

TECHNICKÁ UNIVERZITA V LIBERCI

Hospodářská fakulta

Studijní program: 6209T – Systémové inženýrství a informatika

Studijní obor: Manažerská informatika

Analýza a návrh univerzitního informačního systému

(Analysis and design of university information system)

DP – MI-KIN-2006 10

Jakub Lank

Vedoucí diplomové práce: prof. Ing. Ehleman Jan, CSc.

Odborný konzultant: Ing. Vladimíra Zádová

Rozsah práce

Počet stran textu: 60

Počet obrázků: 20

ZADÁNÍ DIPLOMOVÉ PRÁCE

pro:

Jakub Lank

Studijní program:

Systémové inženýrství a informatika (6209T)

Studijní obor č. M 6209

Manažerská informatika

Vedoucí katedry Vám ve smyslu zákona č. 111/1998 Sb. o vysokých školách a navazujících předpisů určuje tuto bakalářskou práci:

Název tématu:

Analýza a návrh univerzitního informačního systému

Zásady pro vypracování:

1. Přístup k analýze a návrhu IS
2. Analýza požadavků na systém a návrh systému
3. Vyhodnocení nabídky na trhu z hlediska zjištěných požadavků

Rozsah bakalářské práce: 60 - 70 stran
(do rozsahu nejsou započítány úvodní listy, přehled literatury a přílohy)

Doporučená literatura:

Vedoucí bakalářské práce: prof. Ing. Ehleman Jan, CSc.

Odborný konzultant: Ing. Vladimíra Zádová

Termín odevzdání bakalářské práce: 6. 1. 2006

Prof. Ing. Jan Ehleman, CSc. Prof. Ing. Jiří Kraft, CSc.
vedoucí katedry děkan Hospodářské fakulty

V Liberci dne: 31. 3. 2005

Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 - školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li diplomovou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Diplomovou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím diplomové práce a konzultantem.

V Liberci 5.5.2006

Jakub Lank _____

Resumé

Obsahem této diplomové práce je zhodnocení funkčnosti již zavedeného informačního systému STAG Technické univerzity v Liberci. Dále na základě požadavků na tento systém a jeho analýzy navrhnout možná zlepšení v oblastech webového přístupu a poskytovaných funkcí tohoto systému. Protože je informační systém organizace velikosti univerzity příliš složitý na celkovou analýzu, zaměřím na oblast webového rozhraní, která je pro většinu uživatelů tou hlavní částí, se kterou pracují.

Abstract

The content of this diploma work is evaluation of functionality of already implemented information system STAG of Technical university in Liberec. Then suggest possible improvements based on its analysis and users' needs. Because information system of organisation, which is large as university, is too complex for complete analysis, I have focused on its web interface, which is the most used part of system for many users.

Obsah

Resumé	5
Abstract	5
Obsah	6
Seznam obrázků	8
Seznam tabulek	8
Úvod	9
1 Projektování informačního systému	10
1.1 Co je IS?	10
1.2 Úlohy IS	10
1.2.1 Kategorie úloh v aplikační vrstvě	10
1.2.2 Typy úloh v aplikační vrstvě	11
2 Principy analytických metod	13
2.1 Princip abstrakce	13
2.1.1 Top-Down princip	16
2.1.2 Generalizace – specializace v datovém modelu	16
2.1.3 Princip tří architektur (P3A)	17
2.2 Princip modelování	18
2.3 Metody strukturované analýzy a návrhu	20
2.3.1 Význam strukturalizace	21
2.3.2 Obsah strukturovaných metod – modely	21
2.3.3 Integrace modelů	22
2.3.4 Nástroje strukturovaných metod - ERD	22
2.3.5 Nástroje strukturovaných metod - DFD	25
2.3.6 Nástroje strukturovaných metod – Data dictionary	28
2.4 Objektově orientované metody analýzy a návrhu	29
2.4.1 Principy objektového přístupu	32
2.4.2 Třída a objekt	33
2.4.3 Základní vlastnosti objektů a tříd	34
3 Analýza systému a zjištění požadavků	39

3.1	Student	39
3.2	Vyučující	40
3.3	Studijní oddělení	41
3.4	Modely IS	42
3.4.1	Diagram užití	42
3.4.2	Struktura systému	43
4	Univerzitní systém STAG	45
4.1	Autorizovaný přístup	46
4.1.1	Předzápis studentů na předměty	46
4.1.2	Rozvrh	48
4.1.3	Provázanost informací s rozvrhem	49
4.1.4	Výběr předmětů	50
4.1.5	Přihlašování studentů na zkoušky	53
4.1.6	Zadávání údajů diplomové práce studentem	55
4.1.7	Zadávání termínů zkoušek	56
4.1.8	Zadávání známek	57
4.2	Veřejný přístup	58
4.2.1	Tisky	58
4.2.2	Rozvrhy	59
4.2.3	Hledání volných místností	61
4.2.4	Studijní programy a předměty	62
4.2.5	Termíny zkoušek	63
4.2.6	Ostatní	64
5	Systémy na trhu	65
5.1	Vlastní vývoj	65
5.2	Externí vývoj	67
5.3	Porovnání systémů	68
	Závěr	70
	Seznam použité literatury	71

Seznam obrázků

Obrázek 1 - Klasifikace abstrakcí, zdroj: [2] s. 139	15
Obrázek 2 - Třída a její objekt, zdroj: [2] s.226	34
Obrázek 4 Spojení tříd a spojení instancí, zdroj: [2] s. 232	37
Obrázek 5 - Use case diagram, zdroj: vlastní	42
Obrázek 6 - Generalizace a specializace, zdroj: vlastní	43
Obrázek 7 - Úvodní strana systému STAG	46
Obrázek 8 - Předzázpis, základní informace	47
Obrázek 9 – Rozvrh	48
Obrázek 10 - Předzázpis, výběr předmětů	50
Obrázek 11 - Zázpis na bloky	52
Obrázek 12 - Rozvrhové akce předmětu, zapsaní studenti	53
Obrázek 13 - Zázpis na zkoušky	53
Obrázek 14 - Formulář pro vyplnění údajů závěrečné práce	55
Obrázek 15 – Zadání termínů zkoušek	56
Obrázek 16 – Znamky	57
Obrázek 17 - Úvodní stránka veřejného přístupu	58
Obrázek 18 - Kalendář	59
Obrázek 19 - Volné místnosti	61
Obrázek 20 - Vizualizace studijního oboru	62

Seznam tabulek

Tabulka 1 - Metamodely podle MOF	32
--	----

Úvod

Cílem této diplomové práce je analýza informačního systému Technické univerzity v Liberci na základě různých přístupů, které zahrnují jak principy abstrakce a modelování, tak techniky strukturované a objektové analýzy. Dalším cílem je zjištění požadavků na funkce a vlastnosti univerzitního informačního systému od různých skupin uživatelů, včetně zapracování těchto požadavků do návrhu tohoto systému. Vlastní analýza a návrh jsou zaměřené výhradně na webové rozhraní informačního systému univerzity, protože kompletní analýza systému je velmi rozsáhlá a přesahovala by rozsah této práce. Posledním cílem této práce je zvážit finanční hledisko, tedy zda na trhu existují lepší řešení než to, které je použito Technickou univerzitou v Liberci.

1 Projektování informačního systému

Informační systém je velice komplexní záležitost. Jeho tvorba je složená z velkého množství činností.

1.1 Co je IS?

IS je automatický nebo manuální systém, který zahrnuje lidské zdroje, stroje a metody k získávání, zpracování, přenosu a šíření dat, která reprezentují uživatelské informace.

1.2 Úlohy IS

Úlohy v aplikační vrstvě v rámci informačních systémů můžeme klasifikovat na kategorie a typy úloh. Kategorie úloh vyjadřují podíl standardních aplikačních software vzhledem ke specializovanému vývoji aplikací. Typy úloh vyjadřují způsob řešení a provozu aplikací vzhledem k jejich určení (typ uživatelů, na které se orientují), k použitým informačním technologiím, k charakteru aplikačních produktů, metodikám a nárokům na provoz.

1.2.1 Kategorie úloh v aplikační vrstvě

Aplikační vrstvu informačního systému lze řešit několika způsoby:

- a) Vývojem specializovaného software podle potřeb konkrétního zákazníka, tzv. software šitý na míru. Často se jedná o projekty pro státní správu nebo bankovní instituce, tedy projekty, kde je důležitá bezpečnost. Bezpečnost je u nich částečně dosažena tím, že je IS koncipován jako uzavřený.
- b) Nákupem a instalací aplikačního software určitého typu, kde jsou pouze malé nároky na jeho úpravy. Jde o nějaké standardní řešení, které používá více

klientů pouze s malými úpravami. V těchto projektech se nepoužívají téměř žádné nástroje ani speciální metodologie.

- c) Komplexními projekty založenými na výběru velkých aplikačních software, jejich modifikaci, sloučení, s případným doděláním a zavedením těch modulů, které nejsou součástí typových řešení. Způsob běžně používaný u středních a velkých podniků a organizací.

1.2.2 Typy úloh v aplikační vrstvě

Úkolem vymezení těchto typů v aplikační vrstvě je najít charakteristiky těchto úloh a najít v nich odlišnosti v řízení:

- a) *Úlohy pro podporu taktického a operativního řízení (MIS)* - pokrývají oblasti kontroly organizace tam, kde převažují analytické operace. Cílem je neustálá kontrola produkčních procesů a zdrojů podniku.
- b) *Úlohy manažerské (EIS)* - poskytují celkové analýzy aktivit podniku podle mnoha kritérií s cílem usnadnit rozhodování.
- c) *Úlohy datového skladu (DWH)* - shromažďování informací z různých ostatních databází do jednoho místa, umožňující vykonání velkého množství analýz.
- d) *Úlohy pro elektronickou výměnu dat (EDI)* - slouží k zajištění výměny dat mezi obchodními partnery, s cílem dosáhnout zrychlení a zjednodušení tohoto procesu.
- e) *Úlohy pro podporu kancelářských prací (OIS)* - snížení počtu administrativních operací a zvýšení pořádku v dokumentech organizace. Dále zrychlení a zkvalitnění běžné komunikace mezi pracovníky

organizace užitím elektronické pošty a dalších levných zdrojů v síti Internet.

- f) Úlohy výrobního typu CAD/CAM - slouží ke snížení nákladů ve výrobě a zefektivnění výroby obecně.
- g) Úlohy zákaznické (CIS) - mají za úkol zajištění evidence spotřeby a spotřebitelů, fakturace, sledování pohledávek apod.

2 Principy analytických metod

Nyní, když víme co je IS a jaké jsou jeho funkce, se můžeme podívat na principy analýz IS. Projevují se v metodách analýz objektových a (ne)strukturovaných a to v celém návrhu IS a tvoří jádro těchto metod, zatímco nástroje i techniky se mohou dále zdokonalovat. Ustoupení od těchto metod by mohlo vést ke špatnému užití jejich nástrojů a k nežádoucímu výsledku.

Principy se do metod promítají různými způsoby, např. do nástrojů top-down, do konceptuálního a technologického návrhu systému, do definic pojmů, používaných metodami a do technik a návodů k nástrojům.

Základní formy metod analýzy jsou:

- o Abstrakční principy
 - o Top-Down charakter funkční struktury
 - o Generalizace a specializace v datovém modelu
 - o Princip tří architektur
- o Principy modelování

2.1 Princip abstrakce

Hlavním důvodem existence principu abstrakce v metodách analýzy je snaha o rozdělení zkoumané problematiky na myšlenkově zvládnutelné části. IS je velmi rozsáhlý a složitý, stejně jako jeho analýza. Již samotná předmětná oblast zkoumání je ve všech svých detailech, kterými je třeba se zabývat, značně složitá. Data zpracovávaná v IS vyjadřují na jednu stranu celou řadu informací, což je závislé na způsobu jejich zpracování, na druhou stranu jsou ve

vzájemných logických vztazích. To vyvolává potřebu shromažďovat procesy a data IS podle:

- o Obecných logických souvislostí
- o Způsobu jejich zpracování

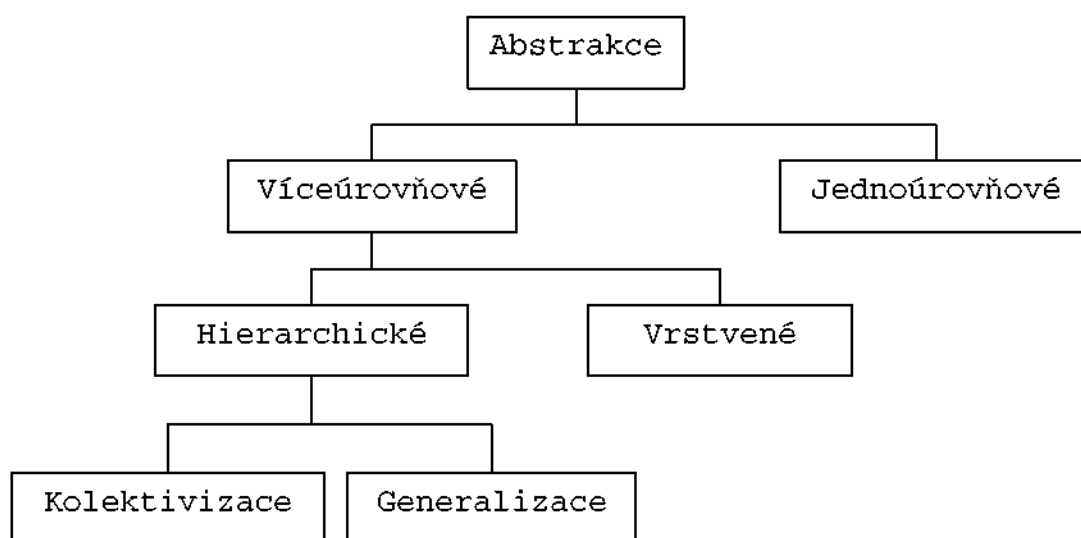
Abstrakce lze rozdělit do dvou základních typů:

- A. Víceúrovňové - kde každý abstraktní prvek může být tvořen konkrétními nebo abstraktními prvky.
- B. Hierarchické (stromové) - kde jsou některé prvky nadřazené jiným.

Podle způsobu tvorby abstraktních pojmů se rozlišují dva typy hierarchických abstrakcí.

- o Kolektivizace, kde abstraktní pojem pouze vyjadřuje účelové sdružení svých prvků a nedefinuje jejich společné vlastnosti. Jako příklad lze uvést „zpracování účetních dokladů“, což zahrnuje mimo jiné „zpracování faktur, příjmových dokladů atd.“, aniž by přitom byly definovány nějaké společné vlastnosti. V metodách strukturované analýzy se jedná o *Top-Down* strukturu.
- o Generalizace, kde abstraktní pojem vyjadřuje sdružení prvků na základě jejich společných vlastností, které jsou všemi prvky povinně děděny. Jako příklad mohu uvést pojem „zaměstnanec“, který zahrnuje velké množství podpojmů jako „dělník“, „manažer“ nebo „úředník“. U těchto konkrétnějších pojmů nás zajímají jejich specifické vlastnosti (řídící nebo řemeslné schopnosti), ale i takové jako je věk, délka pracovního poměru či mzda. V metodách strukturované analýzy se jedná o *generalizaci entit v datovém modelu*.

- C. *Vrstvené* - definují takovou hierarchickou strukturu prvků, kde každý prvek je detailním rozpracováním svého nadřazeného prvku z určitého hlediska s tím, že detaily hledisek, zohledněných u svých nadřazených prvků, přejímá. Každý nižší prvek tak tvoří další vrstvu detailního rozpracování ze svého hlediska, přidanou k vrstvám předchozím. Taková struktura hierarchie není stromová, ale lineární a zohledňované hledisko se vždy týká celé předchozí (vyšší) vrstvy, nikoliv pouze její části. Konkrétně se v metodách analýzy a návrhu IS jedná o *princip tří architektur IS*.
- D. *Jednoúrovňové* - kde celá struktura konkrétních prvků je jednorázově plošně rozdělena na jednotlivé abstraktní oblasti substruktury (úhly pohledu). Každá oblast přitom zdůrazňuje pouze některé charakteristiky a k ostatním nepřihlíží. V metodách analýzy a návrhu IS se jedná o *princip různých pohledů* na vytvářený systém: datový versus funkční model. [2], s. 138-139



Obrázek 1 - Klasifikace abstrakcí, zdroj: [2] s. 139

2.1.1 *Top-Down princip*

Top-Down je tradičním principem, používaným ke zjednodušení pohledu na navrhovaný systém. Tento princip byl použit v metodách strukturovaného programování, navrhování programových systémů a také v metodách analýzy systému, založených na funkčním přístupu. Jde o rozdělení pohledů na zkoumaný systém podle úrovně podrobnosti pohledu. Na nejvyšší úrovni je pohled nejméně podrobný, zato však úplný. Na nižších úrovních je pohled lokálnější a také detailnější.

Smyslem principu top-down jako postupu analýzy návrhu systému je umožnit zkoumání a návrh systému po částech - jednotlivých abstraktních úrovních tak, aby bylo možné zabývat se v jednom okamžiku pouze návrhem jedné úrovně jedné větve stromu.

2.1.2 *Generalizace - specializace v datovém modelu*

Ve strukturovaných metodách se používají dva základní typy hierarchické abstrakce:

- o Abstrakce **část - celek** (kolektivizace, agregace), která se běžně používá například ve funkčním modelu systému, kde se dělí systém na subsystémy, části subsystému atd.
- o Abstrakce **specifický typ - obecný nadtyp** (generalizace), která je naopak typickou hierarchickou abstrakcí v datovém modelu, kde umožňuje jednotlivé entity sdružovat podle své příbuznosti do vyšších celků - nadtypů. [2], s. 143

Zatímco pro agregaci je typická principiální neomezenost dělení a vyšší celek je zcela definován jako souhrn svých částí, v případě generalizace není nadřazený celek definován

jako souhrn podřazených částí, ale jako nositel jejich společných vlastností.

2.1.3 Princip tří architektur (P3A)

Tento princip definuje způsob použití abstrakce pro vývoj informačního systému po jednotlivých vrstvách (vrstvená abstrakce). Jednotlivé vrstvy se zaměřují na tři hlavní aspekty vyvíjeného systému: obsah, technologii a implementační specifika. Tyto aspekty tvoří přirozenou posloupnost: ze specifikace obsahu systému vyplývají možnosti technologického řešení a konkrétní použitá technologie určuje implementační možnosti.

