# 1 Úvod

Na základe schválenia nového vysokoškolského zákona Ministerstva školstva Slovenskej republiky bolo potrebné upraviť schému databázy pre fakultný informačný systém. Tieto zmenené podmienky štúdia (napr. iné podmienky štúdia pre rôzne študijné programy, zavedenie povinne voliteľných predmetov, trojstupňová forma štúdia, iné podmienky pre opakovanie ročníkov...) a tým aj zmenené požiadavky na informačný systém, ktoré už nebolo možné efektívne riešiť prostriedkami formulárov v Informixe. Keďže vystala požiadavka naprogramovať nové formuláre inými prostriedkami, vedenie sa rozhodlo prejsť z klasického textového linuxového klienta na typicky Windows-klienta informačného systému.

Teda cieľom diplomovej práce je navrhnúť a vyvinúť Informačný systém fakulty ako Windows aplikáciu, ktorá bude logikou, funkčnosťou a výzorom odpovedať pôvodnej aplikácii, tak aby prechod od starej verzie k novej verzii bol čo najprijateľnejší pre užívateľov.

Celá architektúra spracovania dát v databáze je založená na architektúre klient – server. Čiže ako každý Informačný systém (IS) založený na tejto architektúre sa aj tento systém skladá z dvoch nosných častí a to zo:

- Systému správy dát centrálna databáza obsahujúca konkrétne dáta
- Aplikácie umožňujúca prístup a prácu s dátami v databáze

### Systém správy dát

Systém sa skladá z jednej centrálnej PostgreSQL databázy uloženej na školskom serveri, ku ktorej aplikácia pristupuje po sieti. Tento systém bol upravený tak, aby spĺňal podmienky dané novým vysokoškolským zákonom.

#### Aplikácia

Používateľská aplikácia bola naprogramovaná vo vývojovom prostredí Borland Delphi 7 s použitím komponentov ZeosLib 6.1.5-stable (build at 2004-04-29 07:03:04). Inštalácia ZeosDBO sa nachádza na priloženom CD.

Medzi základné požiadavky bola zaradená podmienka, aby sa nová aplikácia čo najviac podobala na pôvodný informačný systém fakulty, a to nielen vzhľadom formulárov, ale najmä ovládaním a možnosťou ovládania pomocou zaužívaných klávesových skratiek.

Okrem samotných formulárov aplikácia musí poskytovať možnosti pre jednoduché parametrizovanie prostredia a pridávanie nových výstupných zostáv ako napr. Kontrolu štúdia. Na vytvorenie výstupnej zostavy boli použité komponenty Rave, ktoré sú súčasťou štandardnej inštalácie Delphi 7.

Pre zabezpečenie aplikácie bol použitý algoritmus SHA-1, ktorého implementácia bola použitá zo stránky <u>http://fundementals.sourceforge.net/index.html</u>.

# 2 Vytváranie aplikácií

# 2.1 Vývoj aplikácie

Vývoj čohokoľvek by mal byť vopred premyslený postup, aby sa zamedzilo plytvaniu prostriedkov a najmä času. Vývoj aplikácie je istým druhom kolobehu "života" aplikácie počas jej existencie. Aby sa aplikácia zdokonaľovala a hlavne aby sa približovala predstave akú má koncový používateľ, nemal by sa tento kolobeh zastaviť. Vývoj aplikácie sa skladá zo štyroch fáz (obr. 2.1.):

- 1 Návrh V tejto fáze sa špecifikujú požiadavky na koncový produkt a to v dvoch oblastiach:
  - Analýza systému V tejto oblasti sa definuje aké požiadavky môžeme požadovať od koncovej aplikácie po technickej stránke a taktiež definujeme to, čo už je implementované v existujúcej aplikácii.
  - Požiadavky používateľov Táto oblasť definovania požiadaviek na budúcu aplikáciu je najťažšia, pretože je veľmi ťažké dopracovať sa ku konečnému počtu požiadaviek u používateľov.
- 2 Vývoj Fáza, na ktorej v prevažnej väčšine pracujú už len samotní vývojári.
   Vytvára produkt podľa požiadaviek získaných z prechádzajúcej fázy návrhu.
- 3 Testovanie Do tejto fázy, fázy testovania aplikácie, je podľa môjho názoru dobré zapojiť ak nie samotných používateľov, tak aspoň tretie osoby, ktoré sa slangom nazývajú "testeri". Nie je veľmi vhodné, aby si v tejto fáze programátor testoval vlastnú aplikáciu, pretože najviac chýb v aplikáciách je odhalené tretími osobami pri používaní aplikácie v praxi.
- 4 *Nasadenie* Keď je aplikácia plne funkčná a odskúšaná prichádza na rad samotné nasadenie aplikácie v praxi.



Tento cyklus sa opakuje a tým sa aplikácia stáva komplexnejšia – viac uspokojuje potreby koncového používateľa.

# 2.2 Modulárne spracovanie

V dnešnej dobe vývoja univerzálnych softwarových produktov sa uprednostňuje modulárne programovanie. Toto programovanie má dve výhody. Prvá je tá, že do projektu sa môžu časom doprogramovať moduly s funkciami bez toho, aby sa musela preprogramovať celá aplikácia. Na univerzálnych softwarových aplikáciách pracuje veľké množstvo programátorov. Výhoda programovať modulárne je v tom, že jednotlivé programátorské tímy môžu pracovať len na vývoji jednotlivých modulov. Aplikáciu som programoval sám, ale rozdelil som zdrojový kód do modulov podľa ich funkčnosti. Napríklad všetky funkcie a procedúry týkajúce sa záložky číslo jeden som vložil ako samostatný modul *Tab1.pas* a pod.

# 2.3 Používateľské prostredie

Pre vývoj dobrej aplikácie je veľmi dôležité vytvorenie používateľského prostredia, ktoré musí spĺňať požiadavky koncového používateľa (obr. 2.3.). Zadanie bolo vytvoriť aplikáciu, ktorá sa čo najviac podobá vzhľadom a ovládaním už používanej existujúcej aplikácii. a Ovládanie aplikácie malo byť založené hlavne na používaní klávesových skratiek, ktoré zefektívňujú prácu používateľa.



obr. 2.3.

# 3 Klient/Server architektúra

Klient/Server je sieťová architektúra (obr. 3.), ktorá oddeľuje klienta (zväčša grafické používateľské prostredie) od samotných dát (najčastejšie databázový server). Každá inštancia klientského softwaru môže nezávisle posielať požiadavky na server. Táto architektúra je veľmi prispôsobivá, pretože ktorýkoľvek počítač alebo proces na sieti môže byť klientom alebo serverom. Serverový software zvyčajne, ale nie je to pravidlom, beží na silnejších a špecializovaných počítačoch, aby zvládali požiadavky od viacerých klientov. Naopak, klientské počítače sú zväčša obyčajné počítače.

Vlastnosti servera:

- pasívny prvok v sieti
- čaká na požiadavky
- obsluha požiadaviek a zasielanie odpovedí na požiadavky

Vlastnosti klienta:

- aktívny prvok v sieti
- posielanie požiadaviek
- čakanie na odpovede

V našom prípade v architektúre ako server vystupuje fakultný server, na ktorom je nainštalovaný databázový systém PostgreSQL. V ňom je navrhnutá databáza, ktorá je naplnená dátami. Server odpovedá na požiadavky odoslané inštanciami klientského softwaru spusteného u jednotlivých používateľov.



# 4 Relačný dátový model

# 4.1 Zmeny v dátovom modeli

Na základe schválenia nového vysokoškolského zákona Ministerstva školstva Slovenskej republiky bolo potrebné upraviť schému databázy pre fakultný informačný systém. Tieto zmenené podmienky štúdia (napr. iné podmienky štúdia pre rôzne študijné programy, zavedenie povinne voliteľných predmetov, trojstupňová forma štúdia, iné podmienky pre opakovanie ročníkov...). Väčšina zmien v dátovom modeli bola vykonávaná postupne za chodu systému. Tieto zmeny neboli nikde poznačené a ani inak zdokumentované. Zmeny robili väčšinou správcovia databázy. Ako príklad zmeny, ktorú som robil ja, môžeme uviesť zmeny podmienok pre opakovanie ročníka:

Kým v starom systéme mohol študent opakovať ročník jeden krát počas celého štúdia, v novom systéme môže študent opakovať ročník viackrát. V starom systéme boli na to vyhradené v tabuľke *student* dva atribúty:

- opakovanie nadobúdal hodnotu "a" ak študent opakoval ročník, inak mal hodnotu "n"
- datumop dátum začiatku školského roku, v ktorom študent opakoval ročník

V novom systéme bola na evidovanie opakovaní ročníkov vytvorená samostatná tabuľka *prerusenia*.

# 4.2 Štruktúra jednotlivých častí

V tejto kapitole sú popísané tabuľky, ktoré informačný systém používa a vzťahy medzi nimi. Niektoré tabuľky majú atribúty *uziv* a *dat\_zmen*. Tieto dva atribúty nesú informáciu o tom kto a kedy naposledy pracoval so záznamom. Tieto hodnoty sú dopĺňané a menené pomocou integrovaných triggrov v databáze.

Dátový model bol rozdelený do troch častí:

- osobné údaje

- študent
- zapísané predmety

Jednotlivé časti sú medzi sebou poprepájané. Časť *Osobné údaje* (ods. 4.2.1.) je prepojená s častou *Študent* (ods. 4.2.2.) pomocou prepojenia tabuliek *os\_udaje* a *student* cez atribút *rod\_cislo* (obr.4.2.2.). Časť *Študent* (ods. 4.2.2.) je prepojená s časťou *Zapísané predmety* (ods. 4.2.3.) pomocou prepojenia tabuliek *student* a *zap\_predmety* cez atribút *os\_cislo* (obr. 4.2.3.).

# 4.2.1 Osobné údaje

Túto časť modelu využíva záložka *Študent*.



obr. 4.2.1

#### stat

- zoznam štátov, ktorými sa napĺňa položka Štát na záložke Študent

#### kraj

- zoznam krajov, ktorými sa napĺňa položka Kraj na záložke Študent

#### okres

- zoznam okresov, ktorým sa napĺňa položka Okres na záložke Študent

#### obec

- zoznam obcí a miest, ktorým sa napĺňa položka Obec na záložke Študent

#### st\_prisl

- zoznam štátnych príslušností, ktorým sa napĺňa položka Štátna príslušnosť na záložke
 Študent

#### narodnost

- zoznam národností, ktorým sa napĺňa položka Národnosť na záložke Študent

#### os\_udaje

- hlavná tabuľka uchovávajúca osobné informácie o študentovi

- obc\_preuk číslo občianskeho preukazu. Toto číslo je naplnené len pri osobách so slovenskou štátnou príslušnosťou.
- pas číslo pasu. Tento údaje je evidovaný len pre zahraničných.
   študentov, čiže študentov s inou štátnou príslušnosťou ako je slovenská.
- st\_prisl id štátnej príslušnosti osoby. Tento údaj je cudzím kľúčom z entity *st\_prisl*. Ak je štátna príslušnosť slovenská ako doklad totožnosti pre študenta je evidované číslo občianskeho preukazu, inak číslo pasu.
- psc PSČ mesta trvalého pobytu. Pri vkladaní a úprave záznamu je používateľovi ponúknuté PSČ z tabuľky obec (*obec.psc*). Keďže mestá

a obce majú viacero PSČ, tento údaj **nie je** pevne **viazaný** na výber mesta alebo obce a používateľ ho môže zmeniť.

- ing parameter určujúci či študent je na bakalárskom alebo inžinierskom štúdiu. Ak má hodnotu '\*' znamená to, že študent je na inžinierskom štúdiu.
- ma\_zaujem\_o\_ing ak má študent na bakalárskom štúdiu záujem pokračovať v inžinierskom štúdiu, je nastavená hodnota '\*'.

# 4.2.2 Študent

Táto časť dátového využívajú záložky Zápis študenta, Prerušenia a Opakovania a Kontá študenta.



obr. 4.2.2

#### stupen

- zoznam študijných stupňov (bakalársky, inžiniersky, doktorandsky, magistersky)

#### odbory

- zoznam odborov štúdia, určených ministerstvom školstva (manažment, doprava, informatika, hospodárska informatika, počítačové inžinierstvo,...)

#### st\_program

- entita udržiavajúca informácie o študijných programoch určených školou, ktoré sú akreditované v rámci jednotlivých odborov. Napríklad študijný program *Riadenie dopravných systémov* patrí do odboru *Doprava*. Taktiež sú tu uvedené aj staré študijné odbory (*informačné a riadiace systémy, aplikovaná matematika, informatika*)

#### zameranie

 entita udržiavajúca informácie o zameraniach v rámci jednotlivých študijných programov. Napríklad zameranie *Aplikovaná informatika* patrí pod študijný program *Informačné systémy*.

#### opakovania

 - entita zaoberajúca sa opakovaniami ročníkov. Táto entita bola pridaná pri zmene modelu, pretože v starom systéme mohol študent opakovať len raz ale podľa nového zákona môže opakovať viackrát.

#### prerusenia

- entita udržiavajúca informácie o prerušení štúdia študenta

#### st\_hesla

- entita obsahuje iba prihlasovacie údaje nastavené administrátorom, nie aktuálne hodnoty

#### student

- entita udržiavajúca všetky študentské informácie každého študenta

# 4.2.3 Zapísané predmety

Táto časť sa zaoberá správou zapísaných predmetov. Využíva ju záložka Zápis predmetov.



obr. 4.2.3.

#### predmet\_bod

 - entita zaoberajúca sa parametrami predmetu, ktoré môžu byť iné pre každý školský rok. Je to časť študijného plánu, ktorá je rovnaká pre všetky odbory a zamerania.

#### predmet

- entita zoznamu všetkých predmetov

#### zabezpecuje

táto entita slúži na spojenie entít *premet* a *ucitel*. V nej sa nachádzajú informácie, ktoré sa môžu meniť pre predmet každý rok. Obsahuje dôležité informácie pre nastavenie zodpovedného učiteľa pre tvorbu podkladov pre predmet – plánovanie výuky, zabezpečenie vyplnenia skúšobných správ a pod

#### zap\_predmety

entita udržiavajúca informácie o predmetoch, ktoré boli zapísané jednotlivým študentom

# 4.3 Master-Detail vzťah

Master-Detail vzťah medzi tabuľkami znamená, že **na jediný záznam v tabuľke môže byť "naviazaných" viacej záznamov z inej tabuľky**. Toto previazanie sa vytvára pomocou cudzích kľúčov. Napríklad: V tabuľke *student* sú evidované informácie o jednotlivých študentoch. Každý záznam ma jedinečné osobné číslo študenta. V tabuľke *zap\_predmety* má každý študent informácie o zapísaných predmetoch. Previazanie týchto dvoch tabuliek sa deje použitím cudzieho kľúča *os\_cislo*. Master tabuľka v tomto prípade je *student* a Detail tabuľka je *zap\_predmety*.

Jednotlivé záložky aplikácie pracujú s tabuľkami, takže sú taktiež vo vzťahoch Master-Detail, pri použití hore uvedenej závislosti jednotlivých tabuliek medzi sebou. Na obrázku 4.3. sú Master záložky umiestnené nad Detail záložkami.



obr. 4.3.

# 5 Úvod do aplikácie

Aplikácia bola navrhnutá tak, aby sa čo v najväčšej miere podobala vzhľadom a ovládaním už existujúcej aplikácii napísanej v Informixe. Táto existujúca aplikácia sa členila na viacero záložiek podľa ich funkčnosti. Z tohto návrhu aplikácie bolo vychádzané aj pri navrhovaní novej aplikácie. Funkčnosť aplikácie bola rozdelená do šiestich logických záložiek:

- Študent spracováva osobné údaje každého študenta
- Zápis študenta všeobecné študijné informácie o študentovi
- Zápis predmetov prehľad a práca s predmetmi študenta
- Prerušenia a Opakovania spracovanie prerušení a opakovaní štúdia
- Kontá študenta rieši komunikáciu autorizačných informácií
- Financie kompletná správa finančného vzťahu medzi školou a študentom

Na každej záložke sú funkčné prvky, ktoré slúžia pre hľadanie, editovanie, pridávanie a rušenie jednotlivých záznamov. Taktiež pre každé zobrazenie je deklarovaná samostatná trieda, a tieto deklarácie sú uložené v osobitných súboroch. Napríklad pre zobrazenie "Zápis študenta" je vytvorená trieda cTab2 a ta je deklarovaná v súbore tab2.pas.

# 5.1 Inštalácia aplikácie

# 5.1.1 Klientská inštalácia



obr. 5.1.1.

Inštaláciu softwaru bola vytvorená v programe dodávanom s inštaláciou Delphi spolu InstallShield **Express** Borland Limited Edition. Tento program ponúka veľké množstvo nastavení pri vytváraní inštalačného balíčka. V prípade tejto aplikácie stačilo zabezpečiť nasledovné úlohy:

- vytvorenie adresára kde sa má aplikácia nainštalovať. Používateľ ju môže meniť.
   Štandardne je nastavená [Program Files]\ISFRI
- skopírovanie spustiteľného súboru aplikácie do inštalačného adresára.
- skopírovanie adresára Pictures aj s jeho obsahom do inštalačného adresára.
- vytvorenie odkazov na aplikáciu na pracovnej ploche a v ponuke Štart

Výsledkom vytvorenia inštalačného balíčka je jeden spustiteľný súbor *setup.exe*, ktorý zabezpečí všetky hore uvedené úlohy.

### 5.1.2 Zmeny na serveri

Bezpečnostné nastavenia databázového servera dovoľujú definovať IP adresy alebo rozsahy IP adries, ktoré majú prístup na databázu (viac ods. 6.1.4). Z tohto dôvodu pri inštalácií aplikácie na ďalší počítač, ktorého IP adresa nie je v povolených IP adresách pre prístup na server, musí administrátor pridať túto novú IP adresu do konfiguračného súboru.

### 5.1.3 Inštalácia pre ďalší vývoj aplikácie

Pre ďalší vývoj aplikácie si programátor musí zabezpečiť nasledovné veci:

- štandardná inštalácia Delphi 7 a vyššie.
- Inštalácia komponentov ZeosLib 6.1.5-stable (build at 2004-04-29 07:03:04) a vyššie (inštalácia je na priloženom CD).
- administrátor databázového servera musí povoliť prístup na databázu z jeho IP adresy.
- zdrojové kódy aplikácie sú na priloženom CD spolu s elektronickou verziou diplomovej práce.

# 5.2 Vzhľad a ovládanie aplikácie

Aplikácia pozostáva z jedného pracovného okna (obr. 5.2.), na ktorom sa nachádza hlavné menu(1), zoznam záložiek(2), obsah aktuálnej záložky(3) a informačný pás(4).

# 5.2.1 Hlavné menu

V hlavnom menu si môžeme meniť hlavné parametre aplikácie, prípadne vykonávať všeobecné úlohy.

6 FRKIS v.1.13			
Aplikácia Nastavenia Pomoc	100 1 1		C
1 Student 2 Zapis stud. 3 Zapis predm. 4 Pre	rus&Opako	o Konta s	<u>6</u> Financie
_Osobné údaje	SS PIL	evious	NEXL
Rodné číslo:			
Meno: Pri	ezvisko:		
Ulica: Ob	ec:		
	1940		
PSČ: Okr	95:		
V-t	8		
Kraj: Stat	5		
Štátna príslušnosť Ná	odnosť		-
Telefón: Císlo ob. p.:	<u> </u>		
Stredná škola:			
			_
	-	2	
Query Add Update Re	move		Detail
4			
/yberte si z ponuky odomknutých tlačidiel	Con	nected   15	.4.2006 0:32:19

V nasledujúcom texte sú opísané jednotlivé pod-menu a ich funkčnosť.

