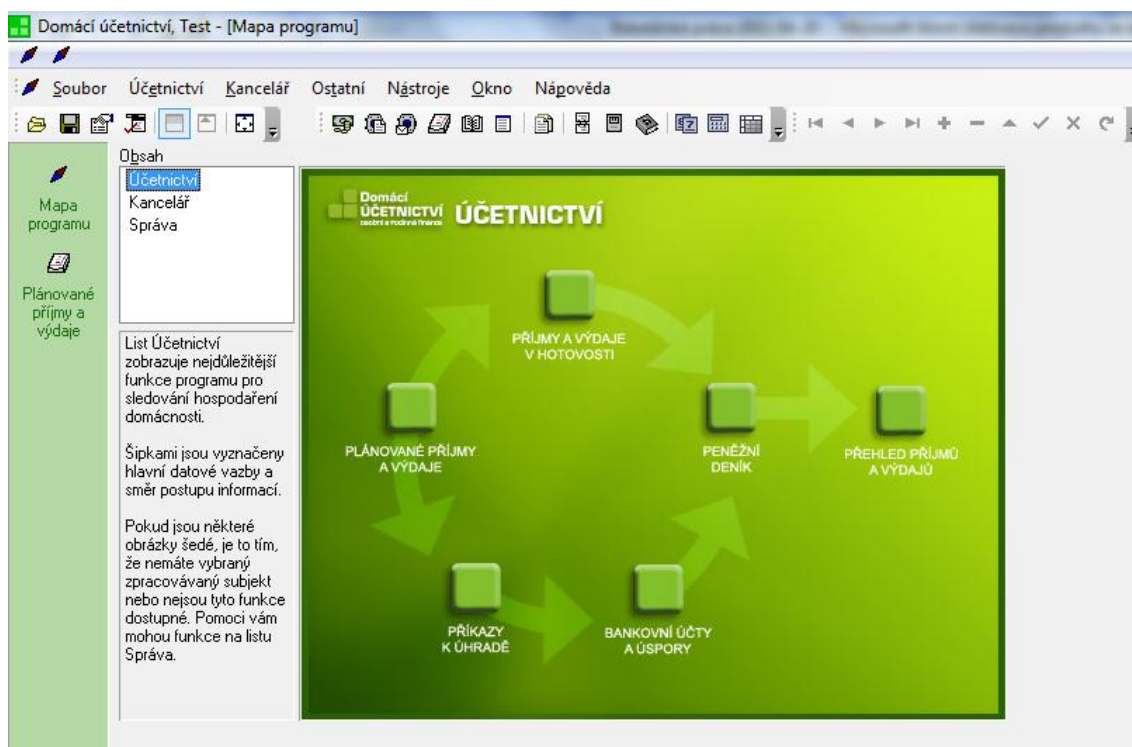
EIS K2 [11] je modulární a velmi přizpůsobivý všem požadavkům. Lze ho tedy využít i pro vedení domácích financí. S ohledem na jeho pořizovací cenu, hardwarovou náročnost, nutnou implementaci výrobcem a zaškolení uživatele, není vhodný pro domácí užití.

## 2.2 Program Domáci účetnictví firmy KASTNER software s.r.o.

Domáci účetnictví firmy KASTNER software s.r.o. [12] je program pro správu rodinných financí. Umožňuje sledovat a plánovat příjmy a výdaje domácnosti, zaznamenávat spotřeby energií, plynu nebo vody, vést knihu jízd, využívat adresář kontaktů nebo vyplňovat nejrůznější formuláře.

Obsahuje tyto agendy a moduly:

### Modul Účetnictví



Obrázek 2 - hlavní obrazovka programu Domáci účetnictví

Agenda ÚČETNICTVÍ zahrnuje moduly hotovost, bankovní účty, příkazy k úhradě, plánované příjmy a výdaje, účetní evidenci a výsledky hospodaření.

Modul HOTOVOST je určen pro vystavování příjmových a výdajových pokladních dokladů, hotovostních úhrad plánovaných příjmů a výdajů a jejich evidenci.

Modul BANKOVNÍ ÚČTY je určen k evidenci bankovních výpisů a položek bankovních výpisů.

Modul příkazy k úhradě slouží k vystavení příkazů k úhradě pro banku, jejich tisku nebo odeslání pomocí homebankingu a evidenci vystavených příkazů.



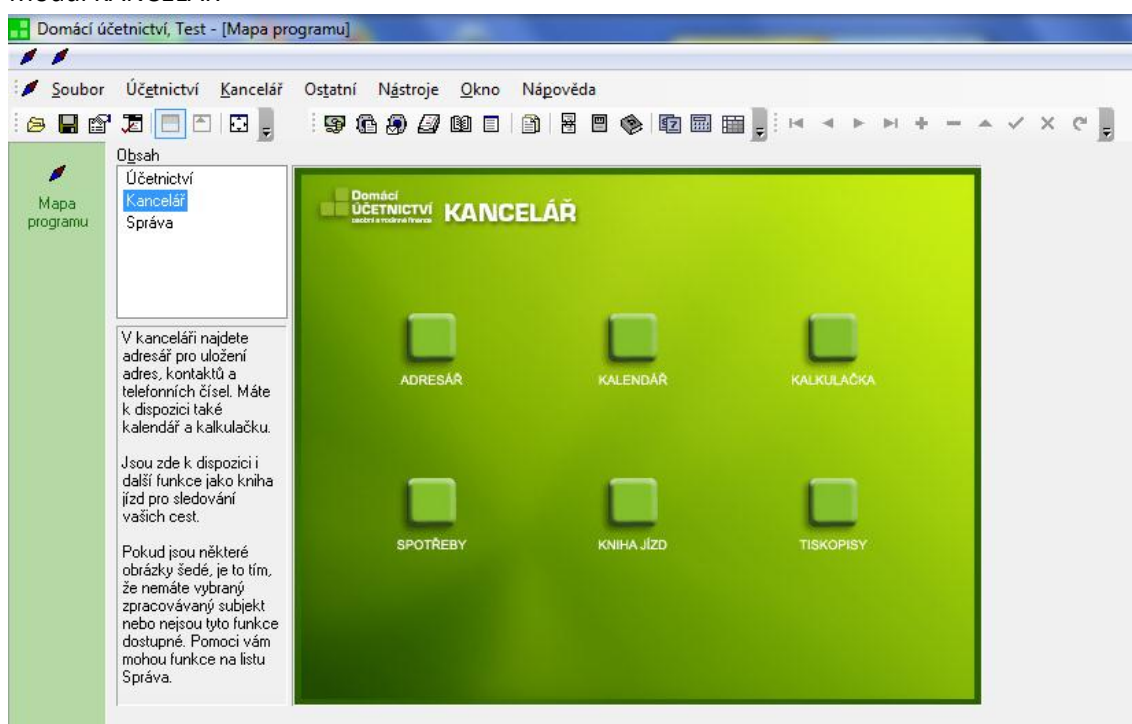
Pomocí modulu plánovaných příjmů a výdajů si lze evidovat příjmy a výdaje, které jsou očekávány v budoucnu. Záznamy v plánovaných příjmech a výdajích můžete využít při vytváření příkazů k úhradě nebo při zadávání jejich plateb v hotovosti nebo bankou. V těchto případech stačí zadat Uhrazovaný doklad a další údaje se převezmou z odpovídajícího záznamu plánovaných příjmů a výdajů.

Modul Účetní evidence zahrnuje peněžní deník a slouží pro spojení záznamů zadaných v agendách Hotovost, Bankovní výpisy. Můžeme v něm prohlížet a hledat záznamy společně pro obě agendy.

Výši příjmů a výdajů můžeme průběžně sledovat podrobně v okně Druhy příjmů a výdajů nebo po skupinách v okně Skupiny příjmů a výdajů (nabídka Účetnictví volba Druhy příjmů a výdajů nebo Skupiny příjmů a výdajů).

Příjmy a výdaje se do seznamu druhů napočítávají průběžně při zadávání pohybů peněz v hotovosti nebo na bankovních účtech. Zaznamenávají se údaje spadající do sledovaného období, které můžete určit v Nastavení domácnosti v části Zpracovávané období. Můžete si tak kdykoliv prohlédnout součty za zadané období. Pro případy, kdy potřebujete vidět jinak vybrané příjmy a výdaje (např. podle osob), zadejte si v Hotovosti nebo v Peněžním deníku požadovaný filtr a výsledky zobrazte pomocí sestav nebo grafů.

### Modul KANCELÁŘ



Obrázek 3 - agenda Kancelář

Agenda KANCELÁŘ zahrnuje moduly adresář, knihu jízd a spotřeby.

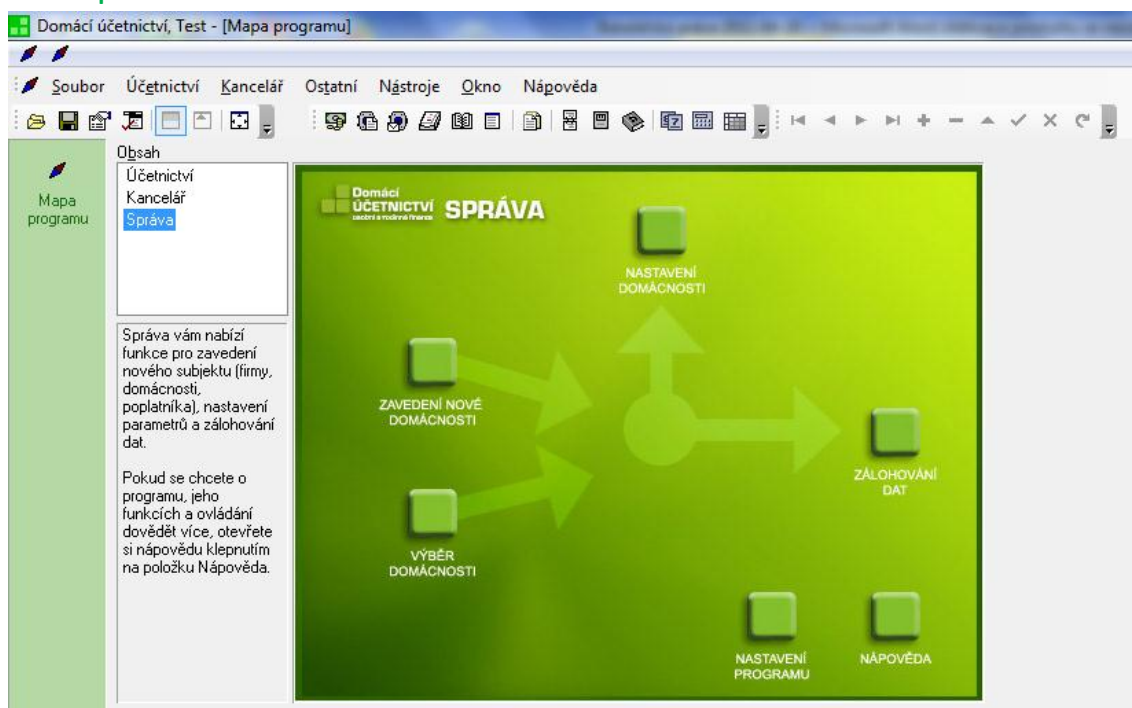
Modul kniha jízd slouží k evidenci pracovních, případně i soukromých cest a vedení knihy jízd. Umožňuje sledovat čerpání a spotřebu pohonných hmot a evidovat náhrady za pracovní cesty.

Model Spotřeba umožňuje sledovat spotřeby elektrické energie, plynu, vody, tepla případně jakýchkoliv dalších měřitelných médií.

Doklad	Datum	Druh	Text	Orientační název firmy	Částka
* h0014					
h0001	1.1.2006	ZÚST	Zůstatek z minulého období		5 000,00
h0002	3.1.2006	NÁJEM	Nájem	Bytový podnik/ Prostějov 1	1 293,00
h0003	5.1.2006	OPRAV	opravy zařízení	Petr Vojříř - instalater/ Prost	358,00
h0004	6.1.2006	BENZ	Benzin, nafta	AGIP	503,00
h0005	7.1.2006	ÚKLID	Úklid domácnosti	Proklid/ Olomouc 2	285,00
h0006	8.1.2006	VEDLE	Vedlejší příjmy		1 500,00
h0007	10.1.2006	PŘEVP	Příjem z jiné pokladny, účtu	taťínek Novák/ Prostějov 1	13 000,00
h0008	11.1.2006	PŘEVP	Příjem z jiné pokladny, účtu	maminka Nováková	8 000,00
h0009	12.1.2006	PŘEVP	Výdej na jinou pokladnu, účet		10 000,00
h0010	12.1.2006	PŘEVP	Výdej na jinou pokladnu, účet	maminka Nováková	3 000,00
h0011	18.1.2006	PŘEVP	Výdej na jinou pokladnu, účet	taťínek Novák/ Prostějov 1	3 000,00
h0012	20.1.2006	PŘEVP	Výdej na jinou pokladnu, účet	taťínek Novák/ Prostějov 1	4 500,00
h0013	26.2.2011	AUTO	Provoz auta	AGIP	200,00

Obrázek 4 - agenda Hotovost

## Modul správa



Obrázek 5 - agenda Správa

## Vyhodnocení

Domácí účetnictví firmy Kastner [12] je poměrně propracovaný program s mnoha funkcemi. Program Domácí účetnictví je vytvářen již několik let a z toho důvodu je poměrně rozsáhlý a obsahuje řadu funkcí. Nevýhodou tohoto programu je využití poměrně zastaralé databáze Paradox. Výběr této databáze byl určitě ovlivněn datem vzniku a cílovou skupinou programu. Za vážnější problém považuji absenci modulu Úkoly. Domnívám se, že Domácímu účetnictví předcházela účetní program zaměřený na malé a střední firmy. S ohledem na velkou konkurenci v této oblasti, firma následně vytvořila program určený pro domácnost. Z toho důvodu je tady základem modul Účetnictví a něj navazují jeho agendy. Přestože je tento program nejvíce zaměřen na domácnost, obsahuje řadu prvků, které vyžadují určité znalosti uživatele a které ztěžují vlastní práci.

K nedostatkům programu dále patří následující:

- Název agendy Účetnictví není vhodný pro domácí užití
- Název agendy Kancelář není vhodný pro domácí užití
- Ovládací prvky jsou spíše vhodné pro uživatele se základními znalostmi práce na PC. V domácích podmínkách je nutno počítat i s uživateli s velmi nízkými znalostmi práce na PC



Obrázek 6 - ovládací prvky

- Symbol diskety je v programu použit pro archivaci dat. Někteří uživatelé mohou zaměnit tuto funkčnost s uložením záznamu.
- Název agendy Plánované příjmy a výdaje není opět vhodný pro domácí užití
- Chybí kontrolní procesy. Pokud je například zadán v zápisu druh „Benzin“, nemělo by být možné zadat činnost „tennis“, ale pouze „provoz auta“. Stejně tak firma BAŤA – obuv produkt benzin nedodává.

Doklad	Datum	Variabilní symbol	Text	Orientační název firmy	Částka	Stav úhrad	Uhrazeno	Datum úhrady	Druh
*pz0005	26.2.2011		Benzin, nafta	BAŤA - obuv	2 000,00	Bez úhrad, po splatnosti	0,00		BENZ
pz0001	14.1.2006		Nové zařízení domácnosti	Elektra, s.t.o./ Bmo	5 000,00	Nastaveno jako uhrazené	5 000,00	1.2.2006	ZARÍZ
pz0002	14.1.2006		Vzdělávání, odborná literatura	Školský úřad/ Prostějov 1	3 500,00	Částečně uhrazeno, zbývá 2800,00 Kč	700,00	1.1.2006	VZDĚL
pz0003	15.1.2006		Spotřeba energií	e-on (elektrina)	4 500,00	Bez úhrad, po splatnosti	0,00		ENER
pz0004	18.1.2006	45555	Spoření	Česká pojišťovna/ Prostějov	800,00	Bez úhrad, po splatnosti	0,00		SPOŘ

Obrázek 7 - agenda Plánované příjmy a výdaje

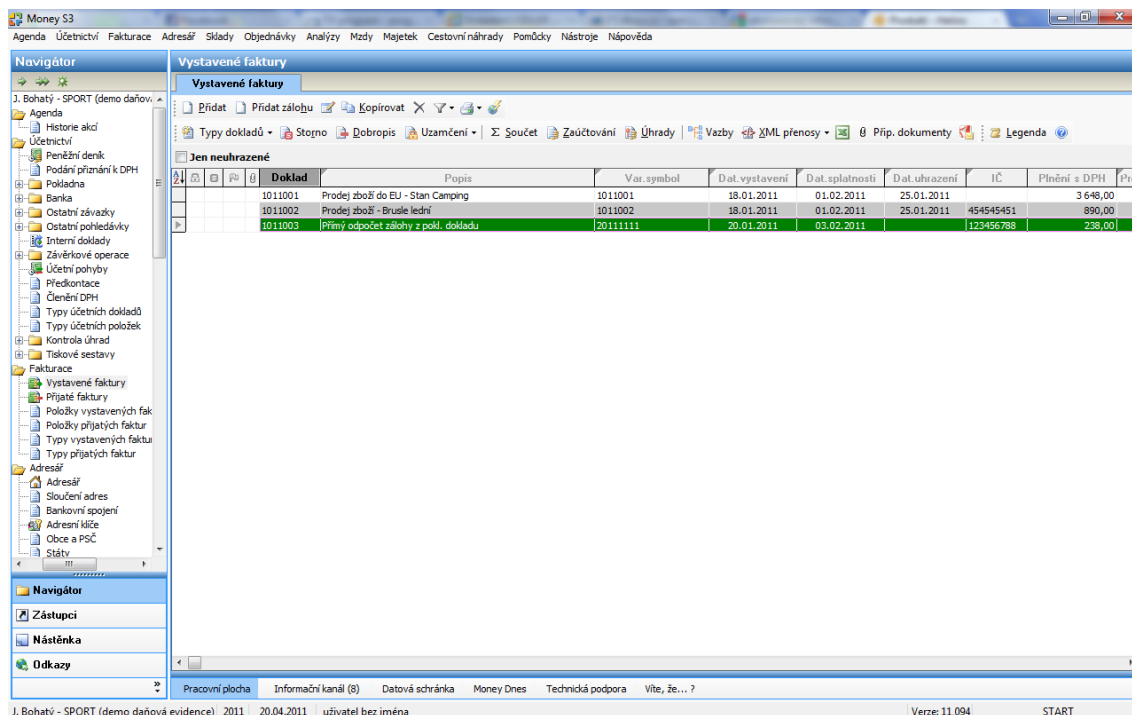
- Agenda Plánovací kalendář vzhledově připomíná velmi rozšířený MS Outlook. Zásadním problémem je neprovázanost s ostatními agendami.

Přes zmíněné nedostatky je tento program s ohledem na funkčnost, cílovou skupinu a pořizovací cenu, ve srovnání s ostatními komerčními programy nejvíce vhodný pro domácí použití.