Návrh systému podle P3A probíhá ve třech po sobě jdoucích architekturách:

- a) *Konceptuální* - zde je vytvořen zcela obecný, čistě obsahový model systému, nezatížený ani technologickou koncepcí ani implementačními specifiky jeho řešení. Tato architektura určuje, co je obsahem systému.
- b) *Technologické* - zde je vytvořen model systému, zohledňující technologickou koncepci řešení, tj. ve strukturovaném pojetí koncepci organizace dat (souborová, stromově, síťově či relačně databázová atd.) a technologickou koncepci jejich zpracování. Tato architektura určuje, jak je obsah systému v dané technologii realizován.
- c) *Implementační* - zde je vytvořen model systému, zohledňující implementační specifika vývojového prostředí (programovacího jazyka, databázového systému). Implementační návrh určuje, čím je technologické řešení realizováno.

Tento koncept slouží pro odstranění nepatřičných hledisek při tvorbě systému a současně se objevuje i ve třech základních, obecně používaných, etapách tvorby systému:

- o Analýza
- o Konstrukce
- o Implementace

Informační systém, navržený v těchto třech architekturách, má následující vlastnosti:

- o Specifikace obsahu systému je nezávislá jak na použitém implementačním prostředí, tak dokonce i na technologické struktuře systému.
- o Stejně technologické řešení lze použít u libovolných implementací v tomto technologickém prostředí.
- o Jakékoliv změny technologického prostředí se netýkají konceptuálního obsahu systému.
- o Potřeba rozdělit specifikaci systému na etapy.
- o Stejný konceptuální návrh lze realizovat v libovolném technologickém prostředí (v databázovém systému nebo souborech, v architektuře klient-server nebo host-terminál).

2.2 Princip modelování

Model znamená:

- o Formální vyjádření zkoumaného jevu, sloužící jako vyjádření skutečnosti.
- o Zjednodušuje zobrazení určitého jevu (systému) pomocí vhodných zobrazovacích prostředků, znázorňujících pouze ty rysy, jež jsou podstatné z hlediska námi sledovaného cíle.

- o Reprodukce charakteristik určitého objektu na objektu jiném, zvláště vytvořeném pro jejich studium.

Smyslem modelování je:

- o Použití abstrakce, která umožňuje odhlížet od nepodstatných rysů reality a tím zjednodušit úlohu analytika systému.
- o Formalizací pojmů vytvořit prostředek dorozumění mezi odborníky rozdílných profesí analytikem systému a odborníkem v dané oblasti.
- o Možnost provádět relativně lacino a bez následků změny v modelu, které provádět přímo v reálném světě by bylo příliš nákladné nebo dokonce neuskutečnitelné.

Při modelování reality se mnohé metodiky uchylují k využití různých výrazových prostředků. Tyto modely se nazývají *sémantické modely*. Základním prostředkem modelování sémantiky (významu) je výše zmíněná abstrakce. K těmto abstrakcím je třeba poznamenat:

Koncept třídy - pro instanci třídy jsou podstatné dvě věci. Za prvé entita musí náležet nějaké třídě, aby mohla existovat a za druhé to, že každá entita má svojí identitu, bez ohledu na hodnoty svých atributů. V relačním modelu by entity, které mají stejné hodnoty atributů primárního klíče, splynuly v jednu.

Asociace - vztah mezi dvěma entitami je jedním ze základních vyjadřovacích prostředků všech modelů. Tyto vztahy se většinou vyjadřují pomocí odkazu, které mají nejčastěji podobu atributu z oboru hodnot identifikátorů entit (primární klíče). Problém definice vztahů není řešen ani objektově

orientovanými jazyky, ani v nich neexistuje žádný koncept explicitně vyjadřující vztah.

Seskupení - Třída je víc, než je množina instancí této třídy. Seskupení (grouping) je právě jen množina entit, které spojuje nějaká charakteristika. Je rozdíl mezi seskupením a vícehodnotovým atributem. Seskupení je určitá entita o níž lze hovořit, např. personál určité budovy.

Generalizace a specializace - je to velmi široký pojem, pod nějž lze ukrýt nejrozličnější vztahy tříd. Třída se vyděluje (specializace) z jiné třídy na základě hodnoty nějakého jejího atributu. Klasická specializace, kdy podtřída je speciálním případem rodičovské třídy, je bohatší o nějaké vlastnosti. Například *zaměstnanec* je *osoba*, u níž se navíc sleduje kde pracuje, jaký má plat. To, do které třídy bude instance patřit, je specifikováno uživatelem při jejím vytváření. Uživatel se musí rozhodnout, zda vytváří pouze osobu nebo zaměstnance. Třída je definována z jiných tříd na základě množinových operátorů. Příkladem budiž třída *historická budova*, která sjednocuje třídy *muzeum* a *palác*. Systém nemůže obsahovat žádnou entitu třídy *historická budova*, vždy půjde buď o *muzeum* nebo *palác*. [2], s. 158

Pochopení a správná aplikace principů metod analýzy a návrhu IS je tím nejdůležitějším, co aktivní zvládnutí těchto metod vyžaduje. Vývoj přináší další techniky, další rozšiřování základního aparátu nástrojů a nové pohledy na různé aspekty návrhu informačního systému.

2.3 Metody strukturované analýzy a návrhu

Pod tímto názvem se skrývá výsledek dlouhodobého vývoje, který vyústil v celou řadu rozličných metod, nástrojů,

technik, úvah a metodik vývoje IS. Patří sem diagram datových toků (data flow diagram), ERA diagram, strukturní diagramy a metody jako Information Engineering, SADT (System Analysis & Design Technique), nebo SSA (Structured System Analysis) apod. Dále sem patří i techniky, jako například normalizace datových struktur či technika analýzy událostí.

2.3.1 Význam strukturalizace

Samotné slovo „strukturované“ dobře charakterizuje základní společnou vlastnost těchto metod: jsou založeny na strukturalizaci jak samotného předmětu svého zkoumání, tak i na způsobu zkoumání. Strukturalizace je jim základní pracovní metodou - prostředkem boje se složitostí, která je pro vývoj IS zcela typická. Jednotlivé metody se vyvíjely odděleně s tím, že každá si pro sebe definovala určitý pohled na věc a postupem vývoje se zdokonalila, a to v mezích svého způsobu vnímání reality.

2.3.2 Obsah strukturovaných metod - modely

Ve strukturovaných metodách jsou rozlišovány dvě základní fáze vývoje IS:

- o *Analýza systému* - výsledek této fáze odpovídá konceptuálnímu modelu.
- o *Konstrukce (design) systému* - výsledkem této fáze je technologický model.

Jako třetí fáze bývá označována implementace, ta však není předmětem zájmu těchto metod.

1. Na konceptuální úrovni (obsahové) je model reality tvořen třemi modely:

- o *Datovým modelem*, který je statickým popisem reality - jakými prvky a jejich vzájemnými vazbami je realita tvořena.
- o *Funkčním modelem*, popisujícím procesy a vztahy mezi nimi.
- o *Modelem řízení*, popisujícím časové následnosti jednotlivých procesů.

Funkční model a model řízení dohromady jsou popisem „dynamiky“ v realitě - jakým způsobem se realita mění (resp. jakými transformacemi procházejí informace o realitě).

2. Na technologické úrovni je model systému tvořen:
 - o Modelem programové struktury systému (procesní část)
 - o Modelem logických datových struktur (datová část)

2.3.3 Integrace modelů

Data dictionary představuje centrální popis datových prvků, které jsou společnou záležitostí všech modelů a je tedy základním nástrojem integrace modelů. Jeho princip vychází z předpokladu, že to, co je pro všechny jednotlivé modely společné, jsou data. Proto je popis datových struktur těmito metodami brán jako základní, místo integrace všech modelů dohromady. [2], s. 162

2.3.4 Nástroje strukturovaných metod - ERD

ERD (Entity Relationship Diagram = diagram vztahů mezi entitami) a datové modelování - při tvorbě datových modelů a při respektování principu tří architektur se obecně dají rozlišovat následující tři úrovně popisu datových struktur:

- o *Technologický (logický) model* - popis způsobu realizace systému v termínech jisté třídy technologického prostředí (lineární, relační atd.). Pro relační databázový model jsou na této úrovni do relačních tabulek doplňovány cizí klíče.
- o *Implementační (fyzický) model* - popis vlastní realizace systému v konkrétním implementačním prostředí.
- o *Konceptuální model* - popis obsahu systému na úrovni, která je nezávislá na vlastním implementačním a technologickém prostředí.

ERD je diagram entit a vztahů a je to grafický nástroj, který patří mezi ERA modely. Používá se k vyjádření datových objektů (entit), jejich vztahů (relationship) a vlastností (attribute). Přesto, že tento model vznikl před 30 lety, používá se téměř nezměněný dodnes.

Každá množina entit musí mít uveden identifikátor, který zajišťuje jednoznačnou identifikaci entit v této množině. Tomuto odpovídá primární klíč. Nejčastěji se rozlišují tyto typy entit:

- o *Obecná*
- o *Silná/kmenová/základní/regulární*
- o *Slabá/popisná* (existenčně závislá na jiných entitách)
- o *Vazební/asociativní* (realizuje vazbu mezi entitami)
- o *Generalizace/nadtyp* (vytvoření entity vyšší úrovně)
- o *Specializace/podtyp* (vytvoření entity z entit vyšší úrovně)

Hlavní charakteristiky vztahů:

- o *Kardinalita* - určuje počet výskytů entity ve vztahu (1:1, 1:N, M:N).
- o *Volitelnost/parcialita* - vztah se nemusí týkat všech výskytů entity.
- o *Výlučnost/exkluzivita* - pro jeden výskyt entity může být realizován právě jeden ze vztahů vzájemně výlučných.

Atribut je vlastnost entity nebo vztahu a je to tedy datový prvek, který blíže charakterizuje entitu nebo vztah.

Hlavní dva typy atributů jsou:

- o *Totální atribut* - pro každý výskyt entity má takový atribut hodnotu
- o *Parciální atribut* - pro každý výskyt entity nemusí mít atribut hodnotu

Postup tvorby ERD zahrnuje několik důležitých kroků:

- o *Výběr nejdůležitějších objektů* - entit a nalezení vztahů mezi nimi. Zajímá nás, jaký je hlavní účel objektu, k čemu slouží, z čeho se skládá a jaký je jeho identifikátor.
- o *Nakreslení entit a jejich vztahů do ERD* - záznam kardinality a parciálních vztahů.
- o *Přidání atributů entitám* - označení parciálních atributů a ověření identifikátorů entit. Při přidávání atributů entitám v hrubém datovém modelu se mohou vyskytnout nové entity, a to tehdy, pokud:
a) jde určitý datový prvek přiřadit jako atribut pouze některým výskytům entity (pak jde nejspíše o podtyp entity vyšší úrovně); b) je nalezen prvek, který může být přiřazen všem výskytům dvou různých entit; c) se zjistí, že prvek popisuje vztah a

nemůže být přímo přiřazen a proto vznikne entita nová, která bude mít vztahy na entity původní.

- o *Kompletace hierarchie* - pro každý typ generalizace/specializace se hledá atribut specializace a zkoumá se, jestli je specializace úplná.
- o *Odstranění tranzitivních vztahů* - vztahy, které lze odvodit z jiných vztahů.
- o *Zaznamenání omezujících podmínek* - pro entity. Je nutné znát limity atributů.
- o *Odstranění nadbytečných entit* - z datového modelu. Může dojít k dále popsaným případům: a) entita má pouze jeden atribut a to je identifikátor - musíme sledovat, zda tento neslouží ke vztahu k jiné entitě; b) entita má pouze jeden výskyt.
- o *Ověření úplnosti datového modelu* - provádí se porovnáním obsahu modelu s požadavky uživatele a s funkčním či procesním modelem.

2.3.5 Nástroje strukturovaných metod - DFD

DFD (Data Flow Diagram = diagram datových toků) - slouží jako grafický prostředek návrhu a zobrazení funkčního modelu systému a je to základní nástroj konceptuálního funkčního modelu. Funkční model popisuje z jakých procesů a jejich návazností se realita skládá a také jaké procesy budou tvořit IS, pokud má být věrným modelem zkoumané reality. Vyvinul se z SADT (Structured Analysis and Design Technique).

V DFD se používají tyto základní prvky:

- o Proces
- o Datový tok (Data Flow)
- o Datový sklad (Data Store)

- o Terminátor (externí entita)

Proces je značen kolečkem či elipsou a znázorňuje reálné dění a transformaci dat, která vede k výstupu.

- o Datový proces (funkce) - vyjadřuje fyzickou transformaci dat, neboli změnu stavu určité části dat, tj. změnu hodnot údajů, vznik nových údajů. Jeho hlavním úkolem je zpracovávat data.
- o Řídící proces - vyjadřuje vzájemné návaznosti (řízení) procesů v reálném čase. Jeho úkolem není zpracovávat data.

Každý proces musí mít v DFD název a jednoznačné číslo. Číslo se odvozuje od nadřazených procesů, podobně jako je číslování nadpisů v této práci. Číslování v rámci jedné úrovně slouží pouze k identifikaci a nevyjadřuje pořadí provádění, narozdíl od této práce.

Datový tok reprezentuje v diagramu přesun dat mezi dvěma částmi systému i mezi systémem a okolím. Je znázorněn šipkou, která určuje směr toku. Datový tok musí mít známý obsah a musí být pojmenovaný, s výjimkou toho do data store. Tento obsah je systémem ukládán. I když byl DFD původně navržen i k popisu toku materiálu, zboží, dokumentů aj., není vhodné k tomu nyní tento diagram používat. Bylo by nutné se vzdát řady pravidel pro konstrukci DFD v IS.

Data store (datový sklad) uchovává data k jejich pozdějšímu použití. Znázorňuje se pomocí dvou rovnoběžek, mezi nimiž je název skladu. Data store může být použit jako soubor, pole nebo i jako kniha. Používá se tehdy, dochází-li k časové prodlevě při přenosu dat (asynchronní). Asynchronnost procesů může vyplývat vždy z jejich podstaty a

ne z formy implementace. Data store, stejně jako datový tok, je abstrakcí jakékoliv formy uložení dat a neříká nic o konkrétní formě tohoto uložení. Pro každý datový sklad musí existovat alespoň jeden datový tok dovnitř a ven.

Terminátor představuje objekty, které nepatří přímo do systému, ale do jeho okolí. Vyjadřuje externí zdroj (počátek nebo konec datového toku) nebo místo určení dat. Jde o místo v okolí systému, s nímž systém komunikuje.

Při tvorbě DFD se musí dodržovat určitá pravidla, zajišťující správnost a přehlednost diagramu.

- o *Číslování procesů* - provádí se shora dolů po úrovních a v rámci jedné úrovně. Identifikuje se tím úroveň rozkladu, do níž funkce patří, a jednak i proces, jehož je daná funkce rozkladem.
- o *Názvy procesů* - jméno by mělo být výstižné a mělo by vyjadřovat funkci procesu (co se děje).
- o *Složitost DFD* - diagram nesmí být příliš komplexní. Musí být pochopitelný pro uživatele, analytika i konstruktéra systému. Neměl by ani obsahovat příliš velké množství procesů a toků. Optimální množství funkcí je mezi třemi a devíti. Méně než tři znamená, že je možná chybný rozklad z vyšší úrovně a více než devět znamená často složitý diagram a mělo by se uvažovat o jeho rozdělení na více částí.
- o *Přehlednost DFD* - diagram musí být technicky správný (přesný, konzistentní), přijatelný a srozumitelný pro uživatele. Jedině tak nezatěžuje mysl čtoucího náhodnými vedlejšími významy (např. větší značky u některých procesů mohou vyvolávat dojem větší významnosti).

- o Konzistence DFD - znamená logickou soudržnost a bezrozpornost diagramu. Ta nemusí být samozřejmá, protože díky hierarchickému rozkladu je jedna skutečnost vyjadřována často ve více diagramech.
 - o Při kontrole konzistence hierarchie je nutné prověřit, zda horní úroveň má pouze podúrovně, středová má nad i podúrovně a spodní má pouze nadúrovně.
- o Pravidla konzistence datových toků určují, že:
 - a) Nesmí existovat transformace, která bez vstupů produkuje data,
 - b) Nesmí existovat proces, který pouze spotřebovává data,
 - c) Datové sklady smějí být pouze propojeny pomocí funkce,
 - d) Datový tok do/z terminátoru musí vždy vést přes proces.
- o Pravidla konzistence funkcí určují, že:
 - a) Neznázorňují se cykly mezi funkcemi,
 - b) Neznázorňují se žádné start/stop procedury,
 - c) Řídící procesy nesmějí zpracovávat data,
 - d) Nelze mít dvě funkce se stejným názvem.

2.3.6 Nástroje strukturovaných metod - Data dictionary

Slovník dat je místem centrálního popisu prvků v systému. Datové prvky jsou společné všem jednotlivým specifickým modelům. V datovém modelu jsou to struktury entit, datové toky a datové sklady ve funkčním modelu.

Obsahově tatáž data se vyskytují v různých strukturách v různých diagramech. To logicky vede k potřebě centrálního popisu, ze kterého bude zřejmé, v jakých všech formách se příslušný datový element v modelu vyskytuje. Tradičně má datový slovník textovou podobu, vycházející z Backus-Naurovy formy. Jedná se o nástroj strukturovaného popisu, se všemi příslušnými vlastnostmi:

- o Jednotlivé abstraktní prvky jsou definovány postupně odshora dolů.
- o Jednotlivé definované struktury jsou definovány jako jeden z možných tří typů: prvky v sekvenci jsou spojeny znaménkem „plus“, varianty selekce jsou odděleny znaménkem „svislá čárka“ a iterace je vyjádřena složenou závorkou.

Aby byl datový prvek definován úplně, měla by definice zahrnovat:

- o Význam datového prvku v rámci uživatelské aplikace
- o Složení datového prvku
- o Hodnoty, kterých může elementární datový prvek nabývat, případně i se specifikací formátu

2.4 Objektově orientované metody analýzy a návrhu

V jedné oblasti programování, v oblasti simulace reálných systémů, vznikly první vlastnosti objektů a princip řízení událostmi. Velmi rychle se rozšířil do oblasti programování vůbec a získal tam velký úspěch. Důvody jsou přibližně tyto:

- o Objektový přístup umožňuje daleko lepší využití kódu než knihovna procedur. Knihovny tříd podstatně zvyšují znovupoužitelnost již jednou naprogramovaných sekvencí příkazů.

