

UNIVERZITA PARDUBICE
FAKULTA EKONOMICKO - SPRÁVNÍ

BAKALÁŘSKÁ PRÁCE

2010

Jakub Špidlen

Univerzita Pardubice
Fakulta ekonomicko - správní

Tvorba modulu pro informační systém Helios Orange

Jakub Špidlen

Bakalářská práce

2010

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky
Akademický rok: 2009/2010

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jakub ŠPIDLEN**

Studijní program: **B6209 Systémové inženýrství a informatika**

Studijní obor: **Informatika ve veřejné správě**

Název tématu: **Tvorba modulu pro podnikový informační systém Helios Orange**

Z á s a d y p r o v y p r a c o v á n í :

1. Problematika IS obecně
2. Popis IS Helios
3. Návrh modulu
4. Implementace modulu
5. Závěr, zhodnocení

Rozsah grafických prací:

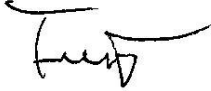
Rozsah pracovní zprávy:

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

Seznam odborné literatury:


- [1] **Helios Orange : Informační systém pro středně velké a menší společnosti.** [s.l.] : [s.n.], 2009. s. 5.
- [2] **Informační systém Helios Orange.** Helios Orange [CD-Rom]. 2008 [cit. 2009-06-28].
- [3] **HALVORSON, Michael.** Microsoft Visual Basic 2008 : krok za krokem. [s.l.] : [s.n.], 2008. 440 s. ISBN 978-80-251-2221-1.
- [4] **KOMÁRKOVÁ, Jitka, et al.** Úvod do informačních systémů. [s.l.] : [s.n.], 2006. 85 s. ISBN 80-7194-870-5.

Vedoucí bakalářské práce:


Ing. Milan Tomeš
Ústav systémového inženýrství a informatiky


Datum zadání bakalářské práce: **5. října 2009**

Termín odevzdání bakalářské práce: **30. dubna 2010**


doc. Ing. Renáta Myšková, Ph.D.

děkanka

L.S.


doc. Ing. Jiří Křepka, Ph.D.

vedoucí ústavu

V Pardubicích dne 30. června 2009

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 28. 4. 2010

Jakub Špidlen

Poděkování:

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Milanovi Tomešovi za jeho rady, názory a především trpělivost během celé mé práce. Dále bych chtěl poděkovat Ing. Michalovi Jarkovskému nejen za poskytnutou podporu technickou, ale i morální. Mé velké díky patří také Ing. Kateřině Jonášové za nápady a připomínky k tvořenému modulu.

ANOTACE

Bakalářská práce je zaměřena na obecnou charakteristiku informačních systémů, včetně popisu produktů firmy Asseco Solutions. Podrobně je rozebrán produkt Informační systém Helios Orange, kde lze kromě výstižné charakteristiky a popisu jeho předností nalézt výčet nejdůležitějších modulů zařazených do konkrétních kategorií. Práce dále zahrnuje návrh a implementaci modulu pro informační systém Helios Orange. Modul Vnitropodniková banka je vytvořen programovacím jazykem Microsoft Visual C#.

KLÍČOVÁ SLOVA

informační systém, modul, vnitropodniková banka, helios

TITLE

The creation of plug-in for the information system Helios Orange

ANNOTATION

This bachelor work deals with a general characteristic of the information systems and the products of Asseco Solutions company as well. There is a detail of the product Helios Orange. Helios Orange description includes its basic properties and its most important plug-ins classified into the different categories. This work also provides a newly created plug-in for the information system Helios Orange. The plug-in is called Vnitropodniková banka and it is made up by Microsoft Visual C# programming language.

KEYWORDS

information system, plug-in, vnitropodniková banka, helios

OBSAH

1	ÚVOD.....	11
2	POJEM INFORMAČNÍ SYSTÉM.....	12
2.1	PODNIKOVÝ INFORMAČNÍ SYSTÉM	12
2.2	POŽADAVKY NA INFORMAČNÍ SYSTÉM.....	12
3	ŽIVOTNÍ CYKLUS INFORMAČNÍHO SYSTÉMU.....	14
3.1	PŘEDBĚŽNÁ ANALÝZA, SPECIFIKACE CÍLŮ	14
3.2	ANALÝZA SYSTÉMU, SPECIFIKACE POŽADAVKŮ	14
3.3	PROJEKTOVÁ STUDIE - NÁVRH	15
3.4	IMPLEMENTACE	15
3.5	TESTOVÁNÍ	15
3.6	ZAVÁDĚNÍ SYSTÉMU.....	15
3.7	ZKUŠEBNÍ PROVOZ.....	15
3.8	PROVOZ A ÚDRŽBA	15
4	KVALITA INFORMAČNÍHO SYSTÉMU	16
4.1	MODEL KVALITY ISO 9126:	17
4.1.1	<i>Funkčnost.....</i>	<i>17</i>
4.1.2	<i>Bezporuchovost</i>	<i>17</i>
4.1.3	<i>Použitelnost</i>	<i>17</i>
4.1.4	<i>Účinnost.....</i>	<i>17</i>
4.1.5	<i>Udržovatelnost.....</i>	<i>17</i>
4.1.6	<i>Přenositelnost</i>	<i>17</i>
5	NÁSTROJE PRO TVORBU MODULU	17
5.1	SQL	17
5.2	MICROSOFT VISUAL STUDIO, VISUAL C#	18
6	ARCHITEKTURA IS A TECHNICKÁ INFRASTRUKTURA.....	18
6.1	ARCHITEKTURA DATABÁZÍ	19
6.1.1	<i>Centrální architektura.....</i>	<i>19</i>
6.1.2	<i>Architektura file-server</i>	<i>19</i>
6.1.3	<i>Architektura klient-server</i>	<i>20</i>
7	O SPOLEČNOSTI ASSECO SOLUTIONS	20
8	INFORMAČNÍ SYSTÉM HELIOS GREEN	21
8.1	PŘEDNOSTI A ZNAKY IS HELIOS GREEN	21
9	INFORMAČNÍM SYSTÉM HELIOS RED	21

9.1	PŘEDNOSTI, ZNAKY IS HELIOS RED.....	22
10	INFORMAČNÍ SYSTÉM HELIOS ORANGE	22
10.1	PŘÍNOSY PRO MENŠÍ FIRMY.....	22
10.2	PŘÍNOSY PRO STŘEDNĚ VELKÉ FIRMY.....	23
10.3	PŘEDNOSTI, ZNAKY IS HELIOS ORANGE.....	23
10.4	MODULY SYSTÉMU HELIOS ORANGE	23
10.4.1	<i>Jádro systému</i>	24
10.4.2	<i>Ekonomika a finance.....</i>	25
10.4.3	<i>Obchod.....</i>	26
10.4.4	<i>Styk se zákazníky.....</i>	26
10.4.5	<i>Lidské zdroje.....</i>	27
10.4.6	<i>Manažerské rozhodování.....</i>	27
10.4.7	<i>Výroba.....</i>	27
10.4.8	<i>Doprava</i>	27
10.4.9	<i>Převážné služby.....</i>	27
10.4.10	<i>Celní software</i>	28
10.4.11	<i>Zemědělství.....</i>	28
10.4.12	<i>Servis.....</i>	28
10.4.13	<i>Řízení projektů</i>	29
10.5	HELIOS SPACE.....	29
11	NÁVRH MODULU VNITROPODNIKOVÁ BANKA	29
11.1	PRAKTICKÁ FUNKCIONALITA TVOŘENÉHO MODULU	29
11.1.1	<i>První spuštění.....</i>	30
11.1.2	<i>Operace zúčít, vložit, vybrat</i>	32
11.1.3	<i>Operace uložit, tisknout, zavřít</i>	32
11.2	TECHNOLOGIE.....	33
12	KOMUNIKACE MODULU S IS HELIOS ORANGE	33
13	PROGRAMOVÁNÍ MODULU VNITROPODNIKOVÁ BANKA	36
13.1	USING REFERENCE	36
13.2	JMENNÉ PROSTORY	37
13.3	TŘÍDA TEST JMENNÉHO PROSTORU PLUGINVNITROBANKA.....	37
13.4	TŘÍDA FORM1 JMENNÉHO PROSTORU FORMULAR.....	38
13.4.1	<i>Přetížený konstruktor Form1.....</i>	39
13.4.2	<i>Metoda Změna data v kalendáři.....</i>	40
13.4.3	<i>Metoda Úrok.....</i>	41
13.4.4	<i>Metody Vklad a Výběr.....</i>	42
13.4.5	<i>Metoda Uložit, Tisk, Zavřít</i>	43

13.4.6	<i>Grafické komponenty</i>	44
14	ZHODNOCENÍ	45
15	ZÁVĚR	45
16	POUŽITÉ ZDROJE	46

Seznam obrázků

OBRÁZEK 1 - VODOPÁDOVÝ ŽIVOTNÍ CYKLUS, ZDROJ: VLASTNÍ	16
OBRÁZEK 2 - MICROSOFT VISUAL STUDIO, ZDROJ: VLASTNÍ	18
OBRÁZEK 3 – CENTRÁLNÍ ARCHITEKTURA, ZDROJ: [5]	19
OBRÁZEK 4 - ARCHITEKTURA FILE-SERVER, ZDROJ: [5]	20
OBRÁZEK 5 - ARCHITEKTURA KLIENT-SERVER, ZDROJ: [5].....	20
OBRÁZEK 6 - KATEGORIE MODULŮ IS HELIOS ORANGE, ZDROJ: [2]	24
OBRÁZEK 7 - MODUL POKLADNA, ZDROJ: VLASTNÍ.....	25
OBRÁZEK 8 - MODUL BANKA, ZDROJ: VLASTNÍ	25
OBRÁZEK 9 - MODUL FAKTURACE, ZDROJ: VLASTNÍ.....	26
OBRÁZEK 10 - MODUL MZDY A PERSONALISTIKA, ZDROJ: VLASTNÍ	27
OBRÁZEK 11 - MODUL PŘEPRAVNÍ SLUŽBY, ZDROJ: VLASTNÍ	28
OBRÁZEK 12 - MODUL SERVIS, ZDROJ: VLASTNÍ	28
OBRÁZEK 13 - USE CASE DIAGRAM, ZDROJ: VLASTNÍ.....	30
OBRÁZEK 14 - MODUL VNITROPODNIKOVÁ BANKA - 1. SPUŠTĚNÍ, ZDROJ: VLASTNÍ	31
OBRÁZEK 15 - MODUL VNITROPODNIKOVÁ BANKA - VKLAD, VÝBĚR, ZDROJ: VLASTNÍ	31
OBRÁZEK 16 - MODUL VNITROPODNIKOVÁ BANKA - ZÚROČIT, ZDROJ: VLASTNÍ	32
OBRÁZEK 17 - IMPORT COM -> .NET, ZDROJ: VLASTNÍ.....	33
OBRÁZEK 18 - KOMPILACE, REGISTRACE PLUGINVNITROBANKA.CS, ZDROJ: VLASTNÍ	34
OBRÁZEK 19 - KOMPILACE FORMULAR.CS, ZDROJ: VLASTNÍ	34
OBRÁZEK 20 - EXTERNÍ AKCE, ZDROJ: VLASTNÍ	35
OBRÁZEK 21 - SPUŠTĚNÍ PLUGINU, ZDROJ: VLASTNÍ.....	36

Seznam použitých zkratk a pojmů

IS	informační systém
ICT	informační a komunikační technologie
ERP	(enterprise resource planning) informační systém, který integruje a automatizuje velké množství procesů souvisejících s produkčními činnostmi podniku. Typicky se jedná o výrobu, logistiku, distribuci, správu majetku, prodej, fakturaci, a účetnictví.
CRM	(customer relationship management) řízení vztahů se zákazníkem
SŘBD	systém řízení báze dat

1 Úvod

Jak celosvětově, tak i v rámci České republiky roste poptávka po větší komplexnosti ERP systémů. Nastavení podnikového informačního systému tak, aby byl co nejvíce komplexní a zároveň, aby některé jeho funkce nebyly spíše na obtíž, je ale velmi složité. Zpravidla mají výhodu menší dodavatelé, kteří bývají v těchto případech flexibilnější. Společnost Asseco Solutions dosahuje komplexnosti množstvím nabízených produktů, ale prakticky je zcela nemožné vyhovět zadavatelským firmám ve všech jejich požadavcích. Výsledek této práce pomůže zkvalitnit služby poskytované uživatelům informačního systému Helios Orange.

Bakalářská práce je rozdělena do několika kapitol, které lze spolehlivě rozdělit do čtyř bloků. V první části jsou kapitoly zaměřeny na obecnou charakteristiku, zejména podnikového, informačního systému. Každý správně fungující informační systém musí splňovat přísná kritéria a požadavky. Tyto požadavky jsou jednak vyjmenovány a jednak i blíže specifikovány. Ke každému IS neodmyslitelně patří i jeho životní cyklus. Respektovat a vhodně naplánovat životní cyklus je základní stavební kámen každého IS. Kromě vyhovění požadavkům je třeba dbát na kvalitu IS. Práce obsahuje návod na dodržení kvality podle hojně využívaného modelu ISO 9126.

Druhá část se již specializuje na podnikové informační systémy Helios od společnosti Asseco Solutions. Vysvětluje rozdíl mezi třemi nabízenými produkty sady Helios a především se zaměřuje na popis modifikace s názvem Orange, pro kterou je modul tvořen. Kromě výčtu znaků a předností IS Helios Orange jsou ještě stručně rozebrány jeho jednotlivé moduly.

Ve třetí části je rozebrán návrh modulu Vnitropodniková banka. Návrh vychází jednak z obecných poznatků zkušenějších autorů a jednak z konkrétních požadavků koncového uživatele Ing. Kateřiny Jonášové, ekonomky firmy Strojon Pardubice. Návrh obsahuje Use case diagram činností modulu včetně stručného a jasného popisu jednotlivých případů užití.

Vrcholem bakalářské práce je fungující modul Vnitropodniková banka, kterého bylo dosaženo programováním kódu jazykem C#. V závěrečné části je popsáno, jak naprogramovaný modul komunikuje s IS Helios Orange, resp. jak se modul v IS spustí. Samozřejmě nechybí ani systematicky na sebe navazující úryvky funkčního kódu s podrobným vysvětlením důležitých metod a proměnných.