### 2.3. Účetní program Money S3

Účetní program Money S3 [13] patří mezi nejrozšířenější ekonomické systémy pro malé a střední firmy v České i Slovenské republice a také patří k nejstarším programům vyvíjeným v České republice. Obsahuje moduly – podvojný účetnictví, daňovou evidenci, adresář, fakturaci, sklady, objednávky, mzdy, homebanking, internetový obchod a další aplikace.

Účetní program Money S3 [13] je využíván v oblastech nákupu a prodeje, ve službách a ve výrobě. Lze ho také využít pro vedení domácích financí.



Obrázek 8 - hlavní obrazovka Money S3

## Vyhodnocení

Stejně jako Domácí účetnictví firmy Kastner [12] je Money S3 [13] poměrně propracovaný program s mnoha funkcemi. Program je také vytvářen již několik let a je také poměrně propracovaný a obsahuje řadu funkcí. Výhodou tohoto programu je velké množství funkcí a doplňků. V porovnání s ekonomickým systémem K2 [11] je méně přizpůsobitelný, avšak cenově velmi dostupný. Pro domácí využití je nevhodný.

### 3. Analýza potřeb domácnosti

Domácí uživatelé mají rozdílné znalosti jak účetní, tak uživatelské s počítači. Program na vedení domácích potřeb, financí musí být jednoduchý, srozumitelný, přehledný a musí mít moderní vzhled. Uživatel musí být schopen velmi rychle zadat potřebná data do programu. Program musí být uživatelsky přívětivý. Naplánovat lze tedy jen předpokládané příjmy a výdeje.

Co tedy běžná domácnost potřebuje? Provedl jsem analýzu v deseti domácnostech. Tato analýza byla založena na odpovědích získaných formou osobního dotazování na připravené otázky. Zjistil jsem, že ve většině analyzovaných domácností jsou řešeny tyto problematiky.

Domácnost	1	2	3	4	5	6	7	8	9	10	Výskyt
domácí účetnictví	A		A			A					3
pohledávky a závazky	A	A	A	A		A		A	A	A	8
stav financí	A	A	A	A		A		A	A	A	8
sledování spotřeby energií v domácnosti	A		A	A	A	A		A	A	A	8
evidence osobního majetku					A				A		2
Evidence záručních listů					A			A			2
Evidence vozidel a spotřeba PHM	A	A		A							3
kuchařské recepty					A		A				2
Evidence pojištění	A	A	A	A		A		A	A	A	7
nákupní lístek							A				1
osobní údaje (také míry a váhy)					A		A				2
strava a pohybová aktivita (příjem a výdej energie)					A		A				2
BMI (index tělesné hmotnosti)					A		A				2
návštěvy lékařů		A			A		A				3
evidence očkování		A			A		A				3
evidence spotřeby léků		A					A				2
sledování tělesné teploty											0
menstruační kalendář					A						1
výpočet alkoholu v krvi									A	A	2
osobní deník	A				A						2
antisklerotik							A				1
archiv dokumentů	A		A	A					A		4

evidence přístupových kódů											0
evidence adres	A		A	A		A			A		5
vytváření korespondence									A		1
plánování úkolů	A		A	A	A	A		A	A	A	8

Tabulka 2 (zdroj: vlastní výzkum)

Získaná data jsou z malého počtu domácností a jsou pouze z určité skupiny obyvatel. Přesto z těchto získaných dat lze zjistit, jaké problematiky je třeba v programu Domácí účetnictví řešit.

V první verzi programu Domácí účetnictví budou řešeny tyto problematiky:

- deník, úkoly
- doklady přijaté
- doklady vydané
- hotovost
- banka
- provoz vozidel
- spotřeby, energie
- pojistné smlouvy

### 3.1. Diference osobních a firemních financí

Základním stavebním prvkem většiny programů pro vedení ekonomiky firem či domácností je zpravidla modul jednoduchého nebo podvojného účetnictví. Na tento modul jsou následně navázány ostatní agendy, jako je evidence faktur, plateb, bankovních výpisů,...

Firmy (podnikatelské subjekty) potřebují zpravidla řešit několik problematik. V první řadě se jedná o účetnictví. Je zde nutné řešit veškeré účetní pohyby, s následnými ekonomickými výsledky. Tyto výsledky jsou nutné mimo jiné pro daňová přiznání. Je také nutné řešit personální a mzdové záležitosti. Modul účetnictví musí zajistit správnost účetních dat a výsledků. Další problematikou firmy je obchod a případně výroba. Tyto moduly musí zajistit rychlou dostupnost informací ohledně dodacích a výrobních termínů, skladových zásob, cen a jiných agend, úzce svázaných s účetnictvím. Je také nutné zajistit legislativní správnost.

Pro domácího uživatele moduly jednoduché a podvojně účetnictví nejsou vůbec stavebním prvkem. Domácí uživatel těmto modulům zpravidla vůbec nerozumí a nutností je pro něj řešení zcela jiných problematik. Domácí uživatel potřebuje získat informace o stavu hotovosti, o stavu financí na bankovních účtech, o stavu pohledávek, závazků, případně potřebuje získat některá data pro daňová přiznání, která jsou však obvykle jiná, než podávají firmy. Domácí firmy obvykle nepodávají daňové přiznání k DPH, spotřební dani, silniční dani, ale například k dani z příjmu, k dani z nemovitosti a jiným daním. Výstupy z programu domácí finance musí být tedy jiné.

## 4. Praktická část

Na základě jednoduché analýzy potřeb domácnosti jsem přikročil k samotnému úkolu a vytvořil vlastní program pro domácí finance. První problém, který bylo nutné vyřešit, byl výběr databáze. Jednalo se o velmi složitý a časově náročný krok. Bylo nutné zohlednit cílového klienta a také možnosti programovacího jazyka. Následným krokem byl výběr programovacího jazyka a po té již došlo k samotné tvorbě programu.

### 4.1. Výběr databáze

Pro realizaci projektu jsem vybral databázi MS Access (plným jménem **Microsoft Office Access**). Tento nástroj na správu relačních databází od společnosti Microsoft je volitelně součástí Microsoft Office. S ohledem na cílového uživatele byly důvodem výběru databáze MS Access mimo jiné snadná dostupnost, jednoduchá editovatelnost dat a snadná přenositelnost. Také není nutno instalovat databázový nástroj, jako například v případě Oracle, Microsoft SQL, Intabase, FireBinder. I to byl důvod výběru MS Access.

Pokud by byl program stavěn pro firemní finance, určitě bych využil některého databázového nástroje, jako je například Microsoft SQL, či Oracle. Pro domácí užití lze samozřejmě tyto databáze také použít, jsou dostupné i zdarma, ale je zde nutná složitější instalace a případná editace dat.

Dalším databázovým nástrojem, o kterém jsem uvažoval, byl nástroj BDE od společnosti Borland [XX]. Pro jeho zastaralost jsem ho také nevybral.

### 4.2. Výběr programovacího jazyka

Pro realizaci projektu Domácí účetnictví jsem použil Embarcadero® RAD Studio 2010, Delphi [1-6]. Důvodem byla především moje znalost tohoto programovacího jazyka. Programovací jazyk Delphi je integrované grafické vývojové prostředí firmy Borland určené pro tvorbu aplikací na platformě MS Windows v jazyce Object Pascal (objektové nástavbě Pascal). Obsahuje systém RAD (Rapid Application Development), který umožňuje vizuální návrh grafického uživatelského rozhraní, na jehož základě je automaticky vytvářena kostra zdrojového kódu, což výrazně urychluje vývojový cyklus.

Programování v Delphi je z velké části založeno na použití komponent. Komponenta je malý program (balíček funkcí), který vykonává určitou činnost (například zobrazuje text nebo obrázky, přehrává multimédia, komunikuje s databází, zprostředkovává FTP přenos atd.).

Velkou předností Delphi proti některým konkurenčním produktům jsou knihovny komponent, které jsou jejich součástí (např. VCL, CLX, Indy ...). Dodávané komponenty významně usnadňují tvorbu aplikací. Další komponenty lze stáhnout z internetu (některé jsou zadarmo, některé se musí koupit). V Delphi lze také vytvářet vlastní komponenty.



### 4.3. Databáze

Hlavní databází programu je soubor data.accdb. Tato databáze obsahuje následující tabulky:

Hlavní tabulky	Popis	Delphi
Kniha	Všechny doklady přijaté, doklady vydané, úkoly	tbKniha
RadaFinance	Řady pokladen a bank	tbRadaHotovost tbRadaBanka
Finance	Kniha pohybu financí	tbHotovost tbBanka
Sablony	Kniha šablon	tbSablona
Vozidla	Kniha vozidel	tbVozidla
Jizdy	Kniha jízd	tbJizdy
Spotreba	Kniha předmětu a místa spotřeby	tbSpotreba
SpotrebaDetail	Kniha záznamů stavu spotřeby	tbSpotrebaDetail
Pojisteni	Kniha předmětu pojištění	tbPojistky
PojisteniDetail	Kniha pojistných smluv	tbPojistkyDetail
Pomocné tabulky (číselníky)		
Typ	Typ záznamu: faktura přijatá, platba přijatá v hotovosti, úkol, bankovní výpis, vozidlo, ...	tbTyp
Skupina	Skupina záznamu: elektrika, plyn, voda, nájem,..	tbSkupina
Druh	Druh záznamu: příjem, výdej, úkol, banka, závazek, dluh, vozidlo	tbDruh
Místo	Druh místa, o kterém je vytvářen záznam	tbMisto
Firma	Firma, osoba	tbFirma
Město	Města a směrovací čísla	tbMesto
Doklad	Typ dokladu	tbDoklad

Tabulka 3 – tabulky v databázi data.accdb

ID	Typ	Zkratka	Popis	Doklad	Skupina	Druh
44	5 U - SERVIS AUT	Úkol - servis auta		6	9	3
45	5 U - NAROZENI	Úkol - Narozeniny babičky Marie dne 25.		6	9	3
46	5 U - SCHUZE	Úkol - Schůze		6		3
47	5	Úkol - vyměnit řidičský průkaz				3
48	5 U -	Úkol - daňové přiznání - daň z příjmu		6	10	3
49	5 U - BAKALARK	Úkol - Odevzdat bakalářskou práci ke kou				3
50	5 U - BAKALARK	Úkol - Odevzdat bakalářskou práci ke kou				3
51	1 FAP -	Faktura přijatá - pojistná smlouva - Chal		1	8	4
52	1 FAP - elektrika	Faktura přijatá - elektrika		1	1	4
54	1 FAP - mobilní t	Faktura přijatá - mobilní telefon		1		4
55	1 FAP - odpad	Faktura přijatá - odpad		1		4
56	1 FAP - voda	Faktura přijatá - voda		1	3	4
57	1 FAP - plyn	Faktura přijatá - plyn		1		4
58	3 PUJCKA -	Splátka poskytnuté půjčky		1		5
59	3 PRONAJEM -	Pronájem garáže leden 2011		1		5
60	3 PRONAJEM -	Pronájem garáže únor 2011		1		5
61	3 PRONAJEM -	Pronájem garáže březen 2011		1		5
62	3 MZDA -	Mzda leden 2011		1		5
63	3 MZDA -	Mzda únor 2011		1		5
64	1 ÚČTENLA - PHM	Účtenka - PHM		1	5	4
65	1 ÚČTENLA - PHM	Účtenka - PHM		1	5	4
*	(Nové)					

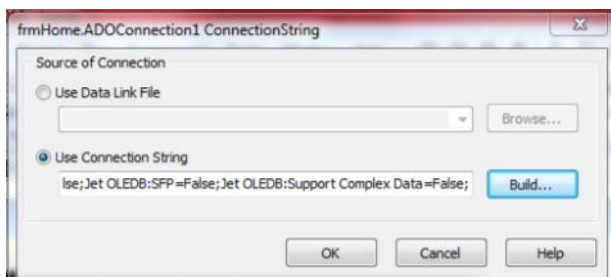
Obrázek 9 – databáze MS Access, tabulky

## Připojení k databázi

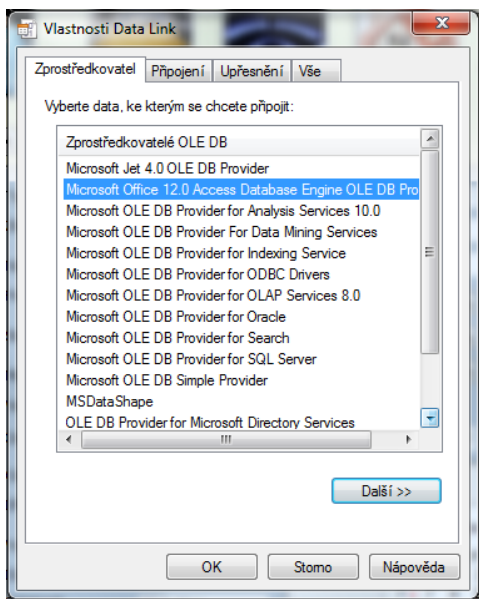
Základní komponentou pro navázání databázové konektivity je ADOConnection. Umožňuje využití celé řady zdrojů, JET, ODBC, nativní ovladače například pro MS SQL či ORACLE, katalogové a indexovací služby. Nejjednodušší příklad můžeme realizovat s databází MS Access. Na formulář umístíme komponentu ADOConnection. Pomocí průvodce pro nastavení vlastnosti ConnectionString vybereme variantu vytvoření řetězce a poskytovatele dat MS Jet OLE DB. Dialog tohoto poskytovatele nám následně umožní zadat cestu k databázovému souboru, jméno a heslo pro přihlášení k databázi.



Obrázek 10 – komponenta ADOConnection



Obrázek 11 – ADOConnection



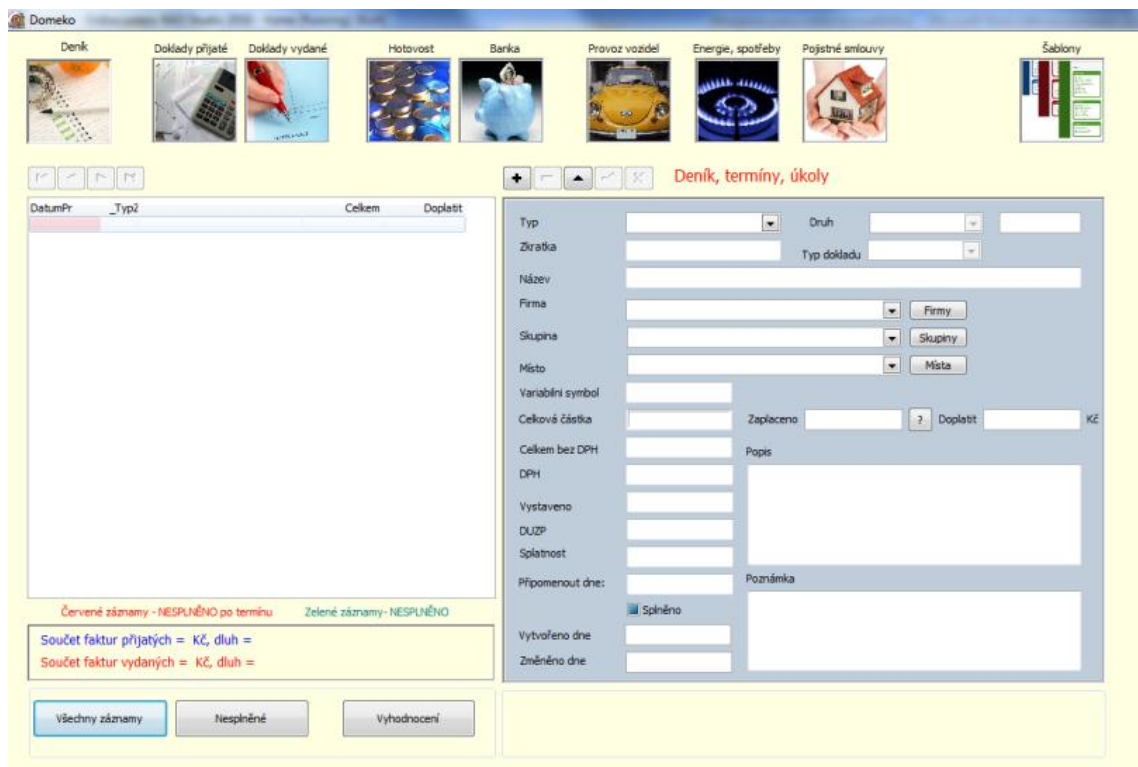
Obrázek 12 – zprostředkovatel spojení s databází MS Access

## 4.4. Hlavní obrazovka programu

Hlavní obrazovku jsem rozdělil na 4 přehledné části. Jak jsem se již několikrát zmínil, při tvorbě programu jsem zohledňoval cílového klienta. Stěžejní tedy bylo, aby hlavní obrazovka byla přehledná, srozumitelná a jednoduchá. Bylo nutno respektovat, že s programem mohou pracovat i uživatelé s velmi nízkými PC znalostmi.

Hlavní obrazovka je rozdělena na tyto části:

- Horní část - ikony jednotlivých modulů
- Střední část levá - seznam dokladů
- Střední část pravá - detail dokladu
- Spodní část - funkční tlačítka



Obrázek 13 – hlavní obrazovka programu

## 4.5. Moduly



modul Deník -



modul Doklady přijaté



modul Doklady vydané



modul Hotovost



modul Banka



modul Provoz vozidel



modul Spotřeba



modul Pojistné smlouvy



modul Šablona

## 4.6. Obecná funkčnost

Při konstrukci programu byly využity některé vestavěné komponenty, například pro pohyb, vkládání, mazání záznamů. Dále jsou zde popsány obecné funkčnosti platné v celém programu.



Obrázek 14 – navigátor pro posun

Objekt Navigator pro posun v tabulce.

Vykonává příkazy Last (poslední záznam), Next (další záznam), First (první záznam), Prior (předchozí záznam)



Obrázek 15 – navigátor pro vkládání, mazání, editaci

Objekt Navigator pro vkládání, mazání, editaci záznamů v tabulce.

Vykonává příkazy Insert (nový záznam), Delete (smazat záznam), Edit (editace záznamu), Cancel (zrušit změnu), Refresh (obnovit tabulku)

Při použití příkazů se obecně vykonají následující činnosti:

Insert – vloží se aktuální datum do pole DatumVy (datum vytvoření)

Edit – vloží se aktuální datum do pole DatumZm (datum změny)

Při startu programu je implicitně nastaven modul Deník.

## 4.7. Modul Deník

### Popis

Modul Deník slouží k evidenci úkolů a především termínů jejich plnění.

V seznamu jsou volitelně zobrazeny všechny úkoly nebo pouze nesplněné. Jednotlivé záznamy jsou barevně rozlišeny.

Červené – nesplněné úkoly po termínu, zelené – nesplněné úkoly před termínem, černé záznamy – splněné úkoly.

