

KATEDRA INFORMATIKY
PŘÍRODOVĚDECKÁ FAKULTA
UNIVERZITA PALACKÉHO

PRŮVODCE BAKALÁŘSKOU PRACÍ

MARTIN DOSTÁL



VÝVOJ TOHOTO UČEBNÍHO TEXTU JE SPOLUFINANCOVÁN
EVROPSKÝM SOCIÁLNÍM FONDEM A STÁTNÍM ROZPOČTEM ČESKÉ REPUBLIKY

Olomouc 2007

Abstrakt

Tento text je stručným a praktickým průvodcem diplomanta při tvorbě bakalářské práce. Čtenář se seznámí se zásadami pro vypracování bakalářské práce, průběhem hodnocení a obhajoby. Text obsahuje také základy práce s typografickým systémem \LaTeX . Stručně je probrána problematika přípravy prezentací.

Cílová skupina

Učební text je určen posluchačům studijního oboru Aplikovaná informatika v kombinované formě vyučovaného na katedře informatiky PřF UP v Olomouci.

Obsah

1	Bakalářská práce	7
1.1	Etapy bakalářské práce	7
1.2	Struktura bakalářské práce	8
2	Obhajoba	10
2.1	Před obhajobou	10
2.2	Průběh obhajoby	10
3	Tvorba odborné části práce	13
3.1	Přehledová část	13
3.2	Použité technologie	13
3.3	Rozbor architektury aplikace, struktury kódu a dynamiky aplikace	13
3.4	Datová základna aplikace	14
3.5	Instalátor aplikace	14
3.6	Bezpečnost	15
3.7	Standardy	15
3.8	Testování aplikace	15
3.9	Pár poznámek k sazbě	15
3.10	Jazyk práce	15
3.11	Práce zaměřená na hardware	16
3.12	Práce zaměřená na počítačovou hru	16
4	Tvorba uživatelské dokumentace	17
4.1	Struktura uživatelské dokumentace	17
4.2	Pracovní scénáře	18
5	Sazba textu práce v typografickém systému \TeX s balíkem maker \LaTeX	19
5.1	Základy	19
5.2	Struktura souboru	20
5.3	Překlad souboru	21
5.4	Základy sazby textu	21
5.4.1	Mezislovní mezery	21
5.4.2	Dělení slov	21
5.4.3	Sazba uvozovek	22
5.4.4	Sazba pomlček, teček a podobných znaků	22
5.4.5	Výpustky v textu	22
5.5	Druhy a velikost písma	23
5.6	Strukturování textu	23
5.7	Křížové odkazy	23
5.8	Poznámky pod čarou	24

5.9	Prostředí v L ^A T _E Xu	24
5.9.1	Výčty	24
5.9.2	Zarovnávání textu	25
5.9.3	Přímý výstup textu	26
5.9.4	Sazba tabulek	26
5.10	Vkládání obrázků	27
5.11	Plovoucí objekty	28
5.11.1	Definice nových příkazů a nových prostředí	29
5.12	Sazba matematických výrazů	29
5.13	Sazba seznamu literatury a citace	30
6	Jak na prezentaci	33
A	Vzor titulních stran	35

Stav textu je první a nedokončená verze učebního textu Průvodce bakalářskou prací.

- neprošlo jazykovou korekturou

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

Úvod

Tento učební text slouží jako praktický průvodce při tvorbě bakalářské práce. Text obsahuje několik na sobě nezávislých témat, která mohou být při tvorbě závěrečné práce užitečná. Jednotlivé kapitoly postihují základy dané problematiky a v žádném případě si nečiní nároky na úplnost.

První kapitola podává základní informace o bakalářské práci, její struktuře, etapách a právním postavení. Druhá kapitola seznamuje čtenáře s průběhem obhajoby bakalářské práce a s tím, co obhajobě samotné předchází. Třetí a čtvrtá kapitola uvádí souhrn zásad pro vypravování odborné části práce, přičemž čtvrtá kapitola je věnována problematice tvorby dokumentace software.

Pátá kapitola seznámí čtenáře se základy sazby dokumentů v typografickém systému L^AT_EX. Šestá, poslední, kapitola je zaměřena na přípravu prezentace během obhajoby.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

1 Bakalářská práce

Bakalářská práce je závěrečnou prací studenta. Osvědčuje jeho odbornou úroveň a schopnost zpracovat zadanou problematiku odborně správným způsobem a její výsledky uceleně formulovat v bakalářské práci.

Téma práce si student vybírá ze seznamu témat, která vypisují jednotliví akademičtí pracovníci. Druhou možností je navrhnout téma vlastní. Při této variantě si student musí najít vhodného vedoucího práce, který návrh posoudí a připraví zadání. Zadání práce musí obsahovat, kromě tématu, jasnou formulaci cílů práce a nároků na ni kladených. Při úvaze o výběru tématu je dobré si položit tyto otázky:

- Čím je téma pro mě zajímavé ?
- Jaký bude význam práce a čeho chci dosáhnout ?
- Jaký bude přínos práce ?
- Jsem schopen téma odborně zvládnout ?
- Je téma pokryto potřebnou literaturou ?

Při výběru tématu je rozumné zvážit své dosavadní zkušenosti, protože by se mohly při volbě vhodného tématu dobře využít. Diplomovou práci je také dobré brát tak trochu jako výzvu — není nutné, abyste problematiku před volbou tématu ovládali, během práce můžete mnoho nových a pro vás přínosných věcí nastudovat.

Přesné podmínky pro zadání, vypracování, odevzdání, obhajobu a zveřejňování bakalářských prací na katedře informatiky upravují předpisy univerzity, fakulty a katedry. Tyto předpisy stanovují nejen povinnosti, ale i práva diplomanta a dalších osob podílejících se na bakalářské práci a souvisejících činnostech. Proto je dobré se s těmito předpisy náležitě seznámit.

Bakalářská práce je autorským dílem a je podobně jako ostatní taková díla chráněna zákonem 121/2000 Sb. „o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů“ (tzv. „autorský zákon“) [5]. V případě bakalářské práce je jedná o takzvané *dílo školní* a jeho režim upravuje §60 zákona.

Z §60 zákona vyplývá právo školy na uzavření licenční smlouvy s autorem díla za účelem užití školního díla a to za obvyklých podmínek. Obvyklými podmínkami se zřejmě rozumí například bezplatné použití díla pro účely související s činností školy. Pokud by škola chtěla využít dílo autora k výdělečným účelům nárok autora na odměnu není nijak dotčen. V případě, že autor odmítne se školou uzavřít licenční smlouvu, může se škola domáhat uzavření licenční smlouvy soudní cestou.

Autor může své dílo užít, nebo také poskytnout jinému licenci pro jeho užití, není-li však v licenční smlouvě se školou sjednáno jinak a není-li to v rozporu s oprávněnými zájmy školy. V případě, že autor školní dílo použije k účelům z nichž mu plyne zisk, pak má škola právo se domáhat přiměřené úhrady nákladů (až do jejich plné výše), které byly na vytvoření díla z prostředků školy vynaloženy. Příkladem takové situace může být třeba software, který vznikl jako součást bakalářské práce a zaměstnavatel autora má zájem o úplatné užití tohoto díla. V tomto případě má škola právo domáhat se přiměřené úhrady svých nákladů.

1.1 Etapy bakalářské práce

Výběr tématu Témata prací jsou zveřejňována podle příslušného předpisu. Student si buď některé z témat vybere, nebo může navrhnout téma vlastní (k tomu

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

má právo podle §62 zákona 111/1998 Sb., tzv. „vysokoškolský zákon“). Témata vypisují jednotliví akademičtí pracovníci, kteří pak vystupují jako vedoucí práce.

Formulace zadání Vedoucí práce připraví studentovi zadání, které stručně formuluje obsah a cíl práce. Zadání práce je důležitým dokumentem při hodnocení práce, jelikož jedním z hlavních kritérií, je zda a do jaké míry diplomant splnil zadání.

Vypracování práce Společně se zadáním práce je studentovi přidělen vedoucí práce, což je akademický pracovník, který studenta vede při tvorbě bakalářské práce a poskytuje konzultace. Vedoucí práce však není jejím spoluautorem, diplomat tedy nemůže očekávat od vedoucího, že za něj vyřeší všechny problémy se kterými si diplomant neví rady.

Konzultace s vedoucím bakalářské práce by měly probíhat průběžně. Vedoucí práce zejména diskutuje s diplomatem o problémech, doporučuje postup a literaturu. Diplomant by se měl na každou konzultaci předem připravit a promyslet si otázky na vedoucího práce. Není korektní očekávat, že váš vedoucí bude provádět jazykovou korekturu práce nebo že bude testerem vaší aplikace. Předkládat text plný gramatických a stylistických chyb dobře vypovídá o úrovni a přístupu diplomanta.

Může se stát, že diplomant se s vedoucím v některých otázkách neshodne. Je vždy na uvážení diplomanta, zda se bude či nebude řídit doporučeními vedoucího.

Odevzdání práce Práce se odevzdává podle pokynů uveřejněných v příslušném předpisu, zejména postup odevzdání, termíny a podobně. Těchto pokynů je třeba striktně dbát, neboť jejich nedodržení může být důvodem pro nepřipuštění k obhajobě.