- o Použitím principu řízení událostmi a objektovým přístupem se elegantně zvládly prostředky zajišťující „okenní“ rozhraní systémů.

K tomu přistupují dva trochu skrytější faktory.

- o Prvním z nich je to, že řešené úlohy změnilý svůj charakter. Z původně většinou dávkových úloh se staly úlohy převážně interakční. Funkce (procedura) má vždy jeden vstup a k němu příslušný jeden výstup. Znamená to, že funkce reaguje pouze na jeden podnět a to vždy na stejný podnět stejně. S objekty je to jinak. Objekt může reagovat na více různých podnětů, na každý jinak. Navíc může mít různé reakce i na tentýž vstup - objekt si může pamatovat svou minulost.
- o Druhý skrytý faktor je metodický. Metodici programování vymysleli mnoho různých doporučení a zásad, které charakterizují správné způsoby programování a jejichž dodržování snižuje pravděpodobnost chyb v programu. Jsou to například abstraktní datové typy, skrývání informací, minimalizace spřaženosti a podobně. Všechna tato doporučení byla převážně verbální a měla malou účinnost. Většina těchto zásad je nyní skryta v definici tříd a v objektovém přístupu a jsou součástí objektových programovacích jazyků. Porušení mnohých těchto zásad je rozpoznáno jako syntaktická chyba a programátor ji chtě nechtě musí opravit. [2], s. 221

Velký úspěch objektů v programování vedl k jejich použití i v návrhu IS. Prvé objektové metody lze označit za pilotní, byly to jen pokusy použít objektový přístup

v návrhu. V podstatě se vycházelo ze strukturovaných postupů, kde byly datové a funkční modely nahrazeny jedním modelem - objektovým. Objektové metody této doby měly zřejmou nevýhodu - závisely na tom, jak člověk mentálně zvládne objektový model. Později se v odborné literatuře a na konferencích objevily metody, které se snaží tento nedostatek odstranit. Jejich množství je velké a celkově metodicky řeší problém, který stál před jejich autory.

Jednotlivé metody se značně liší, pro všechny je společné, že používají objekty, výrazné odlišnosti jsou v následujících oblastech:

- o *Formální zápis* - již v označení jednotlivých tříd jsou značné rozdíly.
- o *Okruh úloh* - Někdy je snaha o odvození objektového modelu z reality, jindy z funkčních požadavků.
- o *Důraz na jednotlivé etapy návrhu* - po různých odlišných metodách byl v roce 1995 vytvořen jazyk UML (Unified Method Language), který slouží jako specifikace pro objektové modelování a jeho účelem bylo sjednotit nástroje CASE. V tomto se také angažuje skupina OMG (Object Management Group), usilující o unifikovaný rozvoj objektového vývoje. Tato skupina vyvinula MOF (Meta Object Facility), což je metamodelování, neboli modelování modelů. Tento metamodel má také sloužit pro přenositelnost a kompatibilitu nástrojů pro vývoj IS.

MOF definuje čtyři úrovně metamodelů:

Úroveň	Popis	Příklad
Meta-metamodel, úroveň MOF	Infrastruktura pro architektury metamodelů. Definice	Meta - Class, Attribute, Operation

	jazyka pro tvorbu metamodelů.	
Metamodel, úroveň UML	Instance meta- metamodelu. Definuje jazyk pro specifikaci modelů.	Class, Attribute, Operation
Model uživatelských objektů	Instance metamodelu. Definuje jazyk pro popis informační domény.	StockShare, askPrice
Uživatelské objekty	Instance modelu. Definuje specifickou informační doménu.	<sell_limit_order>

Tabulka 1 - Metamodely podle MOF, zdroj: [2], s. 223

MOF se zabývá návrhem modelu na úrovni meta-metamodelu, tedy nějakého modelu, s jehož pomocí lze formálně vyjadřovat jiné modely. Původně je MOF spjat s UML v tom smyslu, že MOF má být primárně modelem UML. MOF je však navržen tak, aby s jeho pomocí bylo možno (meta)modelovat i jiné modely, např. DFD, STD atd. MOF je zároveň navržen jako otevřený model, takže je možné jeho rozšíření.

2.4.1 Principy objektového přístupu

Objekty vznikly bez souvislosti s metodikou návrhu a slouží jako výborný prostředek pro usnadnění programování, našly však v metodikách návrhu systému nezastupitelné místo.

Potíže se zvládnutím velkých programů vedly ke tvorbě různých zásad, které by měly zajistit vyšší produktivitu programátorské práce. Nejdůležitější zásady jsou tyto:

- o Programování ve velkém a v malém - Oddělení programových modulů od definice rozhraní a předávání si řízení mezi těmito moduly.

- o *Abstraktní data* - Programátora většinou nezajímá reprezentace dat, ale pouze funkce nad těmito daty. Např. u telefonního seznamu je podstatná změna počtu položek, ale jak jsou reprezentovány položky, není pro návrh programu podstatné.
- o *Utajování informací* - Hierarchie je v objektech samozřejmostí, vyšší vrstva je zobecnění abstraktních dat a má vědět co nejméně o vrstvách nižších.
- o *Platformy* - Každý program je upravován (opravován v nových verzích) pro různá prostředí. Nejlepším řešením je ponechat rozhodování, která povedou k vytvoření nové verze, co nejnižší (co nejblíže k implementaci).

2.4.2 Třída a objekt

U zásad správného programování bylo jejich dodržování závislé na vzdělání, zkušenostech a důslednosti programátora. Objekty obsahují všechny tyto zásady, ale jejich dodržování je nutné, protože jakýkoliv prohřešek je brán jako chyba, nezávisle na vůli a zkušenostech programátora. Základní vlastností objektu je, že obsahuje jak data, tak i procedury. Data jsou nejčastěji nazývána *atributy* nebo *vlastnosti* a procedury *operace* nebo *metody*. Příklad vztahu třída-objekt si můžeme vysvětlit na kočce. „Kočka“ je obecný název a je to tedy třída. Má vlastní funkce, jako je tlukot srdce nebo zažívání a také atributy, například srst. „Micka“ je instance (výskyt) třídy kočka, je to konkrétní objekt. U Micky nabývá atribut srst hodnoty „mourovatá“. Objekt je tedy konkrétním vyjádřením nějaké třídy. Objekt také dědí (obsahuje) veškeré metody třídy, od které je odvozen a nemůže mít jiné metody.

Třída tedy pouze obsahuje deklarace atributu, zatímco objekt má u těchto atributů také hodnoty, což ho dělá konkrétním.

TŘÍDA	OBJEKT
Atribut1	Hodnota atributu1
Atribut2	Hodnota atributu2
Atribut3	Hodnota atributu3
Atribut4	Hodnota atributu4
Metoda1	Obrázek 2 - Třída a její objekt, zdroj: [2] s.226
Metoda2	
Metoda3	
Metoda4	

Obrázek 2 - Třída a její objekt, zdroj: [2] s.226

Na obrázku 2 je schéma třídy a jejího objektu. U třídy jsou uvedeny atributy, ale pouze jejich výčet, ne jejich hodnoty. Objekt obsahuje hodnoty všech atributů, ty musí se u každého objektu uvést, protože se liší objekt od objektu. Metody se u objektů neuvádějí, protože pro všechny objekty jedné třídy jsou stejné a postačí jejich uvedení u třídy.

2.4.3 Základní vlastnosti objektů a tříd

- o *Abstrakce* - soustředění na podstatné, děditelné vlastnosti, tj. co objekt dělá a ne jak je implementován.
- o *Třída* - je identifikovatelný typ entity nebo konceptu v daném prostředí. Třída je skupina datových proměnných a chování (operace, funkce), což je sdíleno instancemi tohoto typu.
- o *Zapouzdření* - skrývání některých informací.

- o *Synergie* - Objekt je kombinace dat a chování - místo dvou přístupů jeden, spojuje vlastnosti a chování. Říkáme, že objekt má synergii, což znamená, že souhrn objektových témat je mohutnější než jejich jednotlivé užití. Vede k odlišování, co vlastně objekt je.
- o *Sdílení kódů* - objektové techniky podporují několik úrovní sdílení (dědění), a to bez redundance. Ještě důležitější je jasnost rozpoznání, které operace jsou skutečně tytéž. Z toho plynou výhody objektů jako nástrojů pro programování.
- o *Objektová struktura* - důraz na to, co objekt je, spíše než jak jej užít. Užití objektů závisí na detailech a často se mění.
- o *Dědění* - objektových hierarchií nad objekty je více druhů, nejpopulárnější je dědění, které můžeme chápat buď jako *generalizaci* nebo jako *specializaci*.
- o *Zpráva* - je požadavek, který jeden objekt posílá druhému. Zpráva vyžaduje, aby přijímací objekt provedl určitou operaci nebo aby poskytl určitá data.
- o *Link* - je spojení mezi objekty, po kterém může být předávána zpráva.
- o *Asociace* - je abstrakce linků, tj. je to spojení mezi třídami, které může (ale nemusí) být realizováno linkem.
- o *Polymorfismus* - je princip, že různé objekty reagují na tutéž zprávu různě, tj. svým vlastním způsobem. Polymorfismus se obvykle realizuje zastíněním. [2], s.227-228

2.4.3.1 Zapouzdření

Objekt obsahuje data a operace, které nejsou z vnějšku viditelné ani přístupné - jsou **soukromé**. Jiná data či operace jsou **veřejné**, lze je z vnějšku použít (změnit nebo vyvolat). To je podstatný přínos objektů. Objekty jsou abstraktní, datové typy a zahrnují i utajování. Jedno z typických použití je to, že data jsou soukromá - jakékoliv změny dat lze dosáhnout pouze vyvoláním veřejné operace, která vstupující data kontroluje a tak zajišťuje, že objekt vždy obsahuje správná a nerozporná data.

Konstruktor objektu určuje, co je soukromé a co je veřejné. Uživatelé to musí respektovat, nelze to obejít. Konstruktor také vloží atributy a metody, které realizují chování objektu. Zapouzdřenost objektu umožňuje, aby chování objektu nebylo narušeno nějakým zásahem zvenčí.

2.4.3.2 Dědění

Objekt je instance třídy. Třídy mohou tvořit hierarchii, kde třída nižší úrovně (potomek) dědí všechny vlastnosti třídy vyšší úrovně (rodič). Zděděné prvky lze doplnit dalšími (speciálními), také je lze nahradit jinými. Hierarchie tříd pro dědění je projevem dvou různých procesů: generalizace a specializace.

Specializace: Navrhnu obecnou třídu a k ní vytvářím potomky (a potomky potomků), kterým přidávám potřebné vlastnosti, až dostanu vhodné třídy, ze kterých vytvářím požadované objekty.

Generalizace: Navrhnu vhodné objekty (jejich třídy). Poté navrhnu rodičovské třídy, které obsahují společné

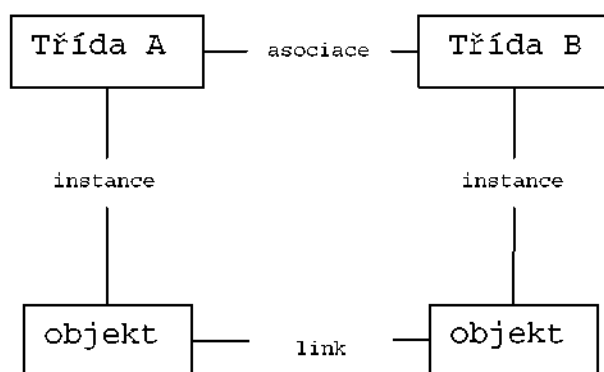
vlastnosti již navržených tříd. Jako příklad si vezmeme auto, letadlo a kolo. Každý z těchto dopravních prostředků má určité vlastnosti, kterými se liší od druhého. Mají ale také vlastnosti stejné, můžeme si tyto třídy zobecnit na "dopravní prostředek" a společné vlastnosti přidělit této třídě. To je generalizace.

2.4.3.3 Link

Komunikace mezi objekty způsobuje, že návrh programu se rozpadá do dvou jasně oddělených rovin:

- o Určení vazeb mezi objekty a z toho vyplývající charakteristiky objektů
- o Konstrukce jednotlivých objektů

Spojení mezi objekty říkáme **link**. Stejnému vztahu mezi třídami říkáme asociace. Link je reálný vztah mezi objekty, je nějakým způsobem programově realizován. Obvyklou realizací je to, že procedura jednoho objektu volá proceduru druhého objektu.



Obrázek 3 Spojení tříd a spojení instancí, zdroj: [2] s. 232

2.4.3.4 Polymorfismus

Různé objekty mohou mít operaci se stejným rozhraním. Mohu vyvolat tuto operaci, aniž přesně vím, jaký objekt volám a objekt udělá to, co umí. Například operace „zobraz“ zobrazí u jednoho objektu obrázek a u druhého text. Mohu volat tuto operaci u různých objektů (například v cyklu) a každý objekt udělá to, za co nese odpovědnost. Toto je polymorfismus.

3 Analýza systému a zjištění požadavků

Tato práce se má zabývat návrhem a analýzou univerzitního informačního systému. Účelem univerzitního systému obecně je umístit všechny prvky týkající se výuky na jedno místo a zlepšit tak přehlednost a uživatelský komfort pro všechny uživatele. To jsou studenti, vyučující, studijní oddělení a další složky školy.

Proto by měl být systém určen pro určité role, neboli skupiny uživatelů. Každá skupina uživatelů, které mohou být rozděleny tak, jak jsem se zmínil o odstavci výše, má jiné požadavky na systém, protože systém jinak využívá. Základem je, aby systém měl pro každého uživatele přiděleno uživatelské jméno a skupinu, do které uživatel patří a podle skupin umožňoval přístup k jednotlivým funkcím. Systém by měl umožňovat přístup nejen pro registrované uživatele, ale i přístup veřejný, který by měl obsahovat informace sloužící hlavně pro mimoškolní osoby. Takový přístup by měl poskytovat údaje o místnostech, rozvrhy učitelů i studentů, kalendář akcí, různé statistiky o předmětech a studentech apod.

Hlavní požadavky skupin na webové rozhraní systému, které budou přístupné převážně v rámci autorizovaného přístupu, následují.

3.1 Student

Student často nevidí do složitosti systému a je koncovým uživatelem, který využívá funkce pracující se zadanými daty, které tam někdo musel vložit. Hlavním zájmem studenta jistě je, aby co nejvíc činností, mimo samotné studium, šlo vykonávat z pohodlí domova, protože velká část studentů

dojíždí a zařizovat zápisy na zkoušky a na předměty bylo pro ně bez univerzitního systému obtížné.

Hlavní funkce systému by měly zahrnovat zápis na předměty, na zápočty i zkoušky a zjišťování jejich výsledků. Bez systému by student pro zjištění známky musel volat vyučujícího nebo si výsledky přečíst na nástěnce katedry. Studenta zajímá také ubytování, informace o předmětech, tzn. jejich sylaby a možnost sestavit si rozvrh. Tato funkce je velmi důležitá, protože se studenti často snaží o co nejmenší počet vyučovacích dní, i za cenu delšího vyučování v rámci jednoho dne.

Pro studenty by bylo jistě výhodné, kdyby existovaly dva STOP termíny na zkoušky. První, zadaný vyučujícím, by zakázal dodatečné odepisování z termínu, zatímco druhý, který by následoval po něm, by již zakazoval dodatečný zápis na termín. Výsledkem by bylo, že například dva dny před zkouškou by byl poslední možný termín na odepsání, zatímco zápis na ni by byl povolen ještě následující den. Proti řešení s jedním STOP termínem to má tu výhodu, že pokud se někdo odepíše na poslední chvíli, je zde ještě trochu času, zajištěného systémem, pro ty, kteří se chtějí zapsat, ale nemohli z důvodu obsazení termínu.

3.2 Vyučující

Vyučující z hlediska systému mají jako hlavní činnost vypisování termínů zkoušek a zapisování známek z těchto zkoušek. K tomu tedy směřují hlavní požadavky na systém. V rámci výpisu termínu je možnost zvolit místnost. Systém by měl při výběru místnosti zkontrolovat, zda je místnost dostatečně velká a neobsazená. Avšak seznam místností by měl nabízet nejdříve ty místnosti, ke kterým má vyučující

přístup, tzn. místnosti katedry a fakulty učitele, aby pak nebyl překvapen, když mu nebude místnost k dispozici. V seznamu by měly být i místnosti učitelů, ale zápis do této místnosti by směli provést pouze učitelé, kteří v této místnosti sedí.