Hlavní cíl bakalářské práce vyplývá přímo z jejího zadání. Úkolem je vytvořit funkční externí řešení (plugin) k informačnímu systému Helios Orange. Helios Orange je vyspělý informační a

ekonomický systém pro malé a středně velké podniky. Tento informační systém je velmi otevřený, umožňuje využít jádro systému pro doprogramování libovolné funkcionality, kterou firma ke své činnosti potřebuje.

2 Pojem informační systém

Na informační systém lze nahlížet z mnoha úhlů pohledu. Proto existuje nespočet definic pojmu informační systém. Každá ho ovšem chápe trochu jinak, podle toho, pro jaké účely chce informační systém definovat. [7]

Pro potřeby studentů vysokých škol lze informační systém definovat podle [7] jako funkční propojení lidí, dat, procesů, rozhraní, sítí a technologií. Jednotlivé prvky spolupracují tak, aby podporovaly a zlepšovaly každodenní operace v organizaci a zároveň aby podporovaly řešení problémů a proces rozhodování v rámci managementu.

Předmětem této práce je programování modulu pro konkrétní podnikový informační systém. Následující kapitola stručně nastíní problematiku podnikových informačních systémů.

2.1 Podnikový informační systém

Vývoj podnikových IS lze sledovat z různých aspektů, jako je jejich měnící se funkcionality, trendy v implementaci, provozování i změny v očekávaných přínosech. K popisu hlavních změn lze symbolicky použít písmena zkratky ERP, užívaná pro označení klíčové aplikace podnikových IS. Na počátku byla důležitá zejména podpora plánování (P – planning), která byla následně vystřídána důrazem na všechny podnikové zdroje (R – resources), tj. hlavně materiálu, kapacit a financí. Současnost do centra pozornosti přesouvá podnik (E – enterprise) a podnikání, zejména pak efektivitu, udržení a rozvoj konkurenceschopnosti podniku. Nejde již jen o jeden samostatný podnik, ale o podnik kooperující a zřetěžený v rámci sítí, o podnik otevřený vůči partnerům a díky informačním systémům rozložitelný v libovolné části globalizovaného světa. [3]

2.2 Požadavky na informační systém

Na informační systém jsou kladeny základní požadavky, které jsou společné pro většinu typů informačních systémů, tedy ať se jedná o podnikový, redakční nebo CRM systém. Podle [7] by měl IS být:

- otevřený
- dynamický

- podporovaný
- komplexní
- kompaktní
- standardizovaný
- stavebnicový
- chráněný
- kompatibilní
- zaveden co nejrychleji
- v co nejvyšší kvalitě

Otevřené systémy mají podstatně snazší přenositelnost (a to jak programů, tak i uživatelů). Snáz probíhá také integrace jednotlivých počítačů do vyšších celků – sítí, proto zvyšuje výraznou měrou efektivitu využívání počítačů a zároveň vytváří tlak na snižování cen jak softwaru, tak i hardwaru. Základní požadavek přenositelnosti otevřených systémů tedy zaručuje vzájemnou slučitelnost komponent od různých výrobců. [8]

Dynamický systém sestává ze stavového prostoru, jehož souřadnice popisují stav systému v daném čase a z dynamických podmínek, které popisují změnu tohoto systému v čase. Stav systému je potom popsán vektorem, který celý leží ve stavovém prostoru. Dynamické podmínky jsou většinou zadány soustavou diferenciálních rovnic, které popisují změnu stavového vektoru v čase. Změna stavu dynamického systému se děje provedením těchto diferenciálních rovnic a nahrazením starého stavového vektoru vektorem novým. Dynamický systém může být deterministický nebo stochastický (náhodný). Deterministický dynamický systém lze poměrně přesně popsat, zatímco u systému stochastického jsme odkázáni pouze na jeho statistické vlastnosti (například střední hodnotu, disperzi, směrodatnou odchylku, centrální moment a jiné). [15]

Podporovaný systém garantuje nejen servis na určitou dobu, ale i zabezpečuje další rozvoj systému. [7]

Komplexní systém může být rozdělen na mnoho částí majících nespočet vazeb mezi sebou. Chování každé části systému se tak odvíjí od chování všech ostatních částí. Mezi charakteristiky komplexních systémů pak lze zařadit: dynamičnost, těsnou svázanost, ovládatelnost zpětnou vazbou a adaptabilitu. [9]

Kompaktní systém má všechny požadované vnitřní vazby mezi jednotlivými subsystemy i jednotlivými daty. Má vytvořené jak vazby horizontální (na stejné rozlišovací úrovni), tak vazby vertikální (na hierarchicky odlišných rozlišovacích úrovních). [7]

Řešení, kterému se přizpůsobují různí výrobci a které tak představuje určitou společnou konvenci, zajišťující vzájemnou kompatibilitu produktů od různých výrobců, si již zaslouží přívlastek standardní. Samotný obsah resp. podstata tohoto řešení se pak v širším slova smyslu označuje jako **standard**. [10]

Stavebnicový systém zajišťuje, že jednotlivé softwarové komponenty lze vyměňovat po blocích, čímž je jednak umožněna realizace systému po etapách, ale i výběr či náhrada jednotlivých komponent IS. [7]

Chráněný je systém jak před zneužitím tak před úmyslným i neúmyslným poškozením techniky, dat a softwarové části. [7]

Prosadit vlastní řešení v oblasti IT, a to ještě se ziskem, si v dlouhodobém výhledu mohly dovolit jen ty největší firmy. Menší firmy, které nemohly nést stále větší náklady na vývoj a marketing vlastních řešení, se ve vlastním zájmu musely přizpůsobit těm řešením, které si zvolily velké firmy. Nešlo přitom ani tak o převzetí technologií či výrobních postupů (které jsou často pečlivě chráněné), jako spíše o převzetí konvencí, parametrů a protokolů, s cílem zajistit **kompatibilitu** (slučitelnost) vlastních produktů s produkty jiných výrobců. [10]

3 Životní cyklus informačního systému

S trochou nadsázky lze říci, že všechny živé i neživé organismy, stroje i lidé mají svůj životní cyklus, na který se ve specifických oblastech bádání a výzkumu nahlíží pokaždé jinak. I každý IS prochází životním cyklem, který lze podle [14] rozdělit do několika fází:

3.1 Předběžná analýza, specifikace cílů

Základem celkového návrhu, vývoje i jakékoli úpravy stávajícího systému jsou požadavky uživatelů a cíle organizace. V této části se musí dané požadavky shromáždit, v hrubých rysech rozebrat a odhadnout doba realizace a náklady. Cílem je pouze sestavit základní rámec požadavků, cílů a funkcí.

3.2 Analýza systému, specifikace požadavků

Tato část cyklu je rozbořením části předchozí. Její důležitost je klíčová, neboť veškeré chyby ve struktuře dat i systému, které se zde neodhalí, jsou později velice obtížně odstranitelné.

3.3 Projektová studie - návrh

Tato část je výsledkem analýzy systému. V této fázi je vytvořen podklad pro obsah smlouvy s externí firmou o návrhu a realizaci IS, časový harmonogram, cena vyvíjeného projektu, konkrétní typ implementace systému, podmínky zavádění v organizaci, záruční servis a podmínky celkového předání IS.

3.4 Implementace

Tato část životního cyklu IS je vlastním programováním, kterého se účastní vybraní experti a analytik nesoucí zodpovědnost za správnost řešení. Na základě získaných faktů z fyzického návrhu se definují vstupy a výstupy jednotlivých operací a určí se způsob jejich modifikace. Naprogramují se veškeré funkce a doladí se jejich vzájemné propojení. Dále se jednotlivé realizované funkce ověří a připraví se testovací data.

3.5 Testování

V této etapě se provádí připravené testy na hotovém IS. Je nutné vyzkoušet veškeré možné reakce systému na zadávaná data a zjištěné nedostatky opravit. Testování se často provádí na systému, který ještě není v reálném prostředí, neboť případné selhání by mohlo mít rozsáhlé následky.

3.6 Zavádění systému

Zaváděním systému je míněna především jeho instalace, zavedení do provozu organizace, transformace původní datové základny tak, aby byla přístupná novému systému, poskytnutí manuálů a školení uživatelům.

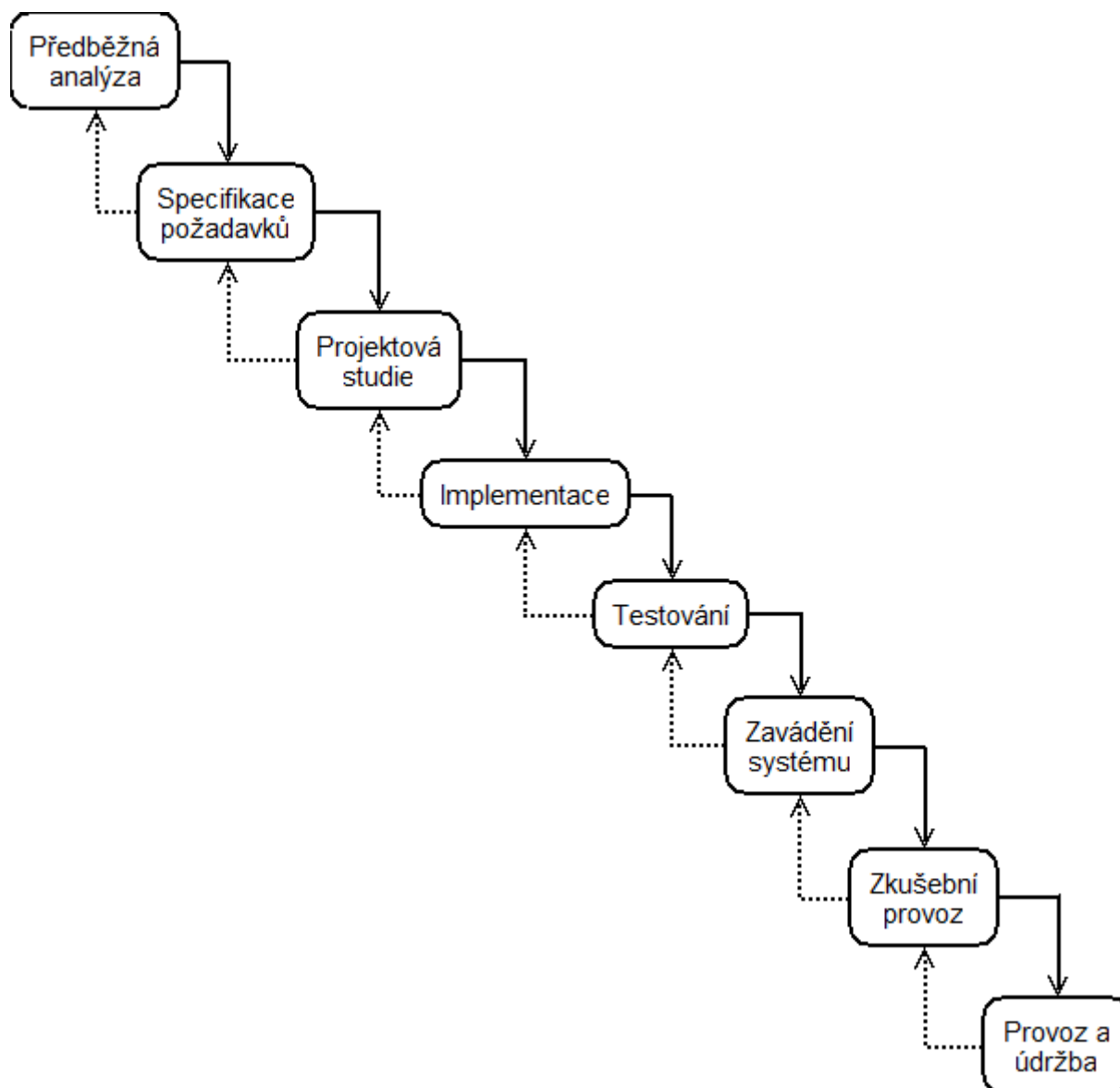
3.7 Zkušební provoz

Zkušební provoz je celková realizace projektu, ve které je poskytovatel povinen zajistit okamžitý servis, odstranit chyby zjištěné během provozu, nebo dořešit dodatečné požadavky uživatelů v rámci původního návrhu.

3.8 Provoz a údržba

Tato etapa je závěrečnou fází projektu, ve které je systém provozován a používán. Do této etapy také spadá údržba systému, tedy zajištění správného provozu, úprava parametrů aplikací nebo změny některých programů tak, aby splňovaly nové požadavky uživatelů.

Zpravidla by měla nejdéle trvat fáze provozu a údržby, jinak by informační systém vůbec nemělo smysl vytvářet. Obrázek 1 ilustruje vodopádový životní cyklus.



Obrázek 1 - Vodopádový životní cyklus, zdroj: vlastní

4 Kvalita informačního systému

Kvalita informačního systému u uživatelů a s ní související kvalita informačního systému jako celku i kvalita jeho programové části se, vzhledem k neustále narůstající závislosti lidské společnosti na informačních a komunikačních technologiích (ICT), stává naprostou nutností. Důvod je zřejmý: ICT jsou využívány i v oblastech, kde může jejich selhání způsobit velmi vysoké ztráty značně různorodého charakteru – například v rámci krizového řízení, v bankovníctví nebo v armádě. [6]

Na kvalitu informačních systémů lze nahlížet z několika různých úhlů pohledu – od komplexního pohledu na straně jedné, k odděleným pohledům uživatelů na straně druhé. Dále je kvalita IS hodnocena nejen podle toho, jak splňují požadavky na svou funkčnost, ale také podle toho, jak splňují ostatní, tzv. nefunkční požadavky. I proto existuje celá řada definic kvality. [6]

Např. Česká novela ISO 9000/2000 definuje kvalitu jako: “Schopnost souboru znaků výrobku, systému nebo procesu plnit požadavky zákazníka a jiných zainteresovaných stran.” [16]

4.1 Model kvality ISO 9126:

Model ISO 9126 obsahuje 6 charakteristik kvality, které hodnotí kvalitu systému velmi důkladně. Protože ale cílem práce není definovat kvalitu IS, jsou charakteristiky modelu vyjmenovány a jen stručně charakterizovány.