Posudek vedoucího a oponenta na práci Odevzdaná práce je hodnocena posudky, které vypracují vedoucí práce a oponent práce. Oponenta práce jmenuje osoba určená příslušným předpisem. Posudek hodnotí zejména postup při řešení, dosažené výsledky a prezentaci výsledků.

Obhajoba práce probíhá před komisí akademických pracovníků. Komise posoudí úroveň práce a rozhodne o hodnocení známkou výborně, velmi dobře, dobře nebo nevyhověle.

1.2 Struktura bakalářské práce

Struktura bakalářské práce není předem pevně určena a může se podle typu a zaměření práce lišit. Přesto lze říci, že základní členění prací bývá obdobné. Nyní ho stručně uvedeme.

Úvod by měl formulovat obsah, cíle práce a motivaci diplomanta. Je třeba jasně vymezit čím se práce zabývá a jaké cíle a výstupy mají být dosaženy.

Odborná část představuje stěžejní část práce. Zde diplomant formuluje východiska práce, postup řešení a dosažené výsledky. Podle zaměření práce bývá také vhodné zpracovat přehledovou část, která stručně popíše aktuální stav řešené problematiky. Pokud je hlavním výsledkem práce softwarová aplikace, pak tato část obsahuje popis řešení aplikace, kde z tohoto popisu musí být jasné, jak je software utvořen, jakými postupy, jaká je jeho vnitřní struktura, jaké byly použité technologie a proč, jaké jsou jeho vlastnosti. Vždy usilujeme o odborný a kritický pohled. Snažíme se vymezit slabé a silné stránky.

Při dokumentaci software je žádoucí se zaměřit na podstatné věci, není nutné podat vyčerpávající souhrn nepodstatných informací (k tomuto účelu je možné

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací. studovat například zdrojový kód) a není vhodné, snad s výjimkou obzvláště důležitých částí, uvádět zdrojové kódy nebo jejich části.

Uživatelská část je součástí práce, pokud je výsledkem něco, co je možné uživatelsky používat, nejčastěji softwarová aplikace nebo nějaké zařízení. V takovém případě je potřeba vytvořit uživatelskou příručku, která bude uživatele informovat o tom, jak věc správně používat.

Závěr je vhodným místem pro shrnutí práce. Je dobré shrnout dosažené výsledky, uvést zda a jak se výsledky práce uplatňují (nebo by se mohly uplatnit) v praxi a jaký by mohl být další vývoj práce.

Text práce musí být psán odborně a kriticky. Autor musí osvědčit svou odbornou úroveň, schopnost kriticky pracovat se zdroji informací, aplikovat vhodné postupy popřípadě přinášet postupy vlastní, originální a odborně přijatelné.

Průvodce studiem

Práce, která přebírá cizí myšlenky bez uvádění zdrojů je plagiátem. Plagiátorství je velmi hrubým přestupkem proti akademické etice.

Autor musí být také dobře obeznámen s aktuálním stavem problematiky. Z práce musí být jasné, jaké zdroje a kdy použil, co je jeho vlastním přínosem a co převzal. Použité zdroje se na příslušných místech v práci citují uvedením zdroje a případně přímou citací dotčené pasáže. Není-li toto hledisko v práci naplněno, bude práce působit neodborným dojmem nebo vznikne podezření, že se jedná o plagiát. Taková práce nemůže být úspěšně obhájena. Použité zdroje se uvádí v seznamu použité literatury.

Častou otázkou je požadovaný rozsah práce. Počet stran není důležitým parametrem, text by měl být zejména výstižný. Přiměřený počet stran se může podle tématu dost lišit, lze říci, že bakalářské práce mají běžně cca 30-50 stran. Není to však žádné pravidlo.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

2 Obhajoba

Obhajoba bakalářské práce je součástí Státní závěrečné zkoušky. Probíhá před komisí složenou z akademických pracovníků. Průběh obhajob řídí její předseda, který je zároveň členem komise. Smyslem obhajoby, jak název napovídá, je osvědčit odbornou úroveň diplomanta prostřednictvím bakalářské práce a obhajoby. Před tím, než popíšeme průběh samotné obhajoby, uvedeme co obhajobě předchází.

2.1 Před obhajobou

Hotovou bakalářskou práci diplomant s předstihem odevzdá a to způsobem, který určuje příslušný předpis. Tento dokument stanovuje jak má být práce provedena (typografie, vazba, provedení příloh, počet výtisků . . .) a kdy, kde a jak má být odevzdána. Poté je pracovištěm stanoven oponent práce.

Oponent práce je akademický pracovník (popřípadě jiný dostatečně fundovaný odborník), který se písemně formou posudku kriticky vyjádří ke kvalitě předložené práce. Oponent hodnotí práci zejména z těchto hledisek:

- jak byly splněny cíle uvedené v zadání
- jak byly zvoleny postupy při řešení práce
- dosažené výsledky a jejich přínos
- jazyková a formální úroveň práce

Součástí posudku mohou být také otázky na diplomanta, které by měly být při obhajobě zodpovězeny. Této možnosti oponent využije nejčastěji v situaci, jsou-li některé části práce oponentovi nejasné. Posudek práce končí návrhem oponenta zda doporučuje nebo nedoporučuje práci k obhajobě a návrhem známky (výborně, velmi dobře, dobře, nevyhověl).

Průvodce studiem

Rozhodně není dobré se spolehnout na odbornou kvalitu práce a ignorovat požadavky na její formální a jazykovou úroveň. V takové situaci bývá nejlepším řešením práci nedoporučit k obhajobě, aby diplomant za jinak vynikající práci neusiloval při obhajobě v lepším případě o známku „dobře“.

Posudek práce vypracovává také vedoucí práce. Hlediska hodnocení jsou obdobná, jako u oponenta, navíc se obvykle vedoucí vyjadřuje také ke spolupráci diplomanta s vedoucím práce (aktivita, samostatnost, schopnost přinášet vlastní a originální řešení . . .). Diplomant má právo se s posudky seznámit (obvykle několik dní) před obhajobou. Konkrétní termíny pro posudky stanovuje příslušný předpis.

Pokud diplomant obdrží zamítavý posudek (nebo posudky) neznámá to, že nesmí jít k obhajobě. Svou situaci musí posoudit každý diplomant sám, užitečné bývá poradit se s vedoucím práce.

2.2 Průběh obhajoby

Jak jsem již uvedli, obhajoba práce probíhá před komisí složenou s akademických pracovníků. Obhajoba je veřejná, takže mohou být přítomny také další osoby. Obvykle

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací. je přítomen také vedoucí a oponent práce. Samotná obhajoba je vlastně řízenou diskusí nad bakalářskou prací, při které diplomant osvědčuje svou odbornou úroveň. Průběh obhajoby se sestává z posloupnosti následujících kroků:

Představení diplomanta - předseda představí diplomanta komisi a stručně uvede bakalářskou práci.

Vystoupení diplomanta - diplomant ve vymezeném čase (stanoví předpis, obvykle 15 minut) komisi představí svou práci. Je na diplomantovi, jakým způsobem si připraví své vystoupení. Rozumné je nejprve uvést práci — co je předmětem práce a proč, jaká byla motivace diplomanta. Další částí je stručný popis práce z odborného hlediska. Zde diplomant uvádí, jaké použil postupy a proč a k jakým došel k výsledkům. Je-li hlavním výsledkem práce softwarová aplikace, pak by ji diplomant měl během svého vystoupení předvést.

Protože čas na vystoupení je dost omezený, je nutné rozvážit co je skutečně podstatné a právě na to se při obhajobě zaměřit. Komise by na základě vystoupení měla získat celkový přehled o obhajované práci — jak z hlediska obsahu, tak formy.

Průvodce studiem

Vystoupení na obhajobě je potřeba si dobře připravit. Je užitečné si rozmyslet alespoň strukturu vystoupení, ještě lepší je však vyzkoušet si obhajobu „nanečisto“. Vymezený čas je poměrně krátký a není-li obhajoba připravena, pak diplomant stihne pouze část vystoupení. Není výjimkou, že se diplomant vůbec nedostane k podstatě práce. Zkušenosti ukazují, že předsedové komise bývají v tomto ohledu nekompromisní a neumožní diplomantovi vystoupit nad rámec vymezeného času.

Seznámení komise s posudkem vedoucího - vedoucí práce seznámí komisi se svým posudkem. V případě nepřítomnosti vedoucího, seznámí přítomné s obsahem posudku některý z členů komise.

Seznámení komise s posudkem oponenta - oponent práce seznámí komisi se svým posudkem. V případě nepřítomnosti oponenta, seznámí přítomné s obsahem posudku některý z členů komise.

Vyjádření diplomanta k posudkům - diplomant má právo vyjádřit s k jakékoliv části libovolného z posudků. Obsahují-li posudky otázky na diplomanta pak se od diplomanta očekává, že na všechny otázky odpoví.