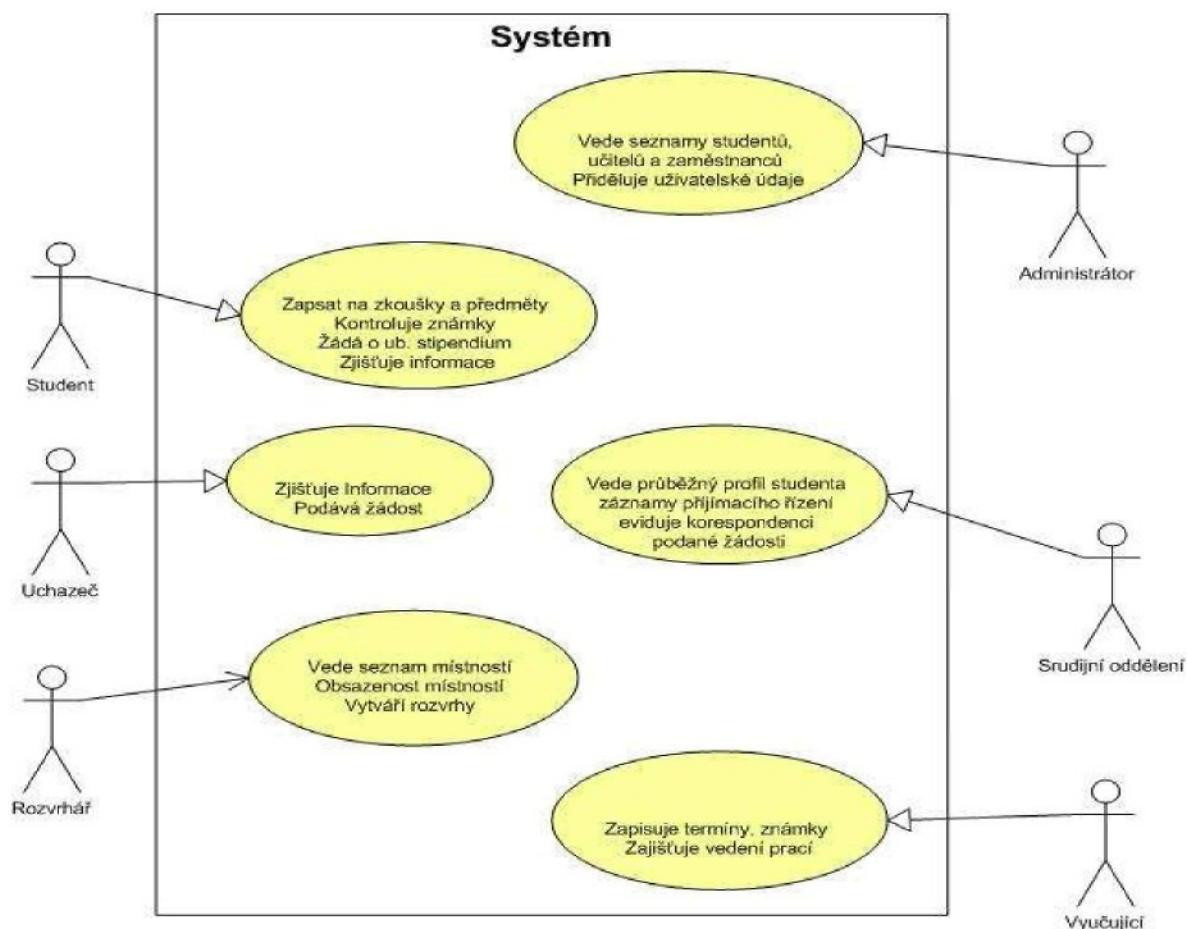
Zrušení termínu by mělo automaticky rozeslat oznámení e-mailem všem zapsaným studentům, pokud mají e-mail v kontaktních údajích a navíc by měl být zvýrazněn, aby bylo zřejmé, že je zrušen. To se týká i zrušení výuky předmětu ať už z důvodu nemoci nebo jiného.

3.3 Studijní oddělení

Studijní oddělení pracuje nejvíce s údaji studentů a proto potřebuje přístup do všech složek. To znamená mít přístup s nejvyššími právy, možnost kontrolovat a opravovat údaje v záznamech. Studijní oddělení také spravuje údaje o přijímacím řízení nových uchazečů o studium, vede záznamy o stipendiích, o závěrečných pracích a další. Pro pohodlnější práci s těmito údaji je vhodnější clientský software, který umožňuje lepší a rychlejší přístup k údajům. Webové rozhraní může být také méně stabilní a zabezpečené.

3.4 Modely IS

3.4.1 Diagram užití

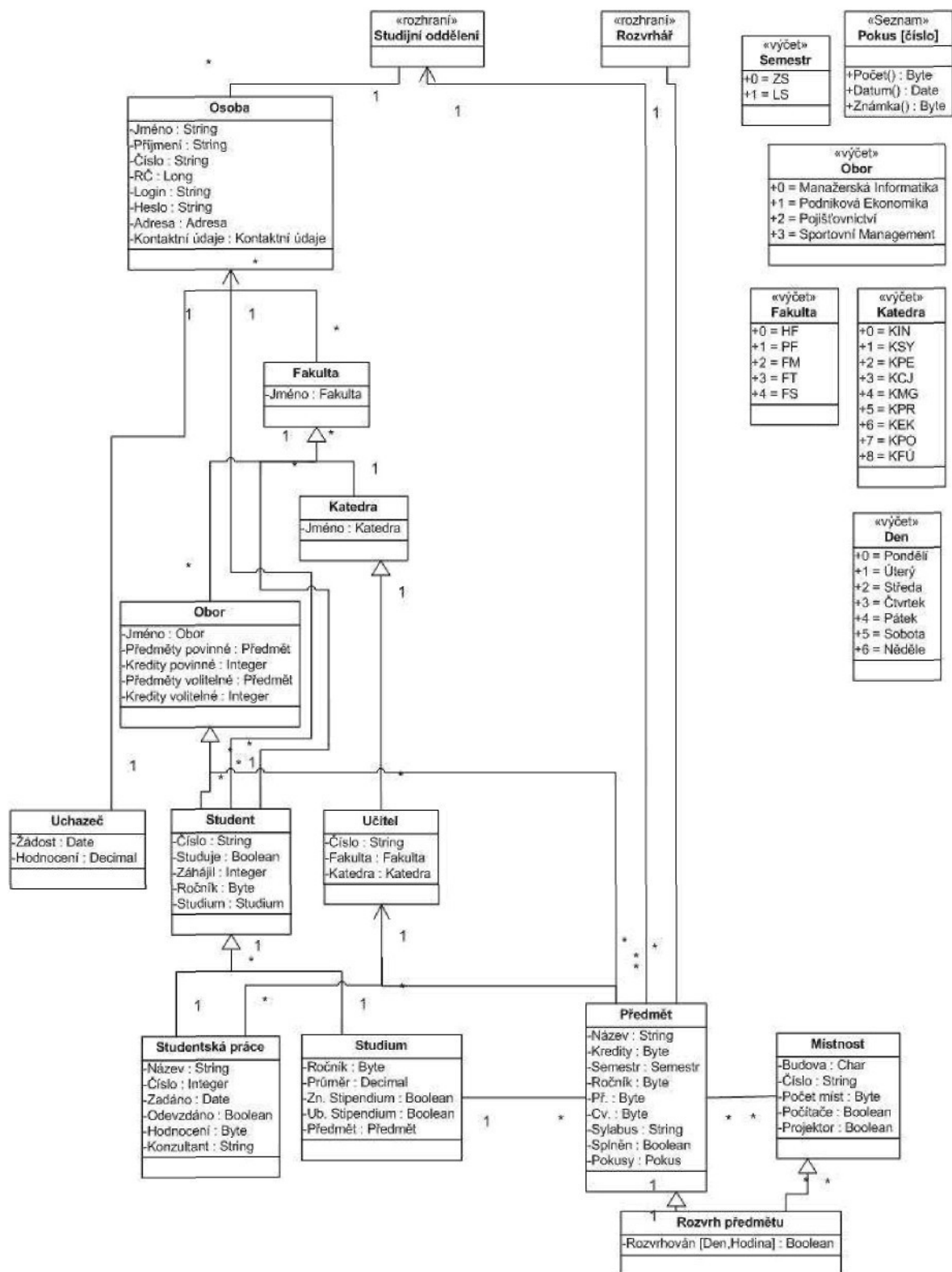


Obrázek 4 - Use case diagram, zdroj: vlastní

V první části analýzy se zaměřím na diagram užití systému a poté na diagram tříd.

Use case diagram, neboli diagram užití, představuje zobrazení systému a vnějších uživatelů. Každý uživatel má nároky na systém a provádí v něm činnosti, které jsou v podobě požadavků sepsány o kapitulu výše. Systém má několik odlišných rolí jako administrátora, studenta, uchazeče o studium, studijní oddělení, rozvrháře, vyučujícího. Rolí může být i více, některé se svými oprávněními příliš neliší. Mohou to být katedra, fakulta, prorektor atd.

3.4.2 Struktura systému



Obrázek 5 - Generalizace a specializace, zdroj: vlastní

Pro analýzu struktury systému použijí diagram tříd. Hlavní třídou v systému je osoba, která je nadtrídou pro velkou část systému a nositel společných vlastností všech podtříd a má mnoho atributů. Osoba tvoří fakulty a katedry nebo se může jednat o osobu mimo školu, jako je uchazeč o studium. V rámci fakulty je několik oborů, které určují studenty a předměty, zatímco katedra určuje učitele. Ze studenta vychází studentská práce, kterých může mít student několik a jež je propojena s učitelem jako vedoucím práce, a také studium, které propojené s předměty zajišťuje přístup ke studentovým studijním výsledkům a informacím. Předmět v kombinaci s místnostmi určuje rozvrh předmětu.

Výhodou je logický přístup, kde je možné se přes třídu osoba dostat hlouběji do systému k dalším informacím. Každá třída je úmyslně pojmenovaná tak, aby bylo jasné o jakou část systému v reálném světě jde a obsahuje ty atributy, které jsou potřeba pro práci systému při získávání informací. Například e-mailová adresa je skrytá ve složeném atributu kontaktních údajů u každého studenta a může sloužit pro příjem informací o zrušených termínech zkoušek, propojení fakult s učiteli zase dokáže zajistit možnost filtrování podle fakult atd.

4 Univerzitní systém STAG

V této části práce bych chtěl popsat konkrétní univerzitní systém. Do této analýzy systému jsem zahrnul i návrhy na systém, které pochází z funkcí jiných systémů a porovnal je s požadavky skupin uživatelů. Srovnávané systémy jsou KOS z ČVUT a Hroch z České zemědělské univerzity v Praze. Univerzitní systém Technické Univerzity v Liberci je systém STudijní AGendy, nazývaný STAG. Tento systém je používán od počátku akademického roku 2003/2004. Webové uživatelské rozhraní, o kterém bude následující analýza a návrhy, je rozděleno na autorizovaný a veřejný přístup. Autorizovaný přístup je dostupný pouze registrovaným uživatelům a vyžaduje jméno a heslo, tzn. studentům, učitelům atd., zatímco veřejný přístup není chráněn heslem.

Úvodní stránka systému STAG se nachází na adrese stag.vslib.cz a kromě přístupů do sekcí obsahuje i základní informace pro práci se systémem a odkaz na uživatelskou příručku. Na stránkách některých fakult chybí do tohoto systému odkaz, např. na stránkách hospodářské fakulty (www.hf.vslib.cz), takže uživatel musí do systému STAG přistupovat přes stránky TUL (www.vslib.cz). Na tuto stránku rovněž není na stránkách HF odkaz, ale připravují se nové stránky, na kterých snad už tyto odkazy budou.



Podle organizační vyhlášky prorektora pro studium a vzdělávání z 18. května 2005, ke stažení ve formátu PDF nebo MS Word, je otevřen předběžný zápis na zimní a letní semestr akademického roku 2005/2006

1. června až 8. července 2005 a 29. srpna až 9. září 2005.

Organizační vyhláška pro ubytovací stipendium za měsíce říjen, listopad, prosinec 2005 prorektora pro studium a vzdělávání z 28. listopadu 2005, ke stažení ve formátu MS Word

Náhradní termín pro elektronické podání žádosti o ubytovací stipendium bude zpřístupněn **od 28. listopadu do 4. prosince 2005.**

Autorizovaný přístup :

- Předzápis studentů na předměty
- Přihlašování studentů na zkoušky
- Zadávání údajů diplomové práce studentem
- Ubytovací stipendium
- Zadávání termínů zkoušek
- Zadávání známek
- Aktualizace klienta
- Výsledky přijímacího řízení uchazeče

Studijní agenda používá k autentizaci uživatelů jejich **uživatelské jméno** a **heslo**. Pro studenta je uživatelské jméno jeho **osobní číslo** (např. A00125147), pro vyučujícího jméno, které mu bylo přiděleno (např. jan_zach). Pokud jste si ve studijní agendě heslo nezměnil(a), je nastaveno na **X+vaše_rodné_číslo** (bez lomítek, pomlček, mezer či jiných oddělovacích znamének např. x8203121216). Pokud jste heslo změnil(a) a nepamätujete si je správně, požádejte Vaše studijní oddělení (pro studenty) popř. uživatele s roli "Katedra" (pro vyučující) ať Vám je změní.

V případě, že se v průběhu předzápisu vyskytnou potíže typu "malá kapacita přednášky", "předmět nemá rozvrh", "předmět z mého studijního plánu v počítači není", obraťte se prosím, na příslušnou katedru případně na Vaše studijní oddělení

V případě **technických** potíží se obraťte na adresu uis@vslib.cz. V dotazu uveďte Vaše osobní číslo, jméno, příjmení, fakultu, studijní program, obor / kombinaci, **nikdy neuvádějte Vaše rodné číslo**. Pokud přistupujete ke STAGu z www, nezapomeňte prosím, uvést typ a verzi Vašeho prohlížeče. V případě, že pracujete s dedikovaným klientem STAGu, nezapomeňte prosím, uvést označení formuláře (vpravo nahoře např. ZA0020) a verzi formuláře (získáte ji z menu "Nápověda" -

Veřejný přístup :

- Graficky zobrazený rozvrh studenta

Obrázek 6 - Úvodní strana systému STAG

4.1 Autorizovaný přístup

Autorizovaný přístup je určen hlavně pro studenty a vyučující. Pro studenty je určen zápis na předměty a na zkoušky. Přístup pro studenty vyžaduje uživatelské jméno a heslo. Uživatelské jméno je studentovo identifikační číslo, které je v systému unikátní a je studentovi přiděleno při přijetí na školu. Spolu s ním je mu přiděleno i heslo, které je pro lepší zapamatování tvořeno rodným číslem.

4.1.1 Předzápis studentů na předměty

Úvodní stránka předzápisu poskytuje základní informace a také odpovědi na často kladené otázky. To je velmi vhodné pro začínající uživatele, protože jim to pomůže se orientovat v základním užívání systému a je zde také zmíněno, jaké jsou obecné přihlašovací údaje, které si většina nových uživatelů

nepamatuje. Vstoupit do systému je možné zmáčknutím jediného tlačítka na stránce, takže přehlednost je zaručena.

Velikou nevýhodou toho, jak systém dnes funguje, je omezení počtu uživatelů v jednom okamžiku na padesát. A i když se uživatel přihlásí, může dojít k situaci, kdy se při odesílání dat na server (zapsání předmětu) vrátí zpráva o tom, že byl odhlášen a že maximální počet uživatelů je padesát. Tzn. i když se uživatel přihlásí na poprvé, nemá jistotu, zda bude moci provést zápis, protože se může stát, že bude odhlášen.

The screenshot shows the 'IS/STAG2 - Předzapis' web interface. At the top, there are links for 'Návod' and 'Změna hesla'. Below this is a table titled 'Kdy mám povolen předzapis' with columns for 'Normální' and 'Kroužkový' registration periods, each with 'Od' and 'Do' sub-columns. The 'Normální' period is from 29.08.2005 00:00 to 09.09.2005 23:59. To the left of the table, the student's name 'Jakub LANK - H02340031' and the academic year 'Ak. rok 2005/2006, Studijní program : M6209' are displayed. To the right of the table, there are input fields for 'Katedra' and 'Zkratka', and a 'Vyhledej předmět' button. Below the table, there is a section 'Celkem zapsáno kreditů' with two rows: 'ZS' with a value of 30 and 'LS' with a value of 0, each with a 'rozvrh' link.

Kdy mám povolen předzapis			
Normální		Kroužkový	
Od	Do	Od	Do
29.08.2005 - 00:00	09.09.2005 - 23:59		

Jakub LANK - H02340031
Ak. rok 2005/2006, Studijní program : M6209

Celkem zapsáno kreditů
ZS ▶ rozvrh 30
LS ▶ rozvrh 0

Katedra : % Zkratka : %
Název : %
Vyhledej předmět

Obrázek 7 - Předzapis, základní informace

Horní část stránky poskytuje základní informace o předzapisu a to datum, kdy má student předzapis umožněn. Tato informace je pro studenta často zásadní. Datum startu předzapisu je důležitý, protože některé předměty jsou velmi žádané a jsou tedy často již v první den nebo spíše v prvních hodinách obsazeny. Naopak poslední den předzapisu je některými studenty sledován s nadějí, zda se neuvolnilo místo v žádaném bloku nebo předmětu celkově. Tato část také přehledně informuje o počtu zapsaných kreditů na letní a zimní semestr včetně možnosti jejich zobrazení.

V této části stránky lze také vyhledávat předmět podle názvu, zkratky nebo katedry. Pokud chce uživatel použít zástupný znak pro libovolný symbol, musí použít „%“ (procento) namísto podle mě častěji používaného znaku „*“ (hvězdička). Přesto myslím, že uživatel není zvyklý používat zástupný znak při vyhledávání slov. Pokud chce např. najít předmět začínající slovem „Daně“, musí vyhledávat „Daně %“, což není obvyklé. Toto hledání vypisuje všechny vyučované předměty na škole a je to jediný způsob, jak se zapsat na předměty, které nejsou ve studentově studijním oboru. Vyhledávání je popsáno v nápovědě, na kterou je odkaz zcela nahoře, kde je i legenda pro všechny obrázky a značky na stránkách předzázpisu.

4.1.2 Rozvrh

Akademický rok												
2004/2005 2005/2006 2006/2007												
Zobrazení letního semestru - Jednoduchá verze pro tisk												
TUL	1	2	3	4	5	6	7	8	9	10	11	12
	07:00-07:45	07:50-08:35	08:50-09:35	09:40-10:25	10:40-11:25	11:30-12:15	12:30-13:15	13:20-14:05	14:20-15:05	15:10-15:55	16:10-16:55	17:00-17:45
	6:00	7:00	8:00	9:00	10:00	11:00	12:00	13:00	14:00	15:00	16:00	17:00
Pondělí			08:50-10:25 <u>KIN/PA3</u> <u>H HB11</u> <u>Arilová Arilová</u> <u>rudý týden</u>		10:40-12:15 <u>KIN/BI1</u> <u>H HB11</u> <u>Zádová Zádová</u>							
Úterý		07:00-08:35 <u>KIN/UM1</u> <u>H HB11</u> <u>Njeđilová Njeđilová</u>					12:30-14:05 <u>KIN/MMA</u> <u>H HB11</u> <u>Voň Voň</u> <u>lidový týden</u>		14:20-15:55 <u>KIN/VMG</u> <u>H HB11</u> <u>Arilová Arilová</u>			
Čtvrtek					10:40-12:15 <u>KIN/MMA</u> <u>H HB11</u> <u>Voň Voň</u>							
Pátek			08:50-10:25 <u>KIN/UM1</u> <u>H HB11</u> <u>Njeđilová Njeđilová</u>									
LEGENDA: Přednáška Cvičení Seminář												
Předměty narozvrhované, ale bez uvedeného času nebo místnosti a nenarozvrhované předměty studenta												
Předmět	Den	Čas	Týden	Týden od-do	Místnost	Místo prvního kontaktu		Učitel				
<u>KIN/PA3</u>			11. týden									
<u>KIN/UM1</u>			11. týden			přihláška ke státní zkoušce - únor 2006						

Obrázek 8 - Rozvrh

Při kliknutí na příslušný semestr se zobrazí přehledný rozvrh, kde jsou barevně rozlišeny přednášky, cvičení i semináře a navíc je zde myšleno i na černobílý tisk, protože přednášky mají navíc tučné písmo. U každého bloku je vidět čas začátku, katedra pod kterou předmět spadá, zkratka předmětu, číslo místnosti s písmenem budovy a vyučující předmětu. Pokud je předmět rozvrhován pouze jednou za čtrnáct dní, je zde i informace o tom, v jakém týdnu. Pod rozvrhem jsou vypsány i předměty nebo akce, které rozvrh nemají.