### Aplikácia

Všeobecné úlohy pre aplikáciu, do ktorých patrí:

Koniec – ukončenie aplikácie v ktoromkoľvek okamžiku jej behu. Táto položka zabezpečí uvoľnenie alokovanej pamäte aplikáciou. Táto položka má aj vlastnú klávesovú skratku (CTRL + E) pre rýchlejší prístup.

#### Nastavenia

Toto pod-menu obsahuje položky pre zmenu globálnych nastavení aplikácie ako sú:

Školský rok – v systéme pri pridávaní nového predmetu študentovi je použitý aktuálny školský rok na vygenerovanie zoznamu predmetov. Ak si praje používateľ aplikácie pridať predmet z iného školského roku, použije túto položku. Pri zmene školského roku sa automaticky zmení zoznam predmetov.

Táto hodnota je udržiavaná ako globálna premenná aplikácie, deklarovaná v triede TMainVariables.

*Počet termínov* – podľa školského poriadku je pevne definovaný maximálny počet termínov pre absolvovanie skúšky. Táto hodnota je kontrolovaná pri pridávaní a aktualizovaní informácie o zapísanom predmete. Jeho hodnota je definovaná v triede TMainVariables. Implicitne je nastavená na hodnotu tri.

#### Pomoc

V tomto pod-menu môže používateľ nájsť pomocnú položku:

- O programe – informatívne údaje o aplikácii

# 5.2.2 Zoznam záložiek

Keďže aplikácia je rozložená do šiestich logických celkov podľa funkčností, a zároveň musí ostať kompaktná, bol na ich zobrazenie použitý komponent *TabSheet*, a tak jednotlivé celky implementovať do jeho záložiek. Prepínanie medzi jednotlivými záložkami je možné pri použití klávesovej skratky ALT + číslo záložky. Napríklad: ALT + 1 je pre prvú záložku, ALT + 2 pre druhú atď. Týmto bol docielený kompaktný vzhľad a zároveň dobrú ovládateľnosť aplikácie. Ďalšie rozširovanie aplikácie o ďalšie záložky je možné cez pridanie ďalšej záložky v *TabSheet*-e.

### 5.2.3 Informačný pás

Informačný pás (*Status bar*) (obr. 5.2.3.) bol vytvorený za účelom informovania používateľa o aktuálnom stave aplikácie, ako aj o spätnú komunikáciu s používateľom. Pre informačný pás bol použitý komponent *StatusBar*. Bol rozdelený do troch menších častí:

Vyberte si z ponuky odomknutých tlačidiel	Connected	25.4.2006 23:29:04
obr. 5.2.3.		

 prvá časť vľavo - slúži pre komunikáciu s používateľom. Informuje v akom stave sa nachádza aplikácia. Napríklad hláškou "*Vyberte si z ponuky* odomknutých tlačidiel" informuje používateľa o stave čakania na stlačenie tlačidla z prístupných tlačidiel.

- stredná časť pásu informuje o tom, či pripojenie aplikácie na databázový server je stále aktívne.
- tretia časť, úplne v pravo, informuje o aktuálnom dátume a čase.

#### 5.2.4 Ovládanie aplikácie

#### Klávesové skratky

Pre ľahšiu prácu s aplikáciou boli použité vo veľkej miere klávesové skratky. Každé tlačidlo má svoju vlastnú klávesovú skratku. Tá bola navrhnutá tak, aby pozostávala zo stlačenia tlačidla ALT a príslušného písmena. Toto písmeno bolo volené tak, aby bolo ovládanie aplikácie intuitívne aj pre laického používateľa, použilo sa prvé písmeno z textu tlačidla alebo ďalšie v poradí, ktoré bolo možné použiť. Napríklad pre tlačidlo Query bolo zvolené tlačidlo Q, čiže sa naň dá pristúpiť klávesovou kombináciou ALT + Q alebo stlačením tlačidla Q. Tieto klávesové skratky pre jednotlivé komponenty sú viditeľné tak, že príslušné písmeno je v texte komponentu podčiarknuté.

#### Klávesy Enter, ESC a Tab

Ak je používateľ v móde hľadania, pridávania alebo zmeny záznamu, pomocou tlačidla Enter spustí vykonanie konkrétnej akcie. Napríklad pri pridávaní nového záznamu, keď používateľ vloží všetky potrebné údaje do položiek a stlačí klávesu **Enter**, **spustí** zapísanie nového záznamu do databázy. Stlačením klávesy **ESC** pri vkladaní údajov na záložke, sa aktuálna akcia **zruší** a aplikácia sa vráti do stavu pre začatím akcie. Tlačidlo **Tab** slúži na **prepínanie** medzi prístupnými položkami na záložke.

#### Tlačidlá

Tlačidlá v aplikácií slúžia viac menej na spúšťanie akcií ako sú vyhľadávanie, upravovanie, vkladanie a mazanie záznamov (*Query*, *Add*, *Update*, *Remove*). Ak

výsledkom vyhľadávania je množina záznamov, tak na prechádzanie po jednotlivých záznamoch nám slúžia tlačidlá *Next* a *Previous*. Ďalšie slúžia na prepínanie medzi jednotlivými záložkami (*Master*, *Detail*), ktorých dáta sú vo vzťahu Master-Detail (viď. ods. 4.3.).

**Tlačidlo** Query – slúži na odomknutie a vyčistenie políčok, pre načítanie podmienok vyhľadávania

**Tlačidlo** – slúži na odomknutie a vyčistenie políčok, pre načítanie nového záznamu, ktorý chce používateľ vložiť

**Tlačidlo** Update – slúži a odomknutie políčok naplnených hodnotami aktuálneho záznamu, ktorý chce používateľ zmeniť

**Tlačidlo** <u>**Remove**</u> – pomocou tohto tlačidla sa vykoná odstránenie aktuálneho záznamu z databázy

**Tlačidlo** <u>Next >></u> – ponúka možnosť prepnutia na ďalší záznam v množine nájdených záznamov

**Tlačidlo <- Previous** – ponúka možnosť prepnutia na predchádzajúci záznam v množine nájdených záznamov

**Tlačidlo** — slúži na prepnutie na záložku typu Master

**Tlačidlo** <u>**Detail**</u> – slúži na prepnutie na záložku typu Detail

### Spracovanie načítaných hodnôt

Ak pri hľadaní používateľ nezadá žiadny reťazec do políčka, podľa ktorého sa dá vyhľadávať, hľadaný reťazec sa nahradí znakom "%", ktorý v SQL syntaxy znamená

reťazec rôznej dĺžky. Pre explicitné definovanie reťazca rôznej dĺžky použijeme znak "\*". Napríklad:

Používateľ chce vyhľadávať osobné záznamy všetkých ľudí, ktorí sa narodili v roku 1982, a ktorých mená sa začínajú na písmeno "O". Vyhľadávanie na položke *Osobne údaje* je možné podľa troch položiek a to podľa rodného čísla, mena a priezviska. Preto používateľ vykoná nasledovné kroky:

- prepne sa na záložku Osobné údaje
- stlačí ALT + Q alebo stlačí tlačidlo Query
- do položky rodné číslo napíše hodnotu "82\*"
- do položky Meno napíše hodnotu "O\*"
- stlačí Enter

Keď používateľ vykoná hore uvedené kroky, na databázový server sa vyšle nasledujúca SQL požiadavka:

### SELECT rod\_cislo,

meno, priezvisko, obc\_preuk, pas, narodnost, st\_prisl, ulica, obec, psc, okres, ing, telefon, ma\_zaujem\_o\_ing FROM os\_udaje WHERE os\_udaje.rod\_cislo LIKE '82%' AND os\_udaje.meno LIKE '0%' AND os\_udaje.priezvisko LIKE '%'; Ako môžeme vidieť v podmienkach WHERE, znaky "\*" sa nahradili znakmi "%" (v podmienkach pre rodné číslo a meno) a nezadaný reťazec (podmienka pre priezvisko) sa celá nahradila znakom "%".

Pri nevyplnení ani jednej položky určenej pre vyhľadávanie a stlačení klávesy *Enter*, aplikácia požiada o potvrdenie od používateľa o zaslaní požiadavky pre vypísanie všetkých záznamov záložky. V tomto prípade by sa vyslala nasledujúca SQL požiadavka:

SELECT rod\_cislo,

meno, priezvisko, obc\_preuk, pas, narodnost, st\_prisl, ulica, obec, psc, okres, ing, telefon, ma\_zaujem\_o\_ing FROM os\_udaje WHERE os\_udaje.rod\_cislo LIKE '%' AND os\_udaje.meno LIKE '%'

# 5.3 Všeobecne o aplikácii

# 5.3.1 DBLookupComboBox-y



použité komponenty typu DBLookupComboBox, ktoré ponúkajú rovnaké použitie ako komponenty s tým že

sú

DBLookupComboBox-y svoje hodnoty získavajú Pre ľahší a rýchlejší beh aplikácie sa tieto komponenty naplnia hodnotami z databázy pri štarte aplikácie. Priebeh inicializácie

sledovať v tzv. logovacom okne, ktoré je zobrazené pri štarte aplikácie (obr. 5.3.1).

# 5.3.2 Spájanie reťazcov v SQL

Pri zobrazení názvu predmetu v DBLookupComboBox-e na záložke Zápis predmetov, bolo potrebné mať v názve predmetu uvedené číslo, názov a staré číslo predmetu. Preto bola použitá operácia spájania reťazcov pomocou operátora "II". Takéto spájanie reťazcov má jednu nevýhodu, že ak je aspoň jeden zo spájaných reťazcov hodnoty NULL, tak celý výsledný reťazec je NULL. Pre odstránenie tohto problému bola použitá funkcia COALESCE(arg1, arg2), ktorá zabezpečí, že ak hodnota argumentu arg1 je NULL, tak vráti argument arg2, inak vráti argument arg1. V mojom prípade funkcia pri NULL hodnote argumentu argl vráti prázdny reťazec: (COALESCE(arg1, '')). Taktiež niektoré nové predmety nemajú definované staré číslo predmetu, preto tu bola použitá funkcia COALESCE pre náhradu NULL hodnôt starého čísla novým číslom predmetu.

# 5.3.3 JOIN

V SQL požiadavke na spojenie dvoch tabuliek cez jeden alebo viacej stĺpcov, použijeme jednoduchú podmienku WHERE. Uvedieme si vysvetlenie na jednoduchom príklade:

Máme tabuľku zamestnancov	( <i>zamestnanec</i> ):
---------------------------	-------------------------

Zamestnanecké_ID	Meno_zamestnanca
01	Jozef Mrkva
02	Juraj Ozimina
03	Petra Hradská
04	Monika Sivá

Máme tabuľku objednávok(objednávka):

Číslo_výrobku	Produkt	Zamestnanecké_ID
124	Tlačiareň	01
541	Stôl	03
741	Stolička	03
313	Kvetina	

Ak chceme spojiť tieto dve tabuľky môžeme použiť SQL príkaz:

SELECT zamestnanec.Meno\_zamestnanca, objednávka.Produkt FROM zamestnanec, objednávka WHERE zamestnanec.Zamestnanecké\_ID = objednávka.Zamestnanecké\_ID

Dostaneme výslednú tabuľku:

Meno_zamestnanca	Produkt
Jozef Mrkva	Tlačiareň
Petra Hradská	Stôl
Petra Hradská	Stolička

Ak by sme chceli vypísať všetkých zamestnancov v tabuľke a k ním produkty, ktoré si objednali a to aj tých, ktorí si nič neobjednali, musíme použiť klauzulu JOIN. Poznáme tri druhy JOIN-ov:

#### **INNER JOIN**

Príklad:

SELECT zamestnanec.Meno\_zamestnanca, objednávka.Produkt FROM zamestnanec INNER JOIN objednávka ON zamestnanec.Zamestnanecké\_ID = objednávka.Zamestnanecké\_ID

Vráti všetky riadky z obidvoch tabuliek kde je zhoda v podmienke, čiže funguje ako obyčajný SELECT, ktorého príklad sme si uviedli hore.

Výsledok:

Meno_zamestnanca	Produkt
Jozef Mrkva	Tlačiareň
Petra Hradská	Stôl
Petra Hradská	Stolička

### **LEFT OUTER JOIN**

Príklad:

SELECT zamestnanec.Meno\_zamestnanca, objednávka.Produkt FROM zamestnanec LEFT JOIN objednávka ON zamestnanec.Zamestnanecké\_ID = objednávka.Zamestnanecké\_ID

Vráti všetky riadky z prvej tabuľky (*zamestnanec*), a to aj v tom prípade, ak nie je zhoda v podmienke s druhou tabuľkou (*objednávka*).

Výsledok:

Meno_zamestnanca	Produkt
Jozef Mrkva	Tlačiareň
Juraj Ozimina	
Petra Hradská	Stôl
Petra Hradská	Stolička
Monika Sivá	

### **RIGHT OUTER JOIN**

Príklad:

SELECT zamestnanec.Meno\_zamestnanca, objednávka.Produkt FROM zamestnanec RIGHT JOIN objednávka ON zamestnanec.Zamestnanecké\_ID = objednávka.Zamestnanecké\_ID

Vráti všetky riadky z druhej tabuľky (*objednávka*), a to aj v tom prípade, ak nie je zhoda v podmienke s prvou tabuľkou (*zamestnanec*).

Výsledok:

Meno_zamestnanca	Produkt
Jozef Mrkva	Tlačiareň
Petra Hradská	Stôl
Petra Hradská	Stolička
	Kvetina

# 5.3.4 Query

Funkcia *Query* zabezpečuje vyhľadávanie záznamov v databáze podľa zadaných podmienok od používateľa. Pre túto funkciu sú vyhradené rovnomenné tlačidlá na jednotlivých záložkách. Táto funkcia sa skladá z troch fáz:

 prvá je prispôsobenie položiek na záložke pre vstup dát od používateľa potrebných pre vytvorenie SQL požiadavky. Položky sa vyprázdnia. Položky, podľa ktorých sa dá vyhľadávať sa nastavia do modu čítania hodnôt, aktivuje sa na nich možnosť prepínania pomocou tlačidla Tab a prvá položka sa nastaví ako aktívna. Tie záložky, ktoré sú nastavené do módu čítania hodnôt sú popísané ďalej pri opisoch jednotlivých záložiek. Teraz je záložka pripravená na vstup dát.

- po vložení podmienok, podľa ktorých chce používateľ vyhľadávať, stlačí klávesu *Enter* a aplikácia pošle SQL požiadavku na server. Pri nevyplnení ani jednej položky určenej pre vyhľadávanie a stlačení klávesy *Enter*, aplikácia požiada o potvrdenie od používateľa o zaslaní požiadavky pre vypísanie všetkých záznamov záložky. Ak pri vkladaní podmienok používateľ stlačí klávesu *ESC* funkcia bude zrušená a aplikácia sa vráti do stavu pred spustením funkcie.
- aplikácia po obdržaní množiny dát od servera vyprázdni položky a obdržané dáta zobrazí. Ak táto množina obsahuje viacero záznamov, po jednotlivých záznamoch sa dá prechádzať pomocou tlačidiel *Next* a *Previous*. Pri zobrazení záznamu sa sprístupnia aj ostatné funkčné tlačidlá (*Remove, Update, Master, Detail*).

Poradie záznamu v celkovom počte záznamov je zobrazený v prvej časti *Informačného pásu* ("*Záznam 2/3"*). Pri obdržaní prázdnej množiny od servera, sa položky vyprázdnia a užívateľ je oboznámený o situácii pomocou *Informačného pásu* ("*Žiaden záznam nenájdený"*).

Niektoré záložky majú SQL požiadavky pospájané z viacerých tabuliek databázy. Toto je nevyhnutné, keď napríklad na záložke *Zápis študenta*, ktorá pracuje s tabuľkou *student*, chceme zobraziť osobné informácie, ktoré sú v tabuľke *os\_udaje*.

### 5.3.5 Add

Funkcia *Add* slúži na pridávanie nových záznamov do databázy. Táto funkcia je aktívna na záložke len vtedy, ak pre danú záložku existuje záznam v Master záložke. Napríklad: zapísať predmet môžeme len zapísanému študentovi, študenta môžeme zapísať ak máme jeho osobné údaje, prerušenia môžeme vkladať len zapísanému študentovi atď. Toto pravidlo neplatí o záložke *Študent*, ktorá nemá Master záložku,

a preto je funkcia pre záložku *Študent* vždy aktívna. Aktívnosť funkcie je reprezentovaná aktívnosť ou rovnomenného tlačidla na jednotlivých položkách. Pridávanie sa taktiež delí na tri fázy:

- v prvej sa zabezpečí, aby položky určené pre vstup dát boli sprístupnené pre používateľa (odomknú sa na zápis, nastaví sa prepínanie pomocou Tab). Taktiež sa za aktívnu položku prehlási prvá položka určená na vstup dát. Na niektorých záložkách sú položky napĺňané hodnotami z Master tabuliek. Napríklad pre záložku *Zápis študenta* je primárna tabuľka *student*. Položky rodné číslo, osobné číslo, meno a priezvisko sú napĺňané hodnotami z tabuľky *os\_udaje*, ktorá je Master tabuľkou pre tabuľku *student*. Preto tieto položky ostávajú nezmenené. Položky, ktoré sa sprístupnia, a ktoré ostanú neprístupné sú popísané pri každom popise záložky samostatne.
- v druhej fáze používateľ vkladá údaje. Pre zrušenie funkcie pridávania a vrátenia sa na funkciu *Query*, môže používateľ stlačiť klávesu *ESC*. Po stlačení klávesy *Enter* na položke, alebo opätovným stlačením tlačidla *Add* (ALT + A) sa spustia testy na integritu vkladaných dát. Tieto testy sú bližšie popísané v kapitole 5.3.8. Ak dáta prejdú testami, prechádza sa na tretiu fázu.
- tretia fáza je samotné zapísanie dát do tabuliek databázy. Pri každej záložke je popísaná kľúčová tabuľka, do ktorej sú dáta zapísané.

Výsledok fázy zápisu je oznámený správou "Nový záznam zapísaný do databázy".

### 5.3.6 Update

Funkcia *Update* slúži na aktualizáciu už existujúcich záznamov v databáze. Táto funkcia je aktívna na záložke až po vykonaní funkcie *Query* a po úspešnom nájdení aspoň jedného záznamu. Aktívnosť funkcie je reprezentovaná aktívnosťou rovnomenného tlačidla na záložke. Funkcia sa prevádza na aktívnom zázname. Aktualizácia sa delí na tri fázy:

- položky, ktoré je možné meniť pri aktualizácii sa v prvej fáze sprístupnia na zápis. Položky, ktoré sa sprístupnia, a ktoré ostanú neprístupné sú popísané pri každom popise záložky samostatne. V položkách ostávajú hodnoty načítané z databázy. Za aktívnu sa prehlási prvá položky v poradí.
- druhá fáza spočíva v samotnom menení dát používateľom. Ak používateľ stlačí klávesu *ESC*, stav funkcie sa vráti do stavu pred spustením funkcie *Update*, čiže do funkcie *Query*. Po stlačení klávesy *Enter*, alebo opätovným stlačením tlačidla *Update* (ALT + U) sa aktivujú testy pre integritu vložených dát do záložky. Tieto testy sú bližšie popísané v kapitole 5.3.8. Po úspešných testoch funkcia prechádza do tretej fázy.
- tretia fáza reprezentuje samotnú zmenu dát v databáze. Pri každej záložke je popísaná kľúčová tabuľka, do ktorej sú zmeny zapísané.