Splněné úkoly jsou rozlišeny datovým polem Splneno. Toto pole je měněno automaticky, např. při úplném zaplacení faktury nebo manuálně. Seznam je řazen podle termínu plnění vzestupně.

### Tabulka

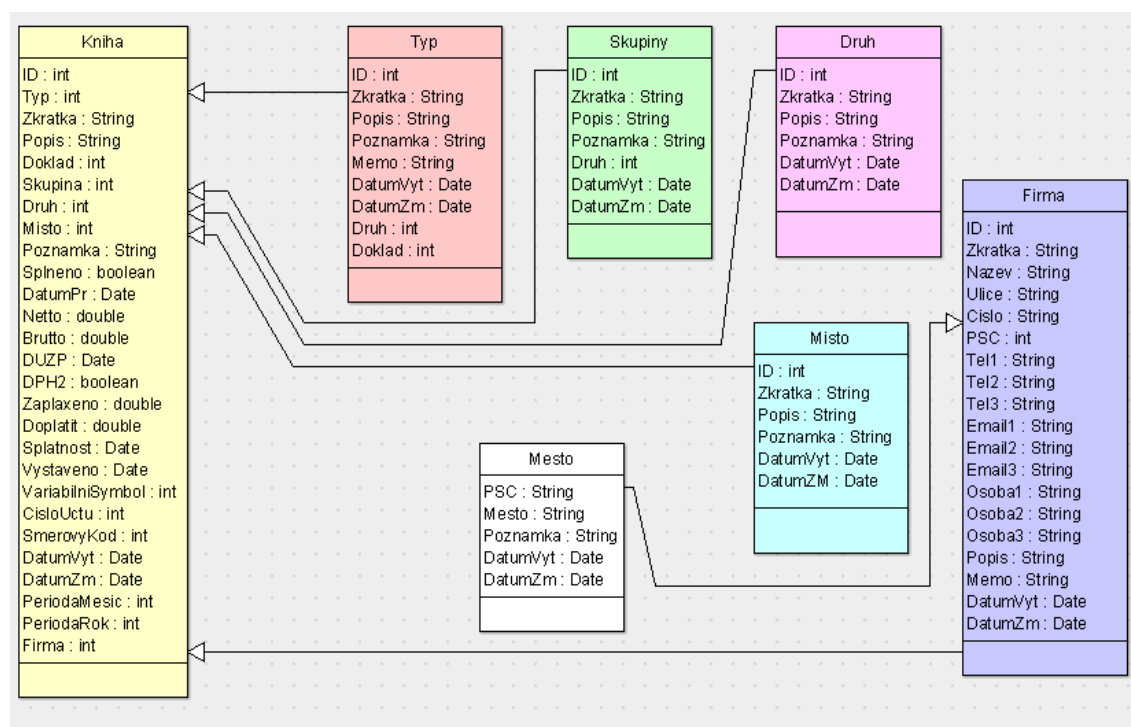
Tabulka Kniha z databáze data.accdb, v programu Delphi tbKniha.

Název pole	Datová typ	Popis
ID	Automatické číslo	Automaticky generované unikátní číslo
Typ	Číslo	Vazba do tabulky Typ
Zkratka	Text	Stručná identifikace úkolu, dokladu
Popis	Text	Podrobnější identifikace úkolu, dokladu
Doklad	Číslo	Vazba do tabulky
Skupina	Číslo	Vazba do tabulky Skupina
Druh	Číslo	Vazba do tabulky Druh
Místo	Číslo	Vazba do tabulky Místo
Poznamka	Memo	Poznámka k záznamu
Splneno	Ano/ne	Ano – úkol splněn / Ne – úkol nesplněn
DatumPr	Datum a čas	Datum připomenutí
Netto	Měna	Částka dokladu bez DPH
Brutto	Měna	Celková částka dokladu
DUZP	Datum a čas	Datum uskutečnitelného zdanitelného plnění
DPH2	Ano/ne	
Zaplaceno	Měna	Celkově zaplacená částka dokladu
Doplatit	Měna	Celkový dluh dokladu
Splatnost	Datum a čas	Datum splatnosti dokladu
Vystaveno	Datum a čas	Datum vystavení dokladu
Firma	Číslo	Vazba do tabulky Firma
VariabilniSymbol	Číslo	Variabilní symbol dokladu přijatého a vydaného
CisloUctu	Číslo	Číslo bankovního účtu
SmerovyKod	Číslo	Směrový kód banky

DatumVyt	Datum a čas	Generované datum dne vytvoření záznamu
DatumZm	Datum a čas	Generované datum dne změny záznamu
PeriodaMesic	Číslo	Generované číslo měsíce, pro potřeby statistik
PeriodaRok	Číslo	Generované číslo roku, pro potřeby statistik

Tabulka 4 – tabulka Kniha

## UML



Obrázek 16 – UML modulu Deník

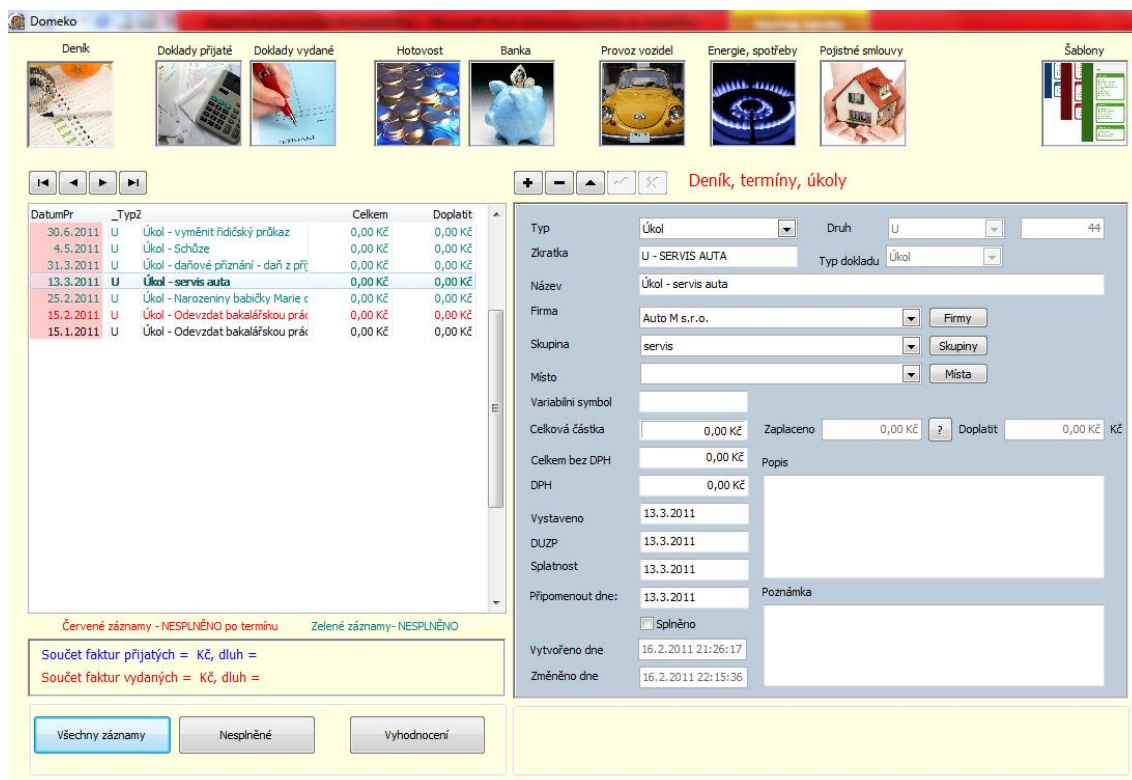
## Modelový příklad

Činnost	Popis
01	Servis auta dne 13. 3. 2011
02	Narozeniny babičky Marie dne 25. 2. 2011
03	Schůze sportovního oddílu 4. 5. 2011 od 19 hod.
04	Vyměnit řidičský průkaz, konec platnosti 31. 6. 2011
05	Podat daňové přiznání z příjmu dne 31. 3. 2011
06	Odevzdat bakalářskou práci ke kontrole 15. 2. 2011

Tabulka 5 – modelový příklad



## Obrázek modulu



Obrázek 17 – obrazovka modulu Deník

## Funkčnost modulu

### Nový záznam (Insert)

Po vložení nového záznamu se nastaví počáteční hodnoty

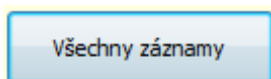
- vložení aktuálního data do pole DatumPr (tbKnihaDatumPr.Text: = DateToStr(Now))
- vložení aktuálního data do pole Splatnost (tbKnihaSplatnost.Text: = DateToStr(Now))
- vložení aktuálního data do pole Vystaveno (tbKnihaVystaveno.Text: = DateToStr(Now))
- vložení aktuálního data do pole DUZP (tbKnihaDUZP.Text: = DateToStr(Now));
- vložení hodnoty 0 do pole Netto (tbKnihaNetto.Value := 0)
- vložení hodnoty 0 do pole Brutto (tbKnihaBrutto.Value := 0)
- vložení hodnoty 0 do pole DPH (tbKnihaDPH.Value := 0)
- vložení hodnoty 0 do pole Zaplacen (tbKnihaZaplaceno.Value := 0)
- vložení hodnoty 0 do pole Doplatit (tbKnihaDoplatit.Value := 0)
- vložení hodnoty Ne do pole Splneno (tbKnihaSplneno.Value := false)
- vložení hodnoty 3 = Úkol do pole Druh (tbKnihaDruh.Value := 3)
- vložení hodnoty 5 = Úkol do pole Typ (tbKnihaTyp.Value := 5)
- vložení textu „Úkol“ do pole Zkratka (tbKnihaPopis.Text: = 'Úkol -')

Po vložení nového záznamu (Insert) je zobrazena kniha Šablon a je zapnut filtr podle typu 5, tedy úkolů.

```
(tbSablona.Filter := 'Typ = 5')
```

```
(tbSablona.Filtered := true)
```

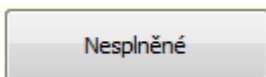
### Zobrazení záznamů



Obrázek 18 – tlačítko všechny záznamy

### Zobrazení všech záznamů je realizováno pomocí SQL příkazu:

```
SELECT  
Kniha.ID, Kniha.Typ, Kniha.Zkratka, Kniha.Skupina, Kniha.Druh, Kniha.Misto, Kniha.Popis,  
Kniha.Memo, Kniha.Poznamka, Kniha.Splneno, Kniha.DatumPr, Kniha.DatumVyt,  
Kniha.DatumZm, Kniha.Netto, Kniha.Brutto, Kniha.DPH, Kniha.DUZP, Kniha.DPH2,  
Kniha.Zaplaceno, Kniha.Doplatit, Kniha.Splatnost, Kniha.Vystaveno, Kniha.Firma,  
Kniha.VariabilniSymbol, Kniha.CisloUctu, Kniha.SmerovyKod, Kniha.Doklad,  
Kniha.PeriodaMesic, Kniha.PeriodaRok  
FROM Kniha  
ORDER BY Kniha.DatumPr DESC;
```



Obrázek 19 – tlačítko nesplněné záznamy

### Zobrazení nesplněných záznamů je realizováno pomocí SQL příkazu a filtru dat přes pole Splneno:

```
SELECT Kniha.ID, Kniha.Typ, Kniha.Zkratka, Kniha.Skupina, Kniha.Druh, Kniha.Misto,  
Kniha.Popis, Kniha.Memo, Kniha.Poznamka, Kniha.Splneno, Kniha.DatumPr, Kniha.DatumVyt,  
Kniha.DatumZm, Kniha.Netto, Kniha.Brutto, Kniha.DPH, Kniha.DUZP, Kniha.DPH2,  
Kniha.Zaplaceno, Kniha.Doplatit, Kniha.Splatnost, Kniha.Vystaveno, Kniha.Firma,  
Kniha.VariabilniSymbol, Kniha.CisloUctu, Kniha.SmerovyKod, Kniha.Doklad,  
Kniha.PeriodaMesic, Kniha.PeriodaRok  
FROM Kniha  
ORDER BY Kniha.DatumPr DESC;
```

```
tbKniha.Filter := 'Splneno = False';  
tbKniha.Filtered := true;
```

### Splněno/Nesplněno

Úkoly jsou rozděleny na splněné a nesplněné. V tabulce je nastavováno pole Splneno na hodnoty Ano= Splněno/Ne=Nesplněno. Změna pole je prováděna manuálně editací záznamu, dvojklikem na grdSeznam a programově při splnění podmínek úkolu.

#### Program pro změnu pole Splneno dvojklikem:

```
procedure TgrdSeznamDbClick(Sender: TObject);  
begin  
  tbKniha.Edit;  
  if tbKnihaSplneno.Value = true then  
    tbKnihaSplneno.Value := false  
  else  
    tbKnihaSplneno.Value := true;  
  tbKniha.Post;  
end;
```

### grdSeznam

Pro přehlednost knihy úkolů jsou záznamy barevně rozlišeny. Splněné úkoly jsou černé, nesplněné po termínu červené a nesplněné před termíny zelené.

#### Program:

```
procedure TgrdSeznamDrawColumnCell(Sender: TObject; const Rect: TRect;  
  DataCol: Integer; Column: TColumn; State: TGridDrawState);  
begin  
  if tbKnihaSplneno.Value = false then  
    begin  
      //  
      if tbKnihaDatumPr.Value >= Now then grdSeznam.Canvas.Font.Color := clTeal  
      else grdSeznam.Canvas.Font.Color := clRed;  
      //  
    end  
  else  
    grdSeznam.Canvas.Font.Color := clBlack;  
  grdSeznam.DefaultDrawColumnCell(Rect, DataCol, Column, State);  
end;
```

## 4.8. Modul Doklady přijaté

### Popis

Modul Kniha dokladů přijatých slouží k evidenci všech přijatých dokladů, dokladů zaplacených v hotovosti, všech závazků vůči dodavatelům zboží, či služeb, dále smluvních závazků, jako jsou například nájmy, leasingy, splátky úvěru. Tento modul zahrnuje evidenci všech závazků a jejich stavu.

Shodně s modulem Deník (tabulka v databázi je stejná), jsou záznamy barevně rozlišeny. Splněné úkoly jsou rozlišeny datovým polem Splněno. Toto pole je měněno automaticky, např. při úplném zaplacení faktury nebo manuálně. Seznam je řazen podle termínu plnění vzestupně.

### Tabulka

Tabulka pro evidenci dokladů přijatých je shodná s tabulkou pro Deník. Jedná se o tabulku Kniha z databáze data.accdb, v programu Delphi tbKniha. Tabulka je filtrována přes Typ = 1.

### Program:

```
tbKniha.Filter := 'Typ = 1';
```

```
tbKniha.Filtered := true;
```

### UML

UML diagram je shodný s UML diagramem v modulu Deník.

### Modelový příklad

Činnost	Popis
01	Faktura přijatá za elektriku, EON, částka 12600 Kč, splatno 7. 3. 2011
02	Vyúčtování pojistného, ČSOB Pojišťovna, částka 1850 Kč, splatno 14. 1. 2011
03	Faktura přijatá – mobilní telefon, VODAFON, částka 1450 Kč, splatno 28. 2. 2011
04	Faktura přijatá za odpad, A.S.A., částka 450 Kč, splatno 31. 3. 2011
05	Faktura přijatá za vodu, ČEVAK, částka 7500 Kč, splatno 31. 1. 2011
06	Faktura přijatá za plyn, EON, částka 3600 Kč, splatno 21. 2. 2011

Tabulka 6 – modelový příklad

## Obrázek modulu



Obrázek 20 – obrazovka Doklady přijaté

## Funkčnost modulu

### Nový záznam (Insert)

Po vložení nového záznamu se nastaví počáteční hodnoty

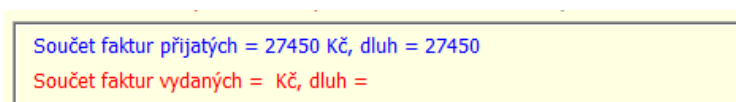
- vložení aktuálního data do pole DatumPr (tbKnihaDatumPr.Text: = DateToStr(Now))
- vložení aktuálního data do pole Splatnost (tbKnihaSplatnost.Text: = DateToStr(Now))
- vložení aktuálního data do pole Vystaveno (tbKnihaVystaveno.Text: = DateToStr(Now))
- vložení aktuálního data do pole DUZP (tbKnihaDUZP.Text: = DateToStr(Now));
- vložení hodnoty 0 do pole Netto (tbKnihaNetto.Value := 0)
- vložení hodnoty 0 do pole Brutto (tbKnihaBrutto.Value := 0)
- vložení hodnoty 0 do pole DPH (tbKnihaDPH.Value := 0)
- vložení hodnoty 0 do pole Zaplaceno (tbKnihaZaplaceno.Value := 0)
- vložení hodnoty 0 do pole Doplatit (tbKnihaDoplatit.Value := 0)
- vložení hodnoty Ne do pole Splneno (tbKnihaSplneno.Value := false)

- vložení hodnoty 5 = Úkol do pole Typ (tbKnihaTyp.Value := 1)
- vložení textu „Úkol“ do pole Zkratka (tbKnihaPopis.Text := 'FAP - ')

Po vložení nového záznamu (Insert) je zobrazena kniha Šablon a je zapnut filtr podle typ = 1, tedy úkolů.

```
(tbSablona.Filter := 'Typ = 1')
(tbSablona.Filtered := true)
```

### Součet faktur přijatých a celkový dluh



Obrázek 21 – součet a dluh faktur přijatých

**Součet faktur přijatých je realizován pomocí SQL (v Delphi tbObratFPSumOfBrutto) :**

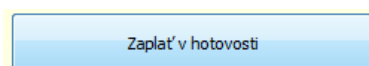
```
SELECT Kniha.Typ, Sum(Kniha.Brutto) AS SumOfBrutto
FROM Kniha
GROUP BY Kniha.Typ
HAVING (((Kniha.Typ)=1));
```

**Součet dluhu u faktur přijatých je realizován pomocí SQL (v Delphi tbDluhFPSumOfDoplatit):**

```
SELECT Kniha.Typ, Sum(Kniha.Doplatit) AS SumOfDoplatit
FROM Kniha
GROUP BY Kniha.Typ
HAVING (((Kniha.Typ)=1));
```

### Hotovostní platba

Veškeré hotovostní i bezhotovostní platby jsou evidovány v tabulce Finance. Platbu faktury přijaté je možné provést pomocí Bankovního výpisu nebo Hotově. Hotovostní platba se realizuje pomocí tlačítka Zaplat' v hotovosti.