Charakteristiky kvality podle [12]:

4.1.1 Funkčnost

je sada atributů, které se opírají o existenci souvisejících funkcí a jejich specifických vlastností.

4.1.2 Bezporuchovost

je soubor vlastností, které souvisí se schopností systému udržovat stejnou úroveň výkonu za stálých podmínek a v daném čase.

4.1.3 Použitelnost

měří množství úsilí uživatelů potřebné k užívání systému a individuální hodnocení použití.

4.1.4 Účinnost

vychází ze vztahu mezi úrovní výkonu systému a množstvím použitých prostředků za daných podmínek.

4.1.5 Udržovatelnost

je vlastnost měřící úsilí potřebné k provedení konkrétních modifikací.

4.1.6 Přenositelnost

je schopnost systému být přenesen mezi prostředími.

5 Nástroje pro tvorbu modulu

Kapitola stručně přibližuje použité technologie v rámci programování modulu.

5.1 SQL

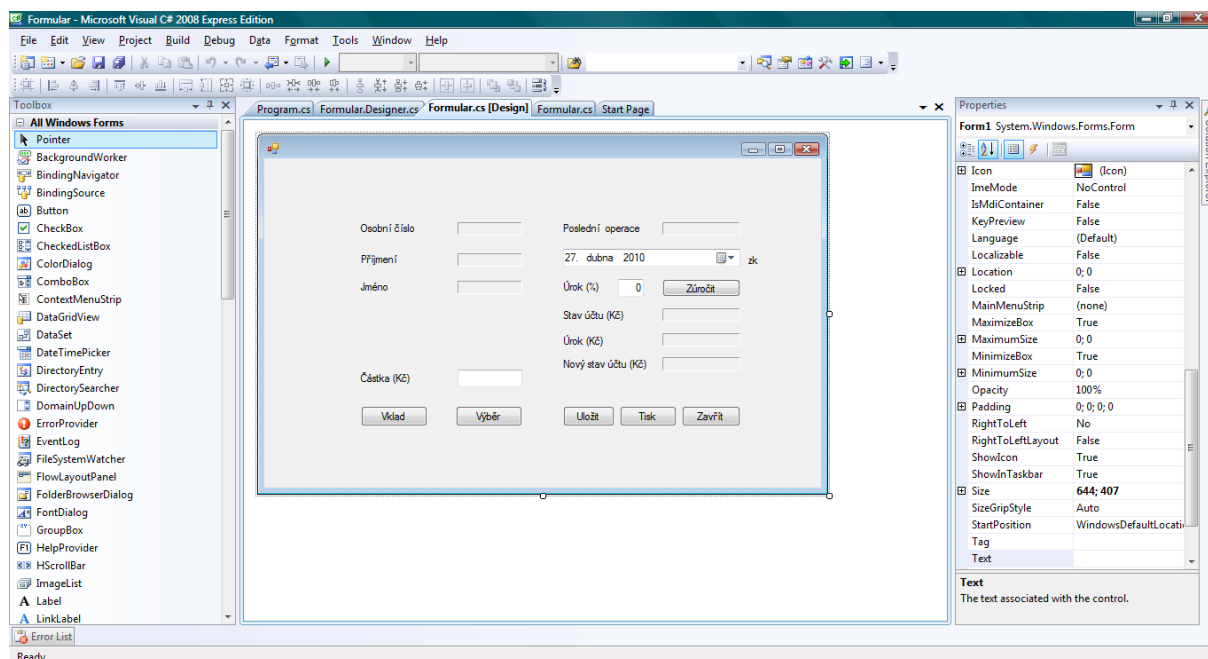
SQL (Structured Query Language) je neprocedurální jazyk. Neprocedurálním jazykem lze říci, co se má provést, kdežto procedurálním jazykem, jak se to má provést. SQL se stalo určitým pojítkem napříč celou databázovou technologií. Na jednu stranu se jedná o jazyk poměrně mocný a současně jednoduchý (příkaz je velice podobný standardní anglické větě – např. „SELECT jméno FROM tabulka WHERE jméno = Josef“ lze vyslovit jako „Vyber záznamy

z tabulky, kde jméno je Josef“. Kromě dotazu SELECT patří do SQL i akce INSERT, DELETE a UPDATE, které nevybírají data z tabulek, ale do tabulek data vkládají, nebo z tabulek data mažou, či je aktualizují. [13]

5.2 Microsoft Visual studio, Visual C#

Microsoft Visual C# je výkonný, ale při tom jednoduchý, jazyk zaměřený především na vývojáře aplikací v platformě .NET Framework. Zdědil velké množství z toho nejlepšího z jazyků C++ a Microsoft Visual Basic, ale jen málo z jejich nesrovnalostí, takže výsledkem je čistší a logičtější jazyk. S příchodem jazyka C# se na scénu dostává několik nových prvků, včetně operátorů, generických typů a rozšiřujících metod. Ve vývojovém prostředí **Microsoft Visual Studio 2008** se tyto výkonné novinky dají velmi snadno používat, produktivitu práce vývojáře pak výrazně zvyšují noví průvodci a různá další vylepšení, která jsou součástí Microsoft Visual Studia 2008. [11]

Obrázek 2 ukazuje prostředí Microsoft Visual Studia. V levé části se nachází Toolbox (panel nástrojů), v pravé tabulka vlastností vybraného objektu, uprostřed grafický formulář, popř. zdrojový kód, dole případně Error list a nahoře standardně tlačítka sloužící k ovládání studia.



Obrázek 2 - Microsoft Visual Studio, zdroj: vlastní

6 Architektura IS a technická infrastruktura

Každý IS má nějakou technickou strukturu. IS je aplikace s datovou, aplikační a prezentační vrstvou. Pro provoz databáze IS je vyčleněn server v příslušné konfiguraci a instalovaným software vybavením. Server je provozován pod operačním systémem a v databázovém

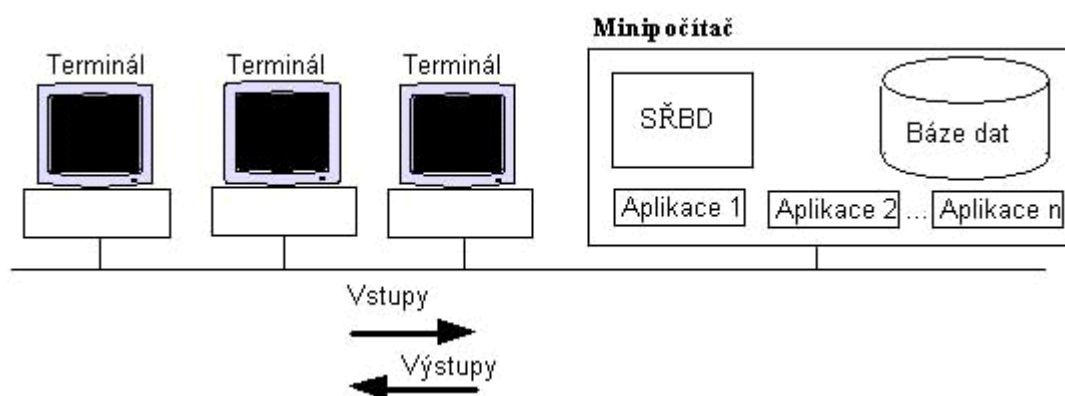
prostředí. Jako klientské stanice mohou být použity osobní počítače se základním vybavením. Propojení klientských stanic se serverem je realizováno prostřednictvím Internetu. Pro komunikaci se využívá sady protokolů TCP/IP. [1]

6.1 Architektura databází

Existuje hned několik architektur, na kterých mohou být IS postaveny.

6.1.1 Centrální architektura

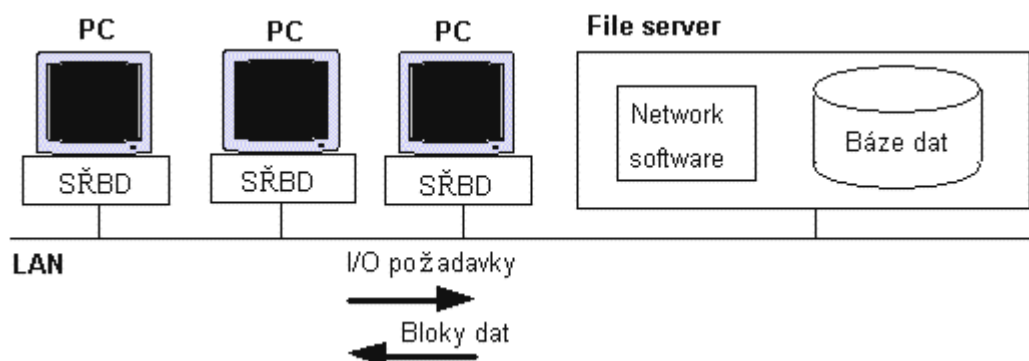
V této architektuře jsou data i SŘBD v centrálním počítači. Tato architektura je typická pro terminálovou síť, kdy se po síti přenáší vstupní údaje z terminálu na centrální počítač do příslušné aplikace, výstupy z této aplikace se přenáší na terminál, jak ukazuje Obrázek 3. Protože aplikační program i vlastní zpracování probíhá na centrálním počítači, který může zpracovávat více úloh, mají odezvy na dotazy určité zpoždění. [5]



Obrázek 3 – Centrální architektura, zdroj: [5]

6.1.2 Architektura file-server

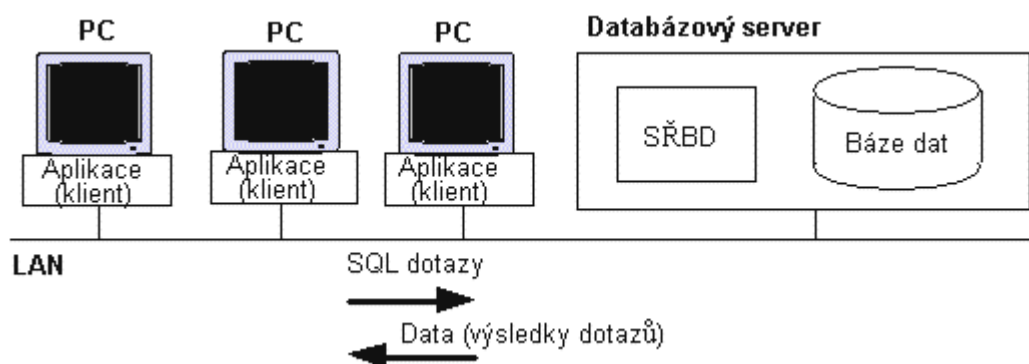
Tato metoda souvisí zejména s rozšířením osobních počítačů a sítí LAN. SŘBD a příslušné databázové aplikace jsou provozovány na jednotlivých počítačích, data jsou umístěna na file-serveru a mohou být sdílena, jak zachycuje Obrázek 4. Aby nedocházelo ke kolizím při přístupu více uživatelů k jedněm datům, musí SŘBD používat vhodný systém zamykání (položek nebo celých tabulek). [5]



Obrázek 4 - Architektura file-server, zdroj: [5]

6.1.3 Architektura klient-server

Architektura klient-server je založena na lokální síti (LAN), personálních počítačích a databázovém serveru. Na personálních počítačích běží program podporující např. vstup dat, formulaci dotazu atd. Dotaz se dále předává pomocí jazyka SQL na databázový server, který jej vykoná a vrátí výsledky zpět na personální počítač, jak ukazuje Obrázek 5. Databázový server je tedy nejvíce zatíženým prvkem systému a musí být tvořen dostatečně výkonným počítačem. Databáze využívá číselníky, tedy seznam objektů s jedinečnými označeními. [5]



Obrázek 5 - Architektura klient-server, zdroj: [5]

Nejčastěji se v IS objevuje architektura klient-server, která vyhovuje i náročným aplikacím a je využívána většinou renomovaných databázových firem. [5]

Tato práce je zaměřena na IS Helios Orange společnosti Asseco, jehož architektura s konkrétními prvky je nastíněna v kapitole 10.4.1.

7 O společnosti Asseco Solutions

Asseco Solutions je certifikovaným partnerem renomovaných společností IBM, Microsoft, Oracle a Hewlett Packard. Ve spolupráci s nimi dokáže uspokojit i poptávku zákazníků například po hardwaru, serverech, technologických novinkách, či v oblasti Client/Server řešení.

V současné době nabízí Asseco Solutions devět produktů – informačních systémů včetně

podpory a školení současných i budoucích uživatelů. Podrobně je rozebrán IS Helios ve třech modifikacích – IS Helios Green, IS Helios Red a IS Helios Orange. Jsou nastíněny rozdíly, stejně tak i znaky a přednosti každého software. Nejdetailněji je zpracován IS Helios Orange, protože právě do této modifikace je naprogramován modul, jehož návrh a tvorba je hlavním cílem této práce. [2]

8 Informační systém Helios Green

Helios Green je určen pro velké a středně velké společnosti, snadno se přizpůsobí konkrétním potřebám firmy a dokáže spolehlivě poskytovat informace podporující strategická manažerská rozhodnutí. Helios Green, kromě standardního jádra, jehož součástí je např. i CRM či Business Intelligence, obsahuje i specializované moduly, tvořící tzv. branžová řešení. [2]

8.1 Přednosti a znaky IS Helios Green

Kapitola obsahuje výčet pozitivních vlastností IS Helios Green podle [2]:

- **Přizpůsobivý** - ač svým rozsahem velký – přizpůsobí Helios Green své uspořádání pro každého z uživatelů tak, aby mu poskytoval náležitou jednoduchost a komfort.
- **Otevřený a kompatibilní** - Systém je tak připraven jak pro jednorázovou, tak pro stálou komunikaci s jinými systémy obchodních partnerů i systémy orgánů státní správy.
- **Standardizovaný** - Informační systém Helios Green je integrován s řadou standardních aplikací firmy Microsoft a respektuje českou legislativu i mezinárodní normy.
- **Technologicky vyspělý** - Helios Green využívá moderní platformu Microsoft .NET, která přináší do tvorby ERP systémů zcela nové možnosti, především v oblastech integrace s jinými aplikacemi, v zabezpečení systému či v datové komunikaci.
- **Bezpečný** - systém disponuje propracovaným přístupem k datům, ve kterém lze přístupová práva přesně definovat.
- **Obsahující branžová řešení** - Vlastní vývojové jádro umožňuje vytvářet nové moduly a funkční celky.
- **Vícejazyčný** - Komplexní systém Helios Green je schopen pracovat ve více jazykových mutacích.