Výsledok fázy aktualizácie je oznámený správou "Zmeny zapísane do databázy".

### **5.3.7 Remove**

Funkcia *Remove* je určená pre odstraňovanie záznamov z databázy. Táto funkcia, ako aj rovnomenné tlačidlo na záložke, je aktivovaná až po spustení funkcie *Query* a nájdení aspoň jedného záznamu. Akcia sa prevádza na aktuálnom zázname z množiny nájdených záznamov. Táto funkcia sa skladá z dvoch fáz:

- prvá fáza vyzve používateľa k potvrdeniu požiadavky o vymazanie záznamu.
   Keď používateľ požiadavku potvrdí, vykoná sa druhá fáza.
- v druhej fáze sa vykonajú testy, ktoré overia či hodnota primárneho kľúča z vymazávanej tabuľky nie je hodnotou cudzieho kľúča v inej tabuľke. Napríklad: Vymazávaný záznam z tabuľky osobných dát (*os\_udaje*), kde primárnym kľúčom je rodné číslo (*rod\_cislo*). Pri vymazávaní sa kontroluje, či v tabuľke študentských dát (*student*) existujú záznamy s uvedeným rodným číslom. Ak existujú nedovolí sa vymazať a používateľ je o tom informovaný správou "Záznam nemôže byť vymazaný pretože študent je už

*zapísaný*" V opačnom prípade sa záznam vymaže a používateľ je informovaný správou "*Záznam bol úspešne vymazaný*".

Po skončení vymazávania sa aplikácia vracia do stavu v akom bol pred začatím vymazávania.

### 5.3.8 Kontrola vstupných dát

Niektoré dáta sú povinné (nastavením stĺpca tabuľky na NOT NULL), iné majú predpísaný formát (dátumy) a niektoré prípustnú hodnotu len z určitej množiny hodnôt. Preto bolo nutné naprogramovať kontrolu vstupných dát od užívateľa pri vkladaní a aktualizácii dát. Použilo sa päť typov testov:

- prvým je test na prítomnosť hodnoty v položke. Tento je zameraný na stĺpce v databáze typu NOT NULL, čo znamená, že musia obsahovať hodnotu. Napríklad rodné číslo pre zápis do tabuľky osobných údajov (*os\_udaje*).
- druhým testom je test na určitý predpis ako má hodnota položky vyzerať.
   Napríklad: Rodné číslo je jedenásť znakový reťaz so znakom "/" na siedmej pozícii.
- tretí test je zameraný na kontrolu položiek, ktoré majú predpísanú množinu prípustných hodnôt. Napríklad: Známka pre predmet musí byť z množiny ['A', 'B', 'C', 'D', 'E', 'FX'].
- štvrtý test je test spojením druhého a tretieho testu, a je to test na správnosť dátumu.
- piaty test kontroluje či zapisovaný alebo aktualizovaný záznam už existuje v databáze.

Ak kontrola dát má negatívny výsledok, konkrétne políčko zmení farbu na červenú a políčko sa stane aktívnym.

# 5.4 Záložka Študent

Táto záložka sa zaoberá spracovaním osobných informácií študentov. Medzi tieto informácie patria identifikačné údaje ako sú rodné číslo, meno a priezvisko, trvalé bydlisko, štátna príslušnosť, národnosť, telefónne číslo a číslo dokladu totožnosti. Táto záložka je východzia pri štarte aplikácie.

# 5.4.1 Vzhľad



Údaje na záložke sú rozdelené do troch logických častí (obr. 5.4.1.).

Prvou časťou (1) sú všeobecné informácie o osobe a to sú rodné číslo, meno, priezvisko, stav *študenta* a *či má záujem* o inžinierske štúdium. Keďže meno a priezvisko sú veľmi dôležité, sú väčšie ako ostatné údaje na formulári. Údaje o tom, či študent je na inžinierskom štúdiu a údaj, či má

obr. 5.4.1.

záujem o inžinierske štúdium bolo riešené textovým odkazom na formulári.

**Druhá časť** (2) je venovaná **trvalému bydlisku** študenta. Okrem Ulice a PSČ, všetky prvky tejto časti sú riešené cez pomocou ComboBox-ov.

**V tretej časti (3)** sú údaje, ktoré nie sú tak často využívané. Informácia o doklade **totožnosti** je pre slovenských a zahraničných študentov riešená odlišne. Pre slovenských študentov tento údaje predstavuje číslo občianskeho preukazu a pre zahraničných študentov číslo pasu.

# 5.4.2 Funkčnosť a spracovanie

### Query

Vyhľadávanie je možné podľa položiek:

- rodné číslo
- meno
- priezvisko
- podmienka či je študent na inžinierskom štúdiu
- podmienka či má študent záujem o inžinierske štúdium

Pre vyhľadávanie sa vytvorí nasledovná SQL požiadavka, ktorá sa odošle na server:

#### SELECT rod\_cislo,

meno, priezvisko, obc\_preuk, pas, narodnost, st\_prisl, ulica, obec, psc, okres, ing, telefon, ma\_zaujem\_o\_ing FROM os\_udaje WHERE os\_udaje.rod\_cislo LIKE ' + Variables.Tab1QueryRodCislo + ' AND os\_udaje.meno LIKE ' + Variables.Tab1QueryMeno + '

AND os\_udaje.priezvisko LIKE ' + Variables.Tab1QueryPriezvisko + '

K týmto podmienkam sa ešte pridružia podmienky, ktoré závisia od zaškrtnutia položiek *Tab1CheckIng* a *Tab1CheckZaujemIng*. Ak je *Tab1CheckIng* zaškrtnuté, pribudne podmienka:

#### AND ing='\*'

a ak je zaškrtnuté Tab1CheckZaujemIng pribudne:

#### AND ma\_zaujem\_o\_ing='\*'

### Add

Jedine na tejto záložke je funkcia a tlačidlo *Add* prístupné po celý čas. Je to kvôli tomu, že všetky študentské záznamy sa musia začať zápisom do tabuľky obsahujúcej osobné údaje študenta, a tento zápis nie je podmienený vytvorením zápisu do inej tabuľky, ako je to pri iných druhoch záznamov.

Položky, ktoré sú prístupné pre vkladanie dát sú:

- rodné číslo
- meno
- priezvisko
- ulica
- obec
- *PSČ*
- okres
- kraj
- štát
- štátna príslušnosť
- národnosť
- telefón
- číslo ob. p/pasu

Po vyplnení a odoslaní požiadavky na zápis záznamu do databázy, sa vykoná niekoľko kontrol vstupných dát:

 kontrola, či zadané rodné číslo už neexistuje v databáze. Je to taktiež ochrana proti duplicite kľúčov v tabuľke databázy.

- niekoľko testov rodného čísla. Tu sa kontroluje dĺžka zadaného reťazca, ktorá musí byť jedenásť znakov a prítomnosť znaku "/", ktorý musí byť na siedmej pozícií. Ďalej sa kontrolujú jednotlivé čísla v zadanom rodnom čísle. Deň môže byť v intervale od jedna do tridsať jeden, mesiac v intervale od jedna do dvanásť alebo od päť desiat jeden do šesť desiat dva v prípade rodných čísel žien.
- kontrolujú sa i tzv. povinné položky, ktoré musia byť vyplnené. Táto kontrola je viazaná priamo na databázu, pretože v tabuľke sú zadané stĺpce, ktoré musia byť vyplnené, čiže sú NOT NULL. Tieto položky sú rodné číslo, meno, priezvisko, obec, štátna príslušnosť, národnosť a doklad totožnosti

Ako doklad totožnosti sa pre slovenských študentov používa číslo občianskeho preukazu a pre študentov zahraničných je to číslo pasu. Táto závislosť sa odvíja od nastavenia národnosti.

Ďalšia vec, ktorú bolo potrebné ošetriť pre toto zobrazenie, bola automatická aktualizácia na sebe závislých ComboBox-ov. Napríklad: Pri zmene obce sa automatický menil okres, kraj a štát. Toto bolo možné vďaka previazanosti jednotlivých tabuliek miest, okresov, krajov a štátov v databáze.

### Update

Pri stlačení tlačidla *Update* sa položky na záložke stávajú prepisovateľnými, to znamená, že sa v nich dajú meniť hodnoty. Toto sa netýka položky rodného čísla. Táto položka je nemeniteľná, keďže je kľúčom v tabuľke *os\_udaje*. Pri odoslaní požiadavky na zápis aktualizovaných dát do databázy sa vykonávajú rovnaké testy na hodnotách vstupných dát ako pri zápise nového záznamu okrem tých, ktoré sa týkali rodného čísla, keďže to sa meniť nemôže.

Previazanosť medzi jednotlivými tabuľkami miest, okresov, krajov a štátov v databáze je i tu využitá napríklad pri zmene obce.

#### Remove

Vykoná sa kontrola, ktorá overí či hodnota primárneho kľúča z vymazávanej tabuľky (*os\_udaje*) nie je hodnotou cudzieho kľúča v inej tabuľke (*student*).

# 5.5 Záložka "Zápis študenta"

Každý študent má svoje osobné študentské číslo podľa, ktorého je jednoznačne identifikovateľný na univerzite. Taktiež každý študent ma svoje zameranie, študijnú skupinu do ktorej patrí, určenú formu štúdia, stupeň štúdia a ďalšie študentské záznamy. Správou týchto študentských údajov sa zaoberá práve táto záložka. Ponúka aj stručný prehľad ďalších doplňujúcich informácií o študentoch, ako je napríklad zobrazenie informácií o poslednom opakovaní ročníka alebo prerušení štúdia.

## 5.5.1 Vzhľad



Záložka je opticky rozdelená do štyroch logických skupín (obr. 5.5.1).

V hornej časti (1) sa nachádzajú všeobecné informácie, ktoré slúžia pre rýchlu identifikáciu osoby. Taktiež v hornej pravej časti je zobrazený počet študentských záznamov pre dané rodné číslo, keďže jedna osoba môže byť zapísaná na školu viackrát. Napríklad ak študent po ukončení bakalárskeho štúdia

nastúpi na inžinierske štúdium, potom dostane nové osobné číslo.

**V strednej časti** záložky (2) sa nachádzajú **študentské dáta**. Medzi ne patrí zameranie, stav, ročník, študijná skupina, forma štúdia, odbor a stupeň štúdia. Taktiež v tejto časti sa nachádza tlačidlo *Kontá*, po ktorého stlačení sa používateľ dostane na záložku *Kontá študentov*.

Časť opakovaní a prerušení (3) je malý prehľad záložky *Prerušení* a *Opakovaní*. Zobrazujú sa tu najnovšie záznamy týkajúce sa opakovaní ročníkov

a prerušení štúdia, ako aj počet záznamov v jednotlivých tabuľkách (*prerusenia* a *opakovania*). Pri kliknutí na ne bude používateľ prepnutý na záložku *Prerušení a Opakovaní*, kde môže s týmito údajmi ďalej pracovať.

**Posledná časť** (4) záložky zobrazuje dátum prvého zápisu na školu a výšku pôžičky, ktorú si študent požičal od školy. V prípade pôžičky je pole zvýraznené červenou farbou.

### 5.5.2 Funkčnosť a spracovanie

#### Query

Vyhľadávanie je možné podľa položiek:

- rodné číslo
- osobné číslo
- ročník
- študijná skupina
- stav študenta

Keďže toto zobrazenie pracuje s dátami zo štyroch tabuliek (*os\_udaje, student, zameranie, st\_program*), SQL dotaz pre vyhľadávanie záznamov je komplexnejší:

#### SELECT os\_udaje.rod\_cislo,

meno,
priezvisko,
CAST(os\_cislo AS varchar) AS os\_cislo,
CAST(student.id\_zamer // student.id\_program // student.forma AS varchar)
AS id\_zamer,
rocnik,
st\_skupina,
datumop,
dat\_prv\_zap,
zameranie.forma,
stav,
st\_program.c\_odboru,
st\_program.stupen

LIKE

CAST(zameranie.id\_zamer || zameranie.id\_program || zameranie.forma AS varchar) AND student.rod\_cislo LIKE ' + Variables.Tab2QueryRodCislo + ' AND student.os\_cislo LIKE ' + Variables.Tab2QueryOsobneCislo + ' AND student.stav LIKE ' + Variables.Tab2QueryStav + ' AND student.rocnik LIKE ' + Variables.Tab2QueryRocnik + ' AND student.st\_skupina LIKE ' + Variables.Tab2QuerySt\_Skupina + ' AND zameranie.id\_program = st\_program.id\_program AND zameranie.forma = st\_program.forma ORDER BY st\_program.c\_odboru DESC, os\_cislo;

#### Add

Položky, ktoré sú prístupné pre vkladanie dát sú:

- osobné číslo
- zameranie
- stav študenta
- ročník
- študijná skupina
- dátum prvého zápisu

Po vyplnení a odoslaní požiadavky na zápis záznamu do databázy, sa vykoná niekoľko kontrol vstupných dát:

 zistí sa, či položka *rodné číslo* nie je prázdna a potom či v databáze už existuje študent s rovnakým osobným číslom

- skontroluje sa hodnota stavu študenta. Táto hodnota musí byť len z množiny povolených hodnôt "S" (študent), "K" (ukončený), "E" (zanechal štúdium), "V" (vylúčený), "P" (prerušený), a "Z" (zahraničný).
- hodnota ročníka musí byť v intervale celých čísel od nula do päť
- dĺžka hodnoty študijnej skupiny musí byť rovná šiestim
- dátum prvého zápisu musí byť dátumom

#### Update

Pri *Update* sa okrem položky *osobné číslo* sprístupnia rovnaké položky ako pri *Add* s tým rozdielom, že hodnoty ostávajú vpísane do položiek. Po odoslaní požiadavky o zmenu dát používateľom, prebehnú rovnaké testy na vstupné dáta ako pri *Add*, ale bez kontroly týkajúcej sa *osobného čísla*.

#### Remove

Vykoná sa kontrola, ktorá overí či hodnota primárneho kľúča z vymazávanej tabuľky (*student*) nie je hodnotou cudzieho kľúča v iných tabuľkách (*zap\_predmety, diplomka, prerusenia, pozicky* a *st\_hesla*).

# 5.6 Záložka Zápis predmetov

Táto záložka je jedna z najdôležitejších, pretože sa zaoberá správou zapísaných predmetov pre študenta. Pomocou nej sa dajú zapisovať nové predmety, vymazávať a editovať už zapísané predmety. Pre každý predmet na tejto záložke je možné zistiť viacero detailov ako je napríklad meno prednášajúceho, počet kreditov, dátum zápočtu a skúšky, známka z predmetu, počet termínov atď.

# 5.6.1 Vzhľad



Záložka je opticky rozdelená do troch častí podľa funkčnosti (obr. 5.6.1.).

V prvej časti (1) sa nachádzajú všeobecné orientačné údaje o študentovi, ktorého údaje si prezeráme. Je tu *meno, priezvisko, osobné číslo* a *školský rok.*, pre ktorý daný predmet zapísaný.

Druhá časť (2) záložky sa venuje predmetu samotnému. Nachádza sa tu číslo

a názov predmetu, číslo a meno prednášajúceho, ktorý pre daný školský rok predmet prednášal a je zodpovedný za vyplnenie skušobnej správy. Ďalej tu nájdeme informácie o počte kreditov za daný predmet, ako aj miesto štúdia predmetu.

**Tretia časť** (3) záložky je venovaná **dosiahnutým výsledkom** študenta na predmete. Nachádzajú sa tu dátumy zápočtu a skúšky. Forma kontroly hovorí o podmienkach absolvovania predmetu. Ďalej tu sú políčka *Známka, Termín* a *Absolvovaný. Termín* udáva na koľký pokus študent absolvoval daný predmet

v školskom roku. Políčko *Absolvovaný* podľa daných podmienok vypíše, či predmet bol absolvovaný alebo nie.

# 5.6.2 Funkčnosť a spracovanie

# Query

Vyhľadávanie je možné podľa položiek:

- osobné číslo
- školský rok
- číslo predmetu
- názov predmetu
- miesto štúdia

Táto záložka pri hľadaní používa štyri tabuľky databázy (*os\_udaje, student, zap\_predmety, predmet*), preto SQL dotaz je obsiahlejší ako ostatné SQL dotazy:

**SELECT** os\_udaje.meno, os\_udaje.priezvisko, CAST(zap\_predmety.os\_cislo AS VARCHAR) AS os\_cislo, CAST(zap\_predmety.skrok AS VARCHAR) AS skrok, zap\_predmety.cis\_predm AS old\_cp, *zap\_predmety.prednasajuci,* zap\_predmety.povinnost, *zap\_predmety.ects*, *zap\_predmety.miesto*, *zap\_predmety.zapocet*, zap\_predmety.datum\_sk, *zap\_predmety.termin,* zap\_predmety.vysledok, predmet\_bod.forma\_kont, student.st\_skupina, zap\_predmety.body AS ects2, predmet.cis\_predm AS new\_cis\_predm

FROM os\_udaje,

student,

(zap\_predmety LEFT OUTER JOIN predmet\_bod ON ( predmet\_bod.cis\_predm = zap\_predmety.cis\_predm AND zap\_predmety.skrok LIKE predmet\_bod.skrok ) ) LEFT OUTER JOIN predmet ON ( predmet.old\_cp = zap\_predmety.cis\_predm) WHERE os\_udaje.rod\_cislo LIKE student.rod\_cislo AND student.os\_cislo LIKE zap\_predmety.os\_cislo AND zap\_prdmety.os\_cislo LIKE '+Variables.Tab3QueryOsobneCislo+' AND zap\_predmety.skrok LIKE ' + Variables.Tab3QuerySkRok + ' AND zap\_predmety.cis\_predm LIKE ' +Variables.Tab3QueCisPred + ' AND zap\_predmety.miesto LIKE ' +Variables.Tab3QueryMiestoStudia+' ORDER BY zap\_predmety.skrok;

Pri každom zobrazení zapísaných predmetov sa políčko informujúce o absolvovaní predmetu aktualizuje. Jeho hodnota závisí od viacerých podmienok. Medzi tie podmienky patrí forma *kontroly predmetu*, *dátum skúšky*, *dátum zápočtu* a *výsledná známka*. Vyhodnotenie sa drží nasledovných pravidiel:

- ak je forma kontroly predmetu typu "e", to znamená, že predmet aby bol uznaný ako absolvovaný, musí byť ukončený skúškou. Ak má dotyčný študent zapísaný dátum skúšky a taktiež známku, ktorá je buď v rozsahu od 1 do 4, alebo v rozsahu od A po E, predmet je vyhodnotený ako absolvovaný.
- pri forme kontroly typu "z", ktorá znamená, že predmet je uznaný za absolvovaný udelením zápočtu, je predmet prehlásený za absolvovaný ak hodnota dátumu zápočtu nie je prázdna hodnota.
- ak spojíme podmienky pre formu kontroly typu "e" s podmienkou pre formu kontroly typu "z" dostanem podmienky pre formu kontroly typu "s".

Ak je predmet vyhodnotený ako neabsolvovaný, pole informujúce o absolvovaní zmení farbu na červenú.