4.1.3 *Provázanost informací s rozvrhem*

Výbornou funkcí je to, že každá informace slouží zároveň jako odkaz na další příslušné informace. Při kliknutí na název místnosti se zobrazí její rozvrh v příslušném semestru, kapacita místnosti a adresa budovy. To je užitečné pro externí uživatele, neboť tyto informace jsou přístupné i ve veřejném režimu.

Po kliknutí na jméno učitele se uživatel může dozvědět rozvrh vyučujícího v každém semestru. Přes tento rozvrh se lze dostat k dalším informacím, jako v původním rozvrhu. Jako poslední lze v rozvrhu zobrazit informace o předmětu, což bude popsáno v další části práce.

4.1.4 Výběr předmětů

Manažerská informatika				Blok : HF-MI - povinné předměty 8.-11.sem. - A (všechny ročníky)										
Název bloku	St.	Kredity			Omezit výběr pouze na ročník									
		Min.	Max.	Zaps.	5									
HF-MI - povinné předměty 1.-7.sem.	A	106	-	0										
HF-MI - povinné předměty 8.-11.sem.	A	85	-	22										
HF-MI - souborní zkoušky	A	49	-	0										
Předměty státní zkoušky	A	0	-	0										
HF-MI - Jazyk - 7. 2.semestr	B	12	-	0										
HF-MI - jazyk 4. ročník	B	4	-	0										
HF-MI - povinné volitelné 3.-7.sem.	B	4	-	0										
HF-MI - volitelný předmět A	B	3	-	0										
HF-MI - volitelný předmět B	B	3	-	0										
HF-MI - volitelný předmět C	B	4	-	4										

Zkratka (► zapsání)	Název (► info o předmětu)	Kr.	Rozsah Pr+Cv+Sem	R. S.	Garant Uk.	ZS Kap.-Obs.	LS Kap.-Obs.
KAI/AFJ	Automaty a formální jazyky	3	2+1+0	4	LS	Zk	-
KAI/OSU	Operační systém UNIX	2	1+1+0	4	LS	Zk	-
KAI/PCT	Počítačové sítě	3	2+1+0	4	LS	Zk	-
KFÚ/FRPPI	Úvod do finančního řízení podniku	5	2+2+0	4	LS HF	Zk	-
KIN/UMT	Úvod do multimediálních technologií	3	2+2+0	4	LS HF	Zk	-
KSY/VSS	Vybrané statistické statistiky	5	2+2+0	4	LS HF	Zk	38-8
KEK/MIE2	Mikroekonomie II	5	2+2+0	5	ZS HF	Zk	-210
KIN/ERO	Elektronická komerce	4	2+1+0	5	ZS HF	Zp	-
KIN/MIS	Manažerské informační systémy	3	2+0+0	5	ZS HF	Zk	-
KIN/TIS	Technické prostředky informačních systémů	3	2+0+0	5	ZS HF	Zk	-
KST/CAD*H	Návrhové prostředky CAD	5	2+2+0	5	ZS	Zp	-
KEK/MK2	Makroekonomie II	5	2+2+0	5	LS HF	Zk	-
KIN/DPI	Diplomová práce I	6	0+3+0	5	LS HF	Zp	-

Obrázek 9 - Předzázpis, výběr předmětů

V levé části zápisu předmětů se nachází jejich tabulka, která má pět sloupců.

- o První sloupec je název bloku, který nese informaci o tom, zda předměty jsou povinné nebo volitelné a v jakém semestru se běžně zapisují.
- o Druhý sloupec rozděluje předměty do bloků na povinné (A), povinně volitelné (B) a volitelné.
- o Další tři sloupce obsahují informace o kreditech, které je nutné v každém bloku získat podle studijního oboru. Minimum je minimální počet kreditů potřebný pro pokračování ve studiu. Sloupec zapsáno informuje o zapsaných kreditech na celý rok. Nevýhodou je, že systém nezobrazuje zapsané kredity z minulých semestrů, takže si student musí pamatovat, zda dosáhl ve všech blocích předepsaného minima.

Tato tabulka bohužel není příliš aktuální, poněvadž pokud dojde ke změně studijního plánu, např. v počtu minimálních kreditů v bloku, tak to v tabulce není zobrazeno a student musí tyto změny studijního plánu sledovat a pamatovat si je. Tato tabulka zobrazuje předměty pouze pro studentův studijní obor.

Po výběru požadovaného bloku se v pravé části zobrazí seznam předmětů, které do něj patří. Pro lepší orientaci je možné zobrazit výpis pouze pro určitý ročník. Sloupce v tomto seznamu jsou:

- o *Zkratka* - jméno katedry a zkratka předmětu. Pod tímto názvem se také zobrazuje v rozvrhu.
- o *Název* - celý název předmětu.
- o *Kr.* - počet kreditů za daný předmět.
- o *Rozsah* - počet vyučovacích hodin v týdnu pro přednášky, cvičení a semináře. Jedna vyučovací hodina je 45 minut. Např. 2+1+0 znamená dvě vyučovací hodiny, které tvoří jeden blok pro přednášky a jeden blok pro cvičení jednou za dva týdny.
- o *R.* - ročník, ve kterém se předmět zapisuje. Platí pro povinné předměty. Pokud je předmět vybrán jako volitelný, není nutné se tohoto údaje držet.
- o *S.* - letní nebo zimní semestr.
- o *Garant* - fakulta garantující předmět.
- o *Uk.* - ukončení předmětu. *Zk.* - předmět končí zkouškou. *Zp.* - předmět končí zápočtem.
- o *Kapacita* - poslední dva sloupce ukazují počet volných a obsazených míst pro každý předmět v rámci semestru pro všechny jeho cvičení a přednášky.

U některých předmětů je poznámka, pro který obor je zápis umožněn, protože je společný pro více oborů, ale závisí to pouze na studentech. Bohužel to systém neumí sám ohlídat, takže studenti se často zapisují i na bloky, které nejsou pro jejich obory, protože mají lepší vyučovací čas.

KIN/RPR		Řízení projektů		3 2+2+0		5 LS HF		Zk	-	
Sem.	Typ	Míst.	Týd.	Kdy	Plán	Obs.	Vyučující	Pozn.	Grupa	
LS	Př	H-H311	Každý	Čt 16:10-17:45	40	13	Antlová Klára, Ing. Ph.D.	-	-	☐
LS	Cv	H-H311	Každý	Út 12:30-14:05		13	Zikmundová Eva, Ing.	-	-	☐
LS	Cv	H-H311	-	neurčeno	40	0	Zikmundová Eva, Ing.	-	-	☐
Proveď zapsání/odzapsání										

Obrázek 10 - Zápis na bloky

Samotný předmět se zapisuje kliknutím na jeho zkratku a poté dojde k výpisu všech dostupných časů pro přednášky i cvičení. U každého bloku je plán, neboli maximální počet míst a množství zapsaných studentů. Pokud je již student zapsán, je u bloku symbol, značící zapsání. Pokud se student chce zapsat, zaškrtně políčka úplně napravo, která ale nejsou nijak popsána, takže začínající uživatelé mohou chvíli tápat, co mají udělat. Poté co student zaškrtně změny, stiskne tlačítko pro potvrzení.

V seznamu předmětů lze kliknout na název předmětu, načež se zobrazí sylabus daného předmětu, obsahující základní informace jako obsah předmětu, předměty na které tento předmět navazuje a které je nutno mít splněné před tím, než se lze na tento předmět zapsat. Bohužel systém STAG tuto posloupnost neumí ohlídat, takže studenti si mohou zapisovat předměty i bez toho, aby splnili předměty, které jsou pro jejich studium podmiňující. Student pak s takto zapsaným předmětem nemá splněno vše potřebné, aniž by o tom věděl.

Obrázek 11 - Rozvrhové akce předmětu, zapsaní studenti

Detaily předmětu obsahují také jeho rozvrh, kde jsou vypsané všechny bloky se všemi informacemi. U každého bloku je také počet zapsaných studentů. Na toto číslo lze kliknout a dojde k výpisu všech zapsaných studentů, takže student ví, kam se zapsali další studenti z kruhu nebo osoby, se kterými by chtěl chodit na hodinu.

4.1.5 Přihlašování studentů na zkoušky

Zapsání / odepsání z termínů zkoušek														
Odhlášení provedete ukončením WWW prohlížeče !!! Změna hesla														
LANK Jakub - H02340031														
Program : M6209 - Systémové inženýrství a informatika - Obor : Manažerská informatika - akad. rok 2005/2006														
Žádné neblokované platné termíny pro tento program / obor nebyly nalezeny...														
Splněné předměty (nebo nesplněné po 3. termínu)														
Semestr	Předmět	Datum	Pokus	Známka										
ZS	KIN/UIIN-2005	01.12.05	1	1										
Pro následující předměty nejsou vypsané termíny :														
ZS KIN/DPII-2005														
ZS KIN/MMA-2005														
ZS KIN/VMG-2005														
ZS KIN/BIL-2005														
ZS KIN/SZZMI-2005														

Obrázek 12 - Zápis na zkoušky

Druhou položkou v autorizovaném přístupu, která je určena pro studenty, je zápis na zkoušky. Zde je seznam všech předmětů, na které se student pro semestr zapsal. Pokud je na

nějaký předmět vypsán termín nebo více termínů, je zobrazen v tabulce. U každého termínu jsou základní informace jako místnost nebo začátek zkoušky. Důležitým sloupcem je stop termín. Ten určuje čas a datum, do kterého je nejpozději možné se zapsat nebo odepsat na termín.

Všechny tyto údaje jsou zadávané buď vyučujícím nebo sekretariátem fakulty. Student se může zapsat logicky pouze na jeden termín z každého předmětu. Některé předměty mají uvedený termín jak zápočtu tak i zkoušky a studenti se mohou zapisovat na oba současně, což asi není nejlepší řešení. Studenti, kteří se zapsali na zkoušku i když nesplnili zápočet, zabírají místo na úkor těch studentů, kteří již zápočet mají, ale nemohli se zapsat včas na, pro ně vhodný, termín. Takže některé termíny zkoušek nemusí být plně obsazené, protože studenti bez zápočtu se narychlo odepíší z termínu a studenti, kteří mohou na tento termín, pak nemají dost času se na tento termín přihlásit. To v konečném důsledku vede k přeplnění pozdějších termínů.

Některé termíny jsou určeny pouze pro některý obor, ovšem tato informace je uložena pouze ve sloupci pro poznámky a systém STAG povolí zápis každému, protože sám nehlídá obor zapisujícího se studenta, ačkoliv by mohl.

Všechny splněné předměty jsou zobrazené v tabulce pod zapisovanými předměty, kde je uvedeno datum splnění, známka a počet pokusů ke splnění. Tyto údaje jsou při zápisu na další akademický rok porovnávány se studentovým výkazem o studiu a také informují studenta o úspěšně vykonané zkoušce, aniž by se musel dotazovat přímo u zkoušejícího, což jistě většina studentů vítá.

4.1.6 Zadávání údajů diplomové práce studentem

[Tisk podkladu pro zadání diplomové práce](#)

PODKLAD PRO ZADÁNÍ DIPLOMOVÉ PRÁCE STUDENTA		
Jméno	Adresa	Osobní číslo
LANIK Jakub	XXXXXXXXXXXX	H02340031

Prosím vyplňte nebo opravte údaje uvedené níže.

1. Pracovní název tématu: (musí být vyplněno)

Analýza a návrh univerzitního informačního systému

Obrázek 13 - Formulář pro vyplnění údajů závěrečné práce

Tato sekce studentského přístupu umožňuje měnit údaje k zadání závěrečné (diplomové, bakalářské) práce nebo doplnit údaje k již zadané práci. Vyplnění údajů se provádí v jednoduchém formuláři, který je na obrázku výše. Student zadá název práce, hlavní body a použitou literaturu.

Takto vyplněný a vytištěný formulář se odevzdává k podpisu a stává se z něho oficiální zadání práce. Špatně řešený je přístup k tomuto formuláři, protože tento formulář musí tisknout student a odevzdat ho na katedru, místo toho, aby katedra nebo vedoucí práce měli možnost k němu přistupovat. Do tohoto formuláře nelze ani vyplnit jméno konzultanta.

4.1.7 Zadávání termínů zkoušek

Povoleno zadání nejzazšího času pro odhlášení: 1 hodin před začátkem termínu
Přelom akademického roku zadán na den: 16.09.2005

TERMÍN

Datum	Od	Do	Místnost	Zápis a odzápis zkoušky možný do:
				Datum: Hodina:
09.12.2005	9	15	H H31 Dohledat [?] H H31-Učeb.do 119/88 H H310-Laboratoř/25 H H311-Učeb.do 44/40 Zadanému "h31" odpovídá více možností, vyberte kliknutím jednu nebo změňte zadání a opakujte hledání...	

PŘEDMĚT

Katedra	Předmět	Zkoušející
KIN	PDS Dohledat [?] Položka "PDS" je v pořádku...	Zkoušející
Typ zkoušky: Zkouška		
Limit: 20		
Semestr: LS		
Platnost: Platný		
Poznámka:		

[Nápověda](#) [\[ZPĚT\]](#)

Obrázek 14 - Zadání termínů zkoušek

Po přihlášení do systému vidí učitel všechny aktuální termíny vypsané katedrou, tento filtr je možné změnit. Pro vypsání termínu je nutné vyplnit formulář, do něhož se zadávají základní údaje jako datum, čas, místnost, stop termín a samozřejmě předmět. Místnost může být vyhledána pohodlněji, než je tomu v sekci [hledání volných místností](#).

Poté co jsou všechny hodnoty zadány a potvrzeny, jsou odeslány do systému. Pokud některá z hodnot není správná, například je plná místnost, systém to oznámí, ale při pokusu o opravu je formulář bohužel vymazán a musí se začít s vyplňováním nanovo.

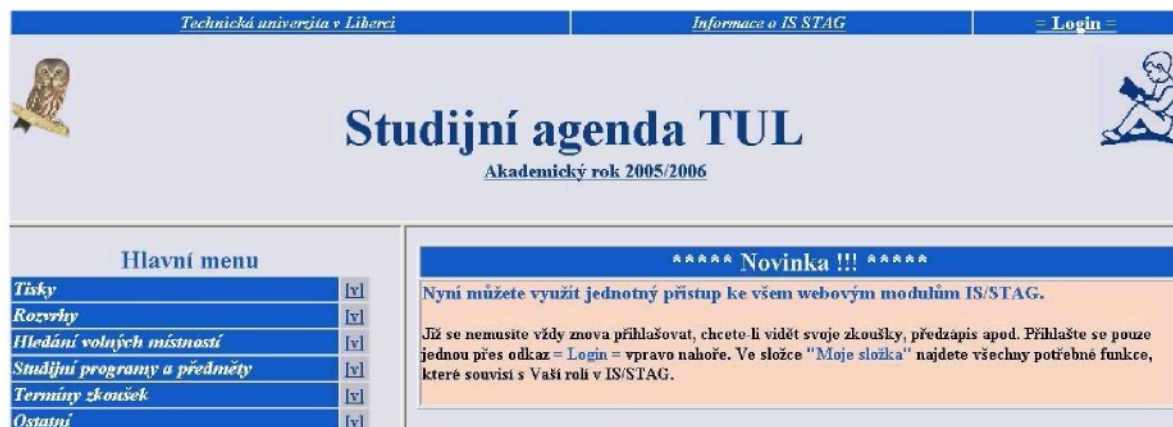
Výhodou je, že lze zadat pro jednoho vyučujícího více termínů, každý pro jiný kruh, takže může zkoušet ve více místnostech současně. Ale na druhou stranu nelze zapsat zkoušku do místnosti, ve které má zkoušející výuku, například v zápočtovém týdnu. Systém také podporuje pouze jeden způsob

4.1.8 Zadávání známek

Obrázek 15 - Známky

57

4.2 Veřejný přístup



Obrázek 16 - Úvodní stránka veřejného přístupu

Veřejný přístup webového rozhraní systému STAG není omezený žádným heslem ani přihlašovacím jménem. Na webové stránce je mnoho odkazů na funkce, ke kterým se lze dostat, ale po kliknutí na jakýkoliv odkaz se uživatel dostane na stejnou stránku. Zřejmě tyto odkazy slouží pouze jako informace, jaké funkce veřejný přístup nabízí.

4.2.1 Tisky

Všechny seznamy získané z této části systému jsou určeny pro tisk, takže neobsahují žádnou nebo jen minimální grafiku.

- o *Počty studentů na předmětech katedry* - vypíše počet studentů na předmětech garantovaných zvolenou katedrou.
- o *Seznamy studentů na předmětu* - vypíše seznam podle zadáných parametrů.
- o *Seznamy studentů na zkoušce* - vypíše aktuální i prošlé termíny vypsané katedrou pro kalendářní rok.
- o *Rozvrh studenta* - tisková verze rozvrhu studenta.
- o *Sylabus předmětu* - zobrazí informace podobné těm, které jsou dostupné v autorizovaném přístupu.