Obrázek 22 – tlačítko Zaplat' v hotovosti

**Program:**

```
procedure TbtnVlozPlatbuClick(Sender: TObject);
begin
    Zaplatit := tbKnihaZaplaceno.Value;
    tbHotovost.Insert;
    tbHotovostIDKniha.Value := tbKnihaID.Value;
    tbHotovostTypFinance.Value := 1;
    tbHotovostDoklad.Value := 2;
    tbHotovostDruh.Value := 2; // Výdej = 2, Příjem = 1, Úkol = 3
    tbHotovostTyp.Value := 4; // PDV = 4
    tbHotovostZkratka.Text := 'Výdej';
    tbHotovostPopis.Text := 'Platba dokladu přijatého';
    tbHotovostCastka.Value := tbKnihaDoplatit.Value;
end;
```

## 4.9. Modul Doklady vydané

### Popis

Modul Doklady vydané slouží k evidenci všech vydaných dokladů, dokladů zaplacených v hotovosti, všech pohledávek vůči odběratelům zboží, či služeb, dále smluvních pohledávek, jako jsou například nájmy, poskytnuté půjčky, mzdy. Tento modul zahrnuje evidenci všech pohledávek a jejich stavu.

Shodně s modulem Deník (tabulka v databázi je stejná), jsou záznamy barevně rozlišeny.

Splnění úkoly jsou rozlišeny datovým polem Splneno. Toto pole je měněno automaticky, např. při úplném zaplacení faktury nebo manuálně. Seznam je řazen podle termínu plnění vzestupně.

### Tabulka

Tabulka pro evidenci dokladů přijatých je shodná s tabulkou pro Deník. Jedná se o tabulku Kniha z databáze data.accdb, v programu Delphi tbKniha. Tabulka je filtrována přes Typ = 2.

### Program:

```
tbKniha.Filter := 'Typ = 2';
```

```
tbKniha.Filtered := true;
```

### UML

UML diagram je shodný s UML diagramem v modulu Deník.

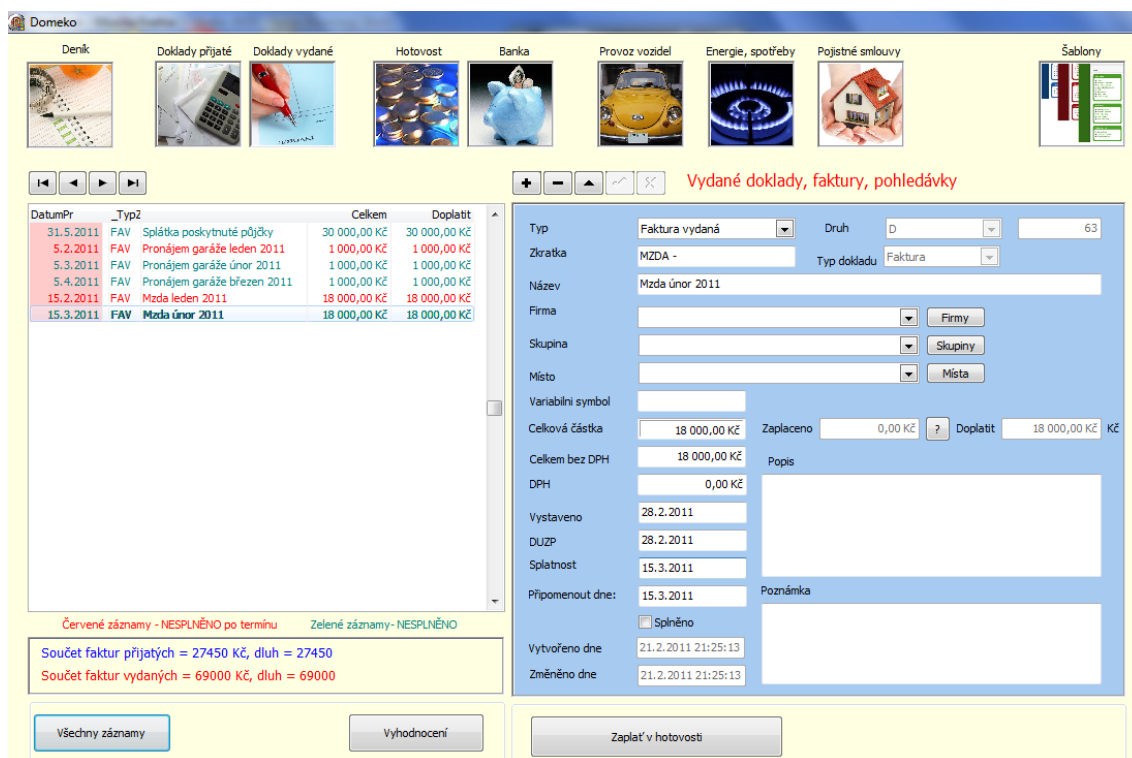
### Modelový příklad

Činnost	Popis
01	Poskytnutá půjčka ve výši 30000 Kč, splatná 31. 6. 2011
02	Pronájem garáže ve výši 1000 Kč, splatný 5. 2. 2011
03	Pronájem garáže ve výši 1000 Kč, splatný 5. 3. 2011
04	Pronájem garáže ve výši 1000 Kč, splatný 5. 4. 2011
05	Mzda ve výši 18000 Kč, splatná 15. 2. 2011
06	Mzda ve výši 18000 Kč, splatná 15. 3. 2011

Tabulka 7 – modelový příklad



## Obrázek modulu



Obrázek 23 – obrazovka Doklady vydané

## Funkčnost modulu

### Nový záznam (Insert)

Po vložení nového záznamu se nastaví počáteční hodnoty

- vložení aktuálního data do pole DatumPr (tbKnihaDatumPr.Text := DateToStr(Now))
- vložení aktuálního data do pole Splatnost (tbKnihaSplatnost.Text := DateToStr(Now))
- vložení aktuálního data do pole Vystaveno (tbKnihaVystaveno.Text := DateToStr(Now))
- vložení aktuálního data do pole DUZP (tbKnihaDUZP.Text := DateToStr(Now);
- vložení hodnoty 0 do pole Netto (tbKnihaNetto.Value := 0)
- vložení hodnoty 0 do pole Brutto (tbKnihaBrutto.Value := 0)
- vložení hodnoty 0 do pole DPH (tbKnihaDPH.Value := 0)
- vložení hodnoty 0 do pole Zaplaceno (tbKnihaZaplaceno.Value := 0)
- vložení hodnoty 0 do pole Doplatit (tbKnihaDoplatit.Value := 0)
- vložení hodnoty Ne do pole Splneno (tbKnihaSplneno.Value := false)

- vložení hodnoty 3 = Úkol do pole Typ z tabulky Typ (tbKnihaTyp.Value := 3)
- vložení textu „Úkol“ do pole Zkratka (tbKnihaPopis.Text := 'FAP -')

Po vložení nového záznamu (Insert) je zobrazena kniha Šablona a je zapnut filtr podle typ 3, tedy úkolů.

```
(tbSablona.Filter := 'Typ = 3')
(tbSablona.Filtered := true)
```

### Součet faktur přijatých a celkový dluh

Součet faktur přijatých = 27450 Kč, dluh = 27450  
 Součet faktur vydaných = 69000 Kč, dluh = 69000

Obrázek 24 – součet a dluh faktur vydaných

**Součet faktur vydaných je realizován pomocí SQL (v Delphi tbObratFPSumOfBrutto) :**

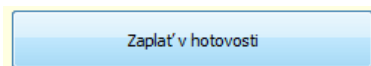
```
SELECT Kniha. Typ, Sum(Kniha.Doplatit) AS SumOfDoplatit
FROM Kniha
GROUP BY Kniha.Typ
HAVING (((Kniha.Typ)=3));
```

**Součet pohledávek u faktur vydaných je realizován pomocí SQL (v Delphi tbDluhFPSumOfDoplatit):**

```
SELECT Kniha.Typ, Sum(Kniha.Brutto) AS SumOfBrutto
FROM Kniha
GROUP BY Kniha.Typ
HAVING (((Kniha.Typ)=3));
```

### Hotovostní platba

Veškeré hotovostní i bezhotovostní platby jsou evidovány v tabulce Finance. Platbu faktury přijaté je možné provést pomocí Bankovního výpisu nebo Hotově. Hotovostní platba se realizuje pomocí tlačítka Zaplat' v hotovosti.



Obrázek 25 – tlačítko Zaplat' v hotovosti

**Program:**

```
procedure TbtnVlozPlatbuClick(Sender: TObject);
begin
    Zaplatit := tbKnihaZaplaceno.Value;
    tbHotovost.Insert;
    tbHotovostIDKniha.Value := tbKnihaID.Value;
    tbHotovostTypFinance.Value := 1;
    tbHotovostDoklad.Value := 2;
    tbHotovostDruh.Value := 1; // Výdej = 2, Příjem = 1, Úkol = 3
    tbHotovostTyp.Value := 2; // PDV = 2
    tbHotovostZkratka.Text := 'Příjem';
    tbHotovostPopis.Text := 'Platba dokladu vydaného';
    tbHotovostCastka.Value := tbKnihaDoplatit.Value;
end;
```

## 4.10. Modul Hotovost

### Popis

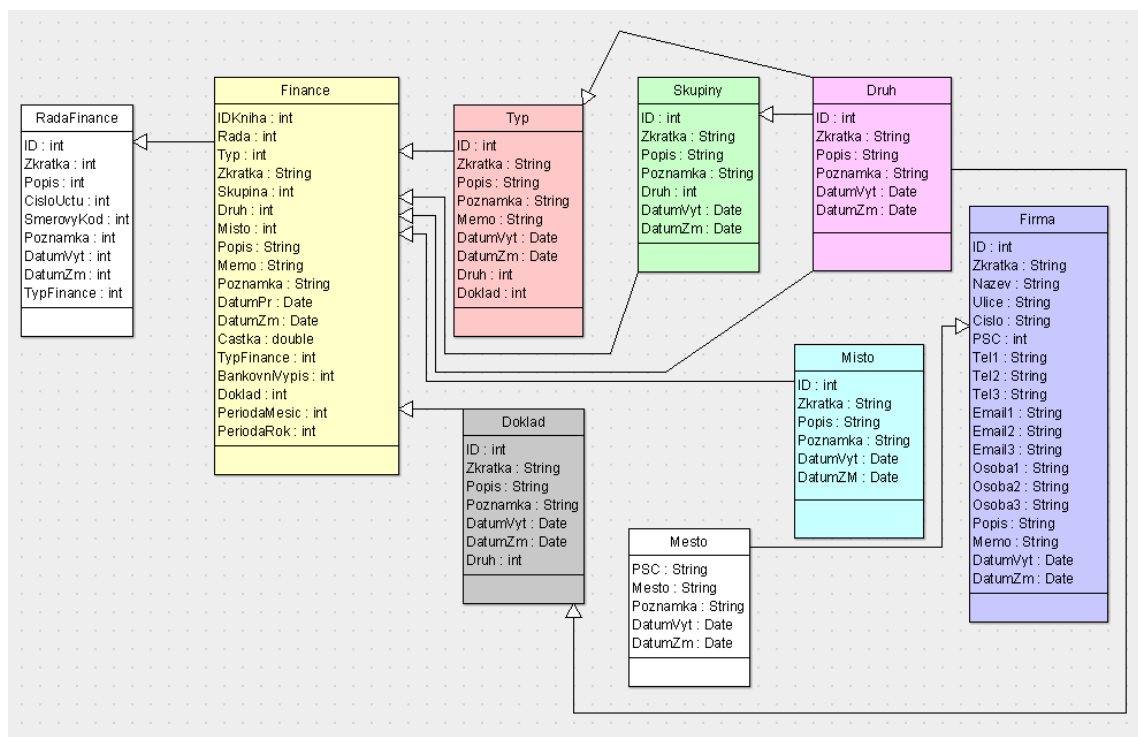
Modul Hotovost slouží k evidenci všech hotovostních finančních pohybů, u kterých uživatel nepožaduje evidenci dokladu. Modul zahrnuje jak příjmy, tak výdaje. Modul dále rozlišuje jednotlivé pokladny. Lze definovat například osobu, nebo místo hotovosti.

### Tabulka

Pro evidenci příjmů a výdajů jsou využity tabulky RadaFinance (v Delphi tbHotovostRa) a Finance (v Delphi tbHotovost) z databáze data.accdb. Tabulky RadaFinance a Finance jsou spojeny.

Tabulka RadaFinance je filtrována přes pole TypFinance = 1 (Příjmy).

### UML



Obrázek 26 – UML modulu Hotovost

## Modelový příklad

Činnost	Popis
01	Večeře v restauraci Bohemia dne 15. 1. 2011, útrata 750 Kč
02	Vstupné do kina dne 17. 1. 2011, 250 Kč
03	Vstupné Aquapark 25. 1. 2011, 450 Kč
04	Nákup potravin v Billa dne 31. 1. 2011, 530 Kč
05	Nákup potravin v InterSpar dne 22. 1. 2011, 850 Kč
06	Dar od rodičů 22. 2. 2011, 5000 Kč

Tabulka 8 – modelový příklad

## Obrázek modulu

Druh	Číslo
Příjem	121

Typ
Platba přijatá v hotovosti

Datum platby: 22.2.2011

Zkratka: Příjem

Popis: Dar od rodičů

Castka: 5 000,00 Kč

Ulož platbu      Zruš platbu

	Vybraná pokladní řada	Všechny řady
Příjmy	5000 Kč	5000 Kč
Výdaje	2830 Kč	2830 Kč
Zůstatek	2170 Kč	2170 Kč

Obrázek 27 – obrazovka modulu Hotovost

## Funkčnost modulu

### Nový záznam (Insert) - výdej

Po vložení nového záznamu se nastaví počáteční hodnoty

tbHotovost.Insert;

```

tbHotovostDruh.Value := 2; // 1= příjem, 2 = výdej
tbHotovostTyp.Value := 4;
tbHotovostZkratka.Text := 'Výdej';
tbHotovostPopis.Text := 'Obecný výdej';
tbHotovostTypFinance.Value := 1;
tbHotovostDoklad.Value := 2;

```

### Nový záznam (Insert) – příjem

Po vložení nového záznamu se nastaví počáteční hodnoty

```

tbHotovost.Insert;
tbHotovostDruh.Value := 1; // 1= příjem, 2 = výdej
tbHotovostTyp.Value := 2;
tbHotovostZkratka.Text := 'Příjem';
tbHotovostPopis.Text := 'Obecný příjem';
tbHotovostTypFinance.Value := 1;
tbHotovostDoklad.Value := 2;
edtCastka.SetFocus;

```

### Součty příjmů a výdajů

	Vybraná pokladní řada	Všechny řady
Příjmy	5000 Kč	5000 Kč
Výdaje	2830 Kč	2830 Kč
Zůstatek	2170 Kč	2170 Kč

Obrázek 28 – součet příjmů a výdajů

#### Součet příjmů (qrySumaPrijemCelkem)

```

SELECT Sum(Finance.Castka) AS SumOfCastka, Finance.Druh, Finance.TypFinance
FROM Finance
GROUP BY Finance.Druh, Finance.TypFinance
HAVING (((Finance.Druh)=1) AND ((Finance.TypFinance)=1));

```

#### Součet příjmů ve vybrané pokladní řadě (qrySumaPrijemRadaCelkem)

```

SELECT Sum(Finance.Castka) AS SumOfCastka, Finance.Druh, Finance.TypFinance,
Finance.Rada
FROM Finance
GROUP BY Finance.Druh, Finance.TypFinance, Finance.Rada
HAVING (((Finance.Druh)=1) AND ((Finance.TypFinance)=1));

```

#### Součet výdajů (qrySumaVydejCelkem)

```

SELECT Sum(Finance.Castka) AS SumOfCastka, Finance.Druh, Finance.TypFinance
FROM Finance
GROUP BY Finance.Druh, Finance.TypFinance
HAVING (((Finance.Druh)=2) AND ((Finance.TypFinance)=1));

```

**Součet výdajů ve vybrané pokladní řadě (qrySumaVydejRadaCelkem)**

```
SELECT Sum(Finance.Castka) AS SumOfCastka, Finance.Druh, Finance.TypFinance,  
Finance.Rada  
FROM Finance  
GROUP BY Finance.Druh, Finance.TypFinance, Finance.Rada  
HAVING (((Finance.Druh)=2) AND ((Finance.TypFinance)=1));
```

## 4.11. Modul Banka

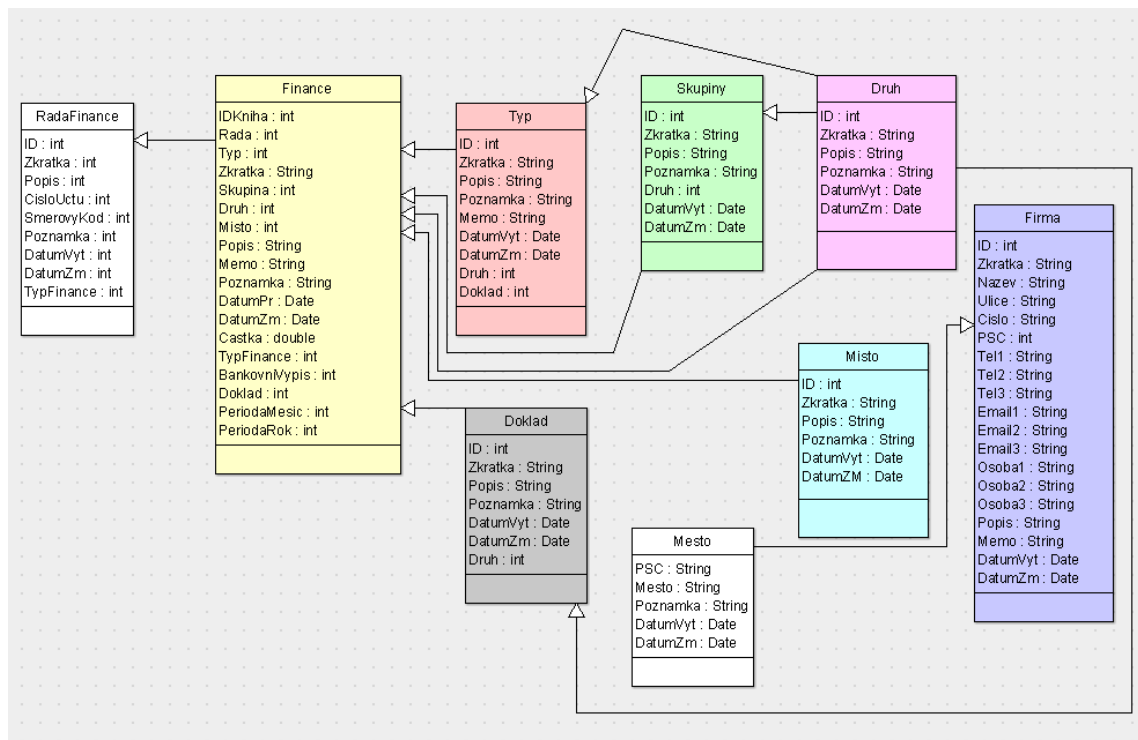
### Popis

Modul Banka slouží k evidenci všech bankovních finančních pohybů. V modulu jsou řešeny platby dokladů přijatých, dokladů vydaných a dalších bankovních operací. Modul tedy zahrnuje příjmy i výdaje. Modul dále rozlišuje jednotlivé banky.