9 Informačním systém Helios Red

Ekonomický a účetní systém Helios Red je moderní modulární účetní program pro zpracování komplexní podnikové agendy podnikatelů, malých a středních firem účtujících v účetnictví nebo

daňové evidenci. Helios Red zahrnuje oblasti finančního účetnictví, lidských zdrojů, skladového hospodářství, oběhu zboží a všech souvisejících a návazných agend. Je hardwarově nenáročný a ve většině případů nevyžaduje žádný upgrade technického vybavení. Zavedení tohoto účetního programu je velmi snadné a zvládne jej i sám uživatel. Datové výstupy lze z tohoto ekonomického softwaru exportovat ve standardních formátech pro další případné zpracování např. v aplikacích MS Office. Rovněž tiskové sestavy v programu je možné uživatelsky modifikovat s použitím integrovaného editoru. Kromě vytváření papírových sestav lze přímo z aplikace odesílat zprávy i e-mailem. [2]

9.1 Přednosti, znaky IS Helios Red

Kapitola nabízí seznam výhod IS Helios Green podle [2]:

- **Snadně konfigurovatelný** – z cenově zvýhodněných produktových balíčků lze snadno vybrat ten, který bude uživateli nejlépe vyhovovat. A v případě, že se po čase rozšíří požadavky, lze jednoduše přikoupit potřebný modul.
- **Podporovaný** - přehledné prostředí, intuitivní účelné ovládání, systém návodů a promyšlená ergonomie zpříjemní a zefektivní práci v systému. Prostřednictvím nejširší sítě technické podpory jsou zajištěny servisní služby na celém území České republiky.
- **Nenáročný na údržbu** - nevyžaduje žádnou průběžnou údržbu. Nástroje pro zálohování a automatickou kontrolu integrity dat zajistí bezpečný provoz.
- **Otevřený** - jakákoliv data a sestavy lze jednoduše exportovat do souborů standardních formátů a využívat je dále např. v aplikacích MS Office (MS Excel, MS Word apod.).

10 Informační systém Helios Orange

Helios Orange je technologicky vyspělý informační a ekonomický systém zefektivňující všechny běžné i vysoce specializované firemní procesy. Poskytuje dokonalý a aktuální přehled o situaci na trhu i uvnitř podniku, automatizaci rutinních operací, zefektivňování provozu, snižování nákladů a účinnou komunikaci. V mnoha úspěšných středně velkých firmách již řadu let pomáhá managementu v řízení všech potřebných oblastí, včetně např. CRM. [2]

10.1 Přínosy pro menší firmy

Menší firmy ocení analytický rozsah informačního systému bez nutnosti jakýchkoliv úprav či dodatečně vyvíjených prvků navyšujících cenu a prodlužujících dobu instalace systému. Velkou výhodou pro menší firmy je možnost pořízení modulů a oblastí, které opravdu potřebují, což výrazně snižuje investici do informačního systému. V případě růstu firmy nabízí informační

a ekonomický systém Helios Orange rozšíření o specializované oblasti a návazná řešení a zvýšení počtu uživatelů pracujících v systému. [2]

10.2 Přínosy pro středně velké firmy

Středně velké společnosti ocení především množství specializovaných oblastí a návazných řešení, které umožňují pokrytí procesů ve firmách podnikajících v různých oborech a odvětvích. Samozřejmostí u tohoto typu informačního systému je možnost jeho přizpůsobení specifickým potřebám a požadavkům řízení těchto firem. [2]

10.3 Přednosti, znaky IS Helios Orange

Podle [2] lze mezi největší přednosti IS Helios Orange zařadit tyto vlastnosti:

Stabilní a prestižní - podnikový systém prověřený časem i množstvím zákazníků mající většinový podíl v segmentu malých a středně velkých firem v ČR.

Komplexní - integrované řešení pro výrobu, dopravu a expedici, servis, řízení projektů, zemědělství a mnoho dalších.

Otevřený - umožňuje propojení s libovolným softwarem a prohlížení dat z jiných programů přímo v systému Helios Orange.

Bezpečný, standardní - podporuje všechny standardy elektronické komunikace.

Bezproblémový – snadná instalace, provoz i ovládání.

10.4 Moduly systému Helios Orange

Jednotlivé moduly informačního systému Helios Orange se podle [2] dělí do několika kategorií. Všechny kategorie zobrazuje Obrázek 6. V každé kategorii lze nalézt vždy alespoň jeden tematický modul.



Obrázek 6 - Kategorie modulů IS Helios Orange, zdroj: [2]

10.4.1 Jádru systému

Technologie klient-server, na které je tento ERP systém vystavěn, zajišťuje dostatečnou stabilitu a bezpečnost dat pro všechny typy firem. Výhradní použití MS SQL databázového serveru zajišťuje maximální rychlost a neomezenou práci s uloženými daty.

Architektura klient-server redukuje přenos dat po síti, protože dotazy jsou prováděny přímo na databázovém serveru a na personální počítač jsou posílány pouze výsledky. Např. pokud je mezi 10 000 záznamy pouze 100 záznamů, které splňují podmínku dotazu, pak na personální počítač putuje pouze těchto 100 záznamů. [5]

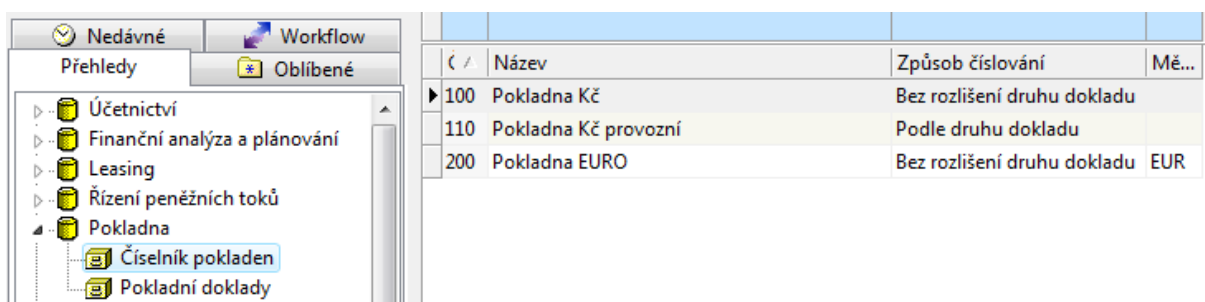
Celá komunikace probíhá podle [5] tímto způsobem:

- uživatel zadává dotaz (buď přímo v SQL nebo musí být do tohoto jazyka přeložen),
- dotaz je odeslán na databázový server,

- databázový server vykoná dotaz,
- výsledek dotazu je poslán zpět na vysílací počítač, kde je zobrazen.

10.4.2 Ekonomika a finance

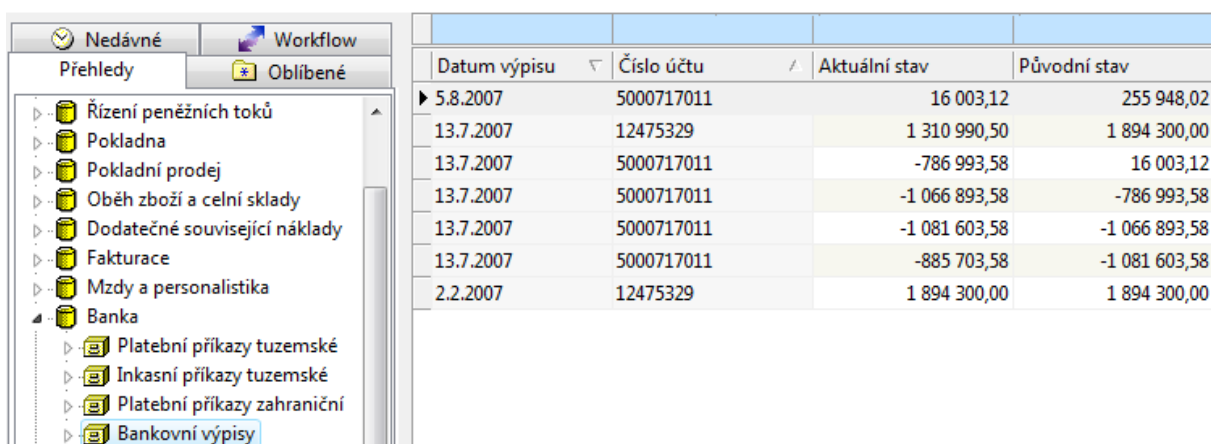
Modul Pokladna slouží k evidenci hotovostních pokladních dokladů v CZK i cizí měně. Umožňuje vést libovolné množství pokladen podle potřeb firmy a interních požadavků zvláště s ohledem na zajištění hmotné odpovědnosti. Obrázek 7 ukazuje několik typů pokladen. Modul Pokladna jednoduše pracuje s cizí měnou. Nabízí možnost automatizovaného účtování a automatického přepočtu bilančním kurzem.



Č /	Název	Způsob číslování	Mě...
▶ 100	Pokladna Kč	Bez rozlišení druhu dokladu	
110	Pokladna Kč provozní	Podle druhu dokladu	
200	Pokladna EURO	Bez rozlišení druhu dokladu	EUR

Obrázek 7 - Modul Pokladna, zdroj: vlastní

Modul Banka slouží k tvorbě platebních příkazů a k evidenci bankovních výpisů v korunách i cizí měně. Poskytuje možnost komplexní oboustranné elektronické komunikace s bankovními ústavy. Jednoduchá vazba na účetnictví zajišťuje snadné automatické dohledávání protistran účtů v případě úhrad předpisů. Modul Banka využívá vzory platebních příkazů tuzemských i zahraničních včetně inkasních plateb. Je vybaven pro práci s třetí měnou, či vytváření sdružených plateb a jejich automatickou identifikaci na řádcích bankovních výpisů. Snadno lze zobrazit i bankovní výpisy, jak ukazuje Obrázek 8.



Datum výpisu	Číslo účtu	Aktuální stav	Původní stav
▶ 5.8.2007	5000717011	16 003,12	255 948,02
13.7.2007	12475329	1 310 990,50	1 894 300,00
13.7.2007	5000717011	-786 993,58	16 003,12
13.7.2007	5000717011	-1 066 893,58	-786 993,58
13.7.2007	5000717011	-1 081 603,58	-1 066 893,58
13.7.2007	5000717011	-885 703,58	-1 081 603,58
2.2.2007	12475329	1 894 300,00	1 894 300,00

Obrázek 8 - Modul Banka, zdroj: vlastní

Modul Ekonomika zajistí, že při rozhodování o klíčových ekonomických záležitostech vaší společnosti budete mít snadný přístup k těm nejdůležitějším informacím.

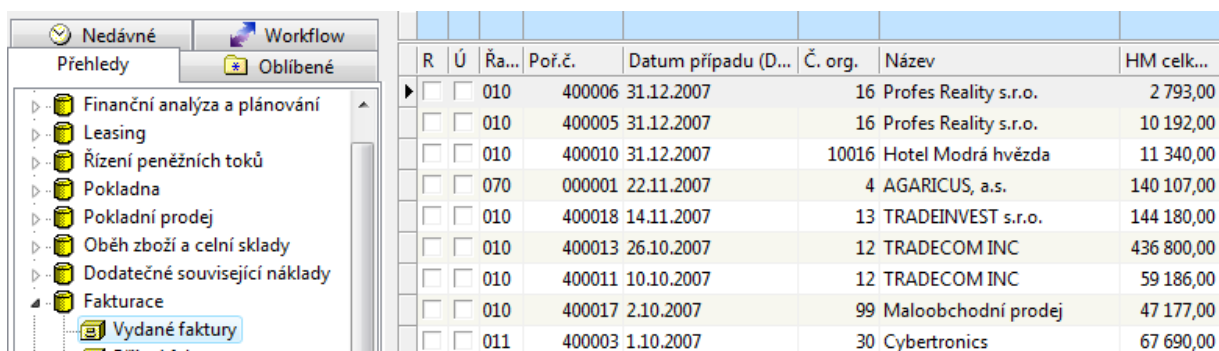
10.4.3 Obchod

Modul Obchod slouží k efektivnímu řízení skladových zásob z pohledu nákupu a prodeje zboží, výrobků, materiálů či služeb a je úzce spojen s modulem Sklad. Řeší problematiku objednávek, rezervací, optimalizace nákupů, cenotvorby apod. Podporuje automatické operace objednávání v závislosti na plánovaných potřebách prodeje či výroby a poskytuje kompletní vyhodnocovací nástroj obchodních aktivit.

10.4.4 Styk se zákazníky

Modul CRM slouží k evidenci a následnému zpracování kontaktů s potenciálními či stávajícími obchodními partnery. Pomáhá navazovat nové obchodní vztahy i udržovat ty se současnými obchodními partnery. Použití modulu CRM je velice široké, neboť není vázáno legislativou, a je tedy jen na zákazníkovi, jakým způsobem tento modul bude využívat.

Modul Fakturace slouží k vystavení faktur v hlavní i cizí měně a k vystavení vzájemných zápočtů. Vystavení souvisejícího dokladu (např. ve skladu) lze zjednodušit, zrychlit a zpřesnit převodem položek (či celých dokladů). Jednoduché generování dobropisu a realizačních dokladů poskytuje komplexní řešení vystavování všech typů dokladů, jak lze vidět na Obrázek 9. Evidence pošty umožňuje vést elektronickou knihu došlé a odeslané pošty včetně faktur.