4.2.2 Rozvrhy

Kalendář studijní agendy							
Kalend. týden	<<	Prosinec 2005					>>
	Po	Út	St	Čt	Pá	So	Ne
48.				01 Zimní semestr	02 Zimní semestr	03 Zimní semestr	04 Zimní semestr
49.	05 Zimní semestr	06 Zimní semestr	07 Zimní semestr	08 Zimní semestr	09 Zimní semestr	10 Zimní semestr	11 Zimní semestr
50.	12 Zimní semestr	13 Zimní semestr	14 Zimní semestr	15 Zimní semestr	16 Zimní semestr	17 Zimní semestr	18 Zimní semestr
51.	19 Zimní semestr	20 Zimní semestr	21 Zimní semestr	22 Zimní semestr	23 Zimní prázdniny	24 Zimní prázdniny	25 Zimní prázdniny
52.	26 Zimní prázdniny	27 Zimní prázdniny	28 Zimní prázdniny	29 Zimní prázdniny	30 Zimní prázdniny	31 Zimní prázdniny	

Obrázek 17 - Kalendář

První položka je kalendář, který u každého dne uvádí, zda probíhá výuka nebo zkouškové období. Bohužel zde nejsou informace o státních závěrečných zkouškách, plánovaných rektorských dnech nebo o dalších akcích týkajících se studia.

Další položky:

- o *Osobní rozvrh studenta* - přístup k rozvrhu studenta podle osobního čísla nebo podle příjmení.
- o *Rozvrhové akce kateder* - přehled všech předmětů a jejich bloků, které jsou vyučovány zvolenou katedrou. Lze zde přistupovat k seznamu zapsaných studentů a pro osoby, které mají práva v IS STAG měnit rozvrh, je zde i informace o tom, kdo vložil příslušná data.
- o *Rozvrh místnosti v semestru* - vypíše rozvrh místnosti. Velkou nevýhodou je zápis zkratky místnosti, protože pokud se uživatel pokouší zadat různé a nutno dodat že správné možnosti, tak systém nevypíše žádnou místnost. Jako příklad mohu napsat

„H31“ a nedostanu žádný výsledek. Jedinou možností je použití zástupných znaků „%“ nebo „*“, které lze také použít v ostatních sekcích veřejného přístupu, takže nový hledaný výraz je „%H31%“ a úspěch se dostaví. Zdá se, že systém neprohledává fulltextově, ale hledá pouze přesně zadaný výraz. Příklad, který je napsán ve vyhledávacím formuláři je tedy chybný a uživateli vůbec nepomůže.

- o *Rozvrh kroužku/ učitele* - ačkoliv některé seznamy omezují výpis na 50 položek, aby hledání netrvalo dlouhou dobu a aby server nebyl příliš vytížen a je na uživateli, aby pozměnil a zopakoval svůj dotaz, tak tyto rozvrhy jsou neomezené, takže výpis může trvat opravdu dlouhou dobu. Navíc co se rozvrhu učitele týče, systém prohledává všechny možnosti a ne pouze přesně hledaný výraz, což je dobře a je to přesně opak toho, jak hledá rozvrh místnosti.

4.2.3 Hledání volných místností

Hledání volných místností

Pro semestr podle vyučovacích hodin

Semestr	Týden	Den	Čas.řada	Hod. od	Hod. do
Zimní semestr	Každý	Po	TUL	14	15
Budova	Typ		Kapacita		
Voroněžská	Učeb. do 44		20		
Ignoruj jednorázové akce:				Ne	
Vyhledej					

Pro celý rok

Datum	Hodin od	Hodin do
07.12.2005	16	20
Budova	Typ	Kapacita
Voroněžská	Učeb. do 44	20
Ignoruj jednorázové akce: Ne		
Ignoruj blokace patřící pracovišti: Neignorovat		
Vyhledej		

Obrázek 18 - Volné místnosti

Velice užitečná funkce, která vyhledá místnosti podle poměrně rozsáhlých kritérií. Často se využívá, pokud se hledá volná místnost v určitém bloku, do kterého se přesune výuka z bloku jiného. Je zde možnost hledání místností podle vyučovacích hodin nebo pro jednorázové užití podle data. Co se týče zadávání vyučovacích hodin, zadává se číslo hodiny v rámci dne od jedné do čtrnácti, což zrovna není příliš intuitivní, protože student si často pamatuje čas a ne, jaká je to vyučovací hodina, která se navíc počítá jako 45 minut. Možná by bylo vhodnější, jelikož je počet hodin přesně omezen, použít seznamy s předvyplněnými časy hodin, kde stačí, aby si uživatel vybral.

4.2.4 Studijní programy a předměty

Tato sekce obsahuje informace o studijních programech, jak platných tak i těch zrušených, podle fakulty a roku. Zajímavou položkou je vizualizace oboru, která zobrazí vybraný obor do tabulky, kde každá buňka tabulky představuje jeden předmět.

ETAPA 1 - VERZE 1													
ROK : 1		ROK : 2		ROK : 3		ROK : 4		ROK : 5		Neudáno			
ZS	LS	ZS	LS	ZS	LS	ZS	LS	ZS	LS	??	ZS	LS	??
KCZ/MK1	KCK/M13*H	KCP/P13	KCP/P12	KMG/SMG	KCK/SEK	KCK/M12	KCK/M12	KCP/DDP	KCP/APP	KCK/SZZE	KMG/MASSZ	KCJ/JA15Z	KPD/UG2
KP-/ZSC*H	KCN/ZN2	KCP/DCL	KCP/FUC2	KCP/ZF2	KCP/PMA	KCP/FRL	KCP/R12	KCK/D12	KCK/D12	KCK/SZPE	KCP/ZP5Z	KCK/KK1Z	KMG/MA5Z
KCN/EN1	KMD/NA2*H	KMG/MKG	KCP/ZP1	KCP/SP2	KCP/SP2	KCP/FMA1	KMG/MMG	KCP/D11	KCP/D12	KPD/SZPE	KCV/SP1	KCP/P5Z	KCV/ST5Z
KMD/NA1*H	KCP/NOP	KCP/UMA	KCV/ST1	KCN/SP2	KCK/RP2	KAP/V3M	KCP/V3M	KCN/D11	KCN/D11	KMG/SZPE	KCP/201H	KPE/MA5Z	KCP/23P
KCP/HGR	KCJ/A15*H	KCV/ST1	KMG/MV2	KCP/SP1	KCP/RP2	KCJ/A10*H	KCJ/A11*H	KMG/D12	KMG/D12	KCP/SZPE	KCJ/N15L	KCK/NO15Z	KCK/VVP
KCJ/A13*H	KCJ/P13*H	KCJ/K10*H	KCJ/A10*H	KCK/SP1	KCP/RP2	KCJ/A20*H	KCJ/A21*H	KCP/D11	KCP/D12	KCP/D12	KCJ/N15C	KCJ/P21*H	KPD/VVP
KCJ/P13*H	KCJ/N13*H	KCJ/N13*H	KCJ/P10*H	KCP/RP1	KMG/SP2	KCJ/P10*H	KCJ/P11*H	KCP/D11	KCP/D12	KCP/D12	KCK/RP1	KTV/TW1*H	KCN/VVP
KCJ/N13*H	KCJ/A21*H	KCP/ZF1	KCN/N10*H	KCP/RP1	KMG/SP2	KCJ/P10*H	KCJ/P11*H	KCP/D11	KCP/D12	KCP/D12	KCP/201	KCJ/N15B	KMG/VVP
KCJ/A21*H	KCJ/P21*H	KCJ/A10*H	KCJ/A20*H	KMG/SP1	KCP/RP2	KCJ/N10*H	KCJ/N11*H	KCP/P11	KCP/V15	KMG/SZPE	KMG/SZP	KCJ/N10	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P20*H	KMG/SP1	KCP/RP2	KCJ/N10*H	KCJ/N11*H	KCP/A11*H	KCP/M11	KCP/D12	KCP/201	KCJ/P13*H	KPD/VVP
KCJ/P21*H	KCJ/N13*H	KCJ/N13*H	KCJ/P20*H	KCP/SP1	KCP/SP2	KCJ/R20*H	KCJ/A21*H	KCP/M11	KCP/M11	KCP/M11	KCP/201	KCJ/P13*H	KCP/VVP
KCJ/P21*H	KCJ/R21*H	KCJ/A21*H	KCJ/N10*H	KCP/SP1	KCP/SP2	KCJ/S10*H	KCP/S11*H	KCP/P11	KCP/REJ	KCP/REJ	KCP/13A*H	KCP/13A*H	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/R20*H	KCP/SP1	KCP/SP2	KMG/PNG	KCP/V11	KCJ/P21*H	KCN/UMK	KMG/REK	KCJ/13C*H	KCJ/13C*H	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/S20*H	KCP/SP1	KCP/SP2	KCP/101V	KMG/CR	KCJ/N11*H	KMG/REK	KMG/REK	KMG/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H	KMG/REK	KMG/REK	KCJ/201	KCJ/201	KCP/VVP
KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCJ/P21*H	KCP/SP1	KCP/SP2	KCJ/P13*H	KCP/101V	KCJ/N11*H</					

Obrázek 19 - Vizualizace studijního oboru

Buňky jsou odlišeny barvou. Fialová pro povinné předměty (předměty nahoře), modrá pro povinně volitelné (ve středu tabulky) a zelená pro volitelné (předměty dole). Při kliknutí na předmět se zobrazí jeho základní informace a pokud má předmět podmiňující předměty, dojde k jejich rozsvícení v tabulce. Předměty, které se vylučují s vybraným předmětem se také rozsvítí, ale jinou barvou. Tato funkce značně urychluje orientaci v tabulce, protože počet předmětů je pro určité obory opravdu velký. Předměty jsou rozděleny podle ročníků, ale je zde také sloupec s předměty, které mohou být zapsány ve více ročnících a nemohly být přesně zařazeny.

Navíc lze zadat přihlašovací údaje do systému a zobrazit tak tuto tabulku pro určitého studenta. Bohužel přístup k této funkci je skrytý tak, že po otevření vizualizace musí



uživatel kliknout na symbol , značící návrat na domácí stránku vizualizace. Zde teprve lze zadat studentovo číslo a poté zobrazit tabulku s předměty. V této tabulce přibudou další barvy, reprezentující zapsané a splněné předměty.

Pokud je zobrazena tabulka konkrétního studenta jsou dostupné i údaje o kreditech. V systému jsou uloženy požadované kredity pro povinné, povinně volitelné předměty a to pro každý blok těchto předmětů zvlášť, pro souhrnné zkoušky a cizí jazyky. Systém však pracuje se současnými údaji a neukládá si údaje v okamžiku jejich vzniku. Jako příklad mohu uvést své studium. Zapsal jsem se na předmět Reklama, který byl jako volitelný v bloku B, ale nyní je jako volitelný blok C, takže podle tohoto výpisu mi chybí kredity v bloku B a přebývají mi kredity v bloku C. Navíc některé předměty mám označené za nesplněné, i když jsou splněné, ale to jsou předměty pouze z prvního a druhého ročníku, kdy systém STAG ještě nebyl implementován, takže se domnívám, že ještě nebyly všechny údaje doplněny. Nebýt těchto nedostatků, je tato tabulka ideálním pomocníkem ve výběru předmětů nutných k získání kreditů.

4.2.5 Termíny zkoušek

Seznam termínů podle katedry, předmětu nebo vyučujícího. Výstup je podobný seznamu studentů na zkoušce v sekci tisky, kterou se zabývám v kapitole 4.2.1.

4.2.6 Ostatní

Tato sekce obsahuje rozdělení počtu studentů na předmětu podle fakult, ale také důležité informace o tom, kteří studenti musí platit za studium. Tento seznam je kvůli ochraně informací omezen pouze na čísla studentů. Jako poslední může uživatel přistoupit do databáze závěrečných prací, a to podle různých kritérií. Nevýhodou je, že pokud uživatel zmáčkne *enter* pro potvrzení vyhledávání, zobrazí se nápověda místo výsledků, takže si uživatel může myslet, že zadal některé údaje chybně.

5 Systémy na trhu

Systém STAG Technické univerzity v Liberci je převzatý ze Západočeské univerzity, která tento systém vyvíjí. Podle oficiálních materiálů univerzity je tento systém používán na devíti českých univerzitách a vysokých školách a vyhrál třetí místo na mezinárodní konferenci o univerzitních systémech v kategorii informační systém podporující kreditní způsob studia. Tento systém má tedy většinu funkcí stejnou jako evropské univerzitní systémy.

Univerzity nepoužívají jeden systém, ale často několik systémů, které se liší podle užití. V průzkumu na přelomu let 1998/1999, který měl formu ankety mezi kvestory vysokých škol a tajemníky fakult, byly uvedeny následující kategorie: finance, studium, knihovna, majetek školy, stravování a kolejje. Některé systémy jsou schopny pokrýt více těchto kategorií, ale na některých vysokých školách je pro každou z nich jiný systém. Podle tohoto průzkumu byl na TUL používán pouze systém na finance.

Systémy jsou vyvíjeny přímo univerzitami nebo na objednávku externími společnostmi. Hlavním cílem univerzit je automatizovat a propojit velké množství činností do jednoho celku tak, aby se ke všem údajům dalo přehledně a rychle přistupovat.

5.1 Vlastní vývoj

Tato automatizace umožňuje univerzitám sledovat přístupy k různým funkcím, díky čemuž lze zjistit, jaké části systému jsou často využívány, aby mohly být v systému posíleny nebo optimalizovány, a které méně, což dává univerzitě podnět

k jejich vylepšení, pokud to lze. Toho se dá nejvíce využít, je-li systém vyvíjen univerzitou, ta pak může provádět zásahy velmi rychle, bez nutnosti předávat údaje vnější firmě. Pokud se systém vyvíjí interně, je lepší kontrola nad bezpečností, protože univerzitní systém obsahuje velké množství důvěrných údajů.

V případě interního vývoje má univerzita lepší kontrolu nad náklady. Například Masarykova univerzita v Brně uvádí ve své výroční zprávě o vývoji IS náklady takto:

- víceleté licence db serveru Oracle v celkové ceně 1.150.077,30 Kč vč. DPH.
- trvalá licence db serveru Oracle pro 1 CPU datového serveru ekonomických aj. agend Inetu MU v celkové ceně 255.573 Kč vč. DPH.
- upgrade SW licence aplikačního serveru v celkové ceně 384.407,58 Kč vč. DPH.
- terminálový server pro systém IS BAPS v ceně 95.200 Kč vč. DPH.
- aplikační servery pro systém Inet MU v ceně v ceně 447.516 Kč vč. DPH.
- roční údržba pro aplikační server WebLogic systému Inet MU v ceně 155.767,30 Kč vč. DPH.
- roční podpora db Oracle pro datové servery systémů IS MU a Inet MU v souhrnné ceně 621.207,89 Kč včetně DPH.
- roční podpora db Informix pro dobíhající datový server systému Inet MU v souhrnné ceně 80.265,05 Kč včetně DPH.

Veškeré jednorázové licence byly hrazeny formou dotací od ministerstva školství, mládeže a tělovýchovy. Součet

nákladů na roční údržbu a roční podporu tedy činí 850 tisíc Kč.

Vlastní vývoj ovšem také vyžaduje znalost tvorby informačních systémů a kvalitní zaměstnance, kteří jsou schopni systém implementovat. Tyto vývojové týmy jsou rozsáhlé, což dokazují informace o vývoji na stránkách univerzit, jako je Mendelova univerzita v Brně.

5.2 Externí vývoj

Vývoj externí společností na objednávku a popřípadě i jeho správa, tzv. outsourcing, má výhody v malých nárocích na univerzitu samotnou, protože velkou část úkonů s tím spojených provádí právě tato společnost a univerzita má svého konzultanta, tj. někoho kdo na tvorbu systému dohlíží a také předává požadavky. Způsoby vývoje mohou být rozděleny i tak, že jednu část (studijní) si vyvíjí univerzita sama a vývojářský tým se řídí požadavky budoucích uživatelů a druhou část, například ekonomickou, si nechává zpracovávat dodavatelsky. Takto je to řešeno například na ČVUT.

Nejpoužívanější systémy pro kategorii financí jsou FIS a EkonFis. První jmenovaný je vyvíjen Brněnskou společností BBM, s. r. o.. Tento systém podporuje mimo běžných finančních částí (mzdy, sklady, majetek) i doplňkovou instalaci na podporu studijní agendy a vědy a výzkumu. Druhý systém je vyvíjen společností PragoData a. s. a nabízí také doplnit instalaci o studijní agendu (komponenta STUDIUM). Jeden z hlavních rozdílů mezi těmito systémy je podporované databázové řešení. V prvním případě se používá databáze Oracle, zatímco v druhém případě to byla databáze Progress. Vysoké školy, které zvolily jedno z těchto řešení, používají

právě jeden z výše jmenovaných databázových systémů a také často využívají i doplněk pro studijní agendu. Je nutné dodat, že od doby toho průzkumu, který jsem zmínil výše, došlo u systému EkonFIS ke změně databází z Progress na Oracle.

Cena systému je kromě nákladů na systém ovlivněna také zvoleným databázovým jádrem. Často je využíván systém Oracle, který je velmi drahý. Jeho cena se běžně odvíjí od počtu procesorů na počítači, na kterých systém běží a pohybuje se v řádech milionů korun. Ale je tu i speciální testovací licence OTN, která má sice velmi omezené vlastnosti, ale umožní zájemcům systém otestovat. Avšak, pokud se univerzita rozhodne systém odzkoušet, je po roční lhůtě, po kterou OTN platí, postavena před rozhodnutím, zda má pokračovat se systémy Oracle za plnou cenu nebo přejít na jiný systém. Přejít na jiný systém je obtížný, pokud již univerzita rozběhla systém na Oracle.

Další používané databáze jsou Progress, Informix, Access nebo FoxPro. Alternativou je i SQL s licencí, která je zdarma, ale nemá tolik možností jako komerční řešení. Některé univerzity dnes zkouší i systém SAP.

5.3 Porovnání systémů

Systém studijní agendy na Technické univerzitě v Liberci je implementován pouze dva a půl roku, ale na některých univerzitách existují déle a mohou obsahovat více funkcí. Při porovnávání webových rozhraní systémů jiných univerzit se systémem STAG jsem funkce, které tyto systémy obsahují navíc a které si myslím jsou přínosné, zahrnul do kapitoly 3.

Co se týče ceny za vývoj systému, tak myslím, že jednoznačnou výhodou systému STAG je to, že byl bezplatně získán od Západočeské univerzity, což ho výrazně zvýhodňuje při porovnání s jinými komerčními řešeními. ZČU má navíc výhodnou smlouvu se společností Oracle a může spolu se systémem STAG distribuovat také databázové jádro. Za nákup licence se platí pouze 35% její původní ceny a roční náklady na její prodloužení byly sníženy na 50% z běžné ceny. Navíc prvotní nákup licence hradí univerzitám ministerstvo školství, mládeže a tělovýchovy, takže proto může být pořízení systému STAG zdarma. Podle smlouvy na dodávku studijního informačního systému mezi ZČU v Plzni a TU v Liberci činí cena za podporu a rozvoj IS STAG 200 000,- Kč ročně.