Průvodce studiem

Obhajoba bakalářské práce je odborná diskuse. Není to „obhajoba před trestním senátem“, kde bývá častá snaha, se ze „všeho nějak vymluvit“. Pokud diplomant prokáže schopnost reflexe vlastní práce, může to být pro něj do jisté míry přínosem před situací, kdy zjevné nedostatky se diplomant snaží „vysvětlit“. Na druhou stranu to vůbec neznamená, že by diplomant, pokud má jiný názor než členové komise, neměl svůj názor obhajovat. Naopak, měl, je-li o něm přesvědčen a má pro to dostatek argumentů.

Diskuze - probíhá zejména mezi členy komise a diplomantem nad bakalářskou prací. Předseda komise může také připustit otázky ostatních přítomných.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.
Hodnocení obhajob probíhá obvykle na závěr, po skončení vystoupení všech diplomantů. Rozhodování komise bývá neveřejné. V případě neúspěchu může diplomant podle platných studijních předpisů obhajobu opakovat.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

3 Tvorba odborné části práce

V této kapitole uvedeme základní doporučení pro strukturu a obsah odborné části bakalářské práce. Čtenář by měl z odborné části získat stručný avšak ucelený přehled o jejím řešení, a to jak z hlediska použitých postupů, tak technologií, standardů a způsobu implementace software, je-li její součástí.

Autor práce musí předpokládat, že čtenář nemusí být s problematikou a použitými technologiemi podrobně seznámen a tento fakt je nutné zohlednit při tvorbě programátorské dokumentace. Odborné pojmy a zkratky, které nemusí být čtenáři obecně známy je třeba v textu náležitě vysvětlit nebo odkázat na příslušnou literaturu.

3.1 Přehledová část

Přehledová část obsahuje stručné shrnutí aktuálního stavu řešené problematiky. Dbáme na vymezení řešené oblasti a důsledně pracujeme s literaturou.

3.2 Použité technologie

Úvodní část programátorské dokumentace by měla přehledovým způsobem uvést technologie, které byly použity pro řešení problematiky bakalářské práce. Patří sem zejména programovací jazyky, informační technologie, systémové platformy (například Java, .NET) a podobně. Výběr použitých technologií je vhodné stručně zdůvodnit, popřípadě technologie, které připadají v úvahu jako alternativy navzájem porovnat. Tato část je, jak jsme uvedli dříve, přehledová a proto je třeba pamatovat na stručnost, jejím cílem zejména není encyklopedickým způsobem popsat každý technický detail v práci obsažený.

3.3 Rozbor architektury aplikace, struktury kódu a dynamiky aplikace

Pro popis toho, jak je aplikace vnitřně organizována a jak je členěn kód je vhodná kombinace textového popisu a diagramů. Pokud je to možné, používáme diagramy, které jsou v odborné veřejnosti zažitě a jsou tak pro čtenáře lépe srozumitelné. Nejpoužívanější jsou v současné době diagramy obsažené v jazyce UML.

Pro popis aplikace z hlediska funkcí, tak jak je vidí uživatel jsou vhodné diagramy případů užití (use-case diagramy). Pro přehledové zobrazení funkcí a jejich hierarchie jsou vhodné grafické use-case diagramy. Podrobnější popis případů užití můžeme provést pomocí textových scénářů k jednotlivým případům užití. Není nutné psát textové scénáře pro všechny případy užití stačí se zaměřit na ty, které jsou z hlediska aplikace podstatné. Zejména není nutné psát textové scénáře u případů užití, které jsou triviální, nepodstatné nebo se často analogicky opakují. Pokud scénáře z nějakého důvodu nevyhovují, je možné použít jiný, třeba vlastní systém popisu.

Strukturu kódu aplikace zachycujeme podle použitého paradigmatu. V případě, že je aplikace implementována objektově, pak je třeba popsat objekty, jejich atributy (v obecném smyslu) a vztahy mezi nimi. Pro tento účel můžeme použít například diagram tříd z jazyka UML, kde zachytíme třídy, jejich vlastnosti a metody, vztahy (včetně kardinality) mezi třídami. Pokud je naše aplikace implementována v některém z méně rozšířených objektových jazyků (například v jazyce založeném na prototypu), mohou být diagramy tříd z jazyka UML nevhodné. UML bylo vytvořeno pro modelování systémů tvořených v populárních, široce používaných jazycích. Dobrou cestou bývá

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací. nahlédnutí do literatury k daném programovacímu jazyku, jak se takové programy dokumentují. Je dobré textově popsat účel jednotlivých tříd, důležité vlastnosti a metody. To napomůže lepší orientaci v diagramech, resp. interpretaci.

Pokud aplikace není implementována pod objektovým paradigmatem, pak aplikaci popíšeme z hlediska použitého paradigmatu - v případě funkcionální paradigmatu z hlediska funkcí a vztahů mezi funkcemi, v případě strukturovaného programování z hlediska procedur, datových struktur a modulů.

Pokud je aplikace implementovaná multiparadigmově, pak je vhodné užít kombinaci různých postupů.

Programátor v aplikaci implementuje řadu procesů (funkcí), z nichž některé jsou z hlediska celku důležité. Takové procesy je třeba popsat v programátorské dokumentaci, aby byly zachyceny algoritmy, které procesy realizují. Nejčastěji se používá grafické znázornění. Jazyk UML [3] nabízí několik diagramů pro popis dynamiky systému a to podle toho, z jakého hlediska chceme proces popsat. Nejčastěji používaným způsobem je popis z hlediska činnosti algoritmu, který lze zaznamenat diagramem aktivity. Pokud je třeba popsat proces z hlediska stavů objektu a vztahů mezi stavy, pak použijeme stavový diagram. Některé procesy je dobré popsat z hlediska časové následnosti. K tomuto účelu je určen stavový diagram.

Jsou-li v návrhu aplikace použity další techniky (například návrhové vzory), je potřebné se o tom v dokumentaci zmínit. V případě návrhových vzorů použité vzory uvedeme a velmi stručně popíšeme.

V řadě případů se při implementaci aplikace využívají komponenty třetích stran. Tuto skutečnost v žádném případě nelze v programátorské dokumentaci zamlčet. Naopak, je nutné uvést autora, účel komponenty, způsob jejího užití v aplikaci a důvody pro její volbu.

Není nutné ani vhodné, aby v programátorské dokumentaci byl uveden zdrojový kód aplikace. Ten bude uložen na doprovodném CD k bakalářské práci. V situacích, kdy je to z hlediska pochopení nezbytně nutné, můžeme k dílčím problémům uvést části zdrojového kódu. Vždy dbáme na stručnost.

3.4 Datová základna aplikace

V úvodu popisu datové základny je vhodné uvést stručný popis použitého řešení, jako je použitý SRBD, dotazovací jazyk, popř. souborová databáze a podobně. Dále by měl následovat popis datového modelu aplikace. Nejpoužívanějším datovým modelem je model entitně-relační. Pokud jej v datové vrstvě aplikace využíváme, pak v programátorské dokumentaci uvedeme ER-diagram, seznam tabulek s popisem sloupců, jejich datových typů a integritních omezení. Není vhodné namísto ER-diagramu použít grafické znázornění tabulek z aplikace Microsoft Access.

Jak známo, datová vrstva může být realizována také jinými, méně často užívanými způsoby: například S-výrazy, XML soubory, objektovou persistencí a podobně. V tomto případě je dobré věnovat pozornost vysvětlení principu fungování takové databázové vrstvy a důvodů pro její volbu.

3.5 Instalátor aplikace

K vytvoření instalačního balíčku aplikace můžeme použít některý ze speciálních nástrojů k tomu určených. V programátorské dokumentaci uvedeme jak byla instalace vytvořena.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

3.6 Bezpečnost

Specifickou oblastí aplikací je bezpečnost. V programátorské dokumentaci je třeba se vyjádřit zejména k zabezpečení uživatelských práv a přístupu k datům. Tato otázka je obzvláště důležitá u webových aplikací. Při popisu bezpečnosti aplikace se věnujeme možným bezpečnostním hrozbám, jejich řešení a přehledu použitých technologií pro zabezpečení aplikace.

3.7 Standardy

V programátorské dokumentaci se autor musí vyjádřit ke standardům. Je vhodné uvést jaké standardy se pro řešení problematiky práce nabízejí, které z nich jsou v aplikaci použity a jak je zajištěno jejich naplnění. Tuto otázku je třeba pečlivě zpracovat zejména u webových aplikací.

3.8 Testování aplikace

Autor by měl uvést jak byla aplikace testována, jaké je její chování v reálném nasazení a jaká je kompatibilita aplikace s operačními systémy a dalšími technologiemi. U webových aplikací je velmi důležitá kompatibilita s webovými prohlížeči.

3.9 Pár poznámek k sazbě

Diagramy, které vkládáme do textu jako obrázky je vhodné vkládat ve vektorových formátech kvůli kvalitě výstupu. V případě, že je obrázek bitmapový (například snímek obrazovky), je třeba aby byl v dostatečném rozlišení tak, aby na tiskovém výstupu nebyl obrázek zkreslený. Také je potřeba dbát na rozumnou míru komprese u ztrátově komprimovaných obrázků, aby nedošlo k citelné degradaci kvality výstupu.