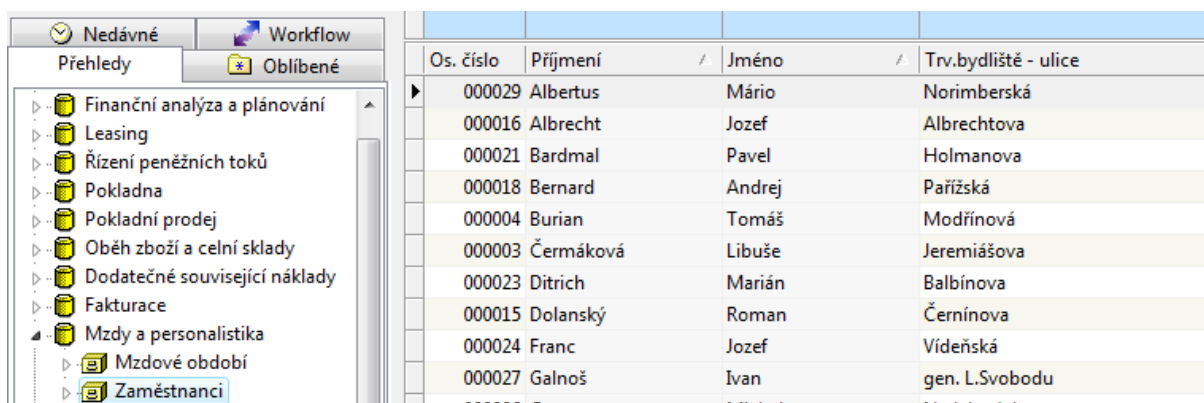
R	Ú	Řa...	Poř.č.	Datum případu (D...	Č. org.	Název	HM celk...
		010	400006	31.12.2007	16	Profes Reality s.r.o.	2 793,00
		010	400005	31.12.2007	16	Profes Reality s.r.o.	10 192,00
		010	400010	31.12.2007	10016	Hotel Modrá hvězda	11 340,00
		070	000001	22.11.2007	4	AGARICUS, a.s.	140 107,00
		010	400018	14.11.2007	13	TRADEINVEST s.r.o.	144 180,00
		010	400013	26.10.2007	12	TRADECOM INC	436 800,00
		010	400011	10.10.2007	12	TRADECOM INC	59 186,00
		010	400017	2.10.2007	99	Maloobchodní prodej	47 177,00
		011	400003	1.10.2007	30	Cybertronics	67 690,00

Obrázek 9 - Modul Fakturace, zdroj: vlastní

Modul Pokladní prodej zajišťuje spojení „pultového prodeje“ se skladovou evidencí. Dovybavením osobního počítače o speciální pokladní periférie vznikne zařízení s funkcemi počítače a kontrolní pokladny. Takové zařízení pak obsahuje peněžní zásuvku, zákaznický displej, tiskárnu pro tisk paragonů, snímač čárového kódu a případně i platební terminál pro bezhotovostní platby kartami.

10.4.5 Lidské zdroje

Modul Mzdy jako nejdůležitější součást řízení lidských zdrojů umožňuje efektivní zpracování všech typů mezd, snadnou evidenci personálních a mzdových údajů zaměstnanců, jak ukazuje Obrázek 10, či tisk veškerých sestav a formulářů pro jednotlivé úřady. Lze jej používat samostatně nebo i v návaznosti na modul Účetnictví a Banka. Vybrané formuláře z modulu lze posílat elektronicky přes Portál veřejné správy.



Os. číslo	Příjmení	Jméno	Trv.bydliště - ulice
000029	Albertus	Mário	Norimberská
000016	Albrecht	Jozef	Albrechtova
000021	Bardmal	Pavel	Holmanova
000018	Bernard	Andrej	Pařížská
000004	Burian	Tomáš	Modřínová
000003	Čermáková	Libuše	Jeremiášova
000023	Ditrich	Marián	Balbínova
000015	Dolanský	Roman	Čerínova
000024	Franc	Jozef	Vídeňská
000027	Galnoš	Ivan	gen. L.Svobodu

Obrázek 10 - Modul Mzdy a personalistika, zdroj: vlastní

10.4.6 Manažerské rozhodování

Manažerské vyhodnocování tvoří několik modulů a nástrojů integrovaných do systému Helios Orange. Umožňují komplexní pohled na data v ERP systému podle potřeb a zvyklostí zákazníka. Konkrétní použití jednotlivého nástroje závisí na požadovaném výsledku.

10.4.7 Výroba

Moduly výroby významně podporují komplexní řízení všech typů výroby. Na technickou přípravu výrobních procesů, která je schopna vytvářet a uchovávat „životopis“ výrobku, navazuje produktivní řízení a plánování efektivní výrobní činnosti. Integrované logistické a finanční moduly přímo podporují ekonomické přínosy plynoucí z nasazení tohoto systému ve výrobních společnostech.

10.4.8 Doprava

Modul je určen pro firmy, které ke své činnosti využívají nákladní nebo osobní vozidla a mechanismy, ať už jako hlavní obor firmy nebo pro režijní účely.

10.4.9 Převravní služby

Modul Převravní služby je určen pro firmy, zajišťující přepravu celokamionových (v terminologii Heliosu „Spedice“) i kusových (v terminologii Heliosu „Sberná služba“) zásilek do zahraničí a po tuzemsku. Modul zaručuje kontinuální sledování zásilek po celou dobu jejich přepravy a

umožňuje vystavit širokou škálu formulářů na základě jediného zadání údajů o zásilce. Kromě pozemních umožňuje také letecké a námořní zásilky. Evidenci firem zachycuje Obrázek 11.

Č. ... /	Název	Místo sídla	Z...	IČO
0	Firma, a.s.	Mělník	CZ	12345678
1	RISH A.S.	Praha 5		45308314
2	RZ, spol. s r.o.	Karviná		27081052
3	Gambosa	Praha 10		45023093
4	AGRICUS, a.s.	Valašské Klobouky		49709895
5	Kotrč, spol. s r.o.	Vrchlabí 1		25109324
6	Calocera	Trutnov		18698025
7	Pestrec			14613132
8	Lošák			18479871
9	COPRINUS, a. s.			13132132

Obrázek 11 - Modul Přepavní služby, zdroj: vlastní

10.4.10 Celní software

Modul Celní software je nástrojem ke kompletní přípravě celního řízení. Velice snadno lze připravit měsíční hlášení, libovolné celní deklarace a provést evidenci všech typů celních daňových skladů.

10.4.11 Zemědělství

Modul Zemědělství je nástrojem pro řízení zemědělství s přihlédnutím ke specifikům způsobeným vlivem počasí na zemědělskou výrobu. Pokrývá všechny procesy v zemědělské výrobě, jako například nedokončenou výrobu a další.

10.4.12 Servis

Modul Servis řeší komplexním způsobem problematiku společností, zabývajících se servisní a opravárenskou činností strojů a zařízení. A to jak ve vazbě na realizovaný prodej, dodávku a instalaci, tak i bez této vazby. Seznam výrobků, strojů, či zařízení ukazuje Obrázek 12. Modul Servis je úzce integrován s ostatními důležitými agendami využívanými v rámci řízení podnikových procesů.

Číslo	Název	Výrobní číslo	Datum výroby
001	Konferenční stůl sklo	H7241	25.1.2007 8:28:40
002	Křeslo čalouněné	A7125	6.2.2007 8:33:02
003	Křeslo čalouněné	A8450	11.4.2007 8:33:48
004	Jídelní stůl	B5043	11.7.2007 8:35:17
005	Křeslo čalouněné	B6078	11.7.2007 8:37:19
006	Křeslo čalouněné	A7214	11.4.2007 8:38:21
007	Jídelní stůl	A6154	5.3.2007 8:41:27
008	Jídelní stůl	B7152	11.5.2007 8:40:01
009	Jídelní stůl	B3046	13.8.2007 8:42:19

Obrázek 12 - Modul Servis, zdroj: vlastní

10.4.13 Řízení projektů

Modul Řízení projektů je efektivní nástroj pro řízení všech druhů zdrojů určených pro úspěšnou realizaci různorodých projektů. Helios Orange nabízí řešení komplexní podpory všem společnostem, jejichž aktivity se potýkají s potřebou řízení rozsáhlých komplexních zakázek, realizačních akcí atd. Řešení může být implementováno do jakékoliv společnosti, která zavádí nebo využívá projektové řízení, se zavedeným systémem jakosti v souladu s normami ISO 9000 - 2000.

10.5 Helios Space

Helios Space otevírá prostor ke spolupráci při rozvoji informačního systému Helios Orange. Je určený společnostem a jednotlivcům, kteří se na základě akceptace smluvních podmínek rozhodnou podílet na tvorbě externích doplňků pro Helios Orange. Členstvím v komunitě Helios Space získají právo vyvíjet a distribuovat tyto doplňky a k jejich vývoji mají k dispozici pro tyto účely zdarma dostupné nástroje. [2]

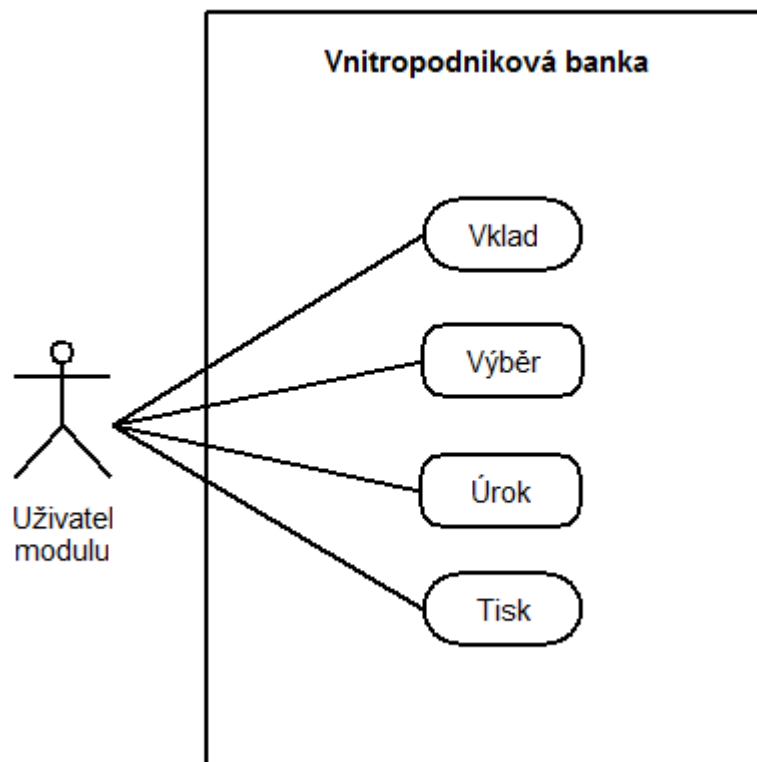
11 Návrh modulu Vnitropodniková banka

V kapitole jsou uvedeny autorovi nabyté zkušenosti z mnoha konzultací na dané téma. Vlastní návrh vznikl z konzultace s ekonomkou firmy Strojon Pardubice Ing. Kateřinou Jonášovou. Právě z její iniciativy vzešel nápad na tvorbu modulu Vnitropodniková banka, protože v IS Helios Orange takový modul chybí. Po zavedení IS Helios Orange ve firmě Strojon Pardubice tak přišla o možnost rychlého a efektivního úročení peněžních částek a vůbec o kontrolu vnitropodnikového účetnictví. Modul Vnitropodniková banka tak nahradí ručně prováděné výpočty v MS Excel a usnadní komunikaci mezi vnitropodnikovým účetnictvím a IS. Pro co největší možnost praktického využití naprogramovaného modulu v praxi byly schůzky sjednány s provozním programátorem společnosti Asseco solutions Ing. Michalem Jarkovským, který poskytl technickou podporu.

11.1 Praktická funkcionality tvořeného modulu

Vnitropodniková banka přeúčtovává výkony vnitropodnikových jednotek mezi sebou. [4]

Funkce, které musí modul podle zadaných požadavků splňovat, jsou vklad, výběr, tisk a úročení uložených peněz vnitropodnikových jednotek úrokem stanoveným firmou, jak zobrazuje Obrázek 13. Popis specifického prvního spuštění a funkčnost jednotlivých tlačítek popisují následující kapitoly.



Obrázek 13 - Use case diagram, zdroj: vlastní

11.1.1 První spuštění

Po prvním načtení formuláře pro každého zaměstnance uvidí uživatel vždy zaměstnancovi základní rozlišovací údaje osobní číslo, příjmení a jméno. Další kolonky ve formuláři, datum poslední operace a stav účtu, vyplněny nebudou, resp. stav účtu bude roven nule. Povinností každého uživatele při prvním spuštění tedy je, aby pomocí přítomného kalendáře navolil datum poslední rekapitulace (nejčastěji 31.12. minulého roku) a kliknutím na tlačítko Zúročit přednastaveným úrokem 0% odblokoval tlačítka Vklad a Výběr. Tuto situaci zachycuje Obrázek 14.

Novák Jan

Osobní číslo Poslední operace

Příjmení Úrok (%)

Jméno Stav účtu (Kč)

Úrok (Kč)

Částka (Kč)

Nový stav účtu (Kč)

Obrázek 14 - Modul Vnitropodniková banka - 1. spuštění, zdroj: vlastní

Dále může uživatel vložit k tomuto datu na účet tolik peněz, kolika zaměstnanec v tu dobu disponoval na vnitropodnikovém účtu firmy, jak ukazuje Obrázek 15.

Novák Jan

Osobní číslo Poslední operace

Příjmení Úrok (%)

Jméno Stav účtu (Kč)

Úrok (Kč)

Částka (Kč) Nový stav účtu (Kč)

Obrázek 15 - Modul Vnitropodniková banka - Vklad, Výběr, zdroj: vlastní

Pomocí tlačítka Uložit hned přesune nově vzniklé hodnoty do IS. Po zavření a opětovném spuštění formuláře se již kromě osobních údajů načtou údaje o stavu účtu a datu poslední změny.

11.1.2 Operace zúročit, vložit, vybrat

Stěžejní operací modulu je správně provedené úročení uložené částky vzhledem k zadanému (nejčastěji dnešnímu) datu. Má-li zaměstnanec na vnitropodnikovém účtu společnosti uloženy peníze, může kdykoliv uživatel, resp. účetní provést jejich zúročení, jak ukazuje Obrázek 16. Výše úroku lze vypočítat podle vzorce: $u = J_0 * i * n$, kde: J_0 = počáteční jistina, i = úroková sazba, n = období v letech.