### Tabulka

Pro evidenci příjmů a výdajů jsou využity tabulky RadaFinance (v Delphi tbBankaRada) a Finance (v Delphi tbRada) z databáze data.accdb. Tabulky RadaFinance a Finance jsou spojeny. Tabulka RadaFinance je filtrována přes pole TypFinance = 2.

### UML



Obrázek 29 – UML modulu Banka

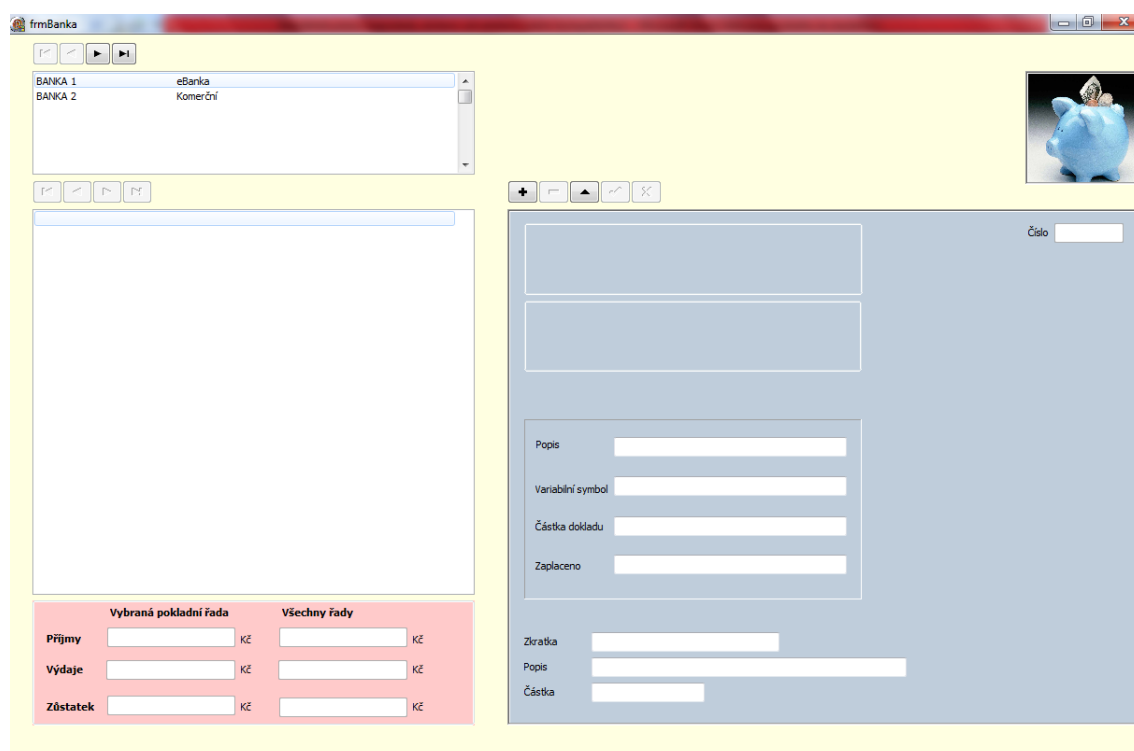


## Modelový příklad

Činnost	Popis
01	Večeře v restauraci Bohemia dne 15. 1. 2011, útrata 750 Kč
02	Vstupné do kina dne 17. 1. 2011, 250 Kč
03	Vstupné Aquapark 25. 1. 2011, 450 Kč
04	Nákup potravin v Billa dne 31. 1. 2011, 530 Kč
05	Nákup potravin v InterSpar dne 22. 1. 2011, 850 Kč
06	Dar od rodičů 22. 2. 2011, 5000 Kč

Tabulka 9 – modelový příklad

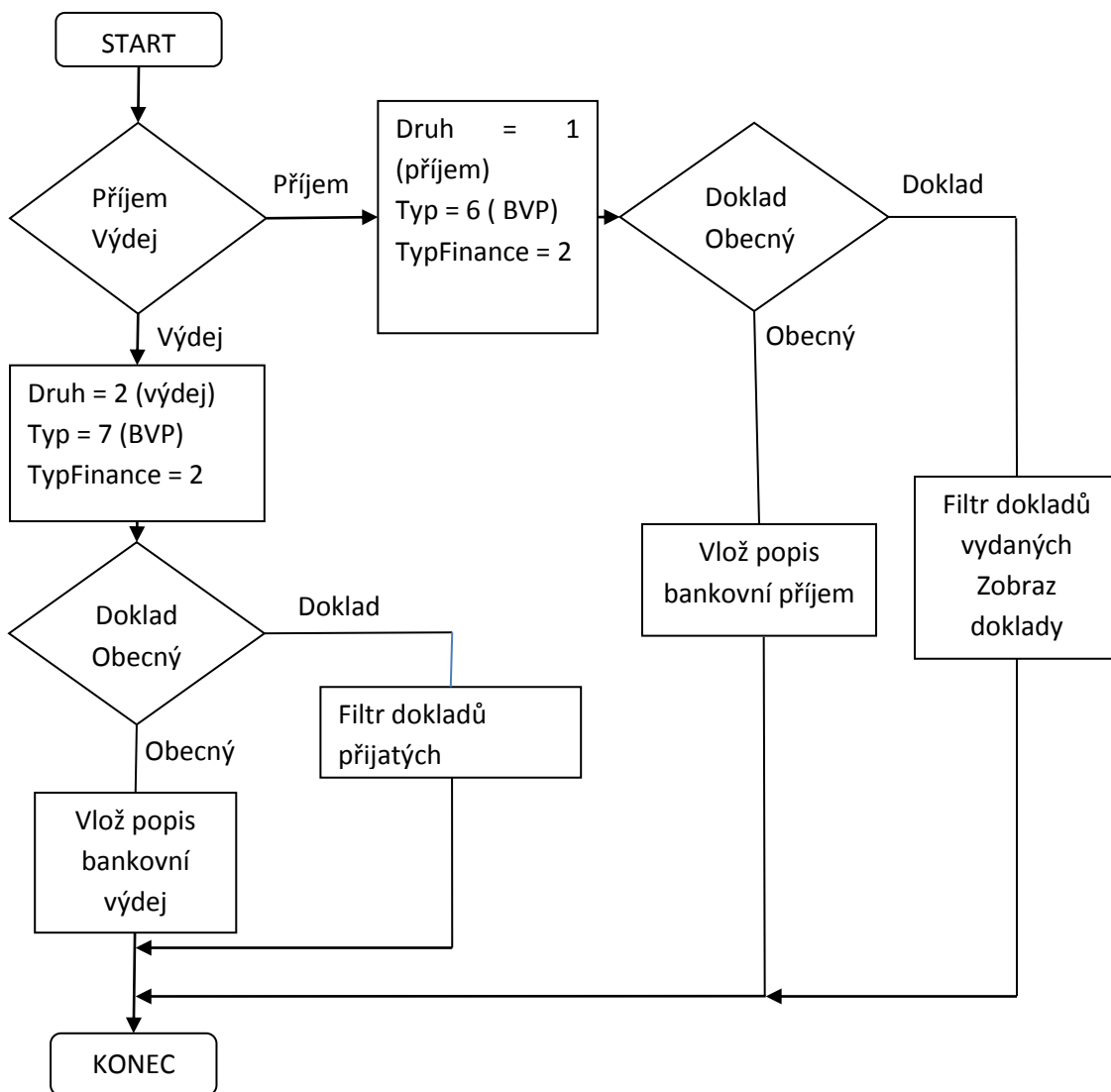
## Obrázek modulu



Obrázek 30 – obrazovka modulu Banka

## Funkčnost modulu

### Nový záznam (Insert)



Obrázek 31 – vývojový diagram pro Nový záznam

### Bankovní výdej

```
procedure TfrmBanka.btnVydejClick(Sender: TObject);  
begin
```

```
    tbBankaDruh.Value := 2;
```

```
    tbBankaTyp.Value := 7;
```

```

tbBankaTypFinance.Value := 2; // 2 = banka
tbBankaKnihaDoplatit.Filter := ('Typ = 1');
Stav := 1
end;

```

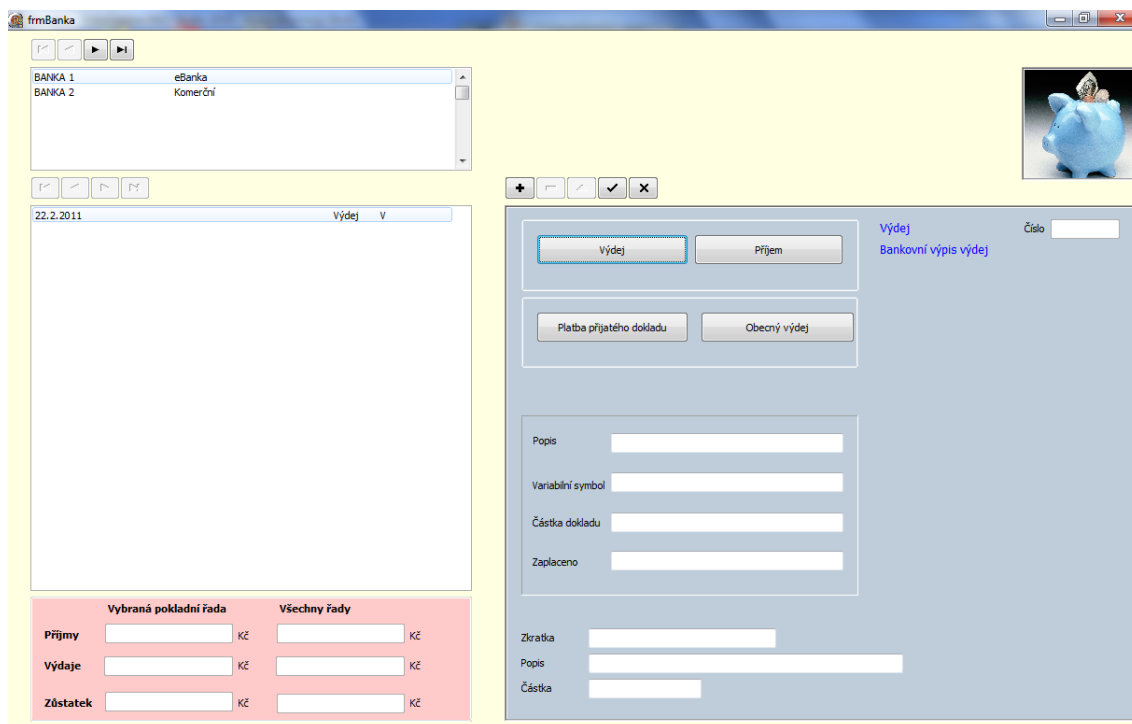
### Bankovní příjem

```

procedure TfrmBanka.btnPrijemClick(Sender: TObject);
begin
    tbBankaDruh.Value := 1;
    tbBankaTyp.Value := 6;
    tbBankaTypFinance.Value := 2; // 2 = banka

    gboTyp.Visible := true;
    tbBankaKnihaDoplatit.Filter := ('Typ = 3');
    Stav := 2;
end;

```



Obrázek 32 – obrazovka modulu Banka

### Platba přijatého ( resp. Vydaného) dokladu

```

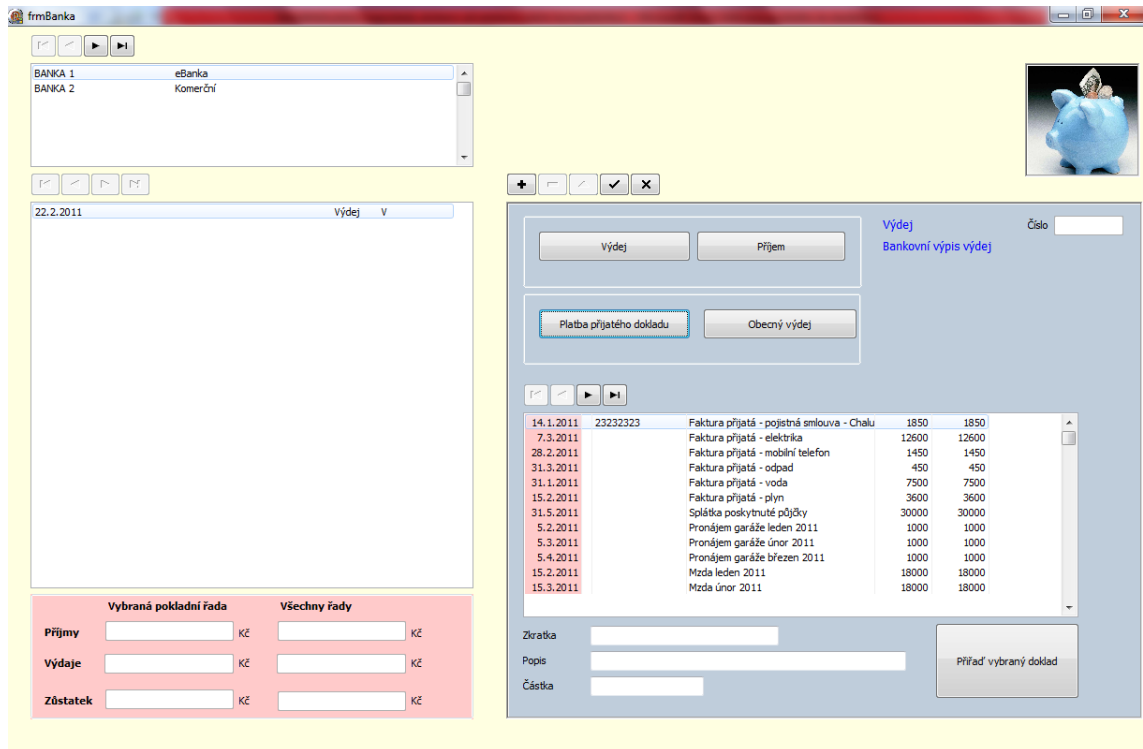
procedure TfrmBanka.btnDokladClick(Sender: TObject);
begin

```

```

tbBankaKnihaDoplatit.Active := False;
tbBankaKnihaDoplatit.Active := True;
nvgKniha.Visible := True;
grdKniha.Visible := True;
btnPriradDoklad.Visible := True;
end;

```



Obrázek 33 – obrazovka modulu Banka

### Obecný výdej

```

procedure TfrmBanka.btnObecnyClick(Sender: TObject);
begin
    case Stav of
    1:    begin
            tbBankaZkratka.Text := 'Výdej';
            tbBankaPopis.Text := 'Bankovní výdej';
        end;
    2:    begin
            tbBankaZkratka.Text := 'Příjem';
            tbBankaPopis.Text := 'Bankovní příjem';
        end;
    end;
end;

```

```
edtCastka.SetFocus;  
end;
```

### Nový záznam (Insert) – příjem

Po vložení nového záznamu se nastaví počáteční hodnoty

```
tbHotovost.Insert;  
tbHotovostDruh.Value := 1; // 1= příjem, 2 = výdej  
tbHotovostTyp.Value := 2;  
tbHotovostZkratka.Text := 'Příjem';  
tbHotovostPopis.Text := 'Obecný příjem';  
tbHotovostTypFinance.Value := 1;  
tbHotovostDoklad.Value := 2;  
edtCastka.SetFocus;
```

### Součty příjmů a výdajů

	Vybraná pokladni řada	Všechny řady
Příjmy	18 000,00 Kč	18 000,00 Kč
Výdaje	9 370,00 Kč	9 370,00 Kč
Zůstatek	8630 Kč	8630 Kč

Obrázek 34 – součet příjmů a výdajů

#### Součet příjmů (qryBankaPrijemCelkem)

```
SELECT Sum(Finance.Castka) AS SumOfCastka, Finance.Druh, Finance.TypFinance  
FROM Finance  
GROUP BY Finance.Druh, Finance.TypFinance  
HAVING (((Finance.Druh)=1) AND ((Finance.TypFinance)=2));
```

#### Součet příjmů ve vybrané bankovní řadě (qryBankaPrijemRadaCelkem)

```
SELECT Sum(Finance.Castka) AS SumOfCastka, Finance.Druh, Finance.TypFinance,  
Finance.Rada  
FROM Finance  
GROUP BY Finance.Druh, Finance.TypFinance, Finance.Rada  
HAVING (((Finance.Druh)=1) AND ((Finance.TypFinance)=2));
```

#### Součet výdajů (qryBankaVydejCelkem)

```
SELECT Sum(Finance.Castka) AS SumOfCastka, Finance.Druh, Finance.TypFinance  
FROM Finance  
GROUP BY Finance.Druh, Finance.TypFinance  
HAVING (((Finance.Druh)=2) AND ((Finance.TypFinance)=2));
```

**Součet výdajů ve vybrané bankovní řadě (qryBankaVydejRadaCelkem)**

```
SELECT Sum(Finance.Castka) AS SumOfCastka, Finance.Druh, Finance.TypFinance,  
Finance.Rada  
FROM Finance  
GROUP BY Finance.Druh, Finance.TypFinance, Finance.Rada  
HAVING (((Finance.Druh)=2) AND ((Finance.TypFinance)=2));
```

## 4.12. Modul Provoz vozidel

### Popis

Modul Provoz vozidel slouží k evidenci vozidel, jízd, tankování a spotřeby.

### Tabulka

Pro evidenci vozidel je využita tabulka Vozidla. Jízdy, tankování, spotřeba jsou zapisovány do tabulky Jizdy. Obě tabulky jsou z databáze data.accdb.