Není vhodné používat software, který do výstupu přidává informace o software, ve které byl obrázek vytvořen. Typické je to pro zkušební verze různých aplikací, které do obrázku například přidají vodoznak, nebo jiné označení. Diagramy a jiné tzv. perovky vkládáme v černobílém provedení, u obrázků, které není možné vkládat černobíle jako například snímky obrazovky používáme stupně šedi. U diagramů je nutné dbát na to aby hrany nepocházely přes vrcholy, nebo textové popisky hran neprocházely přes samotné hrany. Můžeme narazit na situaci, že diagram bude příliš velký pro umístění na stránku. Na prvním místě se vždy zamyslíme, zda to co je na obrázku je v pořádku. Není daný problém vyřešen špatně, když je nutné jej zaznamenat tak složitým diagramem? Pokud je nutné, aby byl diagram tak velký, pak buď vysázíme obrázek na výšku, nebo jej rozdělíme na více obrázků. Části, které mají být spojeny, v obrázku označíme. Nezapomeneme to v textu, kde na obrázek odkazujeme, výslovně zmínit.

Názvy souborů, proměnných, tříd, vlastností a metod sázíme zásadně strojopisným řezem písma.

3.10 Jazyk práce

Bakalářská práce musí mít také náležitě jazykové kvality. Dbáme na dodržování odborného stylu. Charakterizují ho tyto vlastnosti:

- přesnost a jednoznačnost ve vyjadřování

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

- používání odborných termínů
- stručnost a věcnost
- důraz na objektivitu
- používání autorského plurálu
- používání neosobního vyjadřování (pasivní konstrukce)

Autorský plurál (tzv. *plurál skromnosti*) znamená, že text formulujeme v první osobě množného čísla.

Příklad 3.1. Místo „Nyní vysvětlím ...“ použijeme autorský plurál, tedy „Nyní vysvětlíme ...“.

Příklad 3.2. Příkladem neosobního vyjadřování je věta „V předcházející kapitole bylo uvedeno ...“

Autorský plurál (1. osoba č. mn.) ?Pokusíme se vysvětlit?? ??ukážeme si na příkladě??
Inkluzivní plurál (1. osoba č. mn.) ?Věnujme se podrobněji?? ??pokusme se zhodnotit??
Neosobní vyjadřování (pasivní konstrukce) ?V předcházející kapitole bylo naznačeno??
?Donedávna se soudilo?? V 1. osobě. č. jednotného ?Pravdivost tohoto tvrzení dokáží následujícím výpočtem?? ??jsem pevně přesvědčen??

Informatika je typická velkým množstvím cizích slov a používáním anglismů. Není však vhodné anglicismy nadužívat — je-li k dispozici vhodný český termín, pak jej také použijeme.

Příklad 3.3. Například slova jako „downloadovat“ nebo „debugovat“ mají vhodné české protiklady, a to „stáhnout“ a „ladit“.

Vždy je dobré nechat si provést jazykovou korekturu textu. Gramatické chyby jsou špatnou vizitkou diplomanta.

Průvodce studiem

Zásadně se vyvarujeme používání slangových výrazů jako je například výraz „mikrosoftí“, hovoříme-li o výrobku firmy Microsoft.

3.11 Práce zaměřená na hardware

Specifickým druhem prací jsou práce zaměřené na hardware. U těchto prací je pro porozumění problematice důležité dobře popsat použité technologie a principy funkce navrženého zařízení. Je vhodné uvádět schémata zapojení a především je okomentovat co možná nejsrozumitelnějším způsobem. Uvědomme si, že na školách univerzitního typu se problematice hardware věnuje menší pozornost, než na školách technického typu. Proto i míra porozumění odborného textu s takovým obsahem je na školách univerzitního typu nižší.

3.12 Práce zaměřená na počítačovou hru

Jiným specifickým typem bakalářských prací jsou počítačové hry. Zde je podstatné neopomenout popis kompletní mapy anebo scénáře hry včetně postupu jak lze hru dohrát. Dále je třeba se zaměřit na stránku grafických technologií použitých ve hře a technologií použitých pro realizaci strategie hry počítačem (algoritmy a heuristiky).

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

4 Tvorba uživatelské dokumentace

Účelem uživatelské dokumentace je poskytnout uživateli návod na ovládání aplikace. Autor aplikace musí, podobně jako při tvorbě uživatelského rozhraní zohlednit, jaká skupina uživatelů bude s jeho aplikací pracovat a podle toho také tvořit uživatelskou dokumentaci. Obecně je vhodné, aby uživatelské dokumentaci porozuměl uživatel počítače se základními znalostmi o jeho ovládání. Pokud je aplikace určena pro užší skupinu uživatelů (zkušení uživatelé, specialisty na určitou oblast, jako například grafiku, zvuk a podobně), nemusíme volit tak detailní popis ovládání, nebo zpracovávané problematiky jako u aplikací určených pro široké vrstvy uživatelů. Na druhou stranu je třeba dbát, aby se autor ve snaze vše uživateli vysvětlit nekryl s manuálem ke uživatelského rozhraní systému, například není nutné vysvětlovat co je kliknutí myši, že se velká písmena píše pomocí klávesy Shift a podobně.

Při tvorbě dokumentace k ovládání aplikace je vhodné používat snímky oken, popřípadě pokud se chceme zaměřit na detail (část okna), pak je lepší použít výřez okna. Uvědomme si, že uživatelé nečtou příliš rádi čtou manuály a je dobré, když je možné se samotných snímků obrazovky se v dané úloze zorientovat.

Výřez okna je přehlednější a v textu nezabere tolik místa. Je důležité, aby bylo zřejmé, že se jedná o výřez okna, aby uživatel nebyl zmaten tím, že nemůže takové okno v aplikaci najít.

Kromě obrázků používáme textový popis, ve kterém se na jednotlivé obrázky odkazujeme. Názvy ovládacích prvků je vhodné v textu uvádět, protože se zlepšuje orientaci v dialogích. Styl textu by měl být stejný jako v ostatních kapitolách práce, není vhodné se inspirovat žoviálním jazykem některých manuálů aplikací dostupných na trhu. Kromě názvu ovládacího prvku uvádíme také jeho titulek, resp. popisku je-li k ovládacímu prvku přiřazena. Název popisky sázíme buď v uvozovkách, nebo kurzívou.

Příklad 4.1. Dobré: Klikneme na „Časové pásmo“.

Lepší: Klikneme na rozbalovací seznam „Časové pásmo“

Klikneme na tlačítko „OK“.

Do textového pole „Adresa“ uvedeme

4.1 Struktura uživatelské dokumentace

V úvodu programátorské dokumentace nejprve popíšeme instalaci aplikace. Měli bychom zmínit jaké volby při instalaci lze nastavit a co znamenají.

Dále uvedeme jakým způsobem se aplikace spustí (kde se nachází její zástupce a podobně) a pokud je do aplikace třeba přihlášení, jak se lze přihlásit. Zde je vhodné uvést jména a hesla pro uživatele všech rolí v aplikaci, aby je uživatel (což může být také vedoucí práce, nebo její oponent) mohl vyzkoušet.

Pak následuje hlavní část uživatelské dokumentace, která popisuje zpracování jednotlivých úloh (funkcí) v aplikaci. Popis by měl být organizován analogicky s organizací dialogů v aplikaci, tedy od hlavního okna k oknům podřízeným. Pokud je aplikace rozsáhlá, může být užitečné uvést strom dialogů.

Jednotlivé úlohy je vhodné popisovat stylem „krok za krokem“. Popis zpracování uživatelských úloh je vhodné rozdělit z hlediska rolí. Například na popis úloh pro běžného uživatele a správce aplikace.

Poté co je popsáno ovládání aplikace, věnujeme se možnostem nastavení parametrů pro aplikaci. Na závěr uživatelské dokumentace uvedeme, jak je možné aplikaci odinstalovat.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

4.2 Pracovní scénáře

Problematika některých témat nemusí být obecně známa a tak samotná uživatelská příručka nemusí být dostatečná pro pochopení, jakým způsobem software správně používat. V takovém případě je vhodné, aby v úvodu práce diplomant problematiku stručně vysvětlil a v uživatelské příručce uvedl několik scénářů, které krok po kroku umožní vyzkoušet stěžejní funkce aplikace.

Příklad 4.2. Uvažme jako příklad aplikaci pro tzv. vyvolávací systém. To je systém, se kterým se setkáváme na úřadech, kdy si klient vyžádá z počítače podle služby pořadový lístek s číslem, podle kterého je potom volán na příslušné pracoviště, kde splní jeho požadavek. Přestože každý s takovým systémem v praxi přišel do styku, nebude uživateli aplikace, která se o správu klientských požadavků stará, předem jasné, jak vlastně tuto aplikaci používat. Proto je dobré uvést stylem krok po kroku příklad pro vyzkoušení hlavních funkcí takové aplikace. Například vydání lístku, správa fronty, obslužení klienta a podobně.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

5 Sazba textu práce v typografickém systému $\text{T}_{\text{E}}\text{X}$ s balíkem maker $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

V této kapitole se čtenář seznámí se základy tvorby dokumentů v systému $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ a se zásadami sazby bakalářské práce. Tato kapitola je tedy užitečná i pro ty, kteří budou práci případně sázet v jiné aplikaci.