The screenshot shows a window titled "Novák Jan" with a form for calculating interest. The form is organized into two columns. The left column contains input fields for "Osobní číslo" (9), "Příjmení" (Novák), "Jméno" (Jan), and "Částka (Kč)". The right column contains a "Poslední operace" field (27.4.2010 12:01), a date dropdown menu (27. dubna 2010), an "Úrok (%)" field (2), a "Zúročit" button, and three output fields: "Stav účtu (Kč)" (10000), "Úrok (Kč)" (64.11), and "Nový stav účtu (Kč)" (10064.11). At the bottom of the form are five buttons: "Vklad", "Výběr", "Uložit", "Tisk", and "Zavřít".

Obrázek 16 - Modul Vnitropodniková banka - Zúročit, zdroj: vlastní

I když zaměstnanec požaduje pouze vložení nebo výběr hotovosti, musí uživatel přesto nejprve provést zúročení, aby se potenciální vložená, resp. vybraná, částka přičetla, resp. odečetla, již k aktualizovanému stavu účtu se stanoveným úrokem. Jestliže by zaměstnanec nebo uživatel požadoval opakovaný vklad nebo výběr v jednom dni, lze tlačítka Vklad a Výběr odblokovat zúročením částky 0%-ním úrokem.

11.1.3 Operace uložit, tisknout, zavřít

Nedílnou součástí každého formuláře by měly být operace uložit, tisknout a přirozeně zavřít (i bez uložení změn). Tlačítkem Uložit lze v IS přepsat původní načtené hodnoty stav účtu a datum

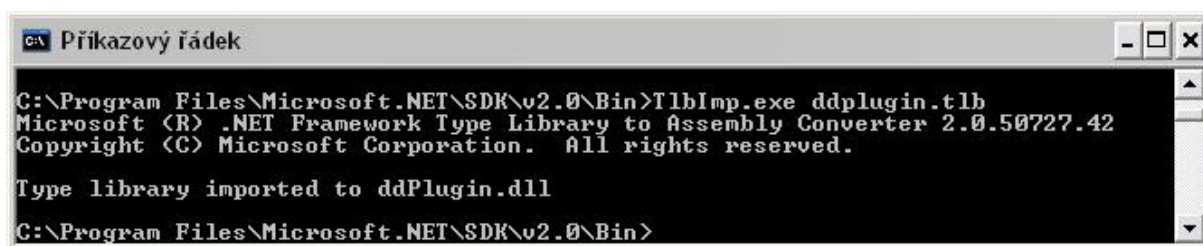
poslední změny. Jestliže se jedná o první zápis, v příslušné tabulce databáze se vytvoří pro načteného zaměstnance nové dva sloupce a hodnoty se po té uloží. Tlačítko Tisk vytiskne formulář v jeho současné podobě. Kdyby uživatel požadoval vytisknutí sestavy nebo přidat či odebrat některé údaje pro tisk z formuláře, lze efektivněji využívat funkci tisku přímo v IS. Tlačítko Zavřít, pakliže nebylo kliknuto na tlačítko Uložit, ukončí formulář bez provedení změn.

11.2 Technologie

Tvorba modulu pro informační systém Helios Orange bude provedena v prostředí MS Visual Studio. Toto programovací prostředí bylo zvoleno na základě dřívějších zkušeností autora ze studijních kurzů Univerzity Pardubice. Ačkoliv je většina modulů IS Helios Orange tvořena v integrovaném grafickém vývojovém prostředí Delphi, modul Vnitropodniková banka bude programován jazykem C#, který je založen na .NET-ové¹ technologii. Kompilací zdrojového kódu lze vytvořit „knihovnu“ (soubor s příponou dll), se kterou již IS, po jejím zaregistrování pověřenou osobou, umí pracovat.

12 Komunikace modulu s IS Helios Orange

Pluginy IS Helios Orange jsou podporovány technologií Microsoft COM². Prakticky je tato technologie popsána ve formátu tlb (ddplugin.tlb), který je nezávislý na vývojovém nástroji. Spolupráci technologie .NET a COM zajišťuje aplikace TlbImp.exe, která přes příkazový řádek importuje COM rozhraní do formátu dll, tedy do knihovny s .NETovým popisem rozhraní, jak ukazuje Obrázek 17. Vzniklý soubor ddPlugin.dll je nezbytný pro správnou funkčnost všech funkcí a procedur vázaných na IS Helios Orange.



```
C:\Program Files\Microsoft.NET\SDK\v2.0\Bin>TlbImp.exe ddplugin.tlb
Microsoft (R) .NET Framework Type Library to Assembly Converter 2.0.50727.42
Copyright (C) Microsoft Corporation. All rights reserved.

Type library imported to ddPlugin.dll
C:\Program Files\Microsoft.NET\SDK\v2.0\Bin>
```

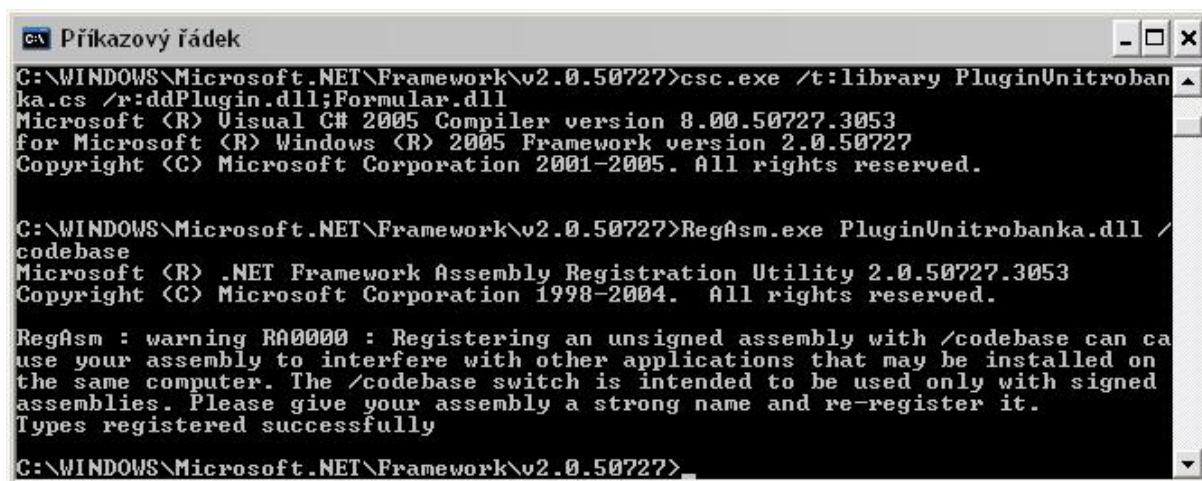
Obrázek 17 - Import COM -> .NET, zdroj: vlastní

Po úspěšném vytvoření knihovny ddPlugin.dll lze začít se samotným programováním navrženého modulu. Po napsání části kódu jej lze uložit do souboru PluginVnitrobanka.cs. IS Helios Orange však podobný typ souboru nerozezná, proto je nutná jeho kompilace pomocí překladače csc.exe. Po nastavení správné cesty a zadání vhodných parametrů pro kompilaci

¹ prostředí potřebné pro běh aplikací a nabízející jak spouštěcí rozhraní, tak potřebné knihovny

² rozhraní umožňující komunikaci programovacího jazyku s informačním systémem

vznikne opět knihovna, tentokrát nazvaná PluginVnitrobanka.dll. Parametr t říká překladači, že nově vzniklý soubor bude typu dll a parametr r odkazuje na jiné knihovny, které překladač při kompilaci souboru využívá. Nově vzniklý plugin je potřeba zaregistrovat, a to jak na počítači, kde má běžet, tak oprávněnou osobou na webu společnosti. Kompilaci i registraci pluginu ukazuje Obrázek 18.



```
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727>csc.exe /t:library PluginUnitroban
ka.cs /r:ddPlugin.dll;Formular.dll
Microsoft (R) Visual C# 2005 Compiler version 8.00.50727.3053
for Microsoft (R) Windows (R) 2005 Framework version 2.0.50727
Copyright (C) Microsoft Corporation 2001-2005. All rights reserved.

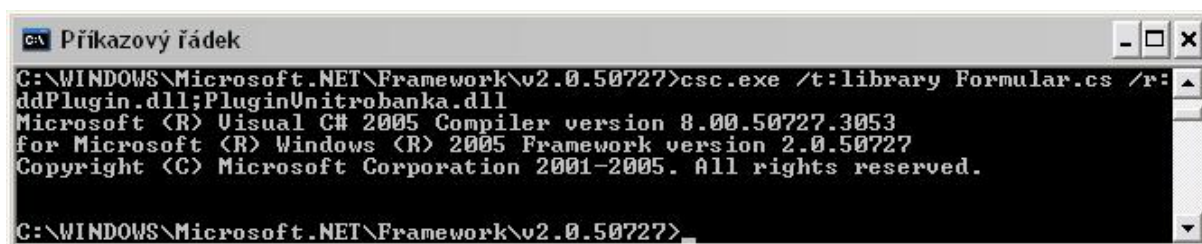
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727>RegAsm.exe PluginUnitrobanka.dll /
codebase
Microsoft (R) .NET Framework Assembly Registration Utility 2.0.50727.3053
Copyright (C) Microsoft Corporation 1998-2004. All rights reserved.

RegAsm : warning RA00000 : Registering an unsigned assembly with /codebase can ca
use your assembly to interfere with other applications that may be installed on
the same computer. The /codebase switch is intended to be used only with signed
assemblies. Please give your assembly a strong name and re-register it.
Types registered successfully

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727>
```

Obrázek 18 - Kompilace, registrace PluginVnitrobanka.cs, zdroj: vlastní

Protože kromě zdrojového souboru PluginVnitrobanka.cs je část kódu obsažena ještě v souboru Formular.cs, je potřeba proces kompilace zopakovat i pro druhý soubor s programovým kódem. Nyní se ale musí do parametru r dosadit kromě souboru ddPlugin.dll ještě nově vzniklý soubor PluginVnitrobanka.dll (Obrázek 19), protože překladač při kompilaci využívá obě knihovny. Knihovnu PluginVnitrobanka jako zdroj volání nového konstruktora, knihovnu ddPlugin jako zdroj rozhraní.

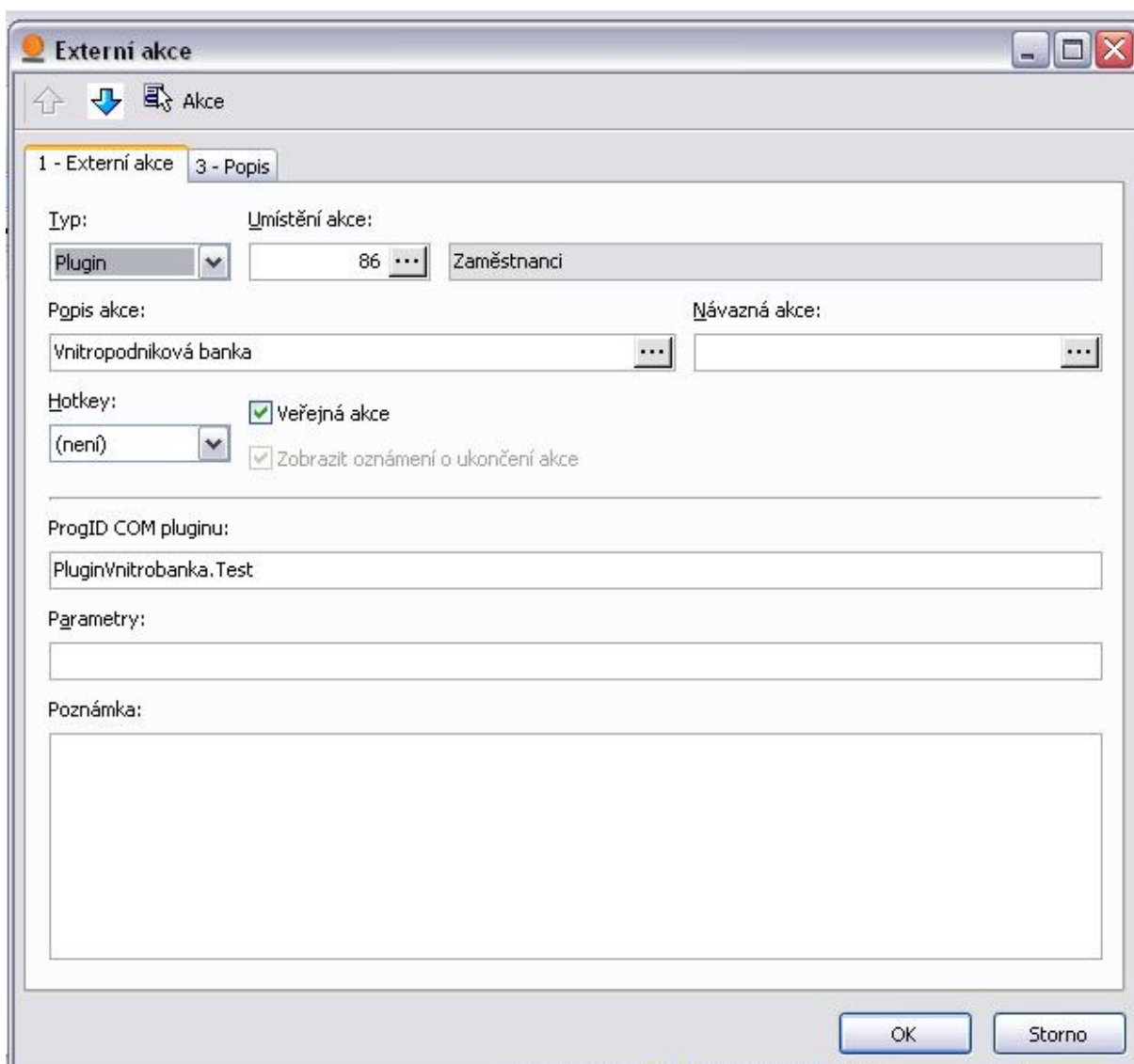


```
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727>csc.exe /t:library Formular.cs /r:
ddPlugin.dll;PluginUnitrobanka.dll
Microsoft (R) Visual C# 2005 Compiler version 8.00.50727.3053
for Microsoft (R) Windows (R) 2005 Framework version 2.0.50727
Copyright (C) Microsoft Corporation 2001-2005. All rights reserved.

C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727>
```

Obrázek 19 - Kompilace Formular.cs, zdroj: vlastní

Ani po zkompilování obou souborů ovšem nebude i sebejednodušší kód v IS Helios Orange fungovat. Pro jeho spuštění nebo testování je potřeba v modulu Nástroje přizpůsobení nadefinovat novou externí akci přesně tak, jak je zobrazeno na Obrázek 20.