Tabulka Vozidla

Pole	Datový typ	Popis
ID	Integer	Automaticky generované číslo
Typ	Číslo	Tabulka Typ, hodnota 8
Skupina	Číslo	Tabulka Skupina, hodnota 6
Druh	Číslo	Tabulka Druh, hodnota 6
Místo	Číslo	
Zkratka	Text	Krátký popis vozidla, např. SPZ
Popis	Text	Podrobný popis vozidla
Poznamka	Memo	Poznámky k vozidlu
DatumVyt	Datum a čas	Automaticky generované datum vytvoření
DatumZm	Datum a čas	Automaticky generované datum změny
Tachometr	Číslo	Aktuální stav tachometru.
PocatecniStav	Číslo	Počáteční stav tachometru
Nakup	Datum a čas	Datum nákupu vozidla
CelkemKm	Číslo	Celkový počet najetých km
CelkemL	Číslo	Celková spotřeba pohonných hmot v litrech
Spotreba	Číslo	Celková průměrná spotřeba

Tabulka 10 – tabulka Vozidla

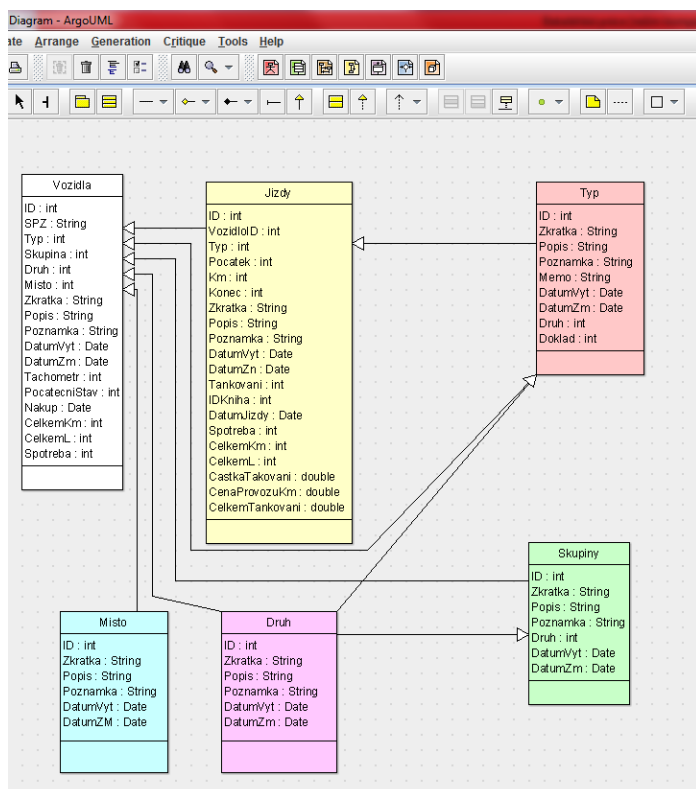
Tabulka Jízdy

Pole	Datový typ	Popis
ID	Integer	Automaticky generované číslo
VozidloID	Číslo	Spojení do tabulky Vozidla
Typ	Číslo	Tabulka Typ, nevyužito
Pocatek	Číslo	Stav tachometru na začátku jízdy
Km	Číslo	Počet najetých km

Konec	Číslo	Stav tachometru na konci jízdy
Zkratka	Text	Krátký popis jízdy
Popis	Text	Podrobný popis jízdy
Poznamka	Memo	Poznámky k jízdě
DatumVyt	Datum a čas	Datum vytvoření záznamu
Tankovani	Číslo	Množství tankování pohonných hmot v litrech
IDKniha	Číslo	
DatumJizdy	Datum a čas	Datum jízdy
Spotreba	Číslo	Propočítaná spotřeba
CelkemKm	Číslo	Počet najetých km celkem
CelkemL	Číslo	Množství spotřebovaných pohonných hmot celkem
CastkaTankovani	Měna	Částka za tankování, přeneseno z knihy
CenaProvozuKm	Měna	Vypočtená cena provozu na km
CenaTankovani	Měna	

Tabulka 11 – tabulka Jizdy

## UML



Obrázek 35 – UML modulu Provoz vozidel



## Modelový příklad

### Vozidla

Činnost	Popis
01	Vozidlo BMW
02	Clio
03	Motorka

Tabulka 12 – modelový příklad Vozidla

### Jízdy

Činnost	Popis
BMW	
01	1. 1. 2011 Jízda chalupa, 70km
02	8. 1. 2011 Jízda chalupa, 70km
03	11. 1. 2011 Jízda chalupa, 70km
04	15. 1. 2011 Jízda lyže Rakousko, 620km
05	22. 1. 2011 Jízda chalupa, 70km
06	24. 1. 2011 Jízda chalupa, 70km

Tabulka 13 – modelová příklad Jízdy

## Obrázek modulu

Obrázek 36 – obrazovka Provoz vozidel

## Funkčnost modulu

### Vozidla - nový záznam

```
procedure TtbVozidlaAfterInsert(DataSet: TDataSet);
begin
    tbVozidlaDatumVyt.Text := DateTimeToStr(Now);
    tbVozidlaTachometr.Value := 0;
    tbVozidlaTyp.Value := 8;
    tbVozidlaDruh.Value := 6;
    tbVozidlaSkupina.Value := 6;
    tbVozidlaCelkemKm.Value := 0;
    tbVozidlaCelkemL.Value := 0;
    tbVozidlaSpotreba.Value := 0;
end;
```

### Jízdy - nový záznam

```
procedure TtbJizdyAfterInsert(DataSet: TDataSet);
begin
    tbJizdyDatumVyt.Text := DateTimeToStr(Now);
    tbJizdyZkratka.Text := 'Jízda';
    tbJizdyPopis.Text := 'Jízda';
    tbJizdyTankovani.Value := 0;
    tbJizdyIDKniha.Value := 0;
    tbJizdyCastkaTankovani.Value := 0;
    tbJizdyDatumJizdy.Text := DateToStr(Now);
    if (tbVozidlaTachometr.Value = 0) OR (tbVozidlaTachometr.Text = '') then
        tbJizdyPocatek.Value := tbVozidlaPocatecniStav.Value
    else
        tbJizdyPocatek.Value := tbVozidlaTachometr.Value;
end;
```

### Přepoččet spotřeby

```
procedure TfrmAuta.btnPrepocitejClick(Sender: TObject);
    Var
        StavTach: Integer;
        SumPHM: Currency;
        SumCastka: Currency;
        SumKm: Integer;
        Spotreba: Currency;
        CenaProvozu: Currency;
        SumCastkaTank: Currency;
        I: Integer;
begin
```

```

tbJizdy.First;
StavTach := tbVozidlaPocatecniStav.Value;
SumPhm := 0;
SumKm := 0;
SumCastkaTank := 0;
Spotreba := 0;
CenaProvozu := 0;
I:= 0;
repeat
I := I+1;
tbJizdy.Edit;
// Přepoččet spotřeby - musí být dříve, než se přičte aktuální stav - heslo natankuješ a
pak jedeš.
// Spotřeba počítána za předpokladu, že bude tankováno při prázdné nádrži
//
if (I = 1) then Spotreba := 0
else
if (tbJizdyTankovani.Value <> 0) then
begin
Spotreba := ((SumPHM/SumKM)*100);
CenaProvozu := (SumCastkaTank/SumKM);
end;
//
tbJizdySpotreba.Value := Spotreba;
tbJizdyCenaProvozuKm.Value := CenaProvozu;
//Přepoččet km
SumKm := SumKm + tbJizdyKm.Value;
tbJizdyCelkemKm.Value := SumKM;
//Přepoččet PHM
SumPHM := SumPHM + tbJizdyTankovani.Value;
tbJizdyCelkemL.Value := SumPHM;
//
//Přepoččet částek
SumCastkaTank := SumCastkaTank + tbJizdyCastkaTankovani.Value;
tbJizdyCelkemTankovani.Value := SumCastkaTank;
//
tbJizdyPocatek.Value := StavTach;
tbJizdyKonec.Value := tbJizdyPocatek.Value + tbJizdyKm.Value;
StavTach := tbJizdyKonec.Value;
tbJizdy.Post;
tbJizdy.Next;

```

```
until (tbJizdy.Eof);  
//  
tbVozidla.Edit;  
tbVozidlaTachometr.Value := StavTach;  
tbVozidlaCelkemKm.Value := SumKm;  
tbVozidlaCelkemL.Value := SumPHM;  
tbVozidlaSpotreba.Value := Spotreba;  
tbVozidla.Post;  
end;
```

### **Kniha dokladů za PHM**

```
SELECT Kniha.ID, Kniha.Zkratka, Kniha.Skupina, Kniha.Popis, Kniha.DatumPr, Kniha.Brutto,  
Kniha.Vystaveno, Kniha.VariabilniSymbol  
FROM Kniha LEFT JOIN Jizdy ON Kniha.[ID] = Jizdy.[IDKniha]  
WHERE (((Kniha.Skupina)=5)  
AND ((Jizdy.IDKniha) Is Null));
```

## 4.13. Modul Spotřeba

### Popis

Modul Spotřeba slouží k evidenci spotřeb, například vody, elektřiny, plynu. Spotřeby lze rozdělit jak podle typu, tak i podle skupiny, či místa. Evidenci spotřeb lze provést jak pro osobní kontrolu, tak i pro potřeby vyúčtování.

### Tabulka

Pro evidenci typu spotřeby je využita tabulka Spotreba. Samotné odečty jsou zaznamenávány v tabulce SpotrebaDetail. Obě tabulky jsou z databáze data.acddb.

Pole	Datový typ	Popis
ID	Automatické číslo	Automaticky generované číslo
Zkratka	Text	Krátký popis
Popis	Text	Podrobný popis
Typ	Číslo	Tabulka Typ
Skupina	Číslo	Tabulka Skupina
Druh	Číslo	Tabulka Druh
Místo	Číslo	Tabulka Místo
DatumPorizeni	Datum a čas	Datum pořízení
Jednotka	Text	Měrná jednotka
CelkemJed	Číslo	
Celkem	Číslo	
Spotreba	Číslo	
Poznamka	Memo	
DatumZm	Datum a čas	
DatumVyt	Datum a čas	

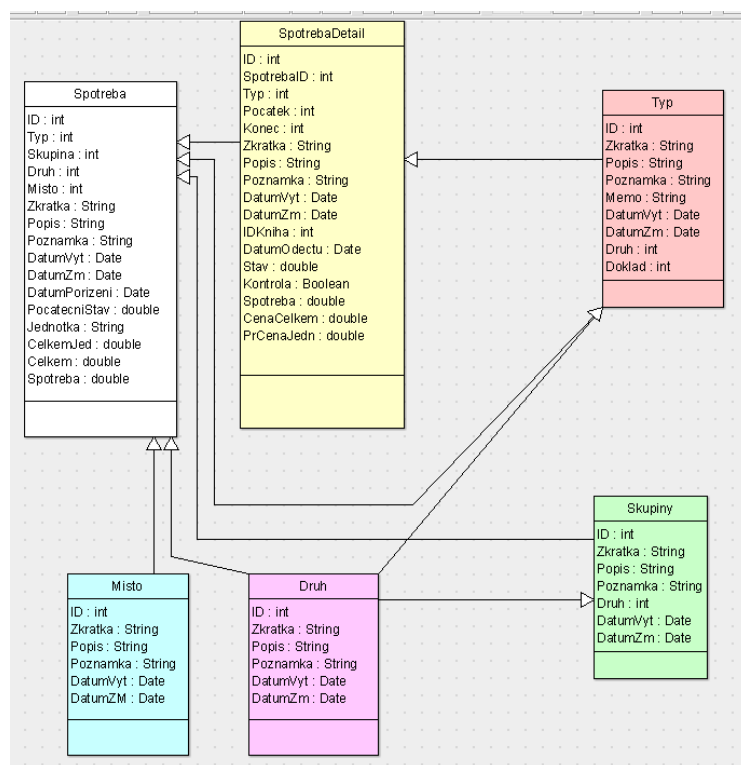
Tabulka 14 – tabulka Spotreba

Pole	Datový typ	Popis
ID	Automatické číslo	
SpotrebaID	Číslo	Spojení do tabulky Spotreba
DatumOdectu	Datum a čas	Datum odečtu
Pocatek	Číslo	Počáteční stav, pro potřeby výpočtu
Typ	Číslo	
Zkratka	Text	Krátký popis

Konec	Číslo	
Popis	Text	Podrobný popis
Stav	Číslo	Stav při odečtu
Kontrola	Ano/Ne	Ano = kontrolní odečet / Ne = účetní odečet
Poznamka	Memo	Poznámka
Spotreba	Číslo	
CenaCelkem	Měna	Cena
DatumVyt	Datum a čas	Datum vytvoření záznamu
DatumZm	Datum a čas	Datum změny záznamu
IDKniha	Číslo	Doklad z tabulky Kniha
PrCenaJed	Měna	Průměrná cena za jednotku

Tabulka 15 – tabulka SpotrebaDetail

## UML



Obrázek 37 – modul Spotřeby

## Modelový příklad

### Spotřeba

Činnost	Popis
Typ spotřeby	
01	Elektrika, Malebná ulice
02	Voda, Malebná ulice

Tabulka 16 – modelový příklad Spotřeba

### SpotřebaDetail

Činnost	Popis
01	Elektrika, Malebná ulice, kontrolní odečet 30. 6. 2010, Stav 180
02	Elektrika, Malebná ulice, účetní odečet 31. 12. 2010, stav 480
03	Elektrika, Malebná ulice, kontrolní odečet 28. 1. 2011, stav 520
04	Voda, Malebná ulice, účetní odečet 31. 12. 2010, stav 162

Tabulka 17 – modelový příklad SpotřebaDetail

## Obrázek modulu

The screenshot shows the 'Spotřeba' software interface. It features a main form with several sections:

- Navigation:** A set of arrows for navigating between records.
- Table:** A table with columns 'Popis' and 'Místo'. It contains two rows: 'Elektrika' at 'Dům Malebná' and 'Voda malebná' at 'Dům Malebná'.
- Form Fields:**
  - Zkratka: ELEKTRIKA M, Číslo: 1, Koupeno dne: 1.1.2010
  - Popis: Elektrika, Počáteční stav: 100
  - Druh: Spotřeba, Druh: (dropdown), Jednotka: kWh
  - Typ: Spotřeba, Typ: (dropdown)
  - Skupina: spotřeba - elektrika, Skupiny: (dropdown)
  - Místo: Dům Malebná, Místa: (dropdown)
- Legend:**
  - Černé záznamy - kontrolní odečet
  - Červené záznamy - účetní odečty
  - Modré záznamy - účetní odečty s daňovým dokladem
- Table of Readings:**

Datum odečtu	Popis	Stav	Cena/jednotku
30.6.2010	Odečet - Elektrika - Dům Malebná	180	0,00 Kč
31.12.2010	Odečet - Elektrika - Dům Malebná	480	65,79 Kč
28.1.2011	Odečet - Elektrika - Dům Malebná	520	142,86 Kč
- Form Fields (Right):**
  - Datum odečtu: 30.6.2010, Číslo: 1
  - Zkratka: Odečet - ELEKTRIKA M
  - Popis: Odečet - Elektrika - Dům Malebná
  - Stav: 180,  kontrolní odečet (nezaškrtnuto - účetní odečet)
  - Datum: (input), Číslo dokladu: 1
  - Zkratka: (input)
  - Variabilní symbol: (input)
  - Částka: (input)
  - Popis: (input)
- Buttons:** 'Přepočítej' at the bottom.

Obrázek 38 – obrázek Spotřeba

## Funkčnost modulu

### Typ spotřeby – Nový záznam

#### Program:

```
procedure TtbSpotrebaAfterInsert(DataSet: TDataSet);
begin
    tbSpotrebaTyp.Value := 9; // 9 = spotřeba
    tbSpotrebaDruh.Value := 7; // 7 = spotřeba
    tbSpotrebaPocatecniStav.Value := 0;
    tbSpotrebaDatumPorizeni.Text := DateToStr(Now);
    tbSpotrebaDatumVyt.Text := DateToStr(Now);
end;
```

### Spotřeba detail – Nový záznam

#### Program:

```
procedure TtbSpotrebaDetailAfterInsert(DataSet: TDataSet);
begin
    tbSpotrebaDetailDatumVyt.Text := DateToStr(Now);
    tbSpotrebaDetailDatumOdectu.Text := DateToStr(Now);
    tbSpotrebaDetailKontrola.Value := False;
    tbSpotrebaDetailZkratka.Text := 'Odečet - ' + tbSpotrebaZkratka.Text;
    tbSpotrebaDetailPopis.Text := 'Odečet - ' + tbSpotrebaPopis.Text + ' - '
    '+tbSpotreba_MistoPopis.Text;
end;

grdSeznam

procedure TfrmSpotreby.grdSeznamDrawColumnCell(Sender: TObject;
    const Rect: TRect; DataCol: Integer; Column: TColumn; State: TGridDrawState);
begin
    // Pokud Kontrola = False (účetní odečet) a není přiřazen doklad
    if (tbSpotrebaDetailKontrola.Value = False) AND (tbSpotrebaDetailIDKniha.Value = 0 ) then
        grdSeznam.Canvas.Font.Color := clRed
    else
        // Pokud Kontrola = False (účetní odečet) a je přiřazen doklad
        if (tbSpotrebaDetailKontrola.Value = False) AND (tbSpotrebaDetailIDKniha.Value <> 0 )
        then grdSeznam.Canvas.Font.Color := clBlue
        else
            // V ostatních případech
            grdSeznam.Canvas.Font.Color := clBlack;
            grdSeznam.DefaultDrawColumnCell(Rect, DataCol, Column, State);
end;
```

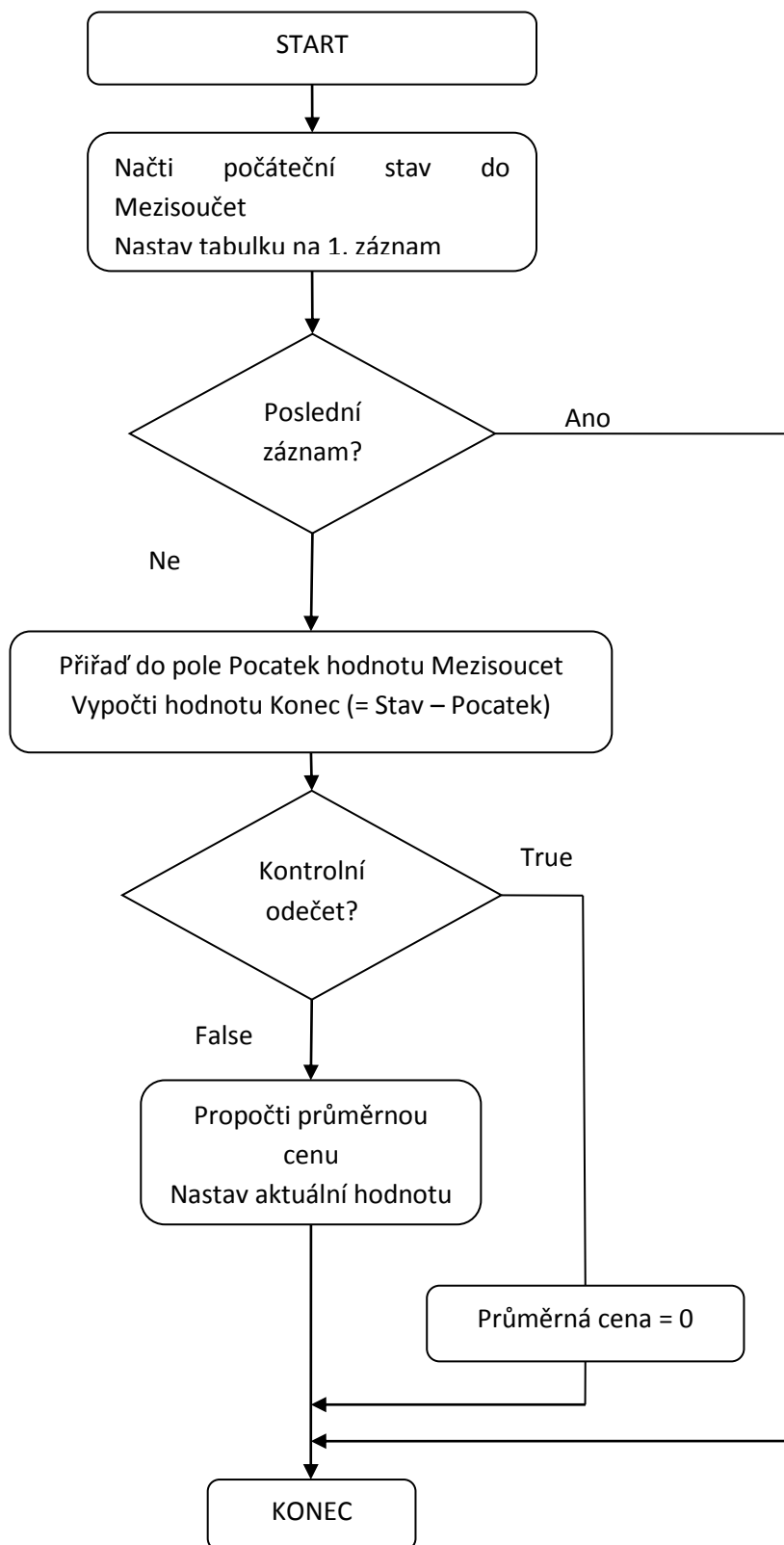


end;

### **funkce Přepočítej (btnPrepocitej)**

#### **Program:**

```
procedure TfrmSpotreby.btnPrepocitejClick(Sender: TObject);
Var
Mezisoucet : Single;
begin
Mezisoucet:= tbSpotrebaPocatecniStav.Value;
tbSpotrebaDetail.First;
repeat
    tbSpotrebaDetail.Edit;
    tbSpotrebaDetailPocatek.Value := Mezisoucet;
    tbSpotrebaDetailKonec.Value := tbSpotrebaDetailStav.Value –
    tbSpotrebaDetailPocatek.Value;
    if tbSpotrebaDetailKontrola.Value = False
    then
    begin
    tbSpotrebaDetailPrCenaJed.Value := tbSpotrebaDetailCenaCelkem.Value/
    tbSpotrebaDetailKonec.Value;
    Mezisoucet :=tbSpotrebaDetailKonec.Value;
    end
    else tbSpotrebaDetailPrCenaJed.Value := 0;
    tbSpotrebaDetail.Post;
    tbSpotrebaDetail.Next;
until tbSpotrebaDetail.Eof;
end;
```



Obrázek 39 – vývojový diagram funkce Přepočítej

## 4.14. Modul Pojištění

### Popis

Modul Pojištění slouží k evidenci pojistných předmětů, např. Dům Malebná ul. Pojištění domu a pojistných smluv.