# Add

Položky, ktoré sa sprístupnia pre vkladanie dát sú:

- číslo a názov predmetu
- typ predmetu
- miesto štúdia
- dátum zápočtu
- dátum skúšky
- známka
- termín

V druhej fáze funkcie *Add*, pred samotným zápisom dát do databázy, prebehnú kontroly zadaných dát:

- kontrola či daný predmet už nemá študent zapísaný.
- kontrola dátumu zápisu a skúšky.
- známka musí byť z množiny "A", "B", "C", "D", "E", "F" alebo "FX". Pri zadaní hodnoty "F" sa do databázy uloží hodnota "FX".
- počet termínov musí byť menší ako počet termínov nastavený globálnou premennou aplikácie. Táto premenná je pri štarte nastavená na hodnotu tri, ale dá sa meniť pomocou hlavného menu v *Nastaveniach*.
- typ predmetu musí byť len z povolenej množiny hodnôt "P" (povinný), "V" (voliteľný), "A" (alternatívny), "Z" (záverečný) alebo "S" (povinne voliteľný).
- hodnota miesta štúdia predmetu musí byť z prístupnej množiny hodnôt "P",
   "R" alebo "Z".

Pri pridávaní záznamu predmetu sa v položke *Číslo a názov predmetu* nachádzajú predmety len z určitého školského roku. Tento školský rok sa dá zmeniť cez položku hlavného menu *Nastavenia*. Pri jeho zmene sa aktualizuje zoznam predmetov. Táto hodnota je pri štarte aplikácie nastavená na aktuálny školský rok.

Pre ľahší výber predmetu slúži aj Edit komponent pred ComboBox komponentom pre položku *Číslo a názov predmetu*. V prípade, že používateľ pozná číslo predmetu, môže ho do Edit komponentu zapísať. Prebehne kontrola, či dané číslo predmetu existuje. V prípade, že existuje, meno predmetu sa doplní do komponentu ComboBox. Samotná implementácia vyzerá tak, že na udalosti komponentu Edit typu OnKeyPress a OnKeyUp je definovaná procedúra, ktorá obsahuje priradenie vpísaného textu v komponente Edit kľúču komponentu ComboBox. Ak sa takýto kľúč nachádza v komponente ComboBox, názov predmetu zobrazí, inak komponent ComboBox ostane prázdny.

Pri výbere predmetu, či už cez výber cez ComboBox, alebo priamym písaním čísla predmetu do položky čísla predmetu, sa aktualizujú informácie súvisiace s predmetom ako sú *počet kreditov* a *meno a číslo prednášajúceho*.

**Miesto štúdia** sa automaticky doplní v závislosti od študijnej skupiny študenta. Samozrejme túto **hodnotu je možné zmeniť**. Keďže do študijnej skupiny študenta je zakomponované miesto štúdia a predpokladá sa, že študent bude navštevovať predmet práve tam. Zoberie sa **druhý znak** zo **študijnej skupiny** a ten sa vloží ako miesto štúdia. Napríklad: Ak študent má študijnú skupinu **5ZA41**, znamená to, že bol zapísaný v Žiline (druhé písmeno v študijnej skupine je "Z"). Predpokladá sa, že študent bude navštevovať predmet v Žiline, a preto sa miesto štúdia prednastaví na hodnotu "Z". Rozličné miesta štúdia môže byť nastavené napríklad ak študenta z Prievidze študuje nejaký predmet v Žiline, ktorý sa v Prievidzi neotvára.

#### Update

Pri *Update* sa sprístupnia všetky položky ako pri *Add* okrem položiek *Číslo a názov predmetu*. Obsahy položiek sa nevymazávajú ale ostávajú naplnené. Pri odoslaní požiadavky o aktualizáciu záznamu v databáze prebehnú rovnaké testy ako pri pridávaní *Add*.

# Remove

Pri vymazávaní sa vyžiada o potvrdenie vymazania od užívateľa. Po potvrdení sa záznam vymaže z databázy. Keďže záložka *Zápis predmetu* nie je Master záložkou pre žiadnu inú záložku, nekontroluje sa existencia cudzieho kľúča v iných tabuľkách.

# 5.7 Záložka Prerušenia a Opakovania

Niektorí študenti opakujú ročníky, iní prerušia štúdium Pre ošetrenie práve týchto prípadov bola vytvorená táto záložka. Ponúka stručný prehľad, tak prerušení štúdia ako aj opakovaní ročníka a manipuláciu s nimi.

# 5.7.1 Vzhľad



obr. 5.7.1.

Záložka je rozdelená na tri optické časti (obr. 5.7.1.).

Horná časť (1) sú informácie všeobecné o študentovi. Nachádza sa tam meno, priezvisko a osobné číslo študenta.

Druhá časť (2) je venovaná prerušeniam. Pri každom prerušení sú informácie evidované školský rok. dátum začiatku prerušenia, dátum konca prerušenia a ročník. Táto časť má vlastné tlačidlá Add.

*Remove* a *Update*, ktorými sa pracuje s prerušeniami.

Spodná časť (3) sa zaoberá opakovaniam. Pre každý záznam opakovania sa vedú informácie o školskom roku a ročníku. Tlačidlá Add, Remove a Update slúžia na manipuláciu s opakovaniami.

# 5.7.2 Funkčnosť a spracovanie

# Query

Vyhľadávanie je možné podľa položiek:

osobné číslo

Toto zobrazenie spája informácie týkajúce sa *prerušení* (tabuľka *prerusenia*) a *opakovaní* (tabuľka *opakovania*). Preto sú pri funkcii *Query* naprogramované dve SQL požiadavky:

- pre prerušenia:

```
SELECT skrok,

datum_od,

datum_do,

rocnik,

os_cislo

FROM prerusenia

WHERE os_cislo LIKE ' + Variables.Tab4ActualOsobneCislo + '

ORDER BY skrok DESC;
```

- pre opakovania:

SELECT skrok, rocnik, os\_cislo FROM opakovania WHERE os\_cislo LIKE ' + Variables.Tab4ActualOsobneCislo + ' ORDER BY skrok DESC;

Pre *Update* a *Remove* konkrétneho záznamu, musí byť tento záznam vybraný klepnutím ľavého tlačidla myši.

#### Add prerušenia

Pridávanie nových záznamov prerušenia sa deje cez sériu vstupných dialógov v poradí: *školský rok, dátum začiatku prerušenia, dátum konca prerušenia* a *ročník*. Pri vkladaní hodnôt, prebiehajú testy týchto hodnôt:

- školský rok sa testuje či má dĺžku štyri znaky a či vložená hodnota je číslom

- kontrola dátum začiatku a dátum konca prerušenia
- *ročník* musí byť celé číslo v intervale od nula do päť

Pred samotným zapísaním nových dát sa vykoná kontrola existencie rovnakého záznamu v databáze.

#### Update prerušenia

Aktualizácia *Update* prebieha obdobne ako pridávanie *Add* s tým rozdielom, že hodnoty záznamu sú predvolené hodnoty vstupných dialógov. Pri vkladaní dát taktiež prebiehajú testy integrity dát ako pri *Add*.

#### Remove prerušenia

Funkcia *Remove* si vyžiada potvrdenie o vymazanie vybraného záznamu a následne ho vymaže.

#### Add opakovania

Aj pridávanie nových záznamov opakovania sa deje cez sériu dialógov v poradí: *školský rok* a *ročník*. Pri vkladaní hodnôt sa vykonávajú testy:

- *školský rok* musí byť reťazec dĺžky štyri a musí to byť číslo
- ročník musí byť celé číslo v intervale od nula do päť

Pre samotným zapísaním záznamu do databázy, sa vykoná test existencie rovnakého záznamu v databáze.

#### Update opakovania

*Update* má taký istý postup ako aj pridávanie *Add* ale dialógy majú preddefinovanú hodnotu a to hodnotu načítanú z meneného záznamu.

#### **Remove opakovania**

Vymazanie *Remove* požiada používateľa o potvrdenie operácie. Po potvrdení sa vybraný záznam vymaže z databázy.

47

# 5.8 Záložka Kontá študenta

Študenti na Fakulte Riadenia a Informatiky majú svoje prihlasovacie meno (*login*) a *heslo* do učební a Internetu na škole. Tieto heslá sú vytvárané systémovým administrátorom. Záložka slúži len pre distribúciu týchto informácií na študijné oddelenia na detašovaných pracoviskách, kde sú ďalej poskytované študentom v prípade vytvorenia nového konta, alebo zmeny hesla administrátorom, ak študent svoje heslo zabudol.

# 5.8.1 Vzhľad



Plocha záložky nie je plne využitá, ale v budúcnosti, v prípade potreby, sa dá zvyšný priestor využiť aj na iné podobné informácie (obr. 5.8.1.).

V prvej časti (1) sú základne informácie o študentovi ako sú: osobné číslo, meno a priezvisko.

Druhá časť (2) je zameraná údajom týkajúcich sa študentských kont. Sú tu položky: *študijná skupina,* 

#### login, heslo a dátum vytvorenia záznamu.

Na tejto záložke nie sú iné tlačidlá ako je *Master* tlačidlo. **Vyhľadávanie** informácii na tejto záložke je možné **len** cez vzťah **Master-Detail** so záložkou *Zápis študenta*. Tlačidlá *Add* a *Update* tu nie sú, pretože táto položka sa používa len ako informatívna záložka pre distribúciu informácií na študijné oddelenia detašovaných pracovísk.

# 5.8.2 Funkčnosť a spracovanie

I keď je záložka jednoduchá, SQL požiadavka spája dáta z troch tabuliek (*student, os\_udaje, st\_hesla*). SQL požiadavka vyzerá nasledovne:

SELECT CAST(student.os\_cislo AS VARCHAR) AS os\_cislo,

meno,
priezvisko,
st\_skupina,
login,
heslo,
datum
FROM student, os\_udaje, st\_hesla
WHERE student.os\_cislo = st\_hesla.os\_cislo
AND student.rod\_cislo = os\_udaje.rod\_cislo
AND student.os\_cislo LIKE ' + Variables.Tab5QueryOsobneCislo + '
AND student.stav = "S";

Tieto údaje je možné poskytovať len študentom, ktorí stále študujú, a preto je v SQL požiadavke aj podmienka pre stále aktívnych študentov:

AND student.stav = "S";

# 5.9 Záložka Financie

Záložka zaoberajúca sa finančnou agendou medzi študentom a školou. Táto agenda spočíva v správe **pôžičiek, štipendií** a **platieb** študenta za nadštandardnú dĺžku štúdia.

# 5.9.1 Vzhľad



Záložka je rozdelená na štyri optické časti (obr. 5.9.1.).

Horná časť (1) sú všeobecné informácie o študentovi. Nachádza sa tu osobné číslo, meno a priezvisko.

Druhá časť (2) je venovaná pôžičkám. Pri každej pôžičke sú evidované informácie ako sú *Suma* a *Školský rok*.

**Tretia časť (3)** je venovaná **štipendiám**. Pri každom štipendiu sú

vedené informácie ako Suma, Školský rok a Typ štipendia.

Štvrtá časť (4) je venovaná platbám študenta škole napríklad za nadštandardnú dĺžku štúdia a pod.

Každá z častí 2, 3 a 4 majú vlastné tlačidlá *Add, Remove* a *Update*, ktoré slúžia na manipuláciu s dátami v jednotlivých častiach.

# 5.9.2 Funkčnosť a spracovanie

# Query

Vyhľadávanie je možné podľa položiek:

- osobné číslo

Toto zobrazenie spája informácie týkajúce sa *pôžičiek* (tabuľka *pozicky*), *štipendií* (tabuľka *stipko*) a *platieb* (tabuľka *platby*). Preto sú pri funkcii *Query* naprogramované tri samostatné SQL požiadavky:

- pre pôžičky:

SELECT skrok,

suma FROM pozicky WHERE os\_cislo LIKE ' + Variables.Tab6ActualOsobneCislo + ' ORDER BY skrok DESC;

- pre štipendiá:

SELECT suma, skrok, typ\_stipka FROM stipko WHERE os\_cislo LIKE ' + Variables.Tab6ActualOsobneCislo + ' ORDER BY skrok DESC;

- pre platby:

SELECT suma, skrok, semester FROM platby WHERE os\_cislo LIKE ' + Variables.Tab6ActualOsobneCislo + ' ORDER BY skrok DESC;

Pre *Update* a *Remove* konkrétneho záznamu, musí byť tento záznam vybraný kliknutím ľavého tlačidla myši.

#### Add pôžičky

Pridávanie nových záznamov pôžičiek sa deje cez sériu vstupných dialógov v poradí: *suma* a *školský rok*. Pri vkladaní hodnôt, prebiehajú testy na týchto hodnotách:

- zisťuje sa či *suma* je číslo
- školský rok sa testuje či má dĺžku štyri znaky a či vložená hodnota je číslom

Pred samotným zapísaním nových dát sa vykoná kontrola existencie rovnakého záznamu v databáze.

#### Update pôžičky

Aktualizácia *Update* prebieha obdobne ako pridávanie *Add* s tým rozdielom, že hodnoty záznamu sú predvolené hodnoty vstupných dialógov. Pri vkladaní dát taktiež prebiehajú testy integrity dát ako pri *Add*.

#### Remove pôžičky

Funkcia *Remove* si vyžiada potvrdenie o vymazanie vybraného záznamu a následne ho vymaže.

#### Add štipendiá

Pridávanie nových záznamov štipendií sa deje cez sériu vstupných dialógov v poradí: *suma, školský rok a typ štipendia*. Pri vkladaní hodnôt, prebiehajú testy na týchto hodnotách:

- zisťuje sa či suma je číslo
- školský rok sa testuje či má dĺžku štyri znaky a či vložená hodnota je číslom
- *typ štipendia* môže byť len "S" (sociálne), "P" (prospechové), "R" (cena rektora),
   "D" (cena dekana) a "M" (mimoriadne)

Pred samotným zapísaním nových dát sa vykoná kontrola existencie rovnakého záznamu v databáze.

#### Update štipendiá

Aktualizácia *Update* prebieha obdobne ako pridávanie *Add* s tým rozdielom, že hodnoty záznamu sú predvolené hodnoty vstupných dialógov. Pri vkladaní dát taktiež prebiehajú testy integrity dát ako pri *Add*.

#### Remove štipendiá

Funkcia *Remove* si vyžiada potvrdenie o vymazanie vybraného záznamu a následne ho vymaže.

#### Add platby

Pridávanie nových záznamov platieb sa deje cez sériu vstupných dialógov v poradí: *suma, školský rok a semester*. Pri vkladaní hodnôt, prebiehajú testy na týchto hodnotách:

- zisťuje sa či *suma* je číslo
- *školský rok* sa testuje či má dĺžku štyri znaky a či vložená hodnota je číslom
- semester určuje či platba bola vykonaná za zimný semester (hodnota 1), za letný semester (hodnota 2) alebo za celý rok (hodnota 0)

Pred samotným zapísaním nových dát sa vykoná kontrola existencie rovnakého záznamu v databáze.

#### Update platby

Aktualizácia *Update* prebieha obdobne ako pridávanie *Add* s tým rozdielom, že hodnoty záznamu sú predvolené hodnoty vstupných dialógov. Pri vkladaní dát taktiež prebiehajú testy integrity dát ako pri *Add*.

#### **Remove platby**

Funkcia *Remove* si vyžiada potvrdenie o vymazanie vybraného záznamu a následne ho vymaže.

# 6 Zabezpečenie

Keďže aplikácia spracováva veľmi citlivé informácie o študentoch ako sú napríklad: rodné čísla, adresy trvalých bydlísk, známky z predmetov a v neposlednom rade aj finančný stav študenta voči škole, musela sa implementovať určitá bezpečnostná politika do aplikácie.

# 6.1 Typy úrovní zabezpečenia

Zabezpečenia sa aplikovali tak v aplikácii ako aj na prenose dát po sieti. Ako **zabezpečenia v aplikácii** sa použili nasledovné techniky:

- vytvorenie možnosti prihlásenia sa každého užívateľa pod vlastným prihlasovacím menom a heslom
- rozdelenie užívateľov a ich privilégií do dvoch úrovní
- použitie algoritmu SHA-1 pre uloženie hesla v databáze

Zabezpečenia pri prenose dát po sieti boli použité techniky:

- možnosť prihlásenia sa aplikácie do databázového servera len z pevne stanovených IP adries
- podpora protokolu SSL pre prenos medzi serverom a klientom

# Aplikada Nestavenia Reporty Pomoc

Samotná aplikácia sa do databázy prihlasuje vlastným prihlasovacím menom a heslom (obr. 6.1.1.), pod ktorým má plné práva na všetky tabuľky, s ktorými pracuje. V tabuľke security sú uložené všetky potrebné informácie pre prihlásenie používateľa do aplikácie. Po prihlásení aplikácie na pokračuje server. následovným postupom:

> vyžiada od používateľa jeho

prihlasovacie meno a heslo

vyžiada si záznam z tabuľky *security* pre zadané prihlasovacie meno SQL požiadavkou:

#### SELECT \*

# FROM security WHERE login = ''' + LoginLogin.Text + ''';

ak záznam s vloženým používateľským menom existuje v tabuľke, v stĺpci password je výsledok algoritmu SHA-1 (pozri ods. 6.1.3.) hesla. Ak nie, tak aplikácia vyhlási správu o neúspešnom prihlásení:" *Zlé prihlasovacie údaje, kontaktujte svojho administrátora*" a čaká na opätovné vloženie prihlasovacích údajov.

# 6.1.1 Prihlasovanie sa do aplikácie

- zadané prihlasovacie heslo sa prevedie taktiež na SHA-1 a porovná sa s načítaným SHA-1 heslom z databázy. Pri nezhode aplikácia vyhlási správu o neúspešnom prihlásení: " *Zlé prihlasovacie údaje, kontaktujte svojho administrátora*" a čaká na opätovné vloženie prihlasovacích údajov. Pri zhode sa pokračuje v ďalšom kroku
- z SQL odpovede sa načíta stĺpec *meno*, v ktorom sa nachádza meno používateľa.
   Toto meno sa pripojí do nadpisu aplikácie pre lepšiu orientáciu používateľa pod akým používateľom pracuje.
- z SQL odpovede sa načíta stĺpec *write*, ktorý určuje privilégia používateľa (pozri ods. 6.1.2.).
- aplikácia pokračuje v štarte

Heslo do aplikácie si môže každý používateľ meniť sám cez položku *Nastavenia* v hlavnom menu.

#### 6.1.2 Privilégia užívateľov

S aplikáciou budú pracovať dve skupiny používateľov. Prvou bude skupina, ktorá má právo **len na čítanie** dát z databázy. Táto skupina ma povolené v aplikácií **len** jedno **funkčné tlačidlo** *Query*. Druhá skupina používateľov má **plné práva** na dáta, takže má sprístupnené **všetky funkčné tlačidla**. Používateľ sa delí do skupín podľa toho akú hodnotu má hodnotu stĺpca *write* v tabuľke *security*, ktorá sa kontroluje pri prihlásení (pozri ods. 6.1.1. krok 6.). Tento stĺpec môže mať dve hodnoty:

- 0 používateľ má právo len na čítanie
- 1 používateľ má plné práva

#### 6.1.3 Použitie algoritmu SHA-1

Pri prvotnom návrhu bezpečností bol implementovaný kryptografický algoritmus MD5, ktorý bol považovaný za bezpečný. Po preskúmaní dostupných informácií na Internete bolo zistené, že bezpečnosť algoritmu MD5 bola spochybnená

17. augusta 2004 pánmi Xiaoyun Wang, Dengguo Feng, Xuejia Lai a Hongbo Yu. Preto bol použitý algoritmus **SHA-1** (**Secure Hash Algorithm**). Tento algoritmus je široko používaný na zabezpečenie aplikácií a protokolov ako sú: **TLS, SSL, PGP, SSH, S/MIME** a **IPSec**. Implementácia SHA-1 je v súbore *cHash.pas*.