$\text{T}_{\text{E}}\text{X}$ (čteme tech) je typografický formátovací software určený pro sazbu dokumentů. Autorem tohoto systému je Donald E. Knuth. $\text{T}_{\text{E}}\text{X}$ byl původně vytvořen pro sazbu technických a matematicky zaměřených textů, ale je možné jej používat i pro sazbu ostatních dokumentů. $\text{T}_{\text{E}}\text{X}$ pracuje jako řádkový překladač, který vstupní soubor, nebo soubory, které obsahují formátovací příkazy přeloží (vysází) do výstupního souboru ve formátu DVI. Soubor ve formátu DVI, je možné snadno převádět například do formátu PostScript pomocí programu `dvips`, nebo do formátu PDF programem `dfipdfm`. Existuje také varianta $\text{T}_{\text{E}}\text{X}$ u zvaná `pdf $\text{T}_{\text{E}}\text{X}$` , která produkuje výstup přímo ve formátu PDF a podporuje specifické možnosti tohoto formátu, jako je sazba hypertextových odkazů, nebo obsahu v PDF souboru.

$\text{T}_{\text{E}}\text{X}$ je volně dostupný software a obvykle se vydává v tzv. distribucích (podobně jako operační systém Linux). Existuje řada distribucí pro nejrůznější platformy a některé distribuce jsou dokonce víceplatformní. Asi nepoužívanější distribucí pro operační systémy z rodiny Windows je $\text{T}_{\text{E}}\text{X}$ Live, která je dostupná na <http://www.tug.org/texlive/>. Pro operační systém Linux se nejčastěji setkáme s verzí $\text{T}_{\text{E}}\text{X}$ u nazvanou `Te $\text{T}_{\text{E}}\text{X}$` (<http://www.tug.org/tetex/>), která se dodává obvykle jako balíček dané distribuce Linuxu a je tedy možné jej jednoduchým a systémovým způsobem nainstalovat. Pro operační systém MacOS X jsou nepoužívanějšími distribucemi `Mac $\text{T}_{\text{E}}\text{X}$` (www.tug.org/mactex/) a `Te $\text{T}_{\text{E}}\text{X}$` . Při instalaci, zejména pod MS Windows, je vhodné prostudovat dokumentaci, kde bývá kromě jiného uvedeno, jak zvolit parametry instalace tak, aby se nainstalovalo také české prostředí pro $\text{T}_{\text{E}}\text{X}$ a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.

Protože zdrojově soubory $\text{T}_{\text{E}}\text{X}$ u jsou vlastně obyčejné textové soubory, můžeme je vytvářet v libovolném textovém editoru, který je schopen ukládat textové soubory. Takový soubor následně přeložíme `TeXem` a získáme výsledný vysázený dokument. Pro $\text{T}_{\text{E}}\text{X}$ existují také specializované editory, které vytváření souborů usnadňují. Tyto editory bývají součástí některých distribucí. Pro systémy z rodiny Windows jsou nepoužívanějšími editory `LeD` (<http://www.latexeditor.org/>), `WindShell` (<http://www.winshell.de/>) a `$\text{T}_{\text{E}}\text{X}$ nicCenter` (<http://www.toolscenter.org/>). V systému Linux se často používá textový editor Emacs, který obsahuje zvláštní editační mód pro `TeX`. V systému MacOS X jsou nepoužívanější editory `$\text{T}_{\text{E}}\text{X}$ Shop` (<http://www.uoregon.edu/~koch/texshop/>) a `i $\text{T}_{\text{E}}\text{X}$ mac` (<http://itexmac.sourceforge.net/>). Po instalaci editoru je třeba provést nastavení parametrů pro překlad souborů `TeXem`, tak aby správně pracovalo české prostředí.

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (čteme latech, popřípadě lejtech) je balík maker pro $\text{T}_{\text{E}}\text{X}$, která usnadňují sazbu běžných dokumentů. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ existuje ve více verzích, v současné době je však nepoužívanější verze $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 2e, na kterou se v tomto textu zaměříme.

Následující popis balíku maker $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ je pouze přehledem těch nejzákladnějších příkazů $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u a je určen úplným začátečníkům. Pro podrobnější popis odkážeme na [4].

5.1 Základy

Znaky „mezera“, „tabulátor“ a „konec řádku“ jsou interpretovány jako jedna mezera. Prázdný řádek, nebo i více prázdných řádků se interpretuje jako konec odstavce. mezery, znaky

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

Mezi slovy bude jedna mezera. Mezi slovy bude jedna mezera.

Zde je nový odstavec. Zde je nový odstavec.

A ještě jeden odstavec. A ještě jeden odstavec.

Znaky $\$$ $\&$ $\%$ $\#$ $_$ $\{$ $\}$ \sim \hat \backslash mají v \LaTeX u speciální význam a tak je nelze sázet prostým zapsáním do textu. Pokud chceme takový znak vysázet, musíme před tímto znakem uvést obrácené lomítko.

$\$$ $\&$ $\%$ $\#$ $_$ $\{$ $\}$ \backslash $\$$ \backslash $\&$ \backslash $\%$ \backslash $\#$ \backslash $_$ \backslash $\{$ \backslash $\}$

Příkazy v \LaTeX u začínají znakem zpětné lomítko. Poté následuje název příkazu. V názvech příkazů se rozlišují malá a velká písmena. Některé příkazy vyžadují ještě parametr, který se uvádí v množinových závorkách.

Poslední slovo bude vysázeno **tučně**. Poslední slovo bude
vysázeno $\text{\textbf{tučně}}$.

Ve vstupním souboru se mohou také vyskytovat komentáře. Vše co se vyskytuje za znakem $\%$ bude až do konce řádku ignorováno.

Moudrý text. Moudrý $\%i$ hloupý
text.

5.2 Struktura souboru

Vstupní soubor musí mít na začátku deklarovaný styl dokumentu příkazem `documentclass`. Styl dokumentu definuje, jak se mají jeho jednotlivé části sázet, takže kromě jiného definuje například rozměry stránky, velikosti písem, nebo řezy písem.

Nejpoužívanějšími styly jsou *article*, dále například *report*, nebo *book*. Po deklaraci stylu dokumentu může následovat příkaz `usepackage`, kterým deklarujeme použití dodatečného balíčku *maker*. Běžně můžeme potřebovat v dokumentu používat více balíčků. Pro nás důležitým balíčkem bude balíček *czech*, který zavede české prostředí a dále balíček *graphicx*, který budeme používat pro práci s obrázky.

Dále může následovat samotný dokument, kde začátek samotného dokumentu je uvozen příkazem `\begin{document}` a konec příkazem `\end{document}`.

Typická hlavička našich dokumentů může vypadat například takto:

```
\documentclass[11pt]{article}
\usepackage{czech}
\usepackage{graphicx}

\begin{document}

\end{document}
```

V případě, že sázíme rozsáhlejší text je výhodné si text organizovat ve více souborech (například po kapitolách) pro lepší přehlednost a orientaci ve zdrojových souborech. Vložení souboru se provádí příkazem `\input{soubor}`, kde parametr *soubor* představuje název souboru, který má být vložen na místo příkazu ve zdrojovém souboru.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

5.3 Překlad souboru

Překlad souboru do výstupního formátu se provádí příkazem `cslatex` s parametrem názvu souboru, který chceme přeložit. Výstupem je soubor ve formátu DVI, který je možné (viz úvod této kapitoly) převádět do dalších formátů. V současné době je velmi rozšířen také `pdfCsLATEX`, který sází výstup přímo do souboru ve formátu PDF. Sazba se provádí příkazem `pdfcslatex` s parametrem názvu souboru, který má být přeložen. Soubory doporučujeme ukládat v kódování ISO-8859-2. Vždy je však potřeba ověřit kódování a možnosti jejího nastavení podle vaší instalace \TeX u a podpůrných aplikací.

5.4 Základy sazby textu

\LaTeX se automaticky stará o řádkový a stránkový zlom. Pokud nemůže provést řádkový zlom mezi slovy, provede rozdělení slova. Řádkový zlom lze také nařídit, ale je vhodné jej používat pouze v nezbytných případech, protože samotné algoritmy \LaTeX u jsou velmi propracované a dokážou dokument zalamovat tak, že výsledná sazba je opticky velmi kvalitní. Neuváženým používáním ručních zlomů kvalitu výstupu zhoršíme. Řádkový zlom se provádí příkazem `\\` nebo `*` kdy navíc zakážeme stránkový zlom po řádkovém zlomu. Stránkový zlom lze nařídit příkazem `\newpage`.

5.4.1 Mezislovní mezery

Pro opticky dobrý výsledek, \LaTeX automaticky zvětšuje, nebo zmenšuje mezery mezi slovy. \LaTeX předpokládá, že tečka následující za velkým písmenem označuje zkrácení slova. V ostatních případech \LaTeX považuje tečku za ukončení věty.