Obrázek 20 - Externí akce, zdroj: vlastní

Kromě Typu, Umístění akce a Popisu akce je potřeba vyplnit políčko ProgID COM pluginu, které je nutné vyplnit podle šablony `jmenný_prostor.třída`, tedy `PluginVnitrobanka.Test`.

Externí akci Vnitropodniková banka, která je nyní připravena k použití, lze spustit v modulu Číselník zaměstnanců pravým tlačítkem myši na konkrétního zaměstnance a výběrem externí akce. Přehledně akci vystihuje Obrázek 21.



Obrázek 21 - Spuštění pluginu, zdroj: vlastní

13 Programování modulu Vnitropodniková banka

Tato kapitola popisuje samotné programování modulu včetně zajištění jeho komunikace s IS.

13.1 Using reference

Základní strukturu kódu v jazyku C# tvoří jmenný prostor a odkazy na některé další datové typy, které hlavní jmenný prostor využívá, ale nejsou v něm přímo definované. Jazyk C# se na podobné datové typy obsažené v jiných jmenných prostorech odkazuje pomocí referencí Using. Některé se implicitně vygenerují na základě zvolené šablony při přípravě projektu. Ostatní je potřeba vhodně doplnit tak, aby při kompilaci nedocházelo k chybám, že překladač některé datové typy nezná nebo neví, kde je najít. Množství referencí závisí na rozsahu kódu a především na použitých komponentech. Níže je vidět několik referencí kódu Formular.cs, kde kromě implicitně vygenerovaných systémových odkazů bylo potřeba doplnit reference na knihovny PluginVnitrobanka.dll a ddPlugin.dll.

```
using System;
using System.Drawing.Printing;
using System.Text;
using System.Windows.Forms;
using PluginVnitrobanka;
using ddPlugin;
```

Obdobným způsobem jsou zpracovány reference souboru PluginVnitrobanka.cs. Systémových referencí je méně, ale odkazy na knihovny ddPlugin.dll a Formular.dll obsahovat musí.

13.2 Jmenné prostory

Jak bylo popsáno výše, napsaný kód pro plugin je rozdělen do dvou souborů PluginVnitrobanka.cs a Formular.cs. Každý soubor má svůj vlastní jmenný prostor. Ty pro snazší orientaci dědí jméno po názvu souboru. Tedy jmenný prostor souboru PluginVnitrobanka.cs nese název PluginVnitrobanka. Stejným způsobem je nadefinován jmenný prostor i pro Formular.cs.

13.3 Třída Test jmenného prostoru PluginVnitrobanka

Každý jmenný prostor obsahuje jednu třídu. Třída pro jmenný prostor PluginVnitrobanka se jmenuje Test kvůli jejímu původnímu účelu. Každá třída má nadefinované rozhraní. Třída Test je naprogramována v rozhraní IHePlugin2. Právě toto rozhraní zajišťuje správnou funkčnost metod v IS Helios Orange. Situaci ilustruje kód níže, na kterém jsou navíc zobrazeny proměnné, definované příslušnými datovými typy. Datový typ IHeQuery, který je součástí rozhraní IHePlugin2 čerpá překladač pomocí reference using ddPlugin z knihovny ddPlugin.dll, které již byly uvedeny.

```
namespace PluginVnitrobanka {
    public class Test : IHePlugin2 {
        public string idecko;
        public IHeQuery zaz;
        public string osCi;
        public string jmeno;
        public string prijmeni;
        public IHeQuery ddd;
        public string datumPosledniZmeny;
        public string stavUctu;
    }
}
```

Datový typ IHeQuery lze použít v případě metod na SQL dotaz.

Třída Test obsahuje dvě metody. Metoda PartnerIdentificaton s návratovým datovým typem string vrací licenci IS, ve které byl modul tvořen. Příslušná licence HEIQ0100-21766 patří Univerzitě Pardubice.

```
public string PartnerIdentification() {
    return @"HEIQ0100-21766";
}
```

Metoda Run s návratovým typem void nevrací žádnou hodnotu, ale je volána Heliosem, když uživatel spustí externí akci typu plugin. Jako parametr je metodě předáno rozhraní IHelios, pomocí kterého kód pluginu komunikuje s Heliosem.

```
public void Run(IHelios Helios) {
    MessageBox.Show(@PartnerIdentification(), @"Verze IS Helios Orange, kde
    byl modul naprogramován");

    idecko = Helios.CurrentRecordID().ToString();
}
```

```

        zaz = Helios.OpenSQL("SELECT Cislo, Jmeno, Prijmeni FROM TabCisZam WHERE
id=" + @idecko);
        osCi = @zaz.FieldValues(0).ToString();
        jmeno = @zaz.FieldValues(1).ToString();
        prijmeni = @zaz.FieldValues(2).ToString();

        ddd = Helios.OpenSQL("SELECT _datumPosledniZmeny, _stavUctu FROM
TabCisZam_EXT WHERE id=" + @idecko);
        datumPosledniZmeny = @ddd.FieldValues(0).ToString();
        stavUctu = @ddd.FieldValues(1).ToString();

        Application.Run(new Form1(osCi, prijmeni, jmeno, datumPosledniZmeny,
stavUctu, idecko, Helios));
    }
}
}

```

Metoda nejprve zavolá výše popsanou metodu PartnerIdentification, poté funkcí CurrentID naplní proměnnou idecko hodnotou odpovídající záznamu, nad kterým je plugin spuštěn a převede ji na text. Následně proměnné zaz a ddd naplní funkcí OpenSQL dotazy, které mají standardní strukturu SELECT ... FROM ... WHERE Potom funkce FieldValues postupně plní všechny nadefinované proměnné tak, že parametr funkce FieldValues představuje pořadové číslo hodnoty nebyté SQL dotazem, ovšem takovým způsobem, že první parametr je zastoupen nulou. Všechny naplněné proměnné mají ve výsledku textovou hodnotu pro jejich snazší předání jako parametru metodou Application.Run přetíženému konstruktoru Form1 ve třídě Form1 ve jmenném prostoru Formular. Viz kapitola 13.4.1.

13.4 Třída Form1 jmenného prostoru Formular

Třída pro jmenný prostor Formular se jmenuje Form1 a je nadefinována v rozhraní Form, které je nejvhodnější pro práci s formulářovými prvky. Následující část kódu ukazuje nadefinování jmenného prostoru Formular, jeho třídy Form1 a všechny proměnné.

```

namespace Formular
{
    public class Form1 : Form
    {
        IHelios H;
        string id;
        double ucet;
        DateTime d;
        DateTime datum;
        TimeSpan r;
        double z;

        double urok;
        double urokCastka;

        double castka;

        string SQL;
    }
}

```

```
PrintDocument printDocument1 = new PrintDocument();
Bitmap memoryImage;
```

Všechny proměnné budou dále použity i vysvětleny.

13.4.1 Přetížený konstruktor Form1

Proměnné mohou být naplněny hodnotami již při jejich deklaraci, ale v kódu, který je předmětem této práce, by to nebylo vhodné a v některých případech ani možné. Většinu proměnných totiž třída Form1 přebírá od třídy Test jako parametry. Tyto parametry je nutno vypsat v přetíženém konstrukturu Form1 třídy Form1 ve stejném pořadí, v jakém byly odeslány metodou Application.Run z třídy Test. Dále je nutné nadefinovat každému parametru jeho datový typ.

```
public Form1(string osCi, string prijmeni, string jmeno, string
    datumPosledniZmeny, string stavUctu, string idecko, IHelios Helios)
{
```

Po předání parametrů již třída Form1 zná hodnoty proměnných, které byly naplněny a může s nimi v rámci konstrukturu Form1 pracovat. Aby mohly být tyto přijaté parametry použity i v jiných metodách třídy Form1 jako proměnné, je potřeba tyto proměnné, které byly nadeklarovány výše, naplnit hodnotami z předaných parametrů.

```
H = Helios;
id = idecko;
```

Proměnná H nyní nese rozhraní IHelios předané parametrem Helios. Proměnná id je naplněna hodnotou ID vybraného zaměstnance. Předání proběhlo přes parametr idecko.

Předání parametru do proměnné ucet je složitější, protože může nastat situace, zejména při prvním spuštění pluginu, kdy parametr stavUctu nebude nést žádnou hodnotu. Tato událost je ošetřena jednoduchou podmínkou s dotazem na prázdnotu parametru stavUctu. Jestliže parametr není prázdny, metoda double.Parse převede textovou hodnotu stavUctu na datový typ double a uloží ji do proměnné ucet. V opačném případě do proměnné ucet uloží nulu. Datový typ double umožní zobrazit a počítat s čísly s desetinnou čárkou.

```
if (stavUctu != "")
    ucet = double.Parse(stavUctu);
else
    ucet = 0;
labelUcetHod.Text = Math.Round(ucet, 2).ToString();
```

Proměnná ucet se poté vypíše, opět jako textová hodnota zaokrouhlená na dvě desetinná místa, do kolonky Stav účtu (Kč).

Velmi podobně probíhá předání parametru `datumPosledniZmeny`. V momentě, kdy parametr nenese prázdnou hodnotu, vypíše se do kolonky Poslední operace a současně se uloží do proměnné `d`, ale již v časovém formátu `DateTime`. V opačném případě se do proměnné `d` uloží dnešní datum.

```
if (datumPosledniZmeny != "") {
    labelPosledniHod.Text = datumPosledniZmeny;
    d = DateTime.Parse(datumPosledniZmeny);
}
else
    d = dateTimePicker1.Value;
VypoctiDny();
```

Následně je volána metoda `VypoctiDny`, jejíž tělo pochopitelně neleží uvnitř konstruktoru, ale pouze ve třídě `Form1` jako ostatní metody. Přesto je její kód rozebrán v této kapitole pro snazší pochopení.

```
Private void VypoctiDny()
{
    datum = dateTimePicker1.Value;
    r = datum - d;
    z = r.Days;
}
```

Hned potom, co se do proměnné `datum` uloží dnešní datum, se provede odečtení data poslední změny od dnešního data. Výsledek zachycuje proměnná `r` v datovém formátu `TimeSpan`, který je právě pro počty s daty nejvhodnější. Výsledek tohoto výpočtu bude využit dále při úročení. Datový typ `TimeSpan` je již ale nevhodný pro klasické numerické výpočty, proto se do numerické proměnné `z` uloží počet dnů z předchozího výpočtu. Není třeba ukládat menší časové jednotky, protože pro výpočet úroku jsou tyto postačující.

Zbytek předaných parametrů se do žádných proměnných neukládá. Prostým způsobem se načtou do příslušných kolonek `Osobní číslo`, `Příjmení`, `Jméno`.

```
labelOsCiHod.Text = osCi;
labelPrijmeniHod.Text = prijmeni;
labelJmenoHod.Text = jmeno;
this.Text = prijmeni + " " + jmeno;
}
```

Zároveň se parametry `prijmeni` a `jmeno` objeví v záhlaví formuláře.

13.4.2 Metoda Změna data v kalendáři

Tato metoda podchycuje případné změny v kalendáři formuláře. K těmto změnám může dojít v případě, kdy uživatel bude chtít např. zpětně vybrat peníze nebo provést pravidelnou rekapitulaci.

```
private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
{
```

```
        VypoctiDny();  
    }
```

Metoda jednoduše zavolá opět metodu `VypoctiDny`, která po případné změně data v kalendáři snadno přepočítá nový počet dnů od poslední operace k vybranému datu.

13.4.3 Metoda Úrok

Při zavolání této metody, kliknutím na tlačítko `Zúročit`, se ihned do proměnné `urok` načte číslo, které je zadáno uživatelem do úrokové kolonky. Pakliže zadaná hodnota není numerická, metoda `TryParse` to rozpozná a vyšle uživateli varovnou hlášku.

```
private void buttonUrok_Click(object sender, EventArgs e)  
{  
    bool result = double.TryParse(textBoxUrok.Text, out urok);  
    if (result == true)  
    {  
        urokCastka = ucet * urok * 0.01 * (z/365);  
        labelUrokCastkaHod.Text = Math.Round(urokCastka, 2).ToString();  
        ucet = ucet + urokCastka;  
        labelUcetNovyHod.Text = Math.Round(ucet, 2).ToString();  
        d = datum;  
        labelPosledniHod.Text = d.ToString();  
        buttonVklad.Enabled = true;  
        buttonVyber.Enabled = true;  
    }  
    else  
        MessageBox.Show("Zadaný úrok není ve správném formátu!");  
}
```

Poté, co proměnná `urok` obdrží číselnou hodnotu od `textBoxUrok`, se provede několik stěžejních operací. Podle vzorce pro výpočet úročené částky naplní proměnnou `urokCastka` odpovídající sumou pro úrok, na který má zaměstnanec nárok vzhledem ke stavu jeho účtu, vypsané úrokové sazbě a počtu dní od vložení peněz. Tato částka se okamžitě promítne v kolonce `Úrok (Kč)`. Současně se také připočte k původnímu stavu účtu a objeví se v kolonce `Nový stav účtu (Kč)`. Obě částky jsou před zobrazením zaokrouhleny na dvě desetinná místa metodou `Math.Round` a převedeny na text metodou `ToString`.

Aby činnost metody `Úrok` byla kompletní, musí aktualizovat proměnnou `d`, která nese datum poslední změny, o proměnnou `datum`, která obsahuje datum nastavené na kalendáři. Aktualizuje se tak datum poslední operace.

Dalším důležitým úkolem této metody je zpřístupnění tlačítek `Vklad` a `Výběr`. Tlačítka nejsou aktivní do doby, kdy uživatel provede operaci úrok. Zamezí tak případné uživatelské chyby, která by mohla nastat v případě, že bude chtít vybrat nebo vložit peníze na nezúročený účet.

13.4.4 Metody Vklad a Výběr

Po odblokování tlačítek Vklad a Výběr kliknutím na tlačítko Zúročit může uživatel do textBoxCastka vyplnit částku, kterou si přeje uživatel na svůj účet vložit nebo z něho vybrat, a právě přes tlačítka Vklad a Výběr provede příslušnou operaci.