Při porovnání nákladů na systém STAG s náklady na systémy, které univerzity vyvíjí, vycházím ze srovnání s pouze jednou univerzitou, což není dostatečné, ale univerzity často informace o nákladech na systém nespokytují nebo uvádějí souhrnné náklady za celou informatiku, které se nedají použít. Ale podle dostupných informací, je zřejmé, že vlastní vývoj obnáší vyšší roční náklady na podporu databázového systému, v tomto případě jsou více než trojnásobné.

Závěr

Cílem této diplomové práce bylo analyzovat univerzitní IS a navrhnout případné změny v jeho rozhraní. Použitím několika analýz struktury systému, požadavků na něj a jeho rozhraní a funkčnosti, jsem došel k závěru, že STAG Technické univerzity v Liberci je kvalitní systém a funkcemi postačuje pro požadavky studijní agendy v rozsahu, v jakém jsem ho analyzoval. A při přihlédnutí k finanční stránce se domnívám, že byl pro TUL zvolen vhodně. Avšak v kapitolách věnovaným požadavkům na systém jsou zmíněny další možnosti pro vylepšení tohoto systému, které by pomohly zvýšit kvalitu a uživatelský komfort pro budoucí uživatele.

Seznam použité literatury

- [1] Joseph Schmuller. Myslíme v jazyku UML, Grada, 2001.
ISBN 80-247-0029-8
- [2] Řepa, Václav. Analýza a návrh informačních systémů.
Vyd. 1. Praha : Ekopress, 1999. ISBN 80-86119-13-0
- [3] CONNOLLY, T. ,BEGG, C., STRACHAN, A.: Database
Systems: A practical Approach to Design,
Implementation, and Management, Second Edition,
Addison-Wesley, 1999. ISBN 0-201-708574
- [4] POKORNÝ, J.: Konstrukce databázových systémů, ČVUT
Praha Vyd. 2.: Vydavatelství ČVUT, 2004. ISBN 80-01-
02898-4
- [5] Voříšek, Jiří. Strategické řízení informačního systému
a systémová integrace. Vyd. 1. Praha : Management
Press, 1997. ISBN 80-85943-40-9
- [6] TVRDÍKOVÁ, M.: Zavádění a inovace informačních systémů
ve firmách, Grada Publishing, 2000, ISBN 80-7169-703-
6.
- [7] <http://stag.zcu.cz/> - 1.12.2005
- [8] www.muni.cz - 1.12.2005
- [9] <http://www.lmcp.jussieu.fr/eunis/> - 1.12.2005
- [10] <http://is.mendelu.cz/dokumentace/uis.pl> - 1.12.2005
- [11] www.eunis.cz - 1.12.2005
- [12] Prof. Ing. Ivan Vrána, DrSc. Metody zavádění
informačních systémů na univerzitách. 1999. ISBN 80-
214-1344-1.



Formulář na vyplnění podkladu pro zadání DP / BP / DISP studentem

[Tisk podkladu pro zadání diplomové práce](#)

PODKLAD PRO ZADÁNÍ DIPLOMOVÉ PRÁCE STUDENTA

Jméno	Adresa	Osobní číslo
LANK Jakub	XXXXXXXXXX	H02340031

Prosím vyplňte nebo opravte údaje uvedené níže.

1. Pracovní název tématu: (musí být vyplněno)

Analýza a návrh univerzitního informačního systému



Podle organizační vyhlášky prorektora pro studium a vzdělávání z 18. května 2005, ke stažení ve formátu [PDF](#) nebo [MS Word](#), je otevřen předběžný zápis na zimní a letní semestr akademického roku 2005/2006

1. června až 8. července 2005 a 29. srpna až 9. září 2005.

Organizační vyhláška pro ubytovací stipendium na učelce Pílen, listopad, prosinec 2005 prorektora pro studium a vzdělávání z 28. listopadu 2005, ke stažení ve formátu [MS Word](#)

Náhradní termín pro elektronické podání žádosti o ubytovací stipendium bude zpřístupněn od 28. listopadu do 4. prosince 2005.

Autorizovaný přístup :

- [Předzápis studentů na předměty](#)
- [Přihlašování studentů na zkoušky](#)
- [Zadávání údajů diplomové práce studentem](#)
- [Ubytovací stipendium](#)
- [Zadávání termínů zkoušek](#)
- [Zadávání známek](#)
- [Aktualizace klienta](#)
- [Výsledky přijímacího řízení uchazeče](#)

Studijní agenda používá k autentizaci uživatelů jejich **uživatelské jméno** a **heslo**. Pro studenta je uživatelské jméno jeho **osobní číslo** (např. A00125147), pro vyučujícího jméno, které mu bylo přiděleno (např. jan_zach). Pokud jste si ve studijní agendě heslo nezměnili(a), je nastaveno na **X+vaše_rodné_číslo** (bez čarkek, pomlček, mezer či jiných oddělovacích znamének např. x8203121216). Pokud jste heslo změnil(a) a nepamätujete si je správně, požádejte Vaše studijní oddělení (pro studenty) popř. uživatele s rolí "Katedra" (pro vyučující) ať Vám je znění

V případě, že se v průběhu předzápisu vyskytnou potíže typu "**malá kapacita předmětů**", "**předmět nemá rozvrh**", "**předmět z mého studijního plánu v počítači není**", obraťte se prosím, na příslušnou katedru případně na Vaše studijní oddělení.

V případě **technických** potíží se obraťte na adresu sis@vslib.cz. V dotazu uveďte Vaše osobní číslo, jméno, příjmení, fakultu, studijní program, obor / kombinaci, **nikdy neuvádějte Vaše rodné číslo**. Pokud přistupujete ke STAGu z www, nezapomeňte prosím, uvést typ a verzi Vašeho prohlížeče. V případě, že pracujete s **dedikovaným** klientem STAGu, nezapomeňte prosím, uvést označení formuláře (vpravo nahoře např. ZA0000) a verzi formuláře (získáte ji z menu "Nápověda" -

Veřejný přístup :

- [Graficky zobrazený rozvrh studenta](#)

Hledání volných místností

Pro semestr podle vyučovacích hodin

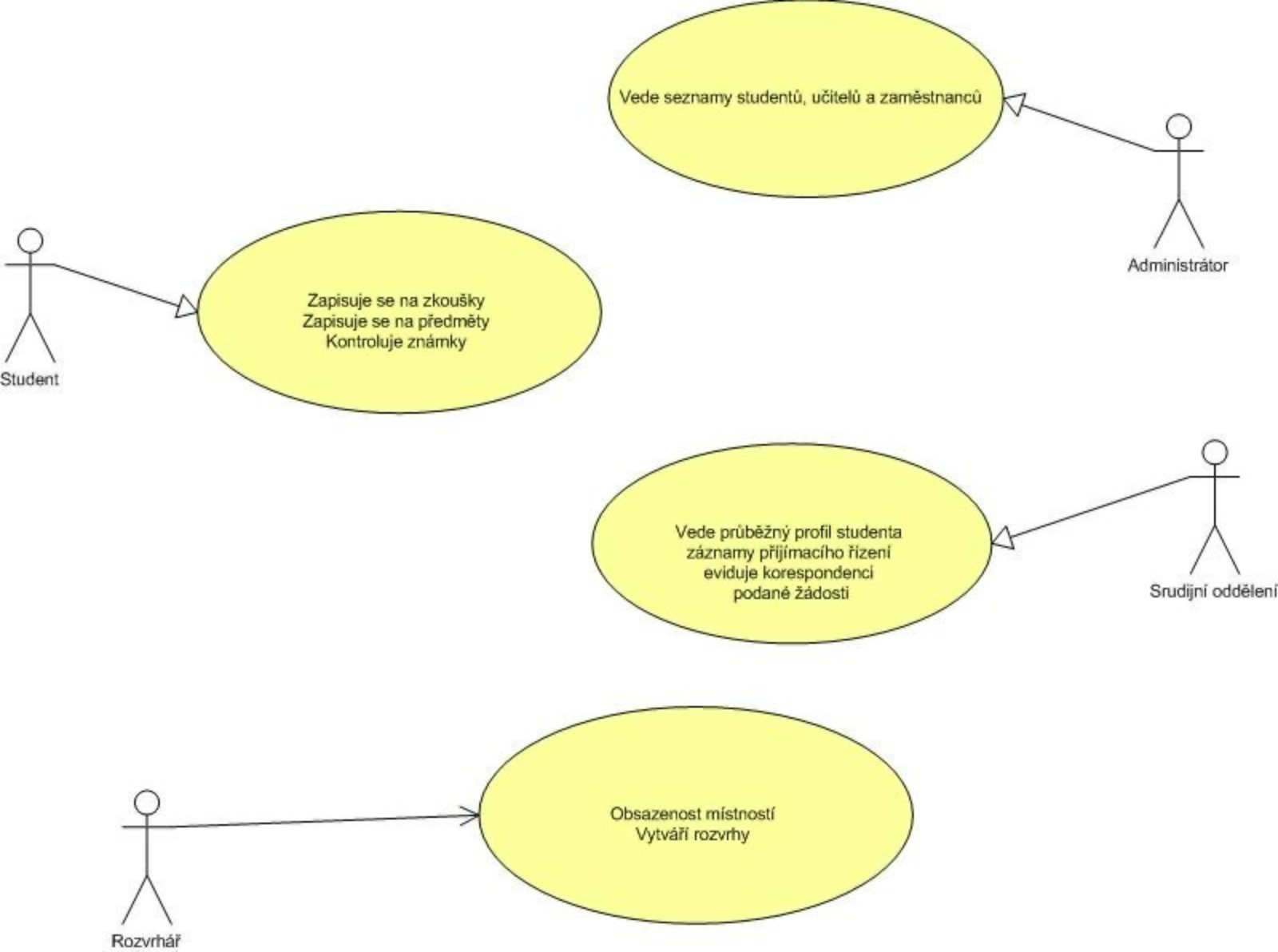
Semestr	Tyden	Den	Čas/řada	Hod. od	Hod. do
Zimní semestr	Každý	Po	TUL	14	16
Budova	Typ			Kapacita	
Varonětská	Učeb.do 44			20	
Ignoruj jednorázové akce:				Ne	

Vyhledej

Pro celý rok

Datum	Hodin od	Hodin do
07.12.2006	16	20
Budova	Typ	
Varonětská	Učeb.do 44	
Ignoruj jednorázové akce: Ne		
Ignoruj blokace patřící pracovníci: Neignorovat		

Vyhledej



IS/STAG2 - Předzapis

[Nápověda](#) [Změna hesla](#)

Kdy mám povolen předzapis			
Normální		Kroužkový	
Od	Do	Od	Do
29.08.2005 - 00:00	09.09.2005 - 23:59		

Celkem zapsané kredity

ZS ▶ rozvrh	30
LS ▶ rozvrh	0

Katedra : % Zkratka : %

Název : %

Jakub LANK - H02340031

Ak. rok 2005/2006, Studijní program : M6209

Omezit výběr pouze na ročník

5

Název bloku	St.	Kredity		
		Min.	Max.	Zaps.
HF-MI - povinné předměty 1.-2.sem.	A	106	-	0
HF-MI - povinné předměty 8.-11.sem.	A	85	-	22
HF-MI - volitelné zkoušky	A	49	-	0
Předměty státní zkoušky	A	0	-	0
HF-MI - jazyk - 1. 2.semestr	B	12	-	0
HF-MI - jazyk 4. ročník	B	4	-	0
HF-MI - povinné volitelné 3.-7.sem.	B	4	-	0
HF-MI - volitelný předmět A	B	8	-	0
HF-MI - volitelný předmět B	B	8	-	0
HF-MI - volitelný předmět C	B	4	-	4

Zkratka
(► zapsáno)

Název
(► info o předmětu)

Kr.

Rozsah
Pr+Cv+Sem

R.

S.

Garant
Uk.

ZS
Kap.-Obs.

LS
Kap.-Obs.

KAVAFJ

Automaty a formální jazyky

3

2+1+0

4 LS | Zk | - | - || KAVOSU | Operační systém UNIX | 2 | 1+1+0 | 4 | LS | Zk | - | - |
KADPCT	Počítačové sítě	3	2+1+0	4	LS	Zk	-	-
KFÚFÉPPI	Úvod do finančního řízení podniku	5	2+2+0	4	LS HF	Zk	-	-
KIN/UMT	Úvod do multimediálních technologií	3	2+2+0	4	LS HF	Zk	-	-
KSYVSS	Vybrané statě ze statistiky	5	2+2+0	4	LS HF	Zk	-	38-8
KEK/ME2	Mikroekonomie II	5	2+2+0	5	ZS HF	Zk	-210	-
KIN/EKQ	Elektronická komerce	4	2+1+0	5	ZS HF	Zp	-	-
KIN/MIS	Manažerské informační systémy	3	2+0+0	5	ZS HF	Zk	-	-
KIN/TIS	Technické prostředky informačních systémů	3	2+0+0	5	ZS HF	Zk	-	-
KSDCAD*H	Návrhové prostředky CAD	5	2+2+0	5	ZS	Zp	-	-
KEK/MK2	Makroekonomie II	5	2+2+0	5	LS HF	Zk	-	-
KIN/DPI	Diplomová práce I	6	0+3+0	5	LS HF	Zp	-	-

Zobrazení letního semestru - Jednoduchá verze pro tisk

TUJ	1	2	3	4	5	6	7	8	9	10	11	12
	07.10-07.11	17.10-17.11	27.10-27.11	06.11-06.11	16.11-16.11	26.11-26.11	06.12-06.12	16.12-16.12	26.12-26.12	05.01-05.01	15.01-15.01	25.01-25.01
	6.00	7.00	8.00	9.00	10.00	11.00	12.00	13.00	14.00	15.00	16.00	17.00
Pondělí			08.10-18.11 LPOVAG M.ML1 Jedinec, Jedinec volný týden		10.11-20.12 KINMTH M.ML1 Jedinec, Jedinec							
Úterý	17.10-17.11 LPOVAG M.ML1 Jedinec, Jedinec						17.10-17.11 KINMTH M.ML1 Jedinec, Jedinec		10.11-20.12 KINMTH M.ML1 Jedinec, Jedinec			
Čtvrtek					10.11-20.12 KINMTH M.ML1 Jedinec, Jedinec							
Pátek			08.10-18.11 KINMTH M.ML1 Jedinec, Jedinec									

LEGENDA: [Vyučování](#) [Cvičení](#) [Semestr](#)

Předmět naryzkovaný, ale bez uvedeného času nebo místnosti a nenařzkovaný předmět studenta.

Předmět	Den	Čas	Učitel	Učitel odle	Adresant	Adres prodeje kromě	Číslo
LPOVAG			Úterý				-
LPOVAG			Úterý			předmět je naryzkovaný - čas 2005	-



Rozvrhové akce KEK/MIE2

Akademický rok:

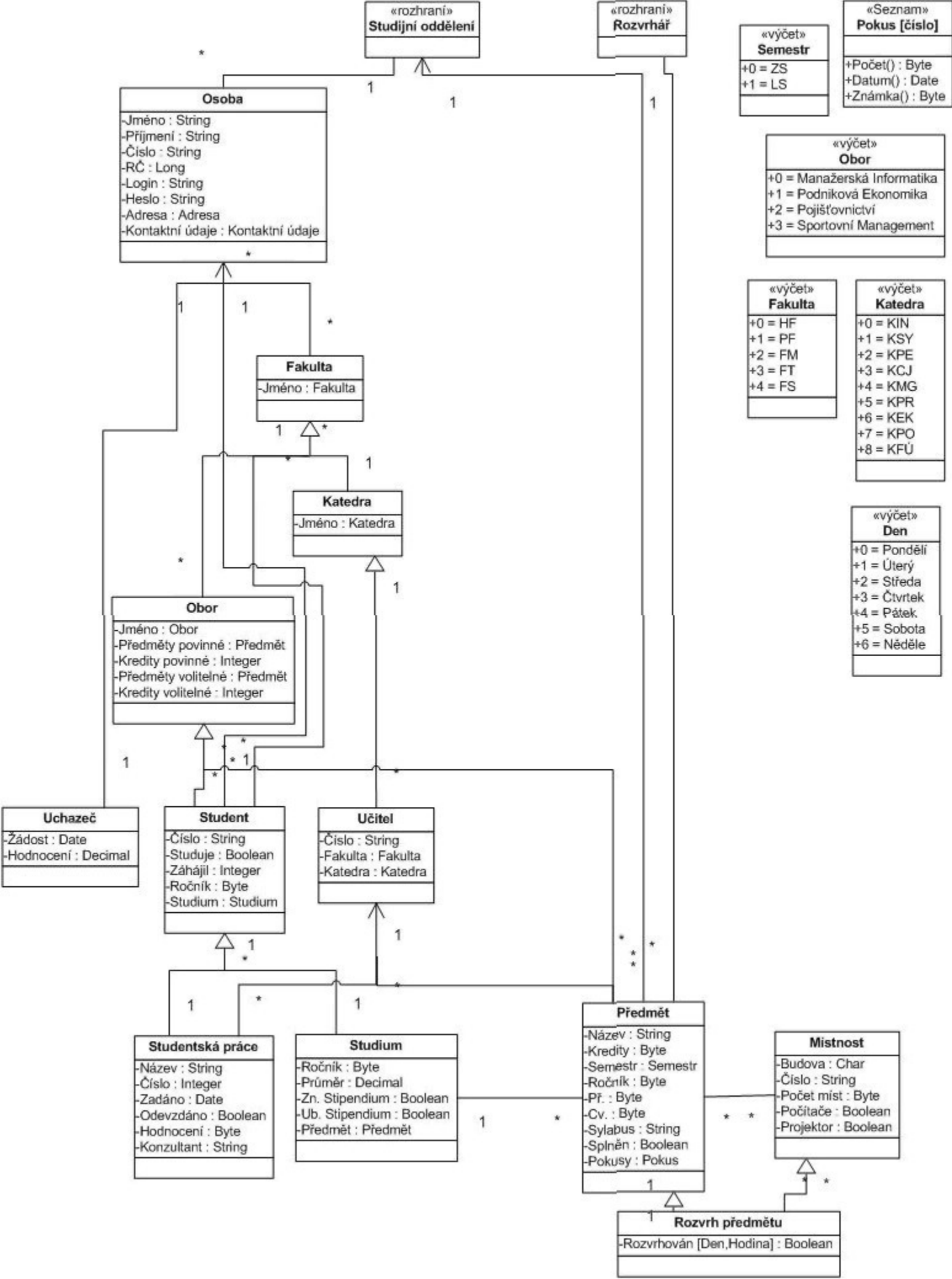
[2004/2005](#) [2005/2006](#) [2006/2007](#)

Kapacita/obsazení přednášky			
	A	B	C
ZS	210	20	2
LS			

Předmět	Míst.	Sem.	Typ	Úd.	Den	Od	Do	Plán	Obs.	Kap.	Platn.	Gr.
KEK/MIE2	P-P13	ZS	Cv	K	Pe	12:30	14:05	30	30	N	A	
KEK/MIE2	E-E9	ZS	HF	K	Pe	14:20	15:55	210	204	N	A	
KEK/MIE2	P-P19	ZS	Cv	K	Pe	16:10	17:45	30	30	N	A	
KEK/MIE2	P-P13	ZS	Cv	K	Pe	16:10	17:45	30	22	N	A	
KEK/MIE2	H-H33	ZS	Cv	K	Út	08:50	10:25	30	30	N	A	
KEK/MIE2	P-P21	ZS	Cv	K	Út	10:40	12:15	30	30	N	A	

Studenti na akci: KEK/MIE2 - PP13 ZS K Po 12:30 14:05 (Cv)

	Příjmení	Jméno	Osobní číslo	Fakulta	Číslo programu	Typ zápisu	Stav
1.	ANDRÉSOVÁ	Gabriela	H05000023	HF	N6208	A	Studuje
2.	BARVÍNKOVÁ	Lenka	H05000025	HF	N6208	A	Studuje
3.	BUBENÍČKOVÁ	Zuzana	H05000027	HF	N6208	A	Studuje
4.	DLOUHÁ	Jana	H05000034	HF	N6208	A	Studuje
5.	DOBKOVÁ	Eva	H02293000	HF	N6208	A	Studuje
6.	DYRCOVÁ	Tereza	H05000006	HF	N6202	A	Studuje



FILTR:

Katedra: KIN

Předmět: %

Vybrat

Zobrazovat jen dnes vložené termíny ☐

Vyučující: %



Počet vybraných termínů: 3. Jste přihlášen(a) jako: uživatel/ role/ katedra/ učitel: VLADIMIRA_ZADOVA/ VY/ KIN/ Zádová Vladimira, Ing.