# 6.1.4 Zabezpečenia pri sieťovom prenose

Ďalšie zabezpečenia sa týkajú samotného prenosu dát po sieti. Služba PostgreSQL, ktorá je spustená na strane serveru má v sebe zabudované zabezpečovacie mechanizmy, pomocou ktorých je zabezpečené, že pripojenie na databázu je povolené len z pevne definovaných IP adries. Druhým spôsobom zabezpečenia je použitie technológie SSL pre zabezpečený prenos dát po sieti.

# 7 Zostava Kontrola štúdia

Zostava je určitý typ výstupu spracovaných dát z databázy. Zostava *Kontrola štúdia* je spracovanie dát tabuľky *zap\_predmety*, ktorej výsledkom je prehľad zapísaných predmetov na jednotlivé školské roky spolu s výsledkami z predmetov a počtom nadobudnutých kreditov. Taktiež sa zobrazí vážený štúdijný priemer za jednotlivé školské roky ako aj celkový výsledný vážený študijný priemer. V obrázkovej prílohe A je príklad zostavy.

# 7.1 Rave Reports

Na generovanie zostavy boli použité komponenty **Rave Reports** dodávané s inštaláciou Delphi. Tieto výstupné zostavy je možné vytvárať dvoma spôsobmi. Jeden spôsob je navrhnúť si v programe **Visual Designer** návrh dokumentu, a potom tento dokument len vypĺňať údajmi načítanými od užívateľa. Druhý spôsob je naprogramovať kód, ktorý vytvorí požadovaný tvar výstupu. Výstup sa programuje po jednotlivých riadkoch. V aplikácii je použitý spôsob programovania kódu.

Celý postup generovania výstupu je v procedúre **TFormMain.RvSystem1Print** v súbore **MainForm.pas**. Použité príkazy:

- SetFont(Font, Veľkosť) nastaví druh Font a veľkosť písma Veľkosť.
- PrintCenter(Text, Pozícia) vypíše reťazec *Text* na aktuálny riadok horizontálne vycentrovaný na pozícií *Pozícia*.
- PrintLeft(Text, Pozícia) vypíše reťazec *Text* na aktuálny riadok horizontálne zarovnaný naľavo na pozícií *Pozícia*.
- PrintRight(Text, Pozícia) vypíše reťazec *Text* na aktuálny riadok horizontálne zarovnaný napravo na pozícií *Pozícia*.
- NewLine vloží koniec riadku na aktuálnom riadku.

- PrintBitmapRect(X1, Y1, X2, Y2, Bitmap) vykreslenie bitmapovej grafiky
   *Bitmap* prispôsobenej veľkosti štvorca definovaného bodmi (X1, Y1) a (X2, Y2).
- Bold ak je hodnota nastavená na *True*, tak text napísaný po nastavení bude tučný.

Pomocou príkazov **ClearTabs**, **SetTabs**, **FinishTabBox** a **PrintTab** boli vytvorené tabuľky v dokumente. Tieto príkazy sú rozpísané ďalej v texte.

# 7.2 Spracovanie

Na začiatku vytvárania dokumentu sa vytvorí hlavička. V hlavičke je *meno, priezvisko, osobné číslo* a *študijná skupina*. Tieto informácie sa získajú z tabuliek *os\_udaje* a *student* nasledovnou SQL požiadavkou:

SELECT meno, priezvisko, os\_cislo, st\_skupina FROM os\_udaje, student WHERE os\_udaje.rod\_cislo = student.rod\_cislo AND student.os\_cislo = ' + Variables.Tab3ActualOsobneCislo + ';

Na získanie všetkých potrebných informácií pre vytvorenie zostavy sa na server pošle nasledovná SQL požiadavka:

#### SELECT zp.skrok,

CAST(COALESCE(zp.cis\_predm,") // ' // COALESCE(pt.nazov,") AS VARCHAR) AS meno zp.vysledok, pb.forma\_kont, zp.datum\_sk, zp.zapocet, zp.ects FROM ( zap\_predmety zp LEFT OUTER JOIN predmet\_bod pb ON ( pb.cis\_predm = zp.cis\_predm AND zp.skrok LIKE pb.skrok ) ) LEFT OUTER JOIN predmet pt ON ( pt.cis\_predm = zp.cis\_predm) WHERE zp.os\_cislo = ' + Variables.Tab3ActualOsobneCislo + ' ORDER BY zp.skrok;

Tieto dáta sú zoradené podľa školského roku, od najmenšieho po najväčší. Procedúra postupne prechádza záznamy. Ak zistí, že záznam, ktorý chce spracovať je z iného školského roku, spustí sa podprocedúra, ktorá vykoná výpočet váženého študijného priemeru známok a súčet kreditov za školský rok. Podprocedúra taktiež ukončí tabuľku školského roku predošlého záznamu a vytvorí novú pre ďalší školský rok. Tento postup sa opakuje až po posledný záznam v množine nájdených záznamov. Na konci sa vypočíta celkový vážený aritmetický priemer a celkový súčet nadobudnutých kreditov. Vypíše sa zoznam neabsolvovaných predmetov za celé štúdium. Na konci dokumentu sa vypíše miesto a dátum vystavenia dokumentu a taktiež príprava na podpis dekana.

Vytváranie dokumentov pomocou Rave komponentu je veľmi jednoduché. Ako môžeme vidieť na príklade nižšie, na vypísanie hlavičky dokumentu nám postačí niekoľko príkazov:

PrintCenter('Žilinská univerzita', 5); NewLine: PrintCenter('Fakulta riadenia a informatiky', 5); NewLine: PrintCenter('Informačné a riadiace systémy - Aplikovaná informatika', 5); NewLine: PrintCenter('Tel.: +421-41-5134108 Fax: +421-41-5652044', 5); NewLine: PrintCenter('e-mail: dekan@fri.utc.sk', 5); *NewLine;NewLine; Bold* := *True*: PrintCenter('KONTROLA ŠTÚDIA', 4); *Bold* := *False*; *Obrazok* := *TBitmap*.*Create*; try Obrazok.LoadFromFile('Pictures/logo.bmp');

```
PrintBitmapRect(1, 0.5, 1 + CalcGraphicWidth(1.5, Obrazok),
                                                                           0.5 +
CalcGraphicHeight(1.5, Obrazok), Obrazok);
finally
  Obrazok.Free;
end:
NewLine;NewLine;
PrintLeft('Osobné číslo: ', 1);
Bold := True;
Print(IntToStr(Data.ReportSource.DataSet.FieldValues['os_cislo']));
Bold := False;
PrintRight('Štúdijná skupina: ', 7);
Bold := True;
Print(Data.ReportSource.DataSet.FieldValues['st_skupina']);
Bold := False;
NewLine:
//
PrintLeft('Meno a priezvisko: ', 1);
Bold := True;
Print(Data.ReportSource.DataSet.FieldValues['meno']
                                                                   1
                                                                          ,
                                                           +
                                                                                 +
Data.ReportSource.DataSet.FieldValues['priezvisko']);
Bold := False;
NewLine:
NewLine;
```

Hore uvedený kód nám zabezpečí vytvorenie hlavičky (obr. 7.2.1.).



Žilinská univerzita Fakulta riađenia a informatiky Informačné a riadiace systémy - Aplikovaná informatika Tel.: +421-41-5134108 Fax: +421-41-5652044 e-mail: dekan@fri.utc.sk

KONTROLA ŠTÚDIA

Osobné číslo: 123456 Meno a priezvisko: Peter Bylina Štúdijná skupina: 5ZA042

obr. 7.2.1

Pri vytváraní tabuliek boli použité príkazy na nastavenie vzhľadu tabuľky:

ClearTabs; SetPen(clBlack, psSolid, 1, pmCopy); SetTab(1, pjCenter, 0.75, 5, BOXLINEALL, 0); SetTab(NA, pjCenter, 0.75, 5, BOXLINEALL, 0); SetTab(NA, pjLeft, 3.5, 5, BOXLINEALL, 0); SetTab(NA, pjCenter, 0.75, 5, BOXLINEALL, 0); SetTab(NA, pjCenter, 0.75, 5, BOXLINEALL, 0); FinishTabBox(1);

**ClearTabs** – procedúra na vymazanie všetkých doteraz nastavených parametrov tabuľky. Predchádzajúce vytvorené tabuľky to neovplyvní.

SetPen – procedúra pre nastavenie parametrov "kresliaceho pera".

SetTab – procedúra na nastavenie parametrov stĺpca tabuľky. Parametre procedúry:

- prvý je pre definovanie východzej pozície stĺpca. Hodnota "NA" znamená, že stĺpec bude začínať tam, kde predchádzajúci skončil.
- druhý určuje zarovnanie stĺpca. Možné hodnoty sú pjLeft, pjRight a pjCenter.
- tretí definuje šírku stĺpca
- štvrtý je konštanta, ktorá definuje ako orámuje bunka tabuľky. Hodnota 5 je pre orámovanie okolo celej bunky.
- piaty je pre tieňovanie buniek.

**FinishTabBox** – procedúra, ktorá nakreslí ukončí tabuľku čiarou. Ako parameter má šírku čiary.

Vpisovanie hodnôt do tabuľky je tiež jednoduché. Napríklad:

PrintTab('Por.'); PrintTab('Šk. rok'); PrintTab('Číslo a názov predmetu'); PrintTab('Známka'); PrintTab('Kredity'); FinishTabBox(2);

#### NewLine;

**PrintTab** – procedúra, ktorá zabezpečí vpísanie textu do aktuálnej bunky tabuľky. Výpis hodnôt do tabuľky je sekvenčné to znamená, že sa zapisuje bunka po bunke v aktuálnom riadku a riadok po riadku v dokumente.

Na ukončenie riadku tabuľky čiarou je potrebné použiť procedúru *FinishTabBox* a presun do nového riadku *NewLine*.

Pri použití horeuvedených príkladov môžeme vytvoriť kód:

// NASTAVENIE ClearTabs; SetPen(clBlack, psSolid, 1, pmCopy); SetTab(1, pjCenter, 0.75, 5, BOXLINEALL, 0); SetTab(NA, pjCenter, 0.75, 5, BOXLINEALL, 0); SetTab(NA, pjLeft, 3.5, 5, BOXLINEALL, 0); SetTab(NA, pjCenter, 0.75, 5, BOXLINEALL, 0); SetTab(NA, pjCenter, 0.75, 5, BOXLINEALL, 0); *FinishTabBox(1);* // HLAVICKA TABULKY PrintTab('Por.'); *PrintTab('Šk. rok');* PrintTab('Číslo a názov predmetu'); PrintTab('Známka'); PrintTab('Kredity'); *FinishTabBox*(2); *NewLine*: // PRVY RIADOK *PrintTab('1.'); PrintTab('2001');* PrintTab('P102 Algebra'); PrintTab('B'); PrintTab('6'); *FinishTabBox*(2); NewLine:

// DRUHY RIADOK PrintTab('2.'); PrintTab('2001'); PrintTab('P103 Matematická analýza 1.'); PrintTab('C'); PrintTab('8'); FinishTabBox(2); NewLine;

Tento kód nám vytvorí tabuľku zobrazenú na obrázku 7.2.2.

Por.	Šk. rok	Číslo a názov predmetu	Známka	Kredity
1	2001	P102 Algebra	В	6
2	2001	P103 Matematická analýza 1.	C	8

# 7.3 Výpočet váženého študijného priemeru

Pre výpočet priemeru známok používame **vážený študijný priemer**, pre ktorého čitateľa a menovateľa platia nasledujúce pravidlá:

- do úvahy sa berú len predmety končiace skúškou, čiže predmety s formou kontroly "e" alebo "s".
- koeficienty známok sú: A 1, B 1.5, C 2, D 2.5, E 3 inak 4.
- čitatel' je rovný súčtu násobkov koeficientov známok a počtu kreditov pre všetky predmety z bodu jeden.
- menovatel' je rovný súčtu počtu kreditov len absolvovaných predmetov z bodu jeden.

# 8 Programátorská príručka

V tejto časti sú podrobne vysvetlené procedúry a funkcie, ktoré sú implementované v programe. Jednotlivé časti programu sú rozdelené do osobitných súborov podľa logického významu:

- MainForm.pas V súbore je uložená implementácia hlavného formulára aplikácie. Nachádzajú sa tu ošetrenia udalostí formulára.
- MainVariables.pas implementácia triedy, ktorá uchováva globálne premenné aplikácie.
- Common.pas implementácia procedúr a funkcií používaných v celej aplikácii.
- DataModule.pas Táto časť obsahuje definíciu zoznamu dátových modulov.
   V ňom sa nachádzajú všetky komponenty zabezpečujúce komunikáciu aplikácie s databázou.
- Tab1.pas implementácia triedy cTab1, ktorá zabezpečuje funkčnosť záložky Študent.
- **Tab2.pas** implementácia triedy cTab2, ktorá zabezpečuje funkčnosť záložky *Zápis študenta*.
- Tab3.pas implementácia triedy cTab3, ktorá zabezpečuje funkčnosť záložky Zápis predmetov.
- **Tab4.pas** implementácia triedy cTab4, ktorá zabezpečuje funkčnosť záložky *Prerušenia a Opakovania*.
- **Tab5.pas** implementácia triedy cTab5, ktorá zabezpečuje funkčnosť záložky *Kontá študentov*.

- **Tab6.pas** implementácia triedy cTab6, ktorá zabezpečuje funkčnosť záložky *Financie*.
- **cHash.pas** knižnica, ktorá obsahuje implementácie hash-ovacích algoritmov pre šifrovanie hesiel.

# 8.1 MainForm.pas

V súbore je uložená implementácia hlavného formulára aplikácie. Nachádzajú sa tu ošetrenia udalostí formulára.

# 8.1.1 Ošetrenie kláves

#### procedure TFormMain.KeyCheckTab2(var Key: Char);

 procedúra ošetrujúca stlačenie kláves Enter a ESC na položkách záložky Zápis študenta. Pre každú záložku je implementovaná vlastná procedúra.

# 8.1.2 Ošetrenie záložiek

# procedure TFormMain.PageControlChange(Sender: TObject);

- procedúra ošetrujúca prepínanie sa medzi jednotlivými záložkami aplikácie. Pri zmene sa podľa určitých podmienok menia nastavenia hodnôt jednotlivých záložiek. Napríklad: Pri prechode zo záložky *Študent* na záložku *Zápis študenta* môžu nastať dva prípady:
  - na záložke Študent nebol aktívny záznam. V tomto prípade sa záložka
     Zápis študenta len vyčistí a nastaví do východzieho stavu
  - na záložke Študent bol aktívny záznam. V tomto prípade, sa položka Zápis študenta vyčistí a následne sa spustí vyhľadávanie študentských záznamov s rodným číslom so záznamu zo záložky Študent.

#### procedure TFormMain.kolskrok1Click(Sender: TObject);

 procedúra ošetrujúca všetky zmeny v aplikácii pri zmene školského roku cez položku *Nastavenia* v hlavnom menu. Týka sa to hlavne záložky *Zapísané* predmety.

# 8.1.3 Zostavy

#### procedure TFormMain.RvSystem1BeforePrint(Sender: TObject);

 procedúra nastavujúca parametre tlače zostavy. Nastavuje sa tu rozmer papiera ako aj okraje:

// page
SelectPaper('A4');
// Margins
MarginTop := 1.0;
MarginLeft := 1.0;
MarginRight := 1.0;
MarginBottom := 1.5;

#### procedure TFormMain.RvSystem1Print(Sender: TObject);

- procedúra, ktorá vytvárajúca zostavu kontroly štúdia. Bližší opis je v kapitole 7.

# 8.1.4 Prihlásenie do aplikácie

#### procedure TFormMain.LoginLoginKeyPress(Sender: TObject; var Key: Char);

procedúra vykonávajúca prihlásenie používateľa do aplikácie. Je v nej použitá implementácia SHA-1 zo súboru cHash.pas pre overenie hesiel. Ak prebehne korektné prihlásenie pokračuje sa procedúrou *TFormMain.PokracujVStarte*.

#### procedure TFormMain.PokracujVStarte;

- po korektnom prihlásení sa používateľa do aplikácie, sa spustí táto procedúra, ktorá zabezpečuje:
  - naplnenie komponentov typu DBLookupComboBox dátami z databázy (viacej v kapitole 5.3.1.).

- vytvára inštancií tried záložiek definovaných v súboroch tab1.pas až tab6.pas.
- napĺňa globálne premenné triedy globálnych premenných Variables definovanej v súbore MainVariables.pas.
- o svojej práci informuje v logovacom okne pri štarte aplikácie (obr. 5.3.1.).

# 8.2 MainVariables.pas

V aplikácii existujú **globálne premenné**, ktoré sú prístupné z ktoréhokoľvek miesta v aplikácií cez triedu globálnych premenných *Variables*. Táto trieda je vytvorená pri štarte aplikácie a hodnoty sú naplnené prednastavenými hodnotami. Jednotlivé hodnoty sú menovite rozdelené podľa toho, s ktorou záložkou súvisia. Napríklad: *vTab1ActualMeno* je hodnota položky *Meno* na prvej záložke *Študent* a *VTab4ActualOsobneCislo* je aktuálna hodnota položky *Osobné číslo* na záložke štyri, *Prerušenia*. Taktiež sa tu nachádzajú premenné, ktoré sú spoločné pre všetky záložky. Napríklad: Premenná *Zápis* je naplnená pri autorizácii používateľa. Je naplnená na hodnotu *True*, ak používateľ má práva zapisovať a meniť údaje v databáze, inak je nastavená na *False*, čo znamená, že má dáta sprístupnené len na čítanie. Viac o bezpečnosti v kapitole 6.1.2.

# 8.3 Common.pas

Niektoré procedúry a funkcie sa používajú v celej aplikácii a nielen v niektorej jej časti. Tento súbor zahŕňa takéto procedúry a funkcie.

#### function JeDatumValidny(tString : string): String;

- dátumy v aplikácii môžu byť zadávané v troch rôznych formátoch:
  - DD.MM.YYYY
  - DD,MM,YYYY
  - DD/MM/YYYY

V prípade vloženia druhého alebo tretieho formátu, si funkcia prevedie dátum na prvý formát, a potom vykoná kontroly. Kontrola spočíva vo využití funkcie *IsValidDate* (z knižnice *DateUtils*). Funkcia *JeDatumValidny* vráti upravený dátum vo formáte DD.MM.YYYY ak je vložený dátum správny, inak vráti prázdny reťazec.

# 8.4 DataModule.pas

Táto časť obsahuje definíciu zoznamu dátových objektov. V ňom sa nachádzajú všetky komponenty zabezpečujúce komunikáciu aplikácie s databázou. Sú to komponenty s balíka ZeosLib.

# 8.5 Tab1.pas

V tomto súbore je definovaná trieda *cTab1*, ktorá zabezpečuje funkčnosť prvej záložky *Študent*.

# 8.5.1 Záložka

#### procedure ZamkniCelyFormular;

- Vyčistenie záložky a nastavenie do východzieho stavu.

#### procedure PSCChange;

- Procedúra pri zmene položky Obec mení položky PSČ, Okres, Kraj a Štát.

#### procedure OkresChange;

- Procedúra pri zmene položky Okres mení položky Kraj a Štát.