### Tabulka

Pro evidenci pojistných předmětů se používá tabulka Pojisteni. Pojistné smlouvy jsou zaznamenávány v tabulce PojisteniDetail. Obě tabulky jsou z databáze data.accdb.

Pole	Datový typ	Popis
ID	Automatické číslo	Automaticky generované číslo
Zkratka	Text	Krátký popis
Popis	Text	Podrobný popis
Typ	Číslo	Tabulka Typ
Skupina	Číslo	Tabulka Skupina
Druh	Číslo	Tabulka Druh
Místo	Číslo	Tabulka Místo
Poznamka	Memo	Poznámka
DatumZm	Datum a čas	Datum změny
DatumVyt	Datum a čas	Datum vytvoření
Pojisteno	Ano/Ne	Ano = pojištěno / Ne = Nepojištěno

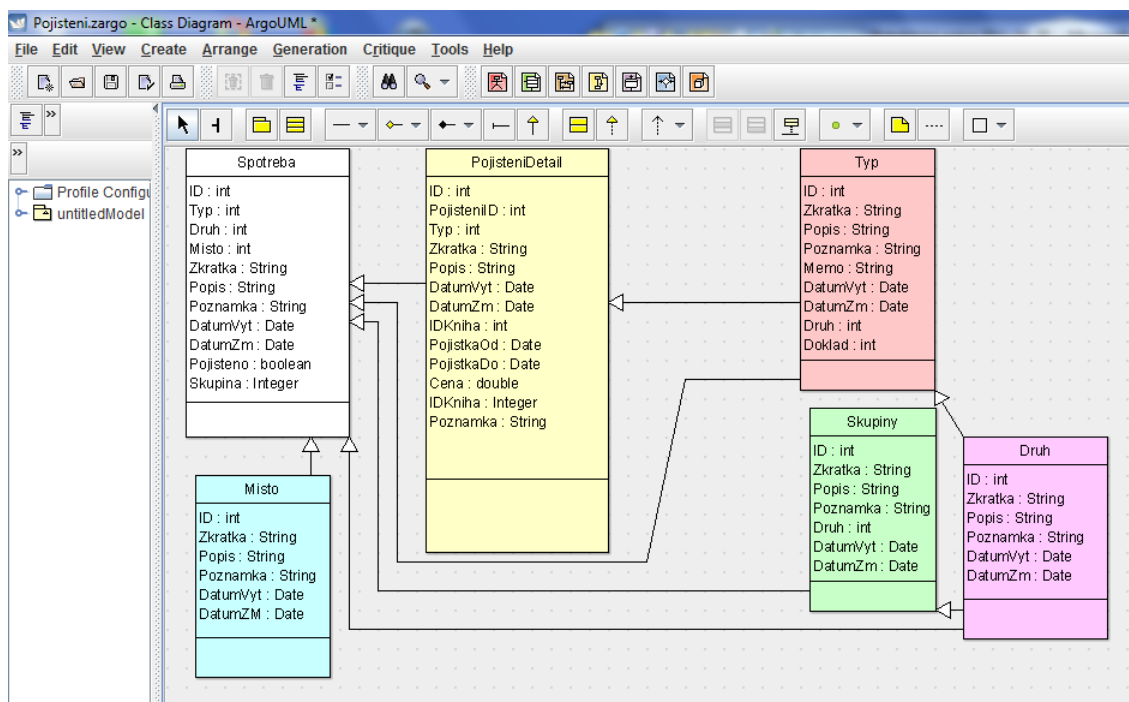
Tabulka 18 – tabulka Pojisteni

Pole	Datový typ	Popis
ID	Automatické číslo	
PojisteniID	Číslo	Spojení do tabulka Pojisteni
DatumPojisteni	Datum a čas	Datum odečtu
Typ	Číslo	
Zkratka	Text	Krátký popis
Popis	Text	Podrobný popis
PojistkaOd	Datum a čas	Pojištěno od
PojistkaDo	Datum a čas	Pojištěno do
Poznamka	Memo	Poznámka
Cena	Měna	Měna
IDKniha	Číslo	

DatumVytv	Datum a čas	Datum vytvoření záznamu
DatumZm	Datum a čas	Datum změny záznamu

Tabulka 19 – tabulka PojisteniDetail

## UML



Obrázek 40 – UML modulu Pojištění

## Modelový příklad

### Pojistný předmět

Činnost	Popis
Typ předmětu	
01	Dům Malebná ulice, pojištění objektu
02	Dům Malebná ulice, pojištění domácnosti

Tabulka 20 – modelový příklad Pojisteni

### Pojistné smlouvy

Činnost	Popis
BMW	
01	Dům Malebná ulice, pojištění objektu – pojištění rok 2010
02	Dům Malebná ulice, pojištění objektu – pojištění rok 2011

Tabulka 21 – modelový příklad Pojištění detail

## Obrázek modulu

Pojistná smlouva	pojištěno od	pojištěno do
Pojistná smlouva - dům Malebná	1.1.2010	31.12.2010
Pojistná smlouva - dům Malebná	1.1.2011	31.12.2011

Obrázek 41 – obrazovka modulu Pojištění

## Funkčnost modulu

### Tabulka Pojistky – Nový záznam

```
procedure TtbPojistkyAfterInsert(DataSet: TDataSet);
```

```
begin
```

```
    tbPojistkyTyp.Value := 10;
```

```
    tbPojistkyDruh.Value := 8;
```

```
    tbPojistkyDatumPorizeni.Text := DateToStr(Now);
```

```
end;
```

### Tabulka PojistkyDetail – Nový záznam

```
procedure TtbPojistkyDetailAfterInsert(DataSet: TDataSet);
```

```
begin
```

```
    tbPojistkyDetailDatumPojisteni.Text := DateToStr(Now);
```

```
    tbPojistkyDetailZkratka.Text := 'POJ.SML';
```

```
    tbPojistkyDetailPopis.Text := 'Pojistná smlouva - ' + tbPojistky_MistoPopis.Text;
```

```
    tbPojistkyDetailPojistkaOd.Text := DateToStr(Now);
```

```
end;
```

## Přepočítej

```
procedure TfrmPojistky.btnPrepocitejClick(Sender: TObject);
Var
Stav: Integer;
begin
tbPojistky.First;
repeat
    begin
    tbPojistkyDetail.First;
    Stav := 0;
        repeat
            if tbPojistkyDetailPojistkaDo.Value >= Now then Stav := Stav + 1;
            tbPojistkyDetail.Next;
        until tbPojistkyDetail.EOF;
    tbPojistky.Edit;
    if Stav > 0 then tbPojistkyPojisteno.Value := True else tbPojistkyPojisteno.Value :=
False;
    tbPojistky.Post;
    end;
tbPojistky.Next;
until tbPojistky.EOF;
tbPojistky.First;
end;
```

## 4.15. Modul Šablony

### Popis

Modul Šablony slouží k definici vzorových záznamů, které jsou následně zobrazeny v modulech Kniha deník, v modulu Doklady přijaté a v modulu Doklady vydané. Šablony slouží k zjednodušení vkládání nových záznamů v příslušných modulech a také zajišťují při správném nastavení vzoru správnost.

### Tabulka

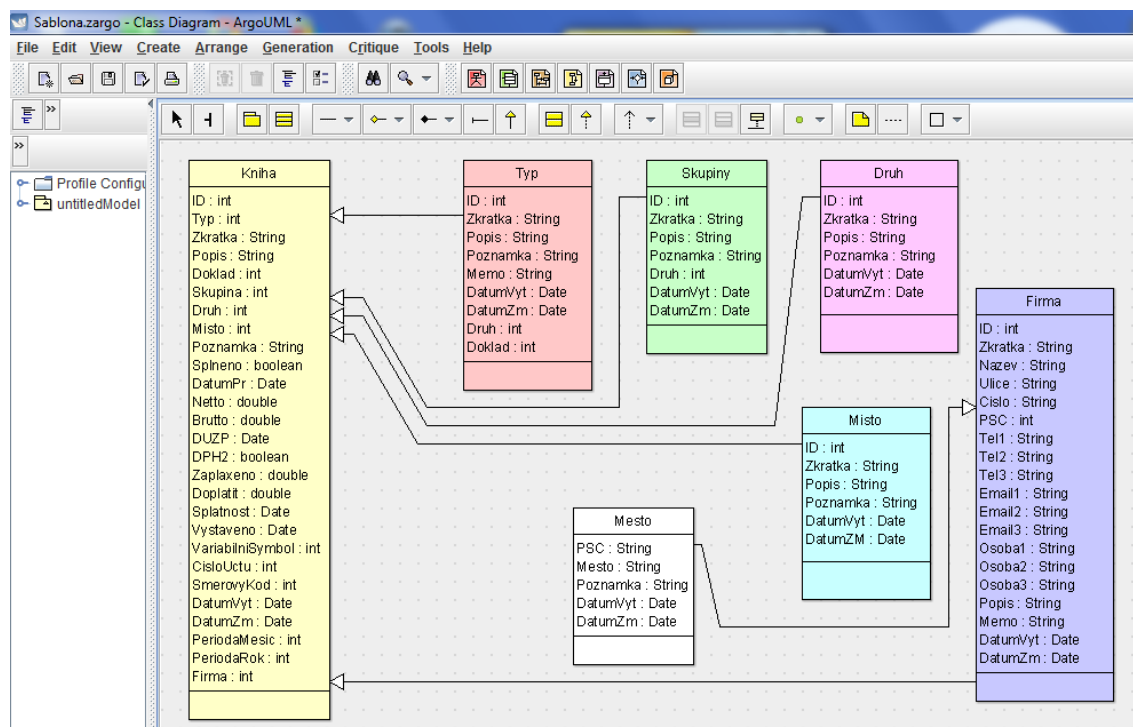
Tabulka Sablony z databáze data.accdb, v programu Delphi tbSablona.

Název pole	Datová typ	Popis
ID	Automatické číslo	Automaticky generované unikátní číslo
Typ	Číslo	Vazba do tabulky Typ
Zkratka	Text	Stručná identifikace úkolu, dokladu
Popis	Text	Podrobnější identifikace úkolu, dokladu
Doklad	Číslo	Vazba do tabulky
Skupina	Číslo	Vazba do tabulky Skupina
Druh	Číslo	Vazba do tabulky Druh
Místo	Číslo	Vazba do tabulky Místo
Poznamka	Memo	Poznámka k záznamu
Splneno	Ano/ne	Ano – úkol splněn / Ne – úkol nesplněn
DatumPr	Datum a čas	Datum připomenutí
Netto	Měna	Částka dokladu bez DPH
Brutto	Měna	Celková částka dokladu
DUZP	Datum a čas	Datum uskutečnitelného zdanitelného plnění
DPH2	Ano/ne	
Zaplaceno	Měna	Celkově zaplacená částka dokladu
Doplatit	Měna	Celkový dluh dokladu
Splatnost	Datum a čas	Datum splatnosti dokladu
Vystaveno	Datum a čas	Datum vystavení dokladu
Firma	Číslo	Vazba do tabulky Firma
VariabilniSymbol	Číslo	Variabilní symbol dokladu přijatého a vydaného
CisloUctu	Číslo	Číslo bankovního účtu
SmerovyKod	Číslo	Směrový kód banky
DatumVyt	Datum a čas	Generované datum dne vytvoření záznamu
DatumZm	Datum a čas	Generované datum dne změny záznamu

PeriodaMesic	Číslo	Generované číslo měsíce, pro potřeby statistik
PeriodaRok	Číslo	Generované číslo roku, pro potřeby statistik

Tabulka 22 – tabulka Sablona

## UML



Obrázek 42 – obrazovka modulu Šablony

## Modelový příklad

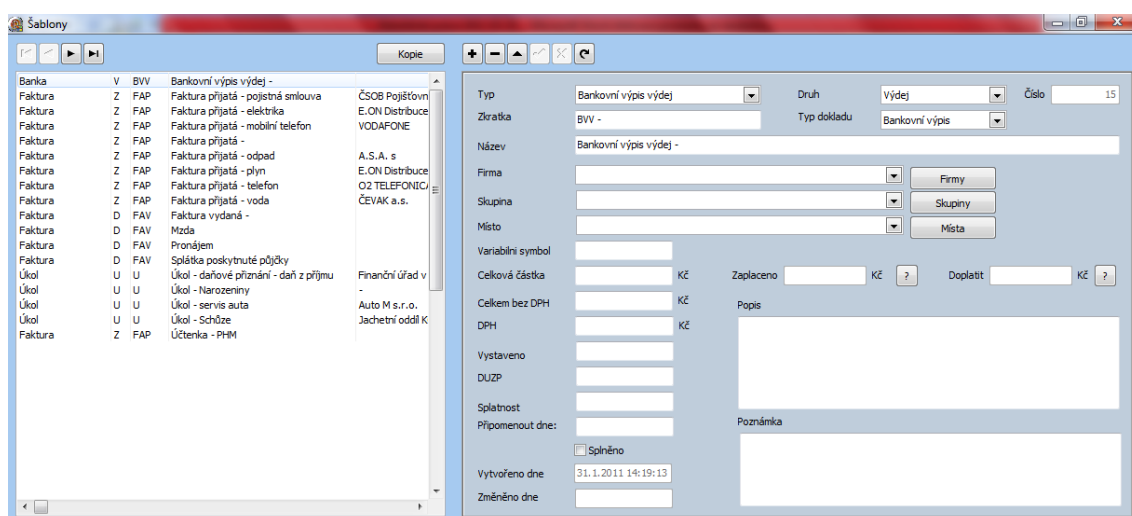
Činnost	Popis
Šablony	
01	Výdej, Bankovní výpis výdej
02	Závazek, Faktura přijatá – pojistná smlouva, ČSOB Pojišťovna
03	Závazek, Faktura přijatá – elektrika, E. ON Distribuce
04	Závazek, Faktura přijatá – mobilní telefon, VODAFONE
05	Závazek, Faktura přijatá –
06	Závazek, Faktura přijatá – odpad, A.S.A
07	Závazek, Faktura přijatá – plyn, E.ON Distribuce
08	Závazek, Faktura přijatá – telefon, O2 Telefonica
09	Závazek, Faktura přijatá – voda, ČEVAK a.s.
10	Pohledávka, Faktura vydaná -



11	Pohledávka, Mzda
12	Pohledávka, Pronájem
13	Pohledávka, Splátka půjčky poskytnuté
14	Úkol, Úkol – daňové přiznání – daň z příjmu
15	Úkol, Úkol – Narozeniny
16	Úkol, Úkol – Servis auta
17	Úkol, Úkol – Schůze
18	Závazek, Účtenka – PHM

Tabulka 23 – modelový příklad Šablona

## Obrázek modulu



Obrázek 43 – obrazovka modulu Šablony

## Funkčnost modulu

### Kopie záznamu

```

procedure TfrmŠablony.btnCopyClick(Sender: TObject);
begin
  CopyTyp := tbSablonaTyp.Text;
  CopySkupina := tbSablonaSkupina.Text;
  CopyDruh := tbSablonaDruh.Text;
  CopyMisto := tbSablonaMisto.Text;
  CopyZkratka := tbSablonaZkratka.Text;
  CopyPopis := tbSablonaPopis.Text;
  CopyMemo := tbSablonaMemo.Text;
  CopyPoznamka := tbSablonaPoznamka.Text;
  CopySplneno := tbSablonaSplneno.Text;
  CopyNetto := tbSablonaNetto.Text;
  CopyBrutto := tbSablonaBrutto.Text;

```

```

CopyDPH := tbSablonaDPH.Text;
CopyDUZP := tbSablonaDUZP.Text;
CopyDPH2 := tbSablonaDPH2.Text;
CopyZaplaceno := tbSablonaZaplaceno.Text;
CopyDoplatit := tbSablonaDoplatit.Text;
CopySplatnost := tbSablonaSplatnost.Text;
CopyVystaveno := tbSablonaVystaveno.Text;
CopyFirma := tbSablonaFirma.Text;
CopyVariabilniSymbol := tbSablonaVariabilniSymbol.Text;
CopyCisloUctu := tbSablonaCisloUctu.Text;
CopyDoklad := tbSablonaDoklad.Text;
//
if tbSablona.State = dsBrowse then tbSablona.Insert;
//
tbSablonaTyp.Text := CopyTyp;
tbSablonaSkupina.Text := CopySkupina;
tbSablonaDruh.Text := CopyDruh;
tbSablonaMisto.Text := CopyMisto;
tbSablonaZkratka.Text := CopyZkratka;
tbSablonaPopis.Text := CopyPopis;
tbSablonaMemo.Text := CopyMemo;
tbSablonaPoznamka.Text := CopyPoznamka;
tbSablonaSplneno.Text := CopySplneno ;
tbSablonaNetto.Text := CopyNetto;
tbSablonaBrutto.Text := CopyBrutto;
tbSablonaDPH.Text := CopyDPH;
tbSablonaDUZP.Text := CopyDUZP;
tbSablonaDPH2.Text := CopyDPH2;
tbSablonaZaplaceno.Text := CopyZaplaceno;
tbSablonaDoplatit.Text := CopyDoplatit;
tbSablonaSplatnost.Text := CopySplatnost;
tbSablonaVystaveno.Text := CopyVystaveno;
tbSablonaFirma.Text := CopyFirma;
tbSablonaVariabilniSymbol.Text := CopyVariabilniSymbol;
tbSablonaCisloUctu.Text := CopyCisloUctu;
tbSablonaDoklad.Text := CopyDoklad;
end;

```

## 4.17. Pomocné číselníky

### 4.17.1. Tabulka Skupina

#### Popis

Pomocný číselník složí k rozlišení skupiny dokladů. Číselník je plněn autorem programu a má vazbu na jednotlivé moduly.