Hodnota vložená uživatelem do textBoxCastka je opět ošetřena metodou TryParse proti případnému chybnému zadání nenumerické hodnoty. Je-li hodnota numerická, uloží ji do proměnné castka datového typu double a poté je kliknutím na tlačítko Vklad připočtena k nejaktuálnější hodnotě proměnné ucet. Nově aktualizovaná proměnná ucet se po zaokrouhlení na dvě desetinná místa vypíše v textovém formátu do kolonky Nový stav účtu (Kč). Po úspěšném provedení této operace se částka z textBoxCastka ihned vymaže. V případě, že zadaná hodnota do textBoxCastka není číslo, metoda uživatele varuje informační hláškou.

```
private void buttonVklad_Click(object sender, EventArgs e)
{
    bool result = double.TryParse(textBoxCastka.Text, out castka);
    if (result == true)
    {
        ucet += castka;
        labelUcetNovyHod.Text = Math.Round(ucet, 2).ToString();
        textBoxCastka.Text = "";
    }
    else
        MessageBox.Show("Zadaná částka není ve správném formátu!");
}
```

Metoda Výběr probíhá velmi podobným způsobem. Málý, ale podstatný rozdíl spočívá v operátoru mezi nově naplněnou proměnnou castka a opět nejaktuálnější hodnotou proměnné ucet. Oproti sčítání v metodě Vklad je nutné odečíst proměnnou castka od proměnné ucet. Pro zajištění korektního výběru se zde nachází ještě jedna podmínka která nedopustí vybrat větší obnos peněz, než jakým uživatel na svém účtu disponuje.

```
private void buttonVyber_Click(object sender, EventArgs e)
{
    bool result = double.TryParse(textBoxCastka.Text, out castka);
    if (result == true)
    {
        if (castka <= ucet)
        {
            ucet -= castka;
            labelUcetNovyHod.Text = Math.Round(ucet, 2).ToString();
            textBoxCastka.Text = "";
        }
        else
            MessageBox.Show("Nedostatečný zůstatek na účtu!");
    }
    else
        MessageBox.Show("Zadaná částka není ve správném formátu!");
}
```

Obě metody lze provést opakovaně i kombinovat, protože proměnná `ucet` obsahuje vždy nejnovější hodnotu.

13.4.5 Metoda Uložit, Tisk, Zavřít

Po správném provedení plánovaných operací na účtu zaměstnance lze nově vzniklé údaje uložit zpět do IS Helios kliknutím na tlačítko Uložit. Při spuštění této metody dojde k naplnění proměnné `SQL` textový řetězcem, který v sobě skrývá podmíněný SQL dotaz. V případě, že záznamy o stavu účtu a datu poslední změny existují, tedy nejedná se o zaevidování těchto údajů nového zaměstnance, proběhne SQL dotaz `UPDATE`, který staré záznamy v IS Helios nahradí aktuálními hodnotami z formuláře. V opačném případě, kdy je třeba poprvé zaevidovat údaje o stavu účtu a datu poslední změny, provede se SQL dotaz `INSERT`, který vytvoří u konkrétního zaměstnance dva nové sloupce a také je naplní hodnotami z formuláře. Použitím dotazu `REPLACE` je zajištěno nahrazení desetinné čárky desetinnou tečkou v proměnné `ucet`. IS Helios totiž s desetinnou čárkou neumí pracovat.

```
private void buttonUlozit_Click(object sender, EventArgs e)
{
    SQL = "IF EXISTS (SELECT * FROM TabCisZam_EXT WHERE ID="+ id +")
        BEGIN UPDATE TabCisZam_EXT SET _stavUctu=REPLACE(' +
ucet.ToString() + "','', '.') , _datumPosledniZmeny=' +
datum.ToString() +"' WHERE id=" + id + "
        END
    ELSE
        BEGIN INSERT TabCisZam_EXT (ID, _datumPosledniZmeny,
_stavUctu) VALUES (" + id + ", '" + datum.ToString() + "','',
REPLACE(' + ucet.ToString() + "','', '.'))
        END";
    H.ExecSQL(SQL);
    H.Refresh(false);
    MessageBox.Show("Data byla úspěšně uložena do IS Helios", "Uložit");
}
```

Proměnná `SQL` je využita hned vzápětí při spuštění metody `ExecSQL`. Tato metoda nevrací žádnou hodnotu, ale jen provede SQL dotaz za pomoci proměnné `SQL`. `ExecSQL` je součástí rozhraní `IHelios`, které je načtené v proměně `H` od jeho předání jako parametru `Helios`.

Předané rozhraní `IHelios` využívá také metoda `Refresh`, která na základě parametru `false` obnoví aktuální řádek s údaji o vybraném zaměstnanci.

Dříve než se metoda `Uložit` ukončí, proběhne ještě informační hláška o úspěšném uložení.

Před tím, než může být zavolána metoda `Tisk`, jsou nadeklarovány dvě metody, které metoda `Tisk` při zavolání sama volá. Metoda `CaptureScreen` uloží aktuální stav formuláře se správnými

velikostmi hran do proměnné `memoryImage`. Obsah proměnné `memoryImage` je dále načten již jako grafický objekt do proměnné `memoryGraphics`.

```
private void CaptureScreen ()
{
    Graphics myGraphics = this.CreateGraphics ();
    Size s = this.Size;
    memoryImage = new Bitmap(s.Width, s.Height, myGraphics);
    Graphics memoryGraphics = Graphics.FromImage(memoryImage);
    memoryGraphics.CopyFromScreen(this.Location.X, this.Location.Y, 0, 0, s);
}
```

Metoda `CaptureScreen` tedy zajistí objekt, který se má tisknout a metoda `printDocument1_PrintPage` vyše příkaz k tisku.

```
private void printDocument1_PrintPage(System.Object sender,
    System.Drawing.Printing.PrintPageEventArgs e)
{
    e.Graphics.DrawImage(memoryImage, 0, 0);
}
```

Po kliknutí na tlačítko `Tisk` se zavolají obě metody a formulář se vytiskne.

```
private void buttonTisk_Click(object sender, EventArgs e)
{
    CaptureScreen ();
    printDocument1.Print ();
}
```

Po ukončení práce s formulářem a případně jeho uložení a vytisknutí nezbyvá uživateli nic, než aplikaci ukončit tlačítkem `Zavřít`.

```
private void buttonZavrit_Click(object sender, EventArgs e)
{
    Close ();
}
```

13.4.6 Grafické komponenty

Protože část kódu je tvořena implicitně na základě výběru prvků a stanovení jejich atributů v grafickém prostředí, je potřeba je zavolat metodu `InitializeComponent`, která všechny komponenty obsahuje. Níže napsaný kód neobsahuje všechny komponenty, ale slouží jen jako ukázka implicitně vygenerovaného kódu pro štítek `Příjmení` a tlačítko `Vklad`.

```
private void InitializeComponent ()
{
    this.labelPrijmeni = new System.Windows.Forms.Label ();
    this.buttonVklad = new System.Windows.Forms.Button ();
```

Atributy, které se automaticky zapíší na základě změny jejich lze pozorovat na části kódu metody `InitializeComponent`.

```
this.labelPrijmeni.AutoSize = true;
this.labelPrijmeni.Location = new System.Drawing.Point(106, 107);
this.labelPrijmeni.Name = "labelPrijmeni";
this.labelPrijmeni.Size = new System.Drawing.Size(48, 13);
this.labelPrijmeni.TabIndex = 0;
this.labelPrijmeni.Text = "Příjmení";
```

```
this.buttonVklad.Enabled = false;
this.buttonVklad.Location = new System.Drawing.Point(109, 279);
this.buttonVklad.Name = "buttonVklad";
this.buttonVklad.Size = new System.Drawing.Size(75, 23);
this.buttonVklad.TabIndex = 16;
this.buttonVklad.Text = "Vklad";
this.buttonVklad.UseVisualStyleBackColor = true;
this.buttonVklad.Click += new
System.EventHandler(this.buttonVklad_Click);
```

Těla ostatních grafických komponentů mají podobnou strukturu. Liší se pouze v nadefinovaných atributech, proto není zapotřebí je všechny uvádět.

14 Zhodnocení

Po dokončení externího řešení Vnitropodniková banka k informačnímu systému Helios Orange byla okamžitě předvedena její funkčnost, byť s fiktivními daty, koncovému uživateli Ing. Kateřině Jonášové, která nad dosaženým výsledkem vyjádřila spokojenost, jak ukazuje Příloha 1. Modul naplnil očekávání a v budoucnu bude moci nabídnout své služby i v jiných firmách a podnicích.

Na základě závěrečného zhodnocení vyvstaly i návrhy na potenciální vylepšení, kterých by modul mohl nabývat. Jedná se o doplnění funkce Uložit. Další usnadnění práce a zpřehlednění výsledku by umožnil zápis nově vytvořených nebo aktualizovaných údajů do nově vytvořené tabulky Historie. Tabulka by měla obsahovat minimálně šest sloupců: datum provedení operace, stav účtu, úrok, vklad, výběr, nový stav účtu.

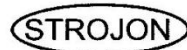
15 Závěr

V bakalářské práci byly popsány teoretické poznatky o informačních systémech. Část práce byla věnována podnikovým informačním systémům, zejména produktům společnosti Asseco solutions. Právě pro IS Helios Orange od Asseco Solutions byl v rámci této bakalářské práce navržen a naprogramován externí modul Vnitropodniková banka, který při testovacím provozu s fiktivními daty obstál a navíc je připraven pro praktické využití nejen firmou Strojon Pardubice. Z toho vyplývá, že cíl práce byl splněn.

16 Použité zdroje

- [1] *ASD Software : podnikové informační systémy a vývoj software* [online]. 2010 [cit. 2010-04-27]. IS DP. Dostupné z WWW: <<http://www.asd-software.cz/index.php?ID=137>>.
- [2] *Asseco Solutions : podnikový informační systém, ekonomický a účetní software* [online]. 2010 [cit. 2010-03-20]. Asseco Solutions. Dostupné z WWW: <<http://www.assecosolutions.eu/cz.html>>.
- [3] BASL, J; BLAŽÍČEK, R. *Podnikové informační systémy*. Praha : Grada Publishing, 2008. 288 s. ISBN 978-80-247-2279-5.
- [4] DVOŘÁK, K. *Účetnictví Dvořák* [online]. 2008 [cit. 2010-04-01]. Účetnictví. Dostupné z WWW: <<http://www.ucto-dvorak.cz/>>.
- [5] FARANA, R. *Databázové systémy* [online]. 1995 [cit. 2010-04-25]. Databázové systémy. Dostupné z WWW: <<http://www.fs.vsb.cz/books/dbacc20/dbacc01.htm#dbacc0223>>.
- [6] KOMÁRKOVÁ, J. *Kvalita webových geografických informačních systémů*. Pardubice: Univerzita Pardubice, 2008. 128 s. ISBN 978-80-7395-056-9.
- [7] KOMÁRKOVÁ, J. *Úvod do informačních systémů*. Pardubice: Univerzita Pradubice, 2006. 85 s. ISBN 80-7194-870-555-740-06.
- [8] MATYSKA, L. *Co jsou otevřené systémy* [online]. 2010 [cit. 2010-02-15]. Co jsou otevřené systémy. Dostupné z WWW: <<http://www.ics.muni.cz/zpravodaj/articles/472.html>>.
- [9] PELÁNEK, R. *Komplexní systémy* [online]. 2008 [cit. 2010-01-21]. Komplexní systémy. Dostupné z WWW: <<http://www.fi.muni.cz/~xpelane/IV109/slidy/komplexni-systemy.pdf>>.
- [10] PETERKA, J. *Jiří Peterka : eArchiv.cz* [online]. 2000 [cit. 2010-02-09]. Standard, norma, doporučení. Dostupné z WWW: <<http://www.earchiv.cz/a92/a209c110.php3>>.
- [11] SHARP, J. *Microsoft Visual C# 2008 : krok za krokem*. Brno : Computer Press, a.s., 2008. 592 s. ISBN 978-80-251-2027-9.
- [12] *Software quality consulting* [online]. 2008 [cit. 2010-02-09]. ISO 9126. Dostupné z WWW: <<http://www.angelfire.com/nt2/softwarequality/ISO9126.pdf>>.
- [13] ŠIMONOVÁ, S. *Databázové systémy II : SQL, přístup k datovým zdrojům*. Pardubice : Univerzita Pardubice, 2006. 100 s. ISBN 80-7194-845-4.

- [14] ŠMÍD, V. *Vladimír Šmíd : Úvod* [online]. 2007 [cit. 2010-03-19]. Životní cyklus informačního systému. Dostupné z WWW: <<http://www.fi.muni.cz/~smid/mis-zivcyk.htm>>.
- [15] TISNOVSKÝ, P. *Dynamický systém* [online]. 1999 [cit. 2010-01-13]. Dynamický systém. Dostupné z WWW: <<http://www.fit.vutbr.cz/~tisnovpa/publikace/diplomka/doc/node29.html>>.
- [16] ZELENÝ, M. *Risk.management.cz* [online]. 2009 [cit. 2010-04-10]. Proč definovat kvalitu. Dostupné z WWW: <<http://www.risk-management.cz/index.php?clanek=33>>.



*Strojon spol. s r.o.
Průmyslová ul.
533 01 Pardubice*

V Pardubicích duben 28, 2010

Posudek k praktické části bakalářské práce p. Jakuba Špidlena

V rámci bakalářské práce byl vytvořen modul Vnitropodniková banka v prostředí informačního systému Helios Orange. Tato část není v rámci daného informačního systému dosud řešena. Vytvořením tohoto funkčního modelu bude mít firma, která zaměstnancům nabízí spoření v podnikové bance, velice ulehčenou práci s administrativou spojenou s touto činností. Možné propojení na stávající informační systém navíc zajistí minimalizaci chyb a transparentnost jednotlivých spořicíh účtů. Ty se dosud evidovali v jiném programovém prostředí, které není s informačním systémem Helios Orange kompatibilní.

Autor akceptoval všechny požadavky kladené na vedení vnitropodnikové banky.

Ing. Kateřina Jonášová

ekonom firmy Strojon spol.s r.o. Pardubice