#### procedure KrajChange;

Procedúra pri zmene položky Kraj mení položku Štát.

# 8.5.2 Tlačidlá

#### procedure ButtonQueryClick;

- Zabezpečenie východzieho stavu záložky pri stlačení tlačidla Query.

#### procedure ButtonPreviousClick;

- Zabezpečí prechod na predchádzajúci záznam v množine nájdených záznamov.

#### procedure ButtonNextClick;

- Zabezpečí prechod na ďalší záznam v množine nájdených záznamov.

#### procedure ButtonRemoveClick;

Vyžiadanie si povolenia od používateľa na vymazanie aktuálneho záznamu z databázy a následne jeho vymazanie.

#### procedure ButtonAddClick;

 Zabezpečenie východzieho stavu záložky pri stlačení tlačidla Add. Pri opätovnom stlačení tlačidla Add sa zavolá procedúra WriteData.

#### procedure ButtonUpdateClick;

Zabezpečenie východzieho stavu záložky pri stlačení tlačidla *Update*. Pri opätovnom stlačení tlačidla *Update* v stave aktualizácie dát, sa zabezpečí kontrola integrity vkladaných dát a následne zápis dát do databázy.

# 8.5.3 Dáta

#### procedure VypisQueryTab1(aPoradie: Longint);

Vykoná poslanie SQL žiadosti na server a výsledok zobrazí na záložke.
 Parametrom *aPoradie* definujeme, ktorý záznam z množiny nájdených záznamov chceme zobraziť.

#### procedure WriteData;

 procedúra zabezpečí kontrolu integrity vkladaných dát a po úspešnej kontrole integrity dát ich zápis do databázy.

# 8.6 Tab2.pas

V tomto súbore je definovaná trieda *cTab2*, ktorá zabezpečuje funkčnosť druhej záložky *Zápis študenta*.

# 8.6.1 Záložka

#### procedure ZamkniCelyFormular;

- Vyčistenie záložky a nastavenie do východzieho stavu.

#### procedure StudProgramChange;

- Procedúra pri zmene položky Zameranie mení položky Forma, Odbor a Stupeň.

# 8.6.2 Tlačidlá

#### procedure ButtonQueryClick;

- Zabezpečenie východzieho stavu záložky pri stlačení tlačidla Query.

#### procedure ButtonPreviousClick;

- Zabezpečí prechod na predchádzajúci záznam v množine nájdených záznamov.

#### procedure ButtonNextClick;

- Zabezpečí prechod na ďalší záznam v množine nájdených záznamov.

#### procedure ButtonRemoveClick;

Vyžiadanie si povolenia od používateľa na vymazanie aktuálneho záznamu z databázy a následne jeho vymazanie.

# procedure ButtonAddClick;

Zabezpečenie východzieho stavu záložky pri stlačení tlačidla *Add*. Pri opätovnom stlačení tlačidla *Add* sa zavolá procedúra *WriteData*.

# procedure ButtonUpdateClick;

Zabezpečenie východzieho stavu záložky pri stlačení tlačidla *Update*. Pri opätovnom stlačení tlačidla *Update* v stave aktualizácie dát, sa zabezpečí kontrola integrity vkladaných dát a následne zápis dát do databázy.

# 8.6.3 Dáta

# procedure VypisQueryTab2(aPoradie: Longint);

Vykoná poslanie SQL žiadosti na server a výsledok zobrazí na záložke.
 Parametrom *aPoradie* definujeme, ktorý záznam z množiny nájdených záznamov chceme zobraziť.

#### procedure WriteData;

 procedúra zabezpečí kontrolu integrity vkladaných dát a po úspešnej kontrole integrity dát ich zápis do databázy.

# 8.7 Tab3.pas

V tomto súbore je definovaná trieda *cTab3*, ktorá zabezpečuje funkčnosť tretej záložky *Zapísané predmety*.

# 8.7.1 Záložka

#### procedure ZamkniCelyFormular;

- Vyčistenie záložky a nastavenie do východzieho stavu.

#### procedure PredmetChange;

 Procedúra pri zmene jednej z položiek Číslo predmetu alebo Názov predmetu mení položky Prednášajúci, Forma kontroly, Kredity a Kredity(st.prog).

#### procedure CisloPredmetuCheck;

 Výber predmetu je realizovaný buď výberom z položky Názov predmetu alebo napísaním čísla predmetu do položky Číslo predmetu. Táto procedúra zabezpečuje toto priradenie napísaného čísla v položke Číslo predmetu položke v menu Názov predmetu.

# 8.7.2 Tlačidlá

#### procedure ButtonQueryClick;

- Zabezpečenie východzieho stavu záložky pri stlačení tlačidla Query.

#### procedure ButtonPreviousClick;

- Zabezpečí prechod na predchádzajúci záznam v množine nájdených záznamov.
#### procedure ButtonNextClick;

- Zabezpečí prechod na ďalší záznam v množine nájdených záznamov.

#### procedure ButtonRemoveClick;

Vyžiadanie si povolenia od používateľa na vymazanie aktuálneho záznamu z databázy a následne jeho vymazanie.

#### procedure ButtonAddClick;

- Zabezpečenie východzieho stavu záložky pri stlačení tlačidla *Add*. Pri opätovnom stlačení tlačidla *Add* sa zavolá procedúra *WriteData*.

#### procedure ButtonUpdateClick;

Zabezpečenie východzieho stavu záložky pri stlačení tlačidla *Update*. Pri opätovnom stlačení tlačidla *Update* v stave aktualizácie dát, sa zabezpečí kontrola integrity vkladaných dát a následne zápis dát do databázy.

## 8.7.3 Dáta

#### procedure VypisQueryTab3(aPoradie: Longint);

Vykoná poslanie SQL žiadosti na server a výsledok zobrazí na záložke.
 Parametrom *aPoradie* definujeme, ktorý záznam z množiny nájdených záznamov chceme zobraziť.

#### procedure WriteData;

 procedúra zabezpečí kontrolu integrity vkladaných dát a po úspešnej kontrole integrity dát ich zápis do databázy.

## 8.8 Tab4.pas

V tomto súbore je definovaná trieda *cTab4*, ktorá zabezpečuje funkčnosť štvrtej záložky *Prerušenia a opakovania*.

## 8.8.1 Záložka

### procedure ZamkniCelyFormular;

- Vyčistenie záložky a nastavenie do východzieho stavu.

## 8.8.2 Tlačidlá

#### procedure ButtonQueryClick;

Zabezpečenie východzieho stavu záložky pri stlačení tlačidla Query.

## procedure ButtonPreviousClick;

- Zabezpečí prechod na predchádzajúci záznam v množine nájdených záznamov.

## procedure ButtonNextClick;

- Zabezpečí prechod na ďalší záznam v množine nájdených záznamov.

## procedure ButtonPreruseniaRemoveClick;

 Vyžiadanie si povolenia od používateľa na vymazanie aktuálneho záznamu prerušenia štúdia z databázy a následne jeho vymazanie.

## procedure ButtonOpakovaniaRemoveClick;

Vyžiadanie si povolenia od používateľa na vymazanie aktuálneho záznamu opakovania ročníka z databázy a následne jeho vymazanie.

## procedure ButtonPreruseniaAddClick;

 Zabezpečenie načítania vstupných dát (pomocou *InputQuery*) pre prerušenie štúdia (*školský rok, dátum začiatku prerušenia, dátum konca prerušenia, ročník*), kontrola integrity týchto dát a zapísanie dát do databázy.

## procedure ButtonOpakovaniaAddClick;

 Zabezpečenie načítania vstupných dát (pomocou *InputQuery*) pre opakovanie ročníka (*školský rok, ročník*), kontrola integrity týchto dát a zapísanie dát do databázy.

#### procedure ButtonPreruseniaUpdateClick;

Zabezpečenie načítania aktualizovaných vstupných dát (pomocou *InputQuery* s prednastavenými hodnotami) pre prerušenie štúdia (*školský rok, dátum začiatku prerušenia, dátum konca prerušenia, ročník*), kontrola integrity týchto dát a zapísanie dát do databázy.

#### procedure ButtonOpakovaniaUpdateClick;

Zabezpečenie načítania aktualizovaných vstupných dát (pomocou *InputQuery* s prednastavenými hodnotami) pre opakovanie ročníka (*školský rok, ročník*), kontrola integrity týchto dát a zapísanie dát do databázy.

## 8.8.3 Dáta

## procedure VypisQueryTab4(aPoradie: Longint);

Vykoná poslanie SQL žiadosti na server a výsledok zobrazí na záložke.
 Parametrom *aPoradie* definujeme, ktorý záznam z množiny nájdených záznamov chceme zobraziť.

## 8.9 Tab5.pas

V tomto súbore je definovaná trieda *cTab5*, ktorá zabezpečuje funkčnosť piatej záložky *Kontá študenta*.

## 8.9.1 Záložka

#### procedure ZamkniCelyFormular;

- Vyčistenie záložky a nastavenie do východzieho stavu.

## 8.9.2 Dáta

#### procedure VypisQueryTab5(aPoradie: Longint);

Vykoná poslanie SQL žiadosti na server a výsledok zobrazí na záložke.
 Parametrom *aPoradie* definujeme, ktorý záznam z množiny nájdených záznamov chceme zobraziť.

# 8.10 Tab6.pas

V tomto súbore je definovaná trieda *cTab6*, ktorá zabezpečuje funkčnosť šiestej záložky *Financie*.

## 8.10.1 Záložka

#### procedure ZamkniCelyFormular;

Vyčistenie záložky a nastavenie do východzieho stavu.

## 8.10.2 Tlačidlá

## procedure ButtonQueryClick;

- Zabezpečenie východzieho stavu záložky pri stlačení tlačidla Query.

#### procedure ButtonPreviousClick;

- Zabezpečí prechod na predchádzajúci záznam v množine nájdených záznamov.

#### procedure ButtonNextClick;

- Zabezpečí prechod na ďalší záznam v množine nájdených záznamov.

## procedure ButtonPozickyRemoveClick;

 Vyžiadanie si povolenia od používateľa na vymazanie aktuálneho záznamu pôžičky z databázy a následne jeho vymazanie.

#### procedure ButtonStipendiaRemoveClick;

 Vyžiadanie si povolenia od používateľa na vymazanie aktuálneho záznamu štipendia z databázy a následne jeho vymazanie.

#### procedure ButtonPlatbyRemoveClick;

 Vyžiadanie si povolenia od používateľa na vymazanie aktuálneho záznamu platby z databázy a následne jeho vymazanie.

#### procedure ButtonPozickyAddClick;

Zabezpečenie načítania vstupných dát (pomocou *InputQuery*) pre pôžičku (*suma*, a *školský rok*), kontrola integrity týchto dát a zapísanie dát do databázy.

#### procedure ButtonStipendiaAddClick;

- Zabezpečenie načítania vstupných dát (pomocou *InputQuery*) pre štipendium (*suma, školský rok* a *typ štipendia*), kontrola integrity týchto dát a zapísanie dát.

## procedure ButtonPlatbyAddClick;

- Zabezpečenie načítania vstupných dát (pomocou *InputQuery*) pre platby (*suma*, *školský rok* a *polrok*), kontrola integrity týchto dát a zapísanie dát do databázy.

## procedure ButtonPozickyUpdateClick;

Zabezpečenie načítania aktualizovaných vstupných dát (pomocou *InputQuery* s prednastavenými hodnotami) pre pôžičku (suma, školský *rok*), kontrola integrity týchto dát a zapísanie dát do databázy.

## procedure ButtonStipendiaUpdateClick;

Zabezpečenie načítania aktualizovaných vstupných dát (pomocou *InputQuery* s prednastavenými hodnotami) pre štipendium (*suma, školský rok* a *typ štipendia*), kontrola integrity týchto dát a zapísanie dát do databázy.

## procedure ButtonPlatbyUpdateClick;

Zabezpečenie načítania aktualizovaných vstupných dát (pomocou *InputQuery* s prednastavenými hodnotami) pre platbu (*suma, školský rok a polrok*), kontrola integrity týchto dát a zapísanie dát do databázy.

## 8.10.3 Dáta

## procedure VypisQueryTab6(aPoradie: Longint);

Vykoná poslanie SQL žiadosti na server a výsledok zobrazí na záložke.
 Parametrom *aPoradie* definujeme, ktorý záznam z množiny nájdených záznamov chceme zobraziť.

# 8.11 cHash.pas

Táto časť bola stiahnutá z Internetu (<u>http://fundementals.sourceforge.net/</u>). Je to knižnica, ktorá obsahuje implementácie hash-ovacích algoritmov. Z celej implementácie boli použité dve funkcie:

## Function CalcHMAC\_SHA1 (const Key, Buf : String) : T160BitDigest;

Funkcia vypočítavajúca SHA-1 hash z daného kľúča *Key* a reťazca *Buf*.
 Používaná je pri autorizácii používateľa do aplikácie. Viac o autorizácii v kapitole 6.1.1.

## Function SHA1DigestAsString (const Digest : T160BitDigest) : String;

- Táto funkcia zabezpečí pretypovanie premennej typu *T160BitDigest* na reťazec *String*.

# 9 Používateľská príručka

# 9.1 Úvod

Aplikácia ISFRI je informačný systém Fakulty riadenia a informatiky pre správu databázy študentských dát.

# 9.2 Inštalácia

Požiadavky na inštaláciu:

- operačný systém Microsoft Windows (všetky verzie)
- 64 MB RAM
- 10 MB voľného miesta na disku

Inštalácia aplikácie sa dodáva ako samostatne spustiteľný súbor *setup.exe*. Po spustení súboru inštalácia prejde nasledovnými krokmi:

# 9.2.1 Inicializácia inštalácie

- inštalácia zisťuje potrebné informácie (obr. 9.2.1.).



obr. 9.2.1.

# 9.2.2 Privítanie

- stlačíme tlačidlo Next (obr. 9.2.2.)



obr. 9.2.2.

## 9.2.3 Zadanie cesty inštalácie

- môžeme zmeniť cestu na disku, kde sa má aplikácia nainštalovať (obr. 9.2.3.)





obr. 9.2.3.

# 9.2.4 Kontrola zadaných parametrov inštalácie

- zosumarizovanie zadaných parametrov inštalácie (obr. 9.2.4.). Ak by sme chceli niečo zmeniť stlačíme tlačidlo **Back** inak stlačíme tlačidlo **Install**.

ady to Install the Program			
The wizard is rea <mark>dy t</mark> o begin installatio	n,		
(f you want to review or change any o exit the wizard.	of your installation	settings, <mark>c</mark> lick Back	. Click Cancel to
Jurrent Settings:			
Setup Type:			
Destination Folder:			
C:\Program Files\ISFR\			
User Information:			
Name: Ondrej Žilinec			
Company:			
Ishield			
	< <u>B</u> ack	Install	Cancel
		Lanuar Hotelstein ann	

# 9.2.5 Proces inštalácie

- samotný priebeh inštalácie (obr. 9.2.5.). Priebeh je možné prerušiť tlačidlom Cancel.

🔂 ISFRI - I	nstallShield Wizard	<u>- 🗆 ×</u>
Installing The prog	I ISFRI gram features you selected are being installed.	-4
ß	Please wait while the InstallShield Wizard installs ISFRI. This may take several minutes. Status: Publishing product information	
InstallShield -	< Back Next >	Cancel
	obr. 9.2.5.	

## 9.2.6 Koniec inštalácie

- aplikácia je nainštalovaná (obr. 9.2.6.). Môžeme zvoliť automatické spustenie aplikácie po skončení inštalácie položkou Launch ISFRI.

- pre skončenie inštalácie stlačíme tlačidlo Finish.



obr. 9.2.6.

## 9.2.7 Zmeny na serveri

Bezpečnostné nastavenia databázového servera dovoľujú definovať IP adresy alebo rozsahy IP adries, ktoré majú prístup na databázu. Z tohto dôvodu pri inštalácií aplikácie na ďalší počítač, ktorého IP adresa nie je v povolených IP adresách pre prístup na server, musí administrátor pridať túto novú IP adresu do konfiguračného súboru.

# 9.3 Odinštalovanie aplikácie

Procedúru na automatické odinštalovanie aplikácie je možné spustiť viacerými spôsobmi:

opätovným spustením inštalačného balíka setup.exe dostaneme ponuku ako na obr. 9.3.1., z ktorej vyberieme ponuku Remove a stlačíme tlačidlo Next.

🙀 ISFRI - Instal	llShield Wizard
Program Main Modify, repair	, or remove the program.
• Modify	Change which program features are installed. This option displays the Custom Selection dialog in which you can change the way features are installed.
C Repair	Repair installation errors in the program. This option fixes missing or corrupt files, shortcuts, and registry entries.
C <u>R</u> emove	Remove ISFRI from your computer.
InstallShield	< Back Next > Cancel
	obr. 9.3.1.

pri inštalácií sa vytvorí zástupca v ponuke Štart na odinštalovanie aplikácie s menom Uninstall ISFRI (obr. 9.3.2.).

🌃 ISFRI
obr. 9.3.2.

 aplikácia sa dá odinštalovať priamo cez Ovládací panel v operačnom systéme Microsoft Windows.

# 9.4 Vzhľad a ovládanie aplikácie

Aplikácia pozostáva z jedného pracovného okna (obr. 9.4.), na ktorom sa nachádza *hlavné menu (1)*, zoznam záložiek (2), obsah aktuálnej záložky (3) a informačný pás (4).

## 9.4.1 Hlavné menu

V hlavnom menu si môžeme meniť hlavné parametre aplikácie, po prípade vykonávať všeobecné úlohy. V nasledujúcom texte sú

FRKIS v.1	.13	0			_0
Aplikácia <u>N</u> a 1 Študent	2 Zápis štud.	3 Zápis predm.	4 Preruš&Opa	ko 5 Kontá š	26 Financie
	Erapio orași	C Copie produit		< Previous	Next >>
Osobné	údaje				
Rodne	cislo:	3			
Meno:			Priezvisko:		
[Trvalé	bydlisko				
Ulica			Obec:		
			Okron		
PSC:		-	Okres:		
Kroin	i.	47	Štát:		
Kraj.		<b>.</b>	Stat.		-
Štáto	a príslušnosť	-	Národnosť		
		×			
Telefón	1	Číslo ob	p.:		
Stredna	á škola:				10
					-
Query	<u>A</u> dd	Update	<u>R</u> emove		<u>D</u> etail
/berte si z po	nuky odomknutých	tlačidiel		Connected	15.4.2006 0:32:1
		ob	r. 9.4.		

opísané jednotlivé pod-menu a ich funkčnosť.

## Aplikácia

Všeobecné úlohy pre aplikáciu, do ktorých patrí:

Koniec – ukončenie aplikácie v ktoromkoľvek okamžiku jej behu. Táto položka zabezpečí uvoľnenie alokovanej pamäte aplikáciou. Táto položka má aj vlastnú klávesovú skratku (CTRL + E) pre rýchlejší prístup.

#### Nastavenia

Toto pod-menu obsahuje položky pre zmenu globálnych nastavení aplikácie ako sú:

Školský rok – v systéme pri pridávaní nového predmetu študentovi je použitý aktuálny školský rok na vygenerovanie zoznamu predmetov. Ak si praje používateľ aplikácie pridať predmet z iného školského roku, použije túto položku. Pri zmene školského roku sa automaticky zmení zoznam predmetov.

 Počet termínov – podľa školského poriadku je pevne definovaný maximálny počet termínov pre absolvovanie skúšky. Táto hodnota je kontrolovaná pri pridávaní a aktualizovaní informácie o zapísanom predmete.