#### Tabulka

Tabulka Skupina databáze data.accdb, v programu Delphi tbSkupina

Název pole	Datová typ	Popis
ID	Automatické číslo	Automaticky generované unikátní číslo
Zkratka	Text	Stručná specifikace
Popis	Text	Podrobnější specifikace
Doklad	Číslo	Vazba do tabulky Doklad
Druh	Číslo	Vazba do tabulky Druh
DatumVyt	Datum a čas	Generované datum dne vytvoření záznamu
DatumZm	Datum a čas	Generované datum dne změny záznamu

Tabulka 24 – pomocná tabulka Skupina

#### Data v číselníku

ID	Zkratka	Název	Druh
1	elektrika	spotřeba - elektrika	7 =
2	plyn	spotřeba -	7 =
3	voda	spotřeba -	7 =
4	nájem	nájem	
5	PHM	vozidla – pohonné hmoty	6 =
8	pojištění	pojistné smlouvy	
9	servis	servis	
10	narozeniny	narozeniny	
11	daň z příjmu	daň z příjmu	

Tabulka 25 – data v tabulce Skupina

## 4.17.2. Tabulka Typ

### Popis

Pomocný číselník slouží k rozlišení typu dokladů. Číselník je plněn autorem programu a má vazbu na jednotlivé moduly.

### Tabulka

Tabulka Typ databáze data.accdb, v programu Delphi tbTyp.

Název pole	Datová typ	Popis
ID	Automatické číslo	Automaticky generované unikátní číslo
Zkratka	Text	Stručná specifikace
Popis	Text	Podrobnější specifikace
Doklad	Číslo	Vazba do tabulky Doklad
Druh	Číslo	Vazba do tabulky Druh
DatumVyt	Datum a čas	Generované datum dne vytvoření záznamu
DatumZm	Datum a čas	Generované datum dne změny záznamu

Tabulka 26 – pomocná tabulka Typ

### Data v číselníku

ID	Zkratka	Název	Využito v modulu	Využito v tabulkách
1	FAP	Faktura přijatá	Faktury přijaté	Kniha
2	PDP	Platba přijatá v hotovosti	Hotovost přijatá	Kniha, Finance
3	FAV	Faktura vydaná	Faktury vydané	Kniha
4	PDV	Platba vydaná v hotovosti	Hotovost vydaná	Kniha, Finance
5	U	Úkol	Deník v tabulce Kniha	Kniha
6	BVP	Bankovní výpis příjem	Bankovní výpis	Finance
7	BVV	Bankovní výpis výdej	Bankovní výpis	Finance
8	VOZ	Vozidlo	Vozidla	Jizdy, Vozidla
9	SPO	Spotřeba	Spotřeba	Spotřeba, SpotřebaDetail
10	POJ	Pojištění	Pojisteni	Pojisteni, Pojisteni Detail

Tabulka 27 – data v tabulce Typ

### 4.17.3. Tabulka Druh

#### Popis

Pomocný číselník slouží k rozlišení typu dokladů. Číselník je plněn autorem programu a má vazbu na jednotlivé moduly.

#### Tabulka

Tabulka Typ databáze data.accdb, v programu Delphi tbDruh.

Název pole	Datová typ	Popis
ID	Automatické číslo	Automaticky generované unikátní číslo
Zkratka	Text	Stručná specifikace
Popis	Text	Podrobnější specifikace
Doklad	Číslo	Vazba do tabulky Doklad
Druh	Číslo	Vazba do tabulky Druh
DatumVyt	Datum a čas	Generované datum dne vytvoření záznamu
DatumZm	Datum a čas	Generované datum dne změny záznamu

Tabulka 28– pomocná tabulka Druh

#### Data v číselníku

ID	Zkratka	Název
1	P	Příjem
2	V	Výdej
3	U	Úkol
4	Z	Závazek
5	D	Dluh, pohledávka
6	VOZ	Vozidlo
7	SPO	Spotřeba
8	POJ	Pojištění

Tabulka 29 – data v tabulce Druh

## 4.17.4. Tabulka Doklad

### Popis

Pomocný číselník slouží k rozlišení formy dokladů. Číselník je plněn autorem programu a má vazbu na jednotlivé moduly.

### Tabulka

Tabulka Typ databáze data.accdb, v programu Delphi tbDoklad.

Název pole	Datová typ	Popis
ID	Automatické číslo	Automaticky generované unikátní číslo
Zkratka	Text	Stručná specifikace
Popis	Text	Podrobnější specifikace
Doklad	Číslo	Vazba do tabulky Doklad
Druh	Číslo	Vazba do tabulky Druh
DatumVyt	Datum a čas	Generované datum dne vytvoření záznamu
DatumZm	Datum a čas	Generované datum dne změny záznamu

Tabulka 30 - -- pomocná tabulka Doklad

### Data v číselníku

ID	Zkratka	Název	Využito v modulu	Využito v tabulkách
1	Faktura	Faktura	Deník, Faktury	Kniha
2	Pokladna	Pokladní doklad	Deník, Pokladní doklady	Finance
3	Smlouva	Smlouva	Pojistné smlouvy	
4	Banka	Bankovní výpis	Banka	Finance
5	Pokladna	Pokladní doklad	Pokladna	
6	Úkol	Úkol	Deník	Kniha

Tabulka 31 – data v tabulce Doklad

## 5. Zhodnocení nově vytvořeného programu

Při tvorbě programu domácí finance jsem měl snahu vyvarovat se chyb svých předchůdců. Začal jsem vytvářet program podle potřeb klienta. Tento přístup byl zcela rozdílný, než je to zvykem u tvorby komerčních programů. Pro ně je zpravidla nosným prvkem modul účetnictví a na tento modul jsou navazovány ostatní agendy. Z toho důvodu mají vždy určitá funkční omezení. Tyto agendy jsou však zpravidla ty nejpoužívanější a měly by být uživatelsky nejpříjemnější.

V mém případě jsem postupoval zcela obráceně. Základním nosným prvkem se stal modul deník. Tento deník obsahuje všechny úkoly a povinnosti řazené podle data. Barevně jsou zde rozlišeny úkoly splněné, nesplněné a nesplněné po termínu. Úkoly je možné vložit individuálně, anebo vytvořením dokladu v některé z agend. Termín plnění se automaticky generuje z data splatnosti a je samozřejmě editovatelný. Zda je úkol splněn, či nesplněn, je opět vytvářeno automaticky či ručně, např. při kompletním doplacení dokladu.

Tyto funkčnosti jsou hlavní výhodou programu.

Okamžitý přehled o termínech neobsahuje žádný z konkurenčních programů.

Další z výhod mého řešení je využití šablon. Tyto lze bez omezení nastavit v jednotlivých agendách. Při vkládání prvotních dokladů jsou tyto šablony nabídnuty k použití. Tato funkčnost opět značně zjednodušuje vkládání záznamů do jednotlivých agend a zrychluje práci s programem.

Tato funkčnost není obvyklá u konkurenčních programů.

Program Domácí finance obsahuje pomocné moduly Provoz vozidel, Energie, spotřeby a Pojistné smlouvy.

Modul Provoz vozidel - eviduje vozidla v domácnosti a jejich provoz. Funkční výhodou je možnost přiřazení zaneseného nákupního dokladu pohonných hmot k jízdě. Nákupní doklady jsou nabídnuty pouze doposud nepřřižené.

Následně je možné propočítat spotřebu.

Modul Spotřeby - eviduje elektřinu, plyn a ostatní energie a spotřeby. Odečty spotřeb je možné provést kontrolně, či účetně. Ke spotřebě je možné přiřadit nákupní doklad. Nákupní doklady jsou nabídnuty pouze doposud nepřřižené. Následně je možné propočítat průměrnou cenu.

Modul Pojištění - eviduje jednotlivé předměty pojištění (pojistné místo, druh) a k nim pojistné smlouvy. K pojistné smlouvě je možné přiřadit nákupní doklad. Nákupní doklady jsou nabídnuty pouze doposud nepřřižené. Přepočtem je možné velmi rychle zjistit, zda aktuálně existuje pojistná smlouva k předmětu pojištění.

Tuto funkčnost nemají konkurenční programy

Program Domáci finance obsahuje řadu dalších funkčností, které zde nejsou popsány. Patří k nim například automatické vyplňování polí a dat. Pro jejich rozsáhlost je nelze v této práci popsat. Nicméně jsou zřejmé ze zdrojového kódu programu.

Program pro domácí využití je velmi intuitivní a jednoduchý. Pokud by tomu tak nebylo, uživatel by program buď nevyužíval, případně by pečlivě nezanašel data do programu.



## 6. Závěr

Úkolem bakalářské práce byla analýza potřeb domácnosti a tvorba programu pro domácí finance. Tento úkol byl úspěšně splněn.

V první etapě práce byla provedena analýza domácího trhu. Na základě této analýzy bylo zjištěno, že existuje velké množství programů určených pro firmy, avšak pro domácnosti téměř neexistují, případně pro ně nejsou dostatečně vhodné. Jednotlivé programy byly prozkoumány, byla zjištěna jejich funkčnost a možnosti využití v domácnosti. Z analýzy bylo zjištěno, že většina z nich není vhodná pro domácí finance. Potřeby domácnosti jsou totiž zcela rozdílné od potřeb firem. Jsou zde řešeny jiné problematiky a také jsou zde jiné požadavky na výstupní data. Tím byl potvrzen prvotní předpoklad nutnosti tvorby nového vhodného programu.

V další etapě práce bylo nutno zanalyzovat potřeby domácnosti. Byl proveden průzkum v několika rozdílných domácnostech a byly určeny jednotlivé priority tvorby programu. Došlo zde ke zjištění, že hlavními prioritami jsou stav financí, stav pohledávek a závazků, sledování spotřeb, evidence pojištění, provoz aut a především plánování úkolů. Tímto směrem byla zaměřena i tvorba programu.

Následná etapa byla již praktická. Byl zde řešen výběr programovacího jazyka a výběr databáze. V této velmi důležité etapě bylo nutno zohlednit cílového klienta. S ohledem na jednoduchost a cenovou přístupnost, byla vybrána databáze MS Access. Programovací jazyk byl vybrán Delphi 2010.

Dále byl již vyvíjen vlastní program. Při jeho tvorbě bylo dbáno především na uživatelskou přívětivost a snadnou dostupnost dat. Byly zde použity zcela jiné principy. Na rozdíl od firemních financí, kde je prvořadé účetnictví, je zde hlavní deník, neboli úkoly, dále pak doklady přijaté a vydané, placené jak hotově, či převodem. Na tyto základní moduly pak navazují ostatní agendy.

Vytvořený program v rámci bakalářské práce je plně funkční a již nyní řeší řadu problematik. Jeho vývoj by se však neměl zastavit v této fázi. Vývoj by měl pokračovat a měl by řešit i další problematiky vzešlé z průzkumu potřeb domácností. Program by bylo vhodné doplnit například o platební příkazy, tisky a jiné agendy. Rovněž by bylo vhodné nabídnout domácímu uživateli také alternativní řešení v podobě některého z databázových nástrojů, jako je například MS SQL, či MySQL. Uživateli je také možné nabídnout řešení přes webové rozhraní, jiný operační systém, případně aplikaci v mobilním telefonu. Vývoj těchto produktů je však časově velmi náročný a obvykle jej řeší programátorské týmy.

Celá bakalářská práce vychází z některých mých dosavadních znalostí, nabytých zkušenostmi v praxi a především studiem oboru Matematické modelování v ekonomii na Jihočeské univerzitě, za značné pomoci vedoucího bakalářské práce Ing. Beránka.

Také bych chtěl závěrem zdůraznit, že časově nejnáročnější byla samotná tvorba programu a tento program tvoří podstatnou část bakalářské práce.

## 7. Summary

The task of the thesis was an analysis of household needs and developing programmes of household finance. This task was successfully completed.

The analysis of the domestic market was done in the first part. On the basis of this analysis was found out that there are a lot of programmes for companies, but almost none for households, or they are not sufficiently suitable. Individual programmes were analysed including their functionality and possibilities of their household use. The analysis showed that most of them are not suitable for household finance. The Household needs are in fact quite different from the needs of companies which are different in their requirements for output data. This analysis confirmed the initial premise of necessity of creation of a new suitable programme.

In the next stage of the work was necessary to analyse the needs of households. A survey was conducted in several different homes which led to defining particular priorities of the programme creation. The following priorities were defined: the state of finances, the state of assets and liabilities, monitoring consumption, insurance, operation of cars and of planning tasks. This was directed towards the creation of the programme.

The follow-up period was finally practical. There was a choice of a programming language and a database selection. At this crucial stage, it was necessary to take into account the target clients. With regard to the simplicity and affordability, MS Access database was selected. As the programming language was chosen Delphi 2010th

In the following part the work the own programme was developed. In the process of creation I paid attention mainly to user-friendliness and easily accessible data. There were used completely different principles. Unlike the corporate finance, where the accounting is first-rate, there is major the diary, or tasks, then the documents received and issued and paid by cash or transfer. These basic modules are then followed by other agendas.

The created programme within the thesis is fully functional and already addresses many issues. Its development, however, should not stop at that stage. The development of the programme should continue and should also address other issues arising from the analysis of household needs. It would be appropriate to add such as payment orders, prints and other agendas to the programme. Also, it would be appropriate to offer home users alternative solutions in the form of a database tools such as MS SQL or MySQL. It is possible to offer to the user the solution through a web interface, another operating system or application in the mobile phone. The development of these products is very time consuming and it is usually developed by programming teams.

The whole bachelor's thesis is based on some of my prior knowledge, experience gained in practice and in the particular study of mathematical modelling in economics at the University of South Bohemia with the considerable assistance of the thesis supervisor Ing. Beránek. I would also like to emphasize here that the most time-consuming part of the work was the programme creation which constitutes the essential part of this thesis.

## 8. Přehled použité literatury

01. SVOBODA, Luděk ; VONEŠ, Petr; KONŠAL, Tomáš; MAREŠ, Miroslav. *1001 Tipů a triků*. Brno : Computer Press, a.s., 2003. 546 s. ISBN 80-7226-488-5.
02. PÍSEK, Slavoj. *Delphi : Začínáme programovat*. Praha : Grada Publishing a.s., 2002. 328 s. ISBN 80-247-0547-8.
03. KADLEC, Václav. *Učíme se programovat v DELPHI a v jazyce OBJECT PASCAL*. Praha : Computer Press, a.s., 2001. 288 s. ISBN 80-7226-245-9.
04. SEDLÁČEK, Jiří; SLABA, Jiří. *DELPHI v kostce*. Praha : BEN technická literatura, 2000. 514 s. ISBN 80-86056-097-X.
05. KADLEC, Václav. *Delphi : Hotová řešení*. Brno : Computer Press, a.s., 2003. 312 s. ISBN 80-251-0017-0.
06. LACKO, Luboslav. *SQL : Hotová řešení*. Brno : Computer Press, a.s., 2003. 298 s. ISBN 80-7226-975-5.
11. K2 *Atmitec* [online]. 2010 [cit. 2011-04-26]. K2 software Svět K2. Dostupné z WWW: <<http://www.k2atmitec.cz/cz/produkty/software/svet.htm>>.
12. KASTNER software [online]. 2010 [cit. 2011-04-27]. Osobní finance a domácí hospodaření. Dostupné z WWW: <<http://www.domaci-ucetnictvi.stereo.cz/>>
13. Cígler software [online]. 2010 [cit. 2011-04-27]. Money S3. Dostupné z WWW: <<http://www.money.cz/money-s3>>
14. Cígler software [online]. 2010 [cit. 2011-04-27]. Money S4. Dostupné z WWW: <<http://www.money.cz/money-s4>>
15. Cígler software [online]. 2010 [cit. 2011-04-27]. Money S5. Dostupné z WWW: <<http://www.money.cz/money-s5>>
16. ABRA G1 [online]. 2010 [cit. 2011-04-27]. ABRA Software. Dostupné z WWW: <<http://www.abra.eu/produkty/abra-g1/>>
17. ABRA G2 [online]. 2010 [cit. 2011-04-27]. ABRA Software. Dostupné z WWW: <<http://www.abra.eu/produkty/abra-g2/>>

18. *ABRA G3* [online]. 2010 [cit. 2011-04-27]. ABRA Software. Dostupné z WWW: <<http://www.abra.eu/produkty/abra-g3/>>
19. *ABRA G4* [online]. 2010 [cit. 2011-04-27]. ABRA Software. Dostupné z WWW: <<http://www.abra.eu/produkty/abra-g4/>>
20. *EIS Vision 32* [online]. 2010 [cit. 2011-04-27]. Vision Praha s.r.o. Dostupné z WWW: <<http://www.vision.cz/>>
21. *Asseco Solutions, a.s.* [online]. 2010 [cit. 2011-04-27]. Helios informační systém. Dostupné z WWW: <<http://www.helios.eu/helios-cs/helios.html>>
22. *SAP Česká republika* [online]. 2010 [cit. 2011-04-27]. Řešení SAP. Dostupné z WWW: <<http://www.sap.com/cz/index.epx>>
23. *Altus software s.r.o.* [online]. 2010 [cit. 2011-04-27]. Altus Vario® - ekonomický software kategorie ERP. Dostupné z WWW: <<http://www.vario.cz/>>

## 9. Přílohy

DVD nosič s tímto obsahem:

- zdrojový kód programu
- program
- UML
- bakalářská práce