Povoleno zadání nejzazšího času pro odhlášení: 1 hodin před začátkem termínu

Přelom akademických roků zadán na den: 16.09.2005

Vložit nový termín

Nápověda

Změna hesla

TERMÍN						PŘEDMĚT														
	Datum	Od	Do	Místnost	Vlastník	Katedra	Př.	Sem	Limit	Obsaz.	Zkoušející	Pozn.	Pl.	Typ zk.	Stop					
++	10.12.2005	09:00	15:00	JC-JC_F1	KLARA_ANTLOVA	KIN	IN1	ZS	50	29	Antlová Klára, Ing. Ph.D.	Jičín	A	Zápočet	09.12.05 22:00				+	Kopie
++	14.01.2006	14:20	16:00	H-HI3	KLARA_ANTLOVA	KIN	IN1	ZS	50	38	Antlová Klára, Ing. Ph.D.	Kombinované studium	A	Zápočet	12.01.06 22:00				+	Kopie
++	18.01.2006	09:00	12:00	H-H311	KLARA_ANTLOVA	KIN	RP2	ZS	20	8	Antlová Klára, Ing. Ph.D.	Obhajoba ročníkových projektů	A	Zápočet	17.01.06 22:00				+	Kopie

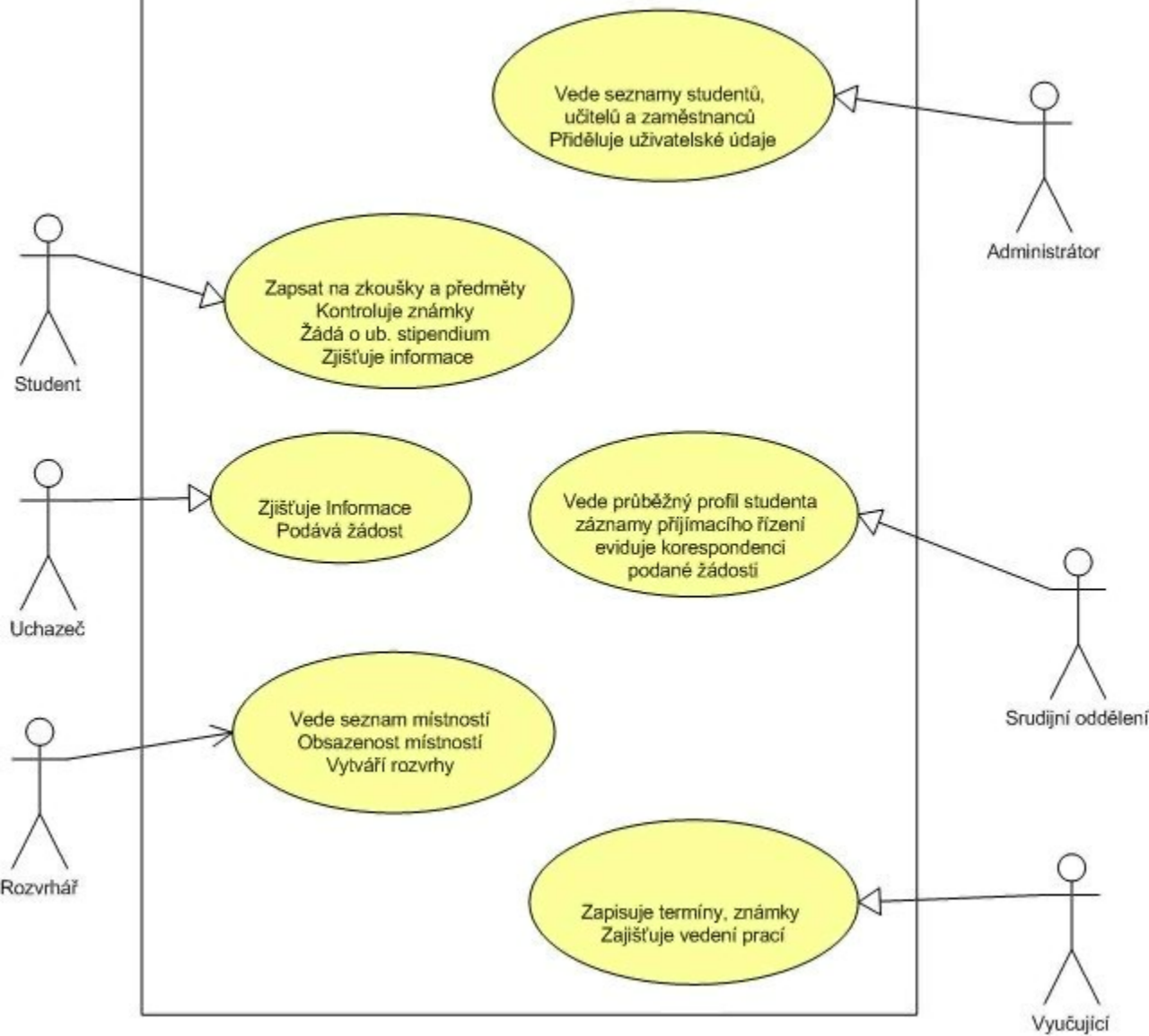
TERMÍN

Datum	Od	-	Do	Místnost	<u>Zápis a odzápis zkoušky možný do:</u>	
					Datum:	Hodina:
09.12.2005	9	-	15	H H31 Dohledat [?] H H31-Učeb.do 119/88 H H310-Laboratoř/25 H H311-Učeb.do 44/40 Zadanému "h31" odpovídá více možností, vyberte kliknutím jednu nebo změňte zadání a opakujte hledání...		

PŘEDMĚT

Katedra	KIN	Předmět	PDS Dohledat [?] Položka "PDS" je v pořádku...	Zkoušející	XXXXXXXXXXXX
Typ zkoušky	Zkouška				
Limit	20				
Semestr	LS				
Platnost	Platný				
Poznámka					

System





Kalendář studijní agendy

Kalend.		Prosince 2005						
tyden		Po	Út	St	Čt	Pá	So	Ne
48.					01 <i>Žitová zarnavař</i>	02 <i>Žitová zarnavař</i>	03 <i>Žitová zarnavař</i>	04 <i>Žitová zarnavař</i>
49.		05 <i>Žitová zarnavař</i>	06 <i>Žitová zarnavař</i>	07 <i>Žitová zarnavař</i>	08 <i>Žitová zarnavař</i>	09 <i>Žitová zarnavař</i>	10 <i>Žitová zarnavař</i>	11 <i>Žitová zarnavař</i>
50.		12 <i>Žitová zarnavař</i>	13 <i>Žitová zarnavař</i>	14 <i>Žitová zarnavař</i>	15 <i>Žitová zarnavař</i>	16 <i>Žitová zarnavař</i>	17 <i>Žitová zarnavař</i>	18 <i>Žitová zarnavař</i>
51.		19 <i>Žitová zarnavař</i>	20 <i>Žitová zarnavař</i>	21 <i>Žitová zarnavař</i>	22 <i>Žitová zarnavař</i>	23 <i>Žitová přednášky</i>	24 <i>Žitová přednášky</i>	25 <i>Žitová přednášky</i>
52.		26 <i>Žitová přednášky</i>	27 <i>Žitová přednášky</i>	28 <i>Žitová přednášky</i>	29 <i>Žitová přednášky</i>	30 <i>Žitová přednášky</i>	31 <i>Žitová přednášky</i>	



Studijní agenda TUL

Akademický rok 2005/2006

Hlavní menu

Tisky	[v]
Rozvrhy	[v]
Hledání volných místností	[v]
Studijní programy a předměty	[v]
Termíny zkoušek	[v]
Ostatní	[v]

***** Novinka !!! *****

Nyní můžete využít jednotný přístup ke všem webovým modulům IS/STAG.

Již se nemusíte vždy znova přihlašovat, chcete-li vidět svoje zkoušky, předpis apod. Přihlaďte se pouze jednou přes odkaz = [Login](#) = vpravo nahoře. Ve složce "Moje složka" najdete všechny potřebné funkce, které souvisí s Vaší rolí v IS/STAG.

[illegible]



Zapsání / odepsání z termínů zkoušek

Odhlášení provedete ukončením **VANM** prohlížeče [Změna hesla](#)

LANK Jakub - H02340031

Program : M6209 - Systémové inženýrství a informatika - Obor : Manažerská informatika - akad. rok 2005/2006

Žádné neblokované platné termíny pro tento program / obor nebyly nalezeny...

Splněné předměty (nebo nesplněné po 3. termínu)

Semestr	Předmět	Datum	Pokus	Zašmka
ZS	KIN/UM-2005	01.12.05	1	1

Pro následující předměty nejsou vypsané termíny :

ZS KIN/DPII-2005

ZS KIN/MMA-2005

ZS KIN/VMG-2005

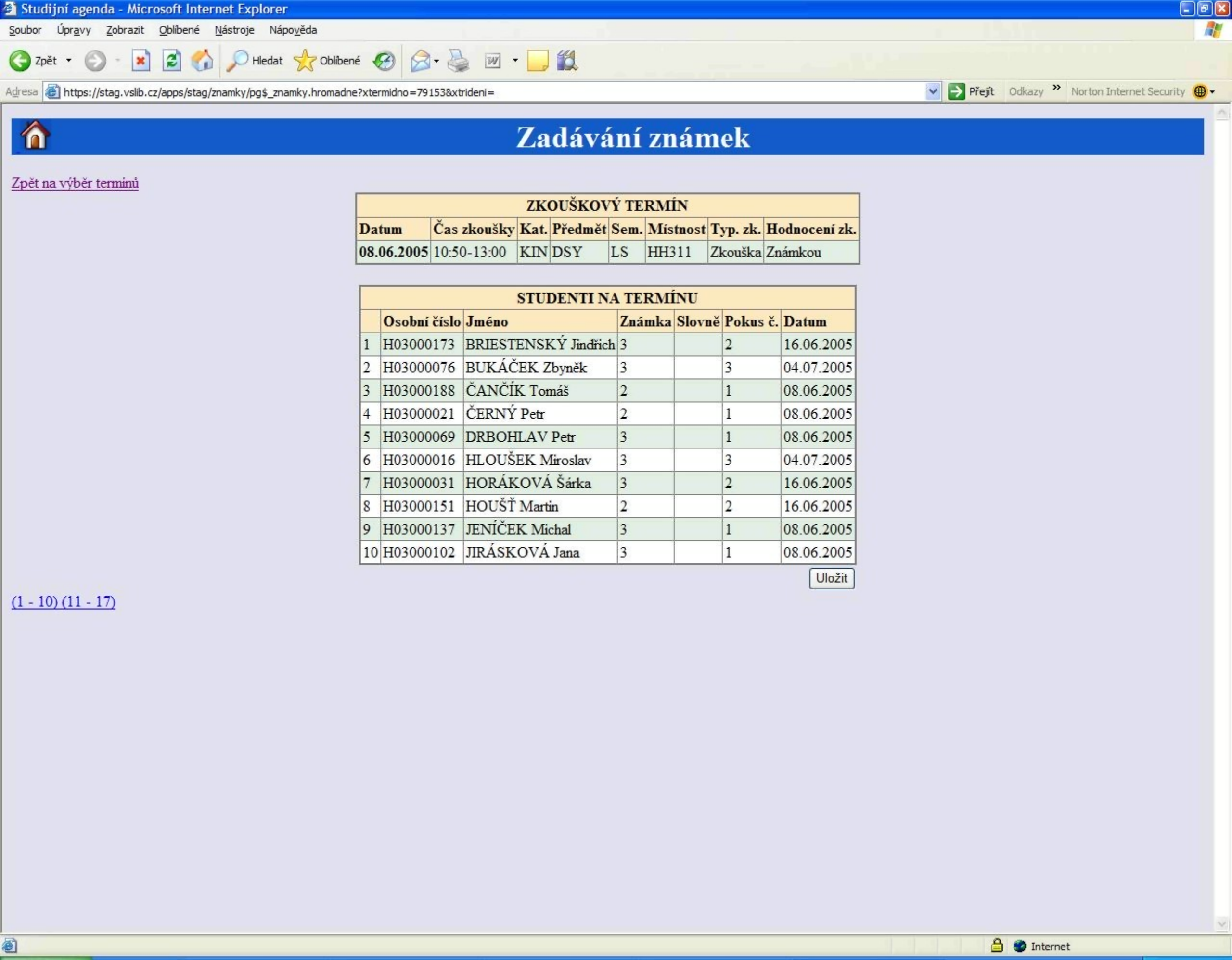
ZS KIN/BIL-2005

ZS KIN/SZOM-2005

Seřadit podle: ☐ data sestupně ☒ předmětu ☐ místnosti Setřídít

ODZKOUŠENÉ TERMÍNY, ZKOUŠEJÍCÍ: ~~XXXXXXXXXXXX~~

Datum	Čas zkoušky	Kat.	Předmět	Sem.	Místnost	Obs.	Typ. zk.	Zapsat známky	Zapsat známky
09.02.2005	09:00-14:00	KIN	BIL	ZS	HH32	9	Zkouška	Hromadně	Jednotlivě
01.02.2005	10:40-15:00	KIN	BIL	ZS	HH32	1	Zkouška	Hromadně	Jednotlivě
25.01.2005	09:00-15:00	KIN	BIL	ZS	HH34	6	Zkouška	Hromadně	Jednotlivě
19.01.2005	09:00-14:00	KIN	BIL	ZS	HH31	13	Zkouška	Hromadně	Jednotlivě
12.09.2005	10:00-13:00	KIN	DBS	ZS	HH311	1	Zkouška	Hromadně	Jednotlivě
08.06.2005	10:50-13:00	KIN	DBS	ZS	HH311	1	Zkouška	Hromadně	Jednotlivě
26.04.2005	19:00-20:30	KIN	DBS	ZS	HH311	4	Zkouška	Hromadně	Jednotlivě
09.02.2005	09:00-14:00	KIN	DBS	ZS	HH32	1	Zkouška	Hromadně	Jednotlivě
01.02.2005	10:40-15:00	KIN	DBS	ZS	HH32	7	Zkouška	Hromadně	Jednotlivě
25.01.2005	09:00-15:00	KIN	DBS	ZS	HH34	5	Zkouška	Hromadně	Jednotlivě
19.01.2005	09:00-14:00	KIN	DBS	ZS	HH31	9	Zkouška	Hromadně	Jednotlivě
07.09.2004	09:00-12:00	KIN	DBS	ZS	HH310	7	Zkouška	Hromadně	Jednotlivě
02.09.2004	09:00-12:00	KIN	DBS	ZS	HH310	1	Zkouška	Hromadně	Jednotlivě



Zadávání známek

[Zpět na výběr terminů](#)

ZKOUŠKOVÝ TERMÍN							
Datum	Čas zkoušky	Kat.	Předmět	Sem.	Místnost	Typ. zk.	Hodnocení zk.
08.06.2005	10:50-13:00	KIN	DSY	LS	HH311	Zkouška	Známkou

STUDENTI NA TERMÍNU						
	Osobní číslo	Jméno	Známka	Slovně	Pokus č.	Datum
1	H03000173	BRIESTENSKÝ Jindřich	3		2	16.06.2005
2	H03000076	BUKÁČEK Zbyněk	3		3	04.07.2005
3	H03000188	ČANČÍK Tomáš	2		1	08.06.2005
4	H03000021	ČERNÝ Petr	2		1	08.06.2005
5	H03000069	DRBOHLAV Petr	3		1	08.06.2005
6	H03000016	HLOUŠEK Miroslav	3		3	04.07.2005
7	H03000031	HORÁKOVÁ Šárka	3		2	16.06.2005
8	H03000151	HOUŠŤ Martin	2		2	16.06.2005
9	H03000137	JENÍČEK Michal	3		1	08.06.2005
10	H03000102	JIRÁSKOVÁ Jana	3		1	08.06.2005

[\(1 - 10\)](#) [\(11 - 17\)](#)