#### Pomoc

V tomto pod-menu môže používateľ nájsť pomocnú položku:

- *O programe* – informatívne údaje o aplikácii

### 9.4.2 Zoznam záložiek

Prepínanie medzi jednotlivými záložkami je možné pri použití klávesovej skratky ALT + číslo záložky. Napríklad ALT + 1 je pre prvú záložku, ALT + 2 pre druhú atď.

## 9.4.3 Informačný pás

Informačný pás (*Status bar*) (obr. 9.4.3.) informuje používateľa o aktuálnom stave aplikácie. Bol rozdelený do troch menších častí.

Vyberte si z ponuky odomknutých tlačidiel	Connected	25.4.2006 23:29:04
obr. 9.4.3.		

- prvá časť vľavo slúži pre komunikáciu s používateľom. Informuje v akom stave sa nachádza aplikácia. Napríklad hláškou "Vyberte si z ponuky odomknutých tlačidiel" informuje používateľa o stave čakania na stlačenie tlačidla z prístupných tlačidiel.
- stredná časť pásu informuje o tom, či pripojenie aplikácie na databázový server je stále aktívne.
- tretia časť, úplne v pravo, informuje o aktuálnom dátume a čase.

#### 9.4.4 Ovládanie aplikácie

#### Klávesové skratky

Pre ľahšiu prácu s aplikáciou boli použité vo veľkej miere klávesové skratky. Každé tlačidlo má svoju vlastnú klávesovú skratku. Tá bola navrhnutá tak, aby pozostávala zo stlačenia tlačidla ALT a príslušného písmena. Toto písmeno bolo volené tak, aby bolo ovládanie aplikácie intuitívne aj pre laického používateľa, použilo sa prvé písmeno z textu tlačidla alebo ďalšie v poradí, ktoré bolo možné použiť. Napríklad pre tlačidlo Query bolo zvolené tlačidlo Q, čiže sa naň dá pristúpiť klávesovou kombináciou ALT + Q alebo stlačením tlačidla Q. Tieto klávesové skratky pre jednotlivé komponenty sú viditeľné tak, že príslušné písmeno je v texte komponentu podčiarknuté.

#### Klávesy Enter, ESC a Tab

Ak je používateľ v móde hľadania, pridávania alebo zmeny záznamu, pomocou tlačidla Enter spustí vykonanie konkrétnej akcie. Napríklad pri pridávaní nového záznamu, keď používateľ vloží všetky potrebné údaje do položiek a stlačí klávesu **Enter, spustí** zapísanie nového záznamu do databázy. Stlačením klávesy **ESC** pri vkladaní údajov na záložke, sa aktuálna akcia **zruší** a aplikácia sa vráti do stavu pre začatím akcie. Tlačidlo **Tab** slúži na **prepínanie** medzi prístupnými položkami na záložke.

#### Tlačidlá

Tlačidlá v aplikácií slúžia viac menej na spúšťanie akcií ako sú vyhľadávanie, upravovanie, vkladanie a mazanie záznamov(*Query*, *Add*, *Update*, *Remove*). Ak výsledkom vyhľadávania je množina záznamov, na prechádzanie po jednotlivých záznamoch nám slúžia tlačidla *Next* a *Previous*. Ďalšie slúžia na prepínanie medzi jednotlivými záložkami(*Master*, *Detail*), ktorých dáta sú vo vzťahu Master-Detail.

**Tlačidlo** Query – slúži na odomknutie a vyčistenie políčok, pre načítanie podmienok vyhľadávania

**Tlačidlo** – slúži na odomknutie a vyčistenie políčok, pre načítanie nového záznamu, ktorý chce používateľ vložiť

**Tlačidlo** <u>Update</u> – slúži a odomknutie políčok naplnených hodnotami aktuálneho záznamu, ktorý chce používateľ zmeniť

**Tlačidlo** <u>**Remove**</u> – pomocou tohto tlačidla sa vykoná odstránenie aktuálneho záznamu z databázy

**Tlačidlo** <u>Next >></u> – ponúka možnosť prepnutia na ďalší záznam v množine nájdených záznamov

**Tlačidlo** <- <u>Previous</u> – ponúka možnosť prepnutia na predchádzajúci záznam v množine nájdených záznamov

**Tlačidlo** — Master – slúži na prepnutie na záložku typu Master

**Tlačidlo** <u>**Detail**</u> – slúži na prepnutie na záložku typu Detail

#### Spracovanie načítaných hodnôt

Ak pri hľadaní používateľ nezadá žiadny reťazec do políčka, podľa ktorého sa dá vyhľadávať, hľadaný reťazec sa nahradí znakom "%", ktorý v SQL syntaxy znamená reťazec rôznej dĺžky. Pre explicitné definovanie reťazca rôznej dĺžky použijeme znak "\*". Napríklad:

Používateľ chce vyhľadávať osobné záznamy všetkých ľudí, ktorí sa narodili v roku 1982, a ktorých mená sa začínajú na písmeno "O". Vyhľadávanie na položke *Osobne údaje* je možné podľa troch položiek a to podľa rodného čísla, mena a priezviska. Preto používateľ vykoná nasledovné kroky:

- prepne sa na záložku Osobné údaje

- stlačí ALT + Q alebo stlačí tlačidlo *Query*
- do položky rodné číslo napíše hodnotu "82\*"
- do položky Meno napíše hodnotu "O\*"
- stlačí Enter

Pri nevyplnení ani jednej položky určenej pre vyhľadávanie a stlačení klávesy *Enter*, aplikácia požiada o potvrdenie od používateľa o zaslaní požiadavky pre vypísanie všetkých záznamov záložky.



9.4.5 Prihlasovanie sa do aplikácie

Pri spustení aplikácie je používateľ vyzvaný ku prihláseniu do aplikácie (obr. 9.4.5.). prihlásenie Toto ie z dôvodu bezpečnosti prístupu ku citlivým dátam a tiež z dôvodu implementácie rôznych úrovní prístupu ku dátam.

Heslo do aplikácie si môže každý používateľ meniť sám cez položku *Nastavenia* v hlavnom menu.

# 9.4.6 Privilégia užívateľov

S aplikáciou budú pracovať dve skupiny používateľov. Prvou bude skupina, ktorá má právo len na čítanie dát z databázy. Táto skupina ma povolené v aplikácií len jedno funkčné tlačidlo, *Query*. Druhá skupina používateľov má plné práva na dáta, takže má sprístupnené všetky funkčné tlačidla.

# 9.5 Záložka Študent

Táto záložka sa zaoberá spracovaním osobných informácií študentov. Medzi tieto informácie patria identifikačné údaje ako sú rodné číslo, meno a priezvisko, trvalé bydlisko, štátna príslušnosť, národnosť, telefónne číslo a číslo dokladu totožnosti. Táto záložka je východzia pri štarte aplikácie.

## 9.5.1 Vzhľad



Údaje na záložke sú rozdelené do troch logických častí (obr. 9.5.1.).

Prvou časťou (1) sú všeobecné informácie o osobe a to sú rodné číslo, meno, priezvisko, stav študenta a či má záujem o inžinierske štúdium. Keďže meno a priezvisko sú veľmi dôležité, sú väčšie ako ostatné údaje na formulári. Údaje o tom, či študent je na inžinierskom štúdiu a údaj, či má záujem

o inžinierske štúdium je riešené textovým odkazom na formulári.

Druhá časť (2) je venovaná trvalému bydlisku študenta.

**V tretej časti (3)** sú údaje o doklade **totožnosti** pre slovenských a zahraničných študentov riešené odlišne. Pre slovenských študentov tento údaje predstavuje číslo občianskeho preukazu a pre zahraničných študentov číslo pasu.

## 9.5.2 Vyhľadanie záznamov

- 1. Stlač tlačidlo Query (ATL + Q).
- 2. Vyplň prístupné položky pre vyhľadávanie.
- 3. Stlač klávesu Enter.

Prístupné položky pre hľadanie:

- rodné číslo
- meno
- priezvisko
- podmienka či je študent na inžinierskom štúdiu
- podmienka či má študent záujem o inžinierske štúdium

## 9.5.3 Pridávanie záznamu

- 1. Stlač tlačidlo  $\underline{Add}$  (ALT + A).
- 2. Vyplň prístupné položky pre vkladanie nového záznamu.
- 3. Stlač klávesu **Enter** alebo tlačidlo (ALT + A).

Prístupné položky pre vkladanie nového záznamu:

- rodné číslo
- meno
- priezvisko
- ulica
- obec
- PSČ
- okres

- kraj
- štát
- štátna príslušnosť
- národnosť
- telefón
- číslo ob. p/pasu

Povinné položky pre vkladanie nového záznamu:

- rodné číslo
- meno
- priezvisko
- obec
- štátna príslušnosť
- národnosť
- číslo ob. p/pasu

Po vyplnení a odoslaní požiadavky na zápis záznamu do databázy, sa vykoná niekoľko kontrol vstupných dát:

- kontrola, či zadané rodné číslo už neexistuje v databáze. Je to taktiež ochrana proti duplicite kľúčov v tabuľke databázy.
- niekoľko testov rodného čísla. Tu sa kontroluje dĺžka zadaného reťazca, ktorá musí byť jedenásť znakov a prítomnosť znaku "/", ktorý musí byť na siedmej pozícií. Ďalej sa kontrolujú jednotlivé čísla v zadanom rodnom čísle. Deň môže byť v intervale od jedna do tridsať jeden, mesiac v intervale od jedna do dvanásť alebo od päť desiat jeden do šesť desiat dva v prípade rodných čísel žien.

Ako doklad totožnosti sa pre slovenských študentov používa číslo občianskeho preukazu a pre študentov zahraničných je to číslo pasu. Táto závislosť sa odvíja od nastavenia národnosti.

## 9.5.4 Aktualizácia záznamu

- aktualizácia záznamu je možná len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.5.2).

- 1. Na vybranom aktuálnom zázname stlač tlačidlo Update (ALT + U).
- 2. Zmeň požadované prístupné položky pre aktualizáciu nového záznamu.
- 3. Stlač klávesu **Enter** alebo tlačidlo **Update** (ALT + U).

Prístupné položky pre aktualizáciu záznamu:

- meno
- priezvisko
- ulica
- obec
- PSČ
- okres
- kraj
- štát
- štátna príslušnosť
- národnosť
- telefón
- číslo ob. p/pasu

Povinné položky pre aktualizáciu záznamu:

- meno
- priezvisko
- obec
- štátna príslušnosť
- národnosť
- číslo ob. p/pasu

Pri odoslaní požiadavky na zápis aktualizovaných dát do databázy sa vykonávajú rovnaké testy na hodnotách vstupných dát ako pri zápise nového záznamu okrem tých, ktoré sa týkali rodného čísla, keďže to sa meniť nemôže.

## 9.5.5 Odstránenie záznamu

- odstránenie záznamu je možné len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.5.2).

- 1. Na vybranom aktuálnom zázname stlač tlačidlo (ALT + R).
- 2. Potvrď súhlas o odstránení aktuálneho záznamu z databázy (obr. 9.5.5.).

Warn	ing	2
	Chete naozaj vymazať šti Bylina s rod. číslom 82030	udenta menom Peter 3/1234?
	ОК	Cancel

Po stlačení tlačidla OK je záznam vymazaný, inak nie.

# 9.6 Záložka "Zápis študenta"

Každý študent má svoje osobné študentské číslo podľa, ktorého je jednoznačne identifikovateľný v škole. Taktiež každý študent má svoje zameranie, študijnú skupinu do ktorej patrí, určenú formu štúdia, stupeň štúdia a ďalšie študentské záznamy. Správou týchto študentských údajov sa zaoberá práve táto záložka. Ponúka aj stručný prehľad ďalších doplňujúcich záznamov o študentoch, ako je napríklad zobrazenie informácií o poslednom opakovaní ročníka alebo prerušení štúdia.

## 9.6.1 Vzhľad



Záložka je opticky rozdelená do štyroch logických skupín (obr. 9.6.1.).

V hornej časti (1) sa nachádzajú všeobecné informácie, ktoré slúžia pre rýchlu identifikáciu osoby. Taktiež v hornej pravej časti je zobrazený študentských počet záznamov pre dané rodné číslo, keďže jedna osoba môže byť zapísaná na školu viackrát.

V strednej časti záložky (2) sa nachádzajú

študentské dáta. Medzi ne patrí zameranie, stav, ročník, študijná skupina, forma štúdia, odbor a stupeň štúdia. Taktiež v tejto časti sa nachádza tlačidlo Kontá, po ktorého stlačení sa používateľ dostane na záložku Kontá študentov.

Časť opakovaní a prerušení (3) je malý prehľad záložky Prerušení a Opakovaní. Zobrazujú sa tu najnovšie záznamy týkajúce sa opakovaní ročníkov a prerušení štúdia, ako aj počet záznamov. Pri kliknutí na ne bude používateľ prepnutý na záložku Prerušení a Opakovaní, kde môže s týmito údajmi ďalej pracovať.

**Posledná časť** (4) záložky zobrazuje dátum prvého zápisu na školu a výšku pôžičky, ktorú si študent požičal od školy. V prípade pôžičky je pole zvýraznené červenou farbou.

## 9.6.2 Vyhľadanie záznamov

- 1. Stlač tlačidlo Query (ALT + Q).
- 2. Vyplň prístupné položky pre vyhľadávanie.
- 3. Stlač klávesu Enter.

Prístupné položky pre hľadanie:

- rodné číslo
- osobné číslo
- ročník
- študijná skupina
- stav študenta

## 9.6.3 Pridávanie záznamu

- 1. Stlač tlačidlo  $\underline{\underline{A}}$  (ALT + A).
- 2. Vyplň prístupné položky pre vkladanie nového záznamu.
- 3. Stlač klávesu **Enter** alebo tlačidlo <u>Add</u> (ALT + A).

Prístupné položky pre vkladanie nového záznamu:

- osobné číslo
- zameranie
- stav študenta
- ročník

- študijná skupina
- dátum prvého zápisu

Povinné položky pre vkladanie nového záznamu:

- osobné číslo
- stav študenta
- ročník
- študijná skupina
- dátum prvého zápisu

Po vyplnení a odoslaní požiadavky na zápis záznamu do databázy, sa vykoná niekoľko kontrol vstupných dát:

- zistí sa, či položka *rodné číslo* nie je prázdna a potom či v databáze už existuje študent s rovnakým osobným číslom
- skontroluje sa hodnota stavu študenta. Táto hodnota musí byť len z množiny povolených hodnôt "S" (študent), "K" (ukončený), "E" (zanechal štúdium), "V" (vylúčený), "P" (prerušený) a "Z" (zahraničný).
- hodnota ročníka musí byť v intervale celých čísel od nula do päť
- dĺžka hodnoty študijnej skupiny musí byť rovná šiestim
- dátum prvého zápisu musí byť dátumom

## 9.6.4 Aktualizácia záznamu

- aktualizácia záznamu je možná len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.6.2).

1. Na vybranom aktuálnom zázname stlač tlačidlo Update (ALT + U).

- 2. Zmeň požadované prístupné položky pre aktualizáciu nového záznamu.
- 3. Stlač klávesu **Enter** alebo tlačidlo **Update** (ALT + U).

Prístupné položky pre aktualizáciu záznamu:

- zameranie
- stav študenta
- ročník
- študijná skupina
- dátum prvého zápisu

Povinné položky pre aktualizáciu záznamu:

- stav študenta
- ročník
- študijná skupina
- dátum prvého zápisu

Po odoslaní požiadavky o zmenu dát, prebehnú rovnaké testy na vstupné dáta ako pri pridávaní záznamu, ale bez kontroly týkajúcej sa *osobného čísla*.

## 9.6.5 Odstránenie záznamu

- odstránenie záznamu je možné len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.6.2).

1. Na vybranom aktuálnom zázname stlač tlačidlo  $\underline{\mathbb{R}}$ emove (ALT + R).

2. Potvrď súhlas o odstránení aktuálneho záznamu z databázy (obr. 9.6.5.).



obr. 9.6.5.

Po stlačení tlačidla OK je záznam vymazaný, inak nie.

## 9.7 Záložka Zápis predmetov

Táto záložka je jedna z najdôležitejších pretože sa zaoberá správou zapísaných predmetov pre študenta. Pomocou nej sa dajú zapisovať nové predmety, vymazávať a editovať už zapísané predmety. Pre každý predmet na tejto záložke je možné zistiť viacero detailov ako je napríklad meno prednášajúceho, počet kreditov, dátum zápočtu a skúšky, známka z predmetu, počet termínov atď.

## 9.7.1 Vzhľad



Záložka je opticky rozdelená do troch častí podľa funkčnosti (obr. 9.7.1.).

V prvej časti (1) sa nachádzajú všeobecné orientačné údaje o študentovi, ktorého údaje si prezeráme. Je tu meno, priezvisko, osobné číslo a školský rok. Školský rok je rok, pre ktorý daný predmet bol zapísaný.

Druhá časť (2) záložky sa venuje samotnému predmetu.

Nachádza sa tu číslo a názov predmetu, číslo a meno prednášajúceho, ktorý pre daný školský rok predmet prednášal. Ďalej tu nájdeme informácie o počte kreditov za daný predmet ako aj miesto štúdia predmetu.

**Tretia časť** (3) záložky je venovaná **dosiahnutým výsledkom** študenta na predmete. Nachádzajú sa tu *dátumy zápočtu* a *skúšky*. *Forma kontroly* hovorí o podmienkach absolvovania predmetu. Ďalej tu sú políčka *Známka, Termín* a *Absolvovaný*. *Termín* je počet absolvovaných skúškových termínov pre daný predmet

ku absolvovaniu predmetu. Políčko *Absolvovaný* podľa daných podmienok vypíše, či predmet bol absolvovaný alebo nie.

## 9.7.2 Vyhľadanie záznamov

- 1. Stlač tlačidlo Query (ALT + Q).
- 2. Vyplň prístupné položky pre vyhľadávanie.
- 3. Stlač klávesu Enter.

Prístupné položky pre hľadanie:

- osobné číslo
- školský rok
- číslo predmetu
- názov predmetu
- miesto štúdia

Pri každom zobrazení zapísaných predmetov sa políčko informujúce o absolvovaní predmetu aktualizuje. Jeho hodnota závisí od viacerých podmienok. Medzi tie podmienky patrí forma *kontroly predmetu*, *dátum skúšky*, *dátum zápočtu* a *výsledná známka*. Vyhodnotenie sa drží nasledovných pravidiel:

- ak je forma kontroly predmetu typu "e", to znamená, že predmet aby bol uznaný ako absolvovaný, musí byť ukončený skúškou. Ak má dotyčný študent zapísaný dátum skúšky a taktiež známku, ktorá je buď v rozsahu od 1 do 4, alebo v rozsahu od A po D, predmet je vyhodnotený ako absolvovaný.
- pri forme kontroly typu "z", ktorá znamená, že predmet je uznaný za absolvovaný udelením zápočtu, je predmet prehlásený za absolvovaný ak hodnota dátumu zápočtu nie je prázdna hodnota.

ak spojíme podmienky pre formu kontroly typu "e" s podmienkou pre formu kontroly typu "z" dostanem podmienky pre formu kontroly typu "s".

Ak je predmet vyhodnotený ako neabsolvovaný, pole informujúce o absolvovaní zmení farbu na červenú.

## 9.7.3 Pridávanie záznamu

- 1. Stlač tlačidlo  $\underline{Add}$  (ALT + A).
- 2. Vyplň prístupné položky pre vkladanie nového záznamu.
- 3. Stlač klávesu **Enter** alebo tlačidlo  $\underline{Add}$  (ALT + A).