Tato pravidla však v některých situacích jsou pro správnou sazbu mezislovních mezer nedostatečná. Proto má \LaTeX definováno několik příkazů pro ovlivnění sazby mezer. Znak `\` následovaný mezerou znamená, že mezera ve výsledném dokumentu nesmí být v případě potřeby rozšířena. Znak `~` označuje mezeru, která nesmí být rozšířena a navíc na jejím místě nesmí dojít k řádkovému zlomu. Příkaz `\@` před tečkou znamená, že tečkou věta končí, přestože je před ní velké písmeno.

Novák a kol. je autorem . . .
Soubor zabere na disku 10 MB.
Prof. Chocholoušek neordinuje.
Konečná trasy metra A. Vystupte.

```
Novák a kol.\ je autorem \ldots\  
Soubor zabere na disku 10~MB.\  
Prof.~Chocholoušek neordinuje.\  
Konečná trasy metra~A\@.  
Vystupte.\  

```

5.4.2 Dělení slov

Již jsme zmínili, že \LaTeX dokáže zařídit dělení slov v případě, že nelze provést opticky vhodný řádkový zlom. K dělení slov se používá příslušný slovník daného jazyka. Dělení slov do dokumentu zavede balíček pro národní prostředí, v našem případě balíček `czech`. Může se však stát, že některá slova nebudou rozdělena správně, například proto, že se dané slovo ve slovníku nevyskytuje a obecný algoritmus \TeX u provede rozdělení slova nesprávně. V takovém případě můžeme použít příkaz `\hyphenation`, kde pomocí znaku pomlčka definujeme místa, kde je možné slovo rozdělit.

```
\hyphenation{Roz-dě-li-tel-ný}
```

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací. Naopak také může nastat situace, že \LaTeX rozdělí slovo, které se však dělit nemá, například název. I k tomuto účelu použijeme příkaz `\hyphenation`, kde nedělitelné slovo uvedeme bez pomlček, tedy nedefinujeme žádný bod k rozdělení slova. V argumentu příkazu `\hyphenation` můžeme uvést i více slov oddělených mezerou.

```
\hyphenation{Common LISP}
```

5.4.3 Sazba uvozovek

Uvozovky se v českém a anglickém textu sázejí odlišně. V anglickém textu se otevírací uvozovka zadávají jako dva znaky „obrácený apostrof“ a uzavírací uvozovky jako dva znaky „apostrof“. V českém textu se sázejí tzv. uvozovky dole a uvozovky nahoře. Pokud použijeme styl `czech.sty`, můžeme sazbu českých uvozovek provést příkazem `\uv{}`. Uvozovky se zásadně nezadávají pomocí znaku “.

Click the “OK” button.

Click the ``OK`` button.

Klikněte na tlačítko „OK“.

Klikněte na tlačítko `\uv{OK}`.

5.4.4 Sazba pomlček, teček a podobných znaků

Typografie rozlišuje několik druhů pomlček. Nejkratší pomlčkou je takzvaný spojovník. Používá se ke spojování slov a označuje se znakem `-`. Dalším typem pomlčky je takzvaná půlčtverčiková mezera, která se používá pro označení intervalu. Tuto pomlčku uvádíme znaky `--`. Třetím typem pomlčky je dlouhá čtverčiková pomlčka. Tu používáme k naznačení přestávky v řeči, nebo k oddělení části textu, významově analogicky jako čárku. V českých textech se dlouhá čárka sází s mezerami z obou stran. Znak `-` se také používá jako znaménko „mínus“ při sazbě matematiky.

Nebude-li pršet, nezmoknem.
 Výkon přístroje je 6–10 W.
 yes—or, no?
 1 nebo –1

Nebude-li pršet, nezmoknem.\\
 Výkon přístroje je 6--10~W.\\
 yes---or, no?\\
 \$1\$ nebo \$-1\$

5.4.5 Výpustky v textu

Výpustky (též elipsy) v textu se v \LaTeX u sázejí příkazem `\ldots`. Nesázejí se pomocí teček, protože mezery mezi tečkami při označení elipsy mají být užší, než mezi obyčejnými tečkami. V následujícím příkladě, je tedy první řádek vysázen nesprávně, zatímco druhý řádek je vysázen správně.

MacOS X, Windows, GNU/Linux ...
 MacOS X, Windows, GNU/Linux ...

MacOS X, Windows, GNU/Linux ...\\
 MacOS X, Windows, GNU/Linux \ldots

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

5.5 Druhy a velikost písma

Druh a velikost použitého písma definují jednotlivé příkazy \LaTeX u, resp. použité styly. Ruční nastavování velikosti písma je sice možné, ale měli bychom ho použít jen výjimečně a ve zvláště odůvodněných případech. Druh písma je také nastavitelný a při sazbě textu se s jeho nastavováním setkáme častěji. Možné druhy písma jsou vyobrazeny níže v příkladu. Připomínáme, že v případě použití jiného stylu dokumentu, než je použit v tomto textu, by mohla být výsledná sazba níže uvedeného příkladu jiná, protože styl může používat jiné řezy písem.

normální písmo	<code>\textrm{normální písmo}\</code>
bezpatkové písmo	<code>\textsf{bezpatkové písmo}\</code>
strojopisné písmo	<code>\texttt{strojopisné písmo}\</code>
střední písmo	<code>\textmd{střední písmo}\</code>
tučné písmo	<code>\textbf{tučné písmo}\</code>
vzpřímené písmo	<code>\textup{vzpřímené písmo}\</code>
<i>kurzíva</i>	<code>\textit{kurzíva}\</code>
<i>skloněné písmo</i>	<code>\textsl{skloněné písmo}\</code>
PÍSMO S KAPITÁLKAMI	<code>\textsc{písmo s kapitálkami}\</code>
<i>zvýrazněné písmo</i>	<code>\emph{zvýrazněné písmo}\</code>

5.6 Strukturování textu

\LaTeX , resp. třídy dokumentů mohou podporovat členění dokumentů pomocí kapitol, podkapitol, sekcí a podobně. V následující části se zaměříme na strukturování textu ve třídě `article`. Kapitoly, podkapitoly jsou automaticky číslovány. Pokud chceme zabránit očíslování oddílu, pak použijeme příkaz, jehož název je ukončen znakem `*`. Neočíslované oddíly se neuvedou v obsahu. Pro členění dokumentu do kapitol a podkapitol slouží následující příkazy.

```
\section{}
\subsection{}
\subsubsection{}
```

Z výše uvedeného vyplývá, že kapitoly a podkapitoly se člení do maximální úrovně 3. Hlubší členění se v textech menšího rozsahu nepoužívá, protože narušuje čitelnost textu. Třída `article` ani hlubší členění nepodporuje. Členění textu do číslovaných oddílů by se také nemělo nadužívat. Pokud chceme zvýraznit, resp. rozčlenit menší část textu, pak použijeme některý z následujících příkazů, který zajistí vysázení tučného, nečíslovaného nadpisu.

```
\paragraph{}
\subparagraph{}
```

Díky strukturování textu můžeme automaticky vygenerovat obsah dokumentu. To se provádí příkazem `\tableofcontents` a na místo příkazu bude vložen obsah. Protože pro generování obsahu \LaTeX používá pomocné soubory do kterých si při překladu ukládá informace o kapitolách a jejich číslování, je nutné pro správné vygenerování obsahu dokument přeložit celkem třikrát.

5.7 Křížové odkazy

V textech se běžně používají odkazy na jiné části textu, obrázky, tabulky, vzorce a podobně. \LaTeX je k tomuto účelu vybaven následujícím aparátem.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

K vytvoření odkazu, tedy „zapamatování si“ značky na daném místě textu se používá příkaz `\label{}`, kde v argumentu uvedeme název návěští odkazu.

Pro vysázení odkazu použijeme příkaz `\ref{}` nebo `\pageref{}`, kde v argumentu uvedeme návěští odkazu, na který chceme odkázat. Příkaz `\ref{}` vysází do textu číslo oddílu, kde se daný odkaz vyskytuje. Příkaz `\pageref{}` vysází do textu číslo stránky, na které se vyskytuje uvedený odkaz. Podobně jako u sazby obsahu je třeba zdrojový soubor přeložit vícekrát.

Tento text je součástí kapitoly 5.7 na straně 24.

```
Tento text \label{odkazy} je
součástí kapitoly \ref{odkazy}
na straně \pageref{odkazy}.
```

5.8 Poznámky pod čarou

Poznámky pod čarou se sázejí příkazem `\footnote{}`. \LaTeX automaticky rozhodne o umístění poznámky pod čarou a provede její očíslování, popř. označení pomocí písmenné řady. Implicitně se poznámky pod čarou číslovají, v následujícím příkladě ale použijeme číselnou řadu, aby nám poznámky pod čarou v tomto textu a příklad navzájem nekolidovaly.

Procesor typu MIMD^a je ...

^aMultiple Instruction Multiple Data

```
Procesor typu MIMD\footnote
{Multiple Instruction
Multiple Data} je \ldots
```

5.9 Prostředí v \LaTeX u

Prostředí slouží ke sazbě specifických částí textu, jako jsou například výčty, tabulky, obrázky, nebo sazba centrovaného textu. Každé prostředí je uvozeno příkazem `\begin{prostředí}` a ukončeno příkazem `\end{prostředí}`, kde parametr prostředí představuje název prostředí. Jednotlivá prostředí musí mohou být do sebe vnořována, nesmí se však křížit.

Nyní budeme text sázet tak, že bude zarovnaný doprava.

```
\begin{flushright}
Nyní budeme text sázet tak,
že bude zarovnaný doprava.
\end{flushright}
```

5.9.1 Výčty

Výčty používáme pro strukturování textu tak, že jednotlivé položky vyjmenujeme a případně stručně, obvykle v rozsahu několika vět rozvedeme. V \LaTeX u máme k dispozici tyto výčtová prostředí: `itemize`, pro sazbu jednoduchých výčtů s odrážkami, `enumerate` pro sazbu výčtu s číslovanými položkami a `description` pro sazbu výčtů s následným popisem každé položky.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

1. Procedurální	<code>\begin{enumerate}</code>
• Pascal	<code>\item Procedurální</code>
• Fortran	<code>\begin{itemize}</code>
	<code>\item Pascal</code>
	<code>\item Fortran</code>
2. Objektivě orientované	<code>\end{itemize}</code>
• Java	<code>\item Objektivě orientované</code>
• C#	<code>\begin{itemize}</code>
	<code>\item Java</code>
	<code>\item C\#</code>
	<code>\end{itemize}</code>
	<code>\end{enumerate}</code>

- Bubble sort	<code>\begin{itemize}</code>
- Insertion sort	<code>\item[-]Bubble sort</code>
- Shell sort	<code>\item[-]Insertion sort</code>
	<code>\item[-]Shell sort</code>
	<code>\end{itemize}</code>

Paměť je zařízení, které umožňuje ukládání a čtení informací.	<code>\begin{description}</code>
Procesor je ústřední výkonnou jednotkou počítače, která čte z paměti instrukce a následně je vykonává.	<code>\item[Paměť] je zařízení, které umožňuje ukládání a čtení informací.</code>
	<code>\item[Procesor] je ústřední výkonnou jednotkou počítače, která čte z paměti instrukce a následně je vykonává.</code>
	<code>\end{description}</code>

5.9.2 Zarovnávání textu

Pro sazbu zarovnaného textu máme v \LaTeX prostředí `flushleft`, které sází text zarovnaný doleva, `flushright`, které zarovnává text vpravo a prostředí `center`, která zarovnává text na střed.

Text je možné zarovnávat doleva, doprava a na také střed.	<code>\begin{flushleft}</code>
	Text je možné zarovnávat doleva, doprava a na také střed.
	<code>\end{flushleft}</code>

Text je možné zarovnávat doleva, doprava a na také střed.	<code>\begin{flushright}</code>
	Text je možné zarovnávat doleva, doprava a na také střed.
	<code>\end{flushright}</code>

Text je možné zarovnávat doleva, doprava a na také střed.	<code>\begin{center}</code>
	Text je možné zarovnávat doleva, doprava a na také střed.
	<code>\end{center}</code>

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

5.9.3 Přímý výstup textu

Pro sazbu textu, který nemá být nijak zpracován \TeX em, ale pouze vysázen tak, je zapsán ve zdrojovém souboru se používá prostředí `verbatim`. Pokud chceme, aby se místo mezer sázel symbol vanička, pak použijeme prostředí `verbatim*`. Tato prostředí jsou zejména vhodná pro sazbu zdrojových kódů. Protože text v prostředí `verbatim`, resp. `verbatim*`, není \TeX em nijak zpracován, nedochází ani k zalamování řádků. Je tedy třeba dbát na to, aby nedocházelo k přetékaní řádků, jsou-li řádky příliš dlouhé.

```
(defun factorial (n)
  (if (<= n 1)
      1
      (* n (factorial (- n 1)))))

\begin{verbatim}
(defun factorial (n)
  (if (<= n 1)
      1
      (* n (factorial (- n 1)))))
\end{verbatim}

ab aab bcde

\begin{verbatim*}
ab aab bcde
\end{verbatim*}
```

Pokud chceme použít přímý výstup pouze pro slovo nebo několik slov na řádku, můžeme použít příkaz `\verb+text+`, kde `+` je oddělovací znak. Místo `+` můžeme použít libovolný jiný oddělovací znak, kromě písmen, znaku `*` a mezery. Podobně jako u prostředí `verbatim`, existuje i u příkazu `\verb+text+` ohvězdičková verze.

Funkce <code>rotate</code> očekává jako svůj argument funkci <code>+</code> .	Funkce <code>\verb+rotate+</code> očekává jako svůj argument funkci <code>\verb + </code> .
---	---

5.9.4 Sazba tabulek

Pro sazbu tabulek je v \LaTeX u určeno prostředí `tabular`. Můžeme nastavovat také horizontální a vertikální linky, které se mají v tabulce vysázet. Šířka sloupce je stanovována automaticky, ale lze ji v případě potřeby ručně nastavit. Prostředí `tabular` zpracovává argument, ve kterém jsou specifikovány parametry sloupců po řadě zleva doprava.

Každý sloupec má udáno zarovnání: `l` znamená zarovnání sloupce vlevo, `r` znamená zarovnání sloupce vpravo a `c` znamená sloupec s centrováním textem. Parametr `p{šířka}` je určen k ručnímu nastavení šířky sloupce s víceřádkovým textem zarovnaným vlevo. Před nebo za specifikací sloupce může být umístěn znak „svíslá čára“ (`|`), který znamená, že se má zleva, resp. zprava sloupce sázet vertikální čára.

Formátování obsahu samotné tabulky probíhá následovně. Hodnoty buněk se zadávají po řádcích, jednotlivé sloupce jsou odděleny znakem `&`. Přejod na další řádek tabulky provedeme řádkovým zlomem, tedy příkazem `\\`. V případě, že se má vysázet vertikální čára, použijeme příkaz `\hline`.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

dec	hex	význam znaku
9	0x9	tabulátor
10	0xA	posuv o řádek (LF)
12	0xC	posuv o stránku (FF)
13	0xD	návrat vozíku (CR)
27	0x1B	escape
32	0x20	mezera

```
\begin{tabular}{|r|r|l|}
\hline
dec & hex & význam znaku\\
\hline \hline
9 & 0x9 & tabulátor\\
10 & 0xA & posuv o řádek (LF)\\
12 & 0xC & posuv o stránku (FF)\\
13 & 0xD & návrat vozíku (CR)\\
27 & 0x1B & escape\\
32 & 0x20 & mezera\\
\hline
\end{tabular}
```

Make commonly used controls larger and distinctive
Use the edges and corners of the screen to make your controls virtually infinite
Never, ever put controls 1 pixel away from a screen edge or corner

```
\begin{tabular}{|p{5cm}|}
\hline
Make commonly used controls larger
and distinctive \\ \hline
Use the edges and corners of the
screen to make your controls
virtually infinite\\ \hline
Never, ever put controls 1 pixel
away from a screen edge or
corner\\
\hline
\end{tabular}
```

Zarovnávaní tabulek se provádí pomocí prostředí pro sazbu na praporek, viz podkapitola 5.9.2. Popisky tabulek se vytvářejí pomocí tzv. plovoucích prostředí, která probereme v následující části.

5.10 Vkládání obrázků

Vkládání obrázků do textu je v \LaTeX realizováno rozšiřujícími balíčky. V této části si popíšeme základní použití balíčku `graphicx`. Připomeňme, že tento balíček je třeba deklarovat v hlavičce zdrojového souboru.

```
\usepackage{graphicx}
```

Již víme, že zdrojový soubor \LaTeX je obyčejný textový soubor a tak nás nepřekvapí, že vkládání obrázků do sázeného textu funguje souborově. V balíčku `graphicx` pro vložení obrázku slouží příkaz `\includegraphics`.

```
\includegraphics[parametry]{soubor}
```

Argument *soubor* představuje cestu k souboru s obrázkem. Název souboru můžeme uvést buď s příponou nebo bez přípony. To má svůj smysl. Záleží totiž na tom, zda zdrojový soubor překládáme \LaTeX em, nebo pdf\LaTeX em. \LaTeX ve spojení s balíčkem `graphicx` umožňuje vkládat obrázky ve formátu `eps`, zatímco ve při překládu pdf\LaTeX em můžeme pomocí balíčku `graphicx` vkládat obrázky ve formátech `png`, `pdf`, `jpg`, `tif` a `meps`. Tedy je patrné že formáty obrázků jsou mezi \LaTeX em a pdf\LaTeX em navzájem nekompatibilní. Pokud však přesto potřebujeme vytvořit zdrojová soubor, který bude možné přeložit oběma programy, pak musíme; za prvé: mít každý obrázek na disku ve formátu `eps` a v některém z formátů pro pdf\LaTeX a za druhé: neuvádět u názvu souboru příponu, protože tu si balíček `graphicx` doplní podle toho, překládá-li se \LaTeX em nebo pdf\LaTeX em.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací. Argument nazvaný *parametry* umožňuje specifikovat dodatečné volby pro obrázek. Jednotlivé parametry s příslušnými hodnotami oddělenými znakem „rovnítko“ jsou navzájem odděleny znakem „čárka“. Nejpoužívanější parametry jsou následující.

angle pro natočení obrázku o zadaný počet stupňů

height pro nastavení výšky obrázku

width pro nastavení šířky obrázku

scale pro nastavení měřítka obrázku

Vložení obrázku může tedy vypadat následovně:

```
\includegraphics[scale=0.4, angle=90]{window_screenshot.pdf}
```

Pro podrobný návod k balíčku `graphicx` odkazujeme na [4].

5.11 Plovoucí objekty

Plovoucí objekty, v našem případě tabulky a obrázky, jsou takové objekty o jejichž přesném umístění rozhodne systém \LaTeX tak, aby byla plocha jednotlivých stránek efektivně využita. Pokud by se objekty sázely pouze na místa kam byly ve zdrojovém textu vloženy, pak by mohlo dojít k například k tomu, že budou části stránek zůstat prázdné, protože na text nezbylo dost místa. Jak název *plovoucí objekty* naznačuje, může se stát že se obrázek nebo tabulka objeví dokonce na jiné stránce než by odpovídalo místu ve zdrojovém souboru. Umisťování objektů \LaTeX em lze do značné míry ovlivnit pomocí specifikace parametru prostředí. V \LaTeX u máme dvě prostředí pro plovoucí objekty: `table` pro tabulky a `figure` pro obrázky. Tato prostředí tedy neslouží pro vkládání plovoucích objektů, ale pouze pro jejich automatické umisťování na vhodné místo. Pro vkládání tabulek máme prostředí `tabular` a pro vkládání obrázků máme například příkaz `\includegraphics` balíčku `graphicx`.

```
\begin{table}[specifikace umístění]
```

```
...
```

```
\end{table}
```

```
\begin{figure}[specifikace umístění]
```

```
...
```

```
\end{figure}
```

Volitelný parametr specifikace umístění slouží k nastavení priorit pro umístění obrázku. Můžeme specifikovat i více hodnot pro umístění, takže pokud první požadavek nelze splnit, pokusí se \LaTeX splnit druhou preferenci a tak dále. Nejvyšší prioritu má hodnota nejvíce vlevo, směrem doprava se priorita požadavku snižuje. Možné hodnoty pro specifikaci požadavků na umístění plovoucího objektu jsou uvedeny v tabulce 1. Pokud nespecifikujeme požadavky na umístění plovoucího objektu, pak \LaTeX bude mít priority `[tbp]`.

Níže uvedený příklad tedy znamená, že preferujeme umístění tabulky na místě, kde byl ve zdrojovém souboru definován, pokud to nebude možné, pak chceme obrázek umístit v horní části stránky.

```
\begin{table} [!ht]
```

```
...
```

```
\end{table}
```

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

hodnota	význam
h	(here) objekt umístit tam, kde je uveden ve zdrojovém souboru
t	(top) objekt umístit v horní části stránky
b	(bottom) objekt umístit v dolní části stránky
p	(page) objekt umístit na samostatné stránce
!	ignorovat některá pravidla sazby

Tabulka 1: Parametry pro umístování plovoucích objektů

Plovoucím objektům je také možné definovat popisné titulky, které se vysází společně s číslem plovoucího objektu a s popisem *Tabulka* pro prostředí `table`, resp. *obrázek* pro prostředí `figure`. Odkaz na tabulku se definuje pomocí již známého příkazu `\label`, který byl popsán v kapitole 5.7.

```
\begin{table}
\begin{center}
\begin{tabular}{c|l}
hodnota & význam \\ \hline
h & (here) objekt umístit tam, kde je uveden ve zdrojovém
souboru \\
t & (top) objekt umístit v horní části stránky \\
b & (bottom) objekt umístit v dolní části stránky \\
p & (page) objekt umístit na samostatné stránce \\
! & ignorovat některá pravidla sazby \\
\end{tabular}
\caption{Parametry pro umístování plovoucích objektů}
\label{float-prefs}
\end{table}
\end{center}
```

Plovoucí objekty se ukládají postupně do zásobníku. Jakmile \LaTeX nalezne místo pro umístění, vybere objekt ze zásobníku. Může se stát, že se \LaTeX u dlouho nebude dařit najít vhodné umístění a objekty budou sázeny příliš daleko od místa, kde byly definovány. V takovém případě můžeme \LaTeX u přikázat at' okamžitě vyprázdní zásobník a umístí objekty. Toto provedeme příkazem `\clearpage`, resp. `\cleardoublepage`, který navíc po umístění plovoucích objektů začne sázet na nové stránce. Použití výše popsaného mechanismu ovšem představuje krajní řešení situace a neměli bychom se k němu uchylovat zbytečně.

Pomocí příkazu `\listoffigures` lze automaticky vygenerovat seznam obrázků a příkazem `\listoftables` lze automaticky vygenerovat seznam tabulek.

5.11.1 Definice nových příkazů a nových prostředí

\LaTeX je do jisté míry programovatelný. Pomocí příkazu `\newcommand` můžeme definovat nový příkaz. Pomocí příkazu `\newenvironment` můžeme definovat nové prostředí. Podrobnější popis těchto příkazů je nad rámec tohoto textu, proto odkazujeme na [4].

5.12 Sazba matematických výrazů

\LaTeX nabízí mimořádně rozsáhlou paletu možností při sazbě matematických výrazů a kvalitou výstupu sazby matematiky jen těžko najde srovnání. Adekvátní popis sazby

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací. matematiky by vydal na samostatnou publikaci. My se mu však budeme naopak věnovat jen velmi povrchně, protože jednak v bakalářských pracech z oblasti aplikací informatiky příliš mnoho matematické sazby nebývá a druhak v případě, že v práci chceme sázet větší množství matematických výrazů, budeme nuceni sáhnout po komplexnějším popisu, než je uvádíme v této části textu. Pro podrobnější popis odkazujeme na [4].

Matematika se sází ve zvláštním módu. Pro sazbu v rámci řádku textu se kód uzavírá mezi znaky $\$$. Pro samostatnou sazbu matematiky jsou k dispozici kromě jiného například prostředí `displaymath`, nebo `equation`.

Platí vztah $d = a * b * c$, což je zřejmé.

Platí vztah $\$d=a*b*c\$,$ což je zřejmé.

$$d = a * b * c$$

```
\begin{displaymath}
d=a*b*c
\end{displaymath}
```

Horní index se sází pomocí znaku \wedge , dolní index znakem $_$. Odmocnina se sází příkazem `\sqrt`, resp. `\sqrt[n]`, kde n je n -tá odmocnina.

$$e^{x^2} \neq e^{x^2}$$

```
\begin{displaymath}
e^{x^2} \neq {e^x}^2
\end{displaymath}
```

$$\sqrt{x} + \sqrt[3]{y}$$

```
\begin{displaymath}
\sqrt{x} + \sqrt[3]{y}
\end{displaymath}
```

Zlomky se sází příkazem `\frac{čitatel}{jmenovatel}`. Suma se sází příkazem `\sum_{dolní-index}{horní-index}{výraz sumy}`.

$$\frac{1}{2} + \frac{1}{3} + \frac{1}{5} \dots$$

```
\begin{displaymath}
\frac{1}{2}+\frac{1}{3}+\frac{1}{5}
\ldots
\end{displaymath}
```

$$\sum_{i=0}^k x_i 2^i$$

```
\begin{displaymath}
\sum_{i=0}^k x_i 2^i
\end{displaymath}
```

5.13 Sazba seznamu literatury a citace

Tvorba seznamu literatury se provádí v prostředí `thebibliography`. Vkládání položek se provádí příkazem `\bibitem{značka}`, kde *značka* označuje jedinečné označení položky literatury. Pomocí této značky a příkazu `\cite{značka}` můžeme v textu literaturu citovat. Protože L^AT_EX používá k sazbě seznamu literatury pomocné soubory (podobně jako u obsahu) je třeba zdrojový soubor přeložit opakovaně.

Způsob citování děl záleží na druhu díla. Uvedeme nejpoužívanější případy.

Citace jednosvazkového díla má následující strukturu:

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

- Příjmení a jméno autora. Jsou-li autoři dva nebo tři, pak se uvádějí všichni autoři a oddělují se čárkou. Je-li autorů více, pak se uvádí pouze jméno prvního autora a „kol.“. Tituly a akademické hodnosti autorů se neuvádějí.
- Název. Píše se v jazyku citované publikace.
- Vydání.
- Nakladatelské údaje. Uvádíme se místo a nakladatel. Je-li známo místo vydání, pak se uvede i toto místo před nakladatelem oddělené dvojtečkou.
- Rok vydání.
- ISBN

Citace článku v časopise má následující strukturu:

- příjmení a jméno autora nebo autorů
- název článku
- název časopisu
- ročník (svazek)
- číslo
- rok vydání
- strany
- ISSN

Citace článku ve sborníku má následující strukturu:

- příjmení a jméno autora nebo autorů
- název článku
- název sborníku za slovem In:
- místo vydání
- nakladatel
- rok vydání
- v případě periodického sborníku ročník (svazek)
- strany
- ISBN

Citace s uvedením zdroje se používá, pokud zdroj dokumentu je jiný než papírová podoba, například CD-ROM nebo webová stránka. V takových případech doplníme obvyklou citaci o zdroj. Ten se uvádí za názvem v hranatých závorkách.