Prístupné položky pre vkladanie nového záznamu:

- číslo a názov predmetu
- typ predmetu
- miesto štúdia
- dátum zápočtu
- dátum skúšky
- známka
- termín

Povinné položky pre vkladanie nového záznamu:

- číslo a názov predmetu
- typ predmetu
- miesto štúdia

Pred samotným zápisom dát do databázy, prebehnú kontroly zadaných dát:

- kontrola či daný predmet už nemá študent zapísaný.

- kontrola dátumu zápisu a skúšky.
- známka musí byť v množine "A", "B", "C", "D", "E", "F" alebo "FX". Pri zadaní hodnoty "F" sa do databázy uloží hodnota "FX".
- počet termínov musí byť menší ako počet termínov nastavený globálnou premennou aplikácie. Táto premenná je pri štarte nastavená na hodnotu tri, ale dá sa meniť pomocou hlavného menu v *Nastaveniach*.
- typ predmetu musí byť len z povolenej množiny hodnôt "P" (povinný), "V" (voliteľný), "A" (alternatívny), "Z" (záverečný) alebo "S" (povinne voliteľný).
- hodnota miesta štúdia predmetu musí byť z prístupnej množiny hodnôt "P", "R" alebo "Z".

Pri pridávaní záznamu predmetu sa v položke *Číslo a názov predmetu* nachádzajú predmety len z určitého školského roku. Tento školský rok sa dá zmeniť cez položku hlavného menu *Nastavenia*. Pri jeho zmene sa aktualizuje zoznam predmetov. Táto hodnota je pri štarte aplikácie nastavená na aktuálny školský rok.

## 9.7.4 Aktualizácia záznamu

- aktualizácia záznamu je možná len na už existujúcom zázname, ktorý bol vybratý cez
 Vyhľadávanie záznamov (ods. 9.7.2).

- 1. Na vybranom aktuálnom zázname stlač tlačidlo Update (ALT + U).
- 2. Zmeň požadované prístupné položky pre aktualizáciu nového záznamu.
- 3. Stlač klávesu **Enter** alebo tlačidlo **Update** (ALT + U).

Prístupné položky pre aktualizáciu záznamu:

- typ predmetu
- miesto štúdia
- dátum zápočtu
- dátum skúšky
- známka
- termín

Povinné položky pre aktualizáciu záznamu:

- typ predmetu
- miesto štúdia

Po odoslaní požiadavky o zmenu dát, prebehnú rovnaké testy na vstupné dáta, ako pri pridávaní záznamu.

## 9.7.5 Odstránenie záznamu

- odstránenie záznamu je možné len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.7.2).

- 1. Na vybranom aktuálnom zázname stlač tlačidlo <u>Remove</u> (ALT + R).
- 2. Potvrď súhlas o odstránení aktuálneho záznamu z databázy (obr. 9.7.5.)



obr. 9.7.5.

Po stlačení tlačidla OK je záznam vymazaný, inak nie.

# 9.8 Záložka Prerušenia a Opakovania

Niektorí študenti opakujú ročníky, iní prerušia štúdium. Pre ošetrenie práve týchto prípadov bola vytvorená táto záložka. Ponúka stručný prehľad tak prerušení štúdia ako aj opakovaní ročníka a manipuláciu s nimi.

## 9.8.1 Vzhľad



Záložka je rozdelená na tri optické časti (obr. 9.8.1).

Horná časť (1) sú všeobecné informácie o študentovi. Nachádza sa tam *meno, priezvisko* a *osobné číslo* študenta.

Druhá časť (2) je venovaná prerušeniam. Pri každom prerušení sú evidované informácie školský rok, dátum začiatku prerušenia, dátum konca prerušenia a ročník. Táto časť má tlačidla Add, vlastné

Remove a Update, ktorými sa pracuje s prerušeniami.

**Spodná časť** (3) sa zaoberá *opakovaniam*. Pre každý záznam opakovania sa vedú informácie o *školskom roku* a *ročníku*. Tlačidlá *Add, Remove* a *Update* slúžia na manipuláciu s opakovaniami.

## 9.8.2 Vyhľadanie záznamov

1. Stlač tlačidlo  $\underline{Query}$  (ALT + Q).

- 2. Vyplň prístupné položky pre vyhľadávanie.
- 3. Stlač klávesu Enter.

Prístupné položky pre hľadanie:

- osobné číslo

# 9.8.3 Pridávanie záznamu prerušenia

- 1. Stlač tlačidlo <u>Add</u> pod časťou zaoberajúcej sa prerušeniami.
- 2. Vyplň vstupné dialógy dátami:

## Školský rok:

	? ×
OK	Canad
	OK

obr. 9.8.3.1.

Dátum začiatku prerušenia:



obr. 9.8.3.2.

Dátum konca prerušenia:

🌈 Dátum		? ×
Do dátumu:		
	DIE	
	QK	Cancel

obr. 9.8.3.3.

#### Ročník:

	? ×
<u>o</u> k	Cancel
	QK

obr. 9.8.3.4.

Pri vkladaní hodnôt, prebiehajú testy na týchto hodnotách:

- školský rok sa testuje či má dĺžku štyri znaky, a či vložená hodnota je číslom
- kontrola dátum začiatku a dátum konca prerušenia
- ročník musí byť celé číslo v intervale od nula do päť

Pred samotným zapísaním nových dát sa vykoná kontrola existencie rovnakého záznamu v databáze.

## 9.8.4 Aktualizácia záznamu prerušenia

- aktualizácia záznamu je možná len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.8.2).

- 1. Vyber si zo zoznamu prerušení a stlač tlačidlo **Update** pod časťou zaoberajúcej sa prerušeniami.
- 2. Vyplň vstupné dialógy dátami, ako pri pridávaní nového záznamu (9.8.3).

Po odoslaní požiadavky o zmenu dát, prebehnú rovnaké testy na vstupné dáta ako pri pridávaní záznamu.

## 9.8.5 Odstránenie záznamu prerušenia

- odstránenie záznamu je možné len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.8.2).

- 1. Vyber si zo zoznamu prerušení a stlač tlačidlo **Remove** pod časťou zaoberajúcej sa prerušeniami.
- 2. Potvrď súhlas o odstránení aktuálneho záznamu z databázy (obr. 9.8.5.).



obr. 9.8.5.

Po stlačení tlačidla OK je záznam vymazaný, inak nie.

## 9.8.6 Pridávanie záznamu opakovania

- 1. Stlač tlačidlo **Add** pod časťou zaoberajúcej sa opakovaniami.
- 2. Vyplň vstupné dialógy dátami:

## Školský rok:

🕻 Školský rok		?
Školský rok:		
		1 123 14
	QK	Cancel

obr. 9.8.6.1.

## Ročník:



obr. 9.8.6.2.

Pri vkladaní hodnôt, prebiehajú testy na týchto hodnotách:

- školský rok sa testuje či má dĺžku štyri znaky, a či vložená hodnota je číslom
- *ročník* musí byť celé číslo v intervale od nula do päť

Pred samotným zapísaním nových dát sa vykoná kontrola existencie rovnakého záznamu v databáze.

## 9.8.7 Aktualizácia záznamu opakovania

- aktualizácia záznamu je možná len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.8.2).

- 1. Vyber si zo zoznamu opakovaní a stlač tlačidlo Update pod časťou zaoberajúcej sa opakovaniami.
- 3. Vyplň vstupné dialógy dátami, ako pri pridávaní nového záznamu (ods. 9.8.6).

Po odoslaní požiadavky o zmenu dát, prebehnú rovnaké testy na vstupné dáta ako pri pridávaní záznamu.

## 9.8.8 Odstránenie záznamu opakovania

- odstránenie záznamu je možné len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.8.2).

- 1. Vyber si zo zoznamu opakovaní a stlač tlačidlo **Remove** pod časťou zaoberajúcej sa opakovaniami.
- 2. Potvrď súhlas o odstránení aktuálneho záznamu z databázy (obr. 9.8.8.).


obr. 9.8.8.

Po stlačení tlačidla OK je záznam vymazaný, inak nie.

## 9.9 Záložka Kontá študenta

Študenti na Fakulte Riadenia a Informatiky majú svoje prihlasovacie meno (*login*) a *heslo* do učební a Internetu na škole. Tieto heslá sú vytvárané systémovým administrátorom. Záložka slúži len pre distribúciu týchto informácií na študijné oddelenia na detašovaných pracoviskách, kde sú ďalej poskytované študentom.

#### 9.9.1 Vzhľad



Plocha záložky nie je plne využitá, ale v budúcnosti, v prípade potreby, sa dá zvyšný priestor využiť aj na iné podobné informácie (obr. 9.9.1.).

V prvej časti (1) sú základne informácie o študentovi ako sú: osobné číslo, meno a priezvisko.

Druhá časť (2) je zameraná údajom týkajúcich sa študentských kont. Sú tu položky: *študijná skupina,* 

#### login, heslo a dátum vytvorenia záznamu.

Na tejto záložke nie sú iné tlačidlá ako je *Master* tlačidlo. **Vyhľadávanie** informácii na tejto záložke je možné **len** cez vzťah **Master-Detail** so záložkou *Zápis študenta*. Tlačidlá *Add* a *Update* tu nie sú, pretože táto položka sa používa len ako informatívna záložka pre distribúciu informácií na študijné oddelenia detašovaných pracovísk.

# 9.9.2 Vyhľadanie záznamov

- 1. Stlač tlačidlo Query (ALT + Q).
- 2. Vyplň prístupné položky pre vyhľadávanie.
- 3. Stlač klávesu Enter.

#### Prístupné položky pre hľadanie:

- osobné číslo

# 9.10 Záložka Financie

Záložka zaoberajúca sa finančnou agendou medzi študentom a školou. Táto agenda spočíva v správe **pôžičiek, štipendií** a **platieb** študenta za nadštandardnú dĺžku štúdia.

## 9.10.1 Vzhľad



Záložka je rozdelená na štyri optické časti (obr. 9.10.1.).

Horná časť (1) sú všeobecné informácie o študentovi. Nachádza sa tu osobné číslo, meno a priezvisko.

Druhá časť (2) je venovaná pôžičkám. Pri každej pôžičke sú evidované informácie ako sú *Suma* a *Školský rok*.

**Tretia časť** (3) je venovaná **štipendiám**. Pri každom štipendiu sú vedené

informácie ako Suma, Školský rok a Typ štipendia.

**Štvrtá časť** (4) je venovaná **platbám** študenta škole napríklad za nadštandardnú dĺžku štúdia a pod.

Každá z častí 2, 3 a 4 majú vlastné tlačidlá *Add, Remove* a *Update*, ktoré slúžia na manipuláciu s dátami v jednotlivých častiach.

#### 9.10.2 Vyhľadanie záznamov

- 1. Stlač tlačidlo  $\underline{Query}$  (ALT + Q).
- 2. Vyplň prístupné položky pre vyhľadávanie.

3. Stlač klávesu Enter.

Prístupné položky pre hľadanie:

- osobné číslo

### 9.10.3 Pridávanie záznamu pôžičky

- 1. Stlač tlačidlo <u>Add</u> pod časťou zaoberajúcej sa pôžičkami.
- 2. Vyplň vstupné dialógy dátami:

#### Suma:

🌈 Suma		? ×
Suma:		
	<u>O</u> K	<u>C</u> ancel
	obr. 9.10.3.1.	

Školský rok:

OV	Crewd
	<u>O</u> K

obr. 9.10.3.2.

Pri vkladaní hodnôt, prebiehajú testy na týchto hodnotách:

- zisťuje sa či suma je číslo
- školský rok sa testuje či má dĺžku štyri znaky, a či vložená hodnota je číslom

Pred samotným zapísaním nových dát sa vykoná kontrola existencie rovnakého záznamu v databáze.

#### 9.10.4 Aktualizácia záznamu pôžičky

- aktualizácia záznamu je možná len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.10.2).

- 1. Vyber si zo zoznamu pôžičiek a stlač tlačidlo Update pod časťou zaoberajúcej sa pôžičkami.
- 2. Vyplň vstupné dialógy dátami, ako pri pridávaní nového záznamu (ods. 9.10.3).

Po odoslaní požiadavky o zmenu dát, prebehnú rovnaké testy na vstupné dáta ako pri pridávaní záznamu.

#### 9.10.5 Odstránenie záznamu pôžičky

- odstránenie záznamu je možné len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (viď ods. 9.10.2).

- 1. Vyber si zo zoznamu pôžičiek a stlač tlačidlo **Remove** pod časťou zaoberajúcej sa pôžičkami.
- 2. Potvrď súhlas o odstránení aktuálneho záznamu z databázy (obr. 9.10.5).



obr. 9.10.5.

Po stlačení tlačidla **OK** je záznam vymazaný, inak nie.

#### 9.10.6 Pridávanie záznamu štipendia

1. Stlač tlačidlo **Add** pod časťou zaoberajúcej sa štipendiami.

2. Vyplň vstupné dialógy dátami:

#### Suma:

	? ×
OK	Cancel
	04

obr. 9.10.6.1.

#### Školský rok:

🌠 Školský rok	<u>?</u> ×
Školský rok:	
[	
	OK Cond

obr. 9.10.6.2.

#### Typ štipendia:

🌈 Typ štipendia		<u>? ×</u>
Typ štipendia [S-sociálne,P-prospe	chové,R-cena rektora,D-cena de	kana,M-mimoriadné]:
	<u>o</u> k	Cancel
1	obr. 9.10.6.3.	

Pri vkladaní hodnôt, prebiehajú testy na týchto hodnotách:

- zisťuje sa či suma je číslo
- *školský rok* sa testuje či má dĺžku štyri znaky, a či vložená hodnota je číslom
- *typ štipendia* môže byť len "S" (sociálne), "P" (prospechové), "R" (cena rektora),
   "D" (cena dekana) a "M" (mimoriadne)

Pred samotným zapísaním nových dát sa vykoná kontrola existencie rovnakého záznamu v databáze.

# 9.10.7 Aktualizácia záznamu štipendia

- aktualizácia záznamu je možná len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.10.2).

- 1. Vyber si zo zoznamu štipendií a stlač tlačidlo **Update** pod časťou zaoberajúcej sa štipendiami.
- 2. Vyplň vstupné dialógy dátami, ako pri pridávaní nového záznamu (ods. 9.10.6).

Po odoslaní požiadavky o zmenu dát, prebehnú rovnaké testy na vstupné dáta ako pri pridávaní záznamu.

#### 9.10.8 Odstránenie záznamu štipendia

- odstránenie záznamu je možné len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (viď ods. 9.10.2).

- 1. Vyber si zo zoznamu štipendií a stlač tlačidlo **Remove** pod časťou zaoberajúcej sa štipendiami.
- 2. Potvrď súhlas o odstránení aktuálneho záznamu z databázy (obr. 9.10.8).



obr. 9.10.8.

Po stlačení tlačidla OK je záznam vymazaný, inak nie.

#### 9.10.9 Pridávanie záznamu platby

- 1. Stlač tlačidlo <u>Add</u> pod časťou zaoberajúcej sa platbami.
- 2. Vyplň vstupné dialógy dátami:

#### Suma:

	? ×
0.00	

obr. 9.10.9.1.

#### Školský rok:

🌈 Školský rok	? )
Školský rok:	
1	
1	

#### obr. 9.10.9.2.

#### Semester:

ОК	Cancel
	<u>O</u> K



Pri vkladaní hodnôt, prebiehajú testy na týchto hodnotách:

- zisťuje sa či suma je číslo
- školský rok sa testuje či má dĺžku štyri znaky, a či vložená hodnota je číslom
- *semester* určuje či platba bola vykonaná za zimný semester (hodnota 1), za letný semester (hodnota 2) alebo za celý rok (hodnota 0)

Pred samotným zapísaním nových dát sa vykoná kontrola existencie rovnakého záznamu v databáze.

## 9.10.10 Aktualizácia záznamu platby

- aktualizácia záznamu je možná len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (ods. 9.10.2).

- 1. Vyber si zo zoznamu platieb a stlač tlačidlo Update pod časťou zaoberajúcej sa platbami.
- 2. Vyplň vstupné dialógy dátami, ako pri pridávaní nového záznamu (ods. 9.10.9).

Po odoslaní požiadavky o zmenu dát, prebehnú rovnaké testy na vstupné dáta ako pri pridávaní záznamu.

#### 9.10.11 Odstránenie záznamu platby

- odstránenie záznamu je možné len na už existujúcom zázname, ktorý bol vybratý cez
Vyhľadávanie záznamov (viď ods. 9.10.2).

- 1. Vyber si zo zoznamu štipendií a stlač tlačidlo **Remove** pod časťou zaoberajúcej sa štipendiami.
- 2. Potvrď súhlas o odstránení aktuálneho záznamu z databázy (obr. 9.10.11).

Chcete naozaj sume 3000 pre	vymazať záznam o <mark>platbách v</mark> študenta Peter Bylina?
OK	Cancel

Po stlačení tlačidla OK je záznam vymazaný, inak nie.

# 9.11 Zostava Kontrola štúdia

Kontrola štúdia je spracovanie dát, ktorého výsledkom je prehľad zapísaných predmetov študentovi. Predmety sú zoskupené podľa jednotlivých školských rokov. Pri jednotlivých predmetoch je zobrazený **výsledok** z predmetu a taktiež počet nadobudnutých **kreditov**. Na konci každého školského roku je vypočítaný **vážený študijný priemer** a **počet kreditov**. Na konci správy je zobrazený celkový vážený študijný priemer, celkový počet kreditov študenta a zoznam predmetov, ktoré študent neabsolvoval.

Pre vytvorenie správy slúži nasledujúci postup:

- prepni sa na záložku Zápis predmetov
- nájdi študenta (ods. 9.7.2.)
- z hlavného menu vyber ponuku Reporty

Na aktuálne nastavenej tlačiarni sa vytlačí požadovaná zostava.

# 10 Záver

Hlavným cieľom diplomovej práce bolo vyvinúť informačný systém fakulty FRI pre prostredie Windows. Táto požiadavka vznikla z dôvodu stále väčších nárokov na existujúci fakultný informačný systém (vytváranie a tlač výstupných zostáv,...).

Navrhnutý a implementovaný informačný systém je využiteľný pre kompletnú agendu týkajúcu sa študenta: osobné údaje, študentské údaje, zapísané predmety, prerušenia štúdia, opakovania ročníkov, platby, atď. Celý tento informačný systém bol vyvíjaný pomocou produktov: Delphi 7 a ZeosLib. Aplikácia bola naprogramovaná použitím objektového prístupu a tým je zabezpečená možnosť doprogramovania ďalších potrebných funkcií aplikácie v budúcnosti.

Aplikácia je odolná len voči známym chybám, ak dôjde k chybe s ktorou som sa počas vývoja aplikácie nestretol, môže dôjsť k neštandardnému správaniu.

# Zoznam príloh

Príloha 1 CD s vytvorenou aplikáciou

# Zoznam použitej literatúry

[1] doc. Ing. Karol Matiaško, PhD.: Databázové systémy, Vydala Žilinská univerzita v Žiline 2002

[2] WWW stránky:

http://fundementals.sourceforge.net/index.html http://en.wikipedia.org/wiki/Md5 http://en.wikipedia.org/wiki/SHA-1 http://www.ipower.sk/dbs/rdbms.html http://www.nevrona.com/Default.aspx?tabid=30 http://www.w3schools.com/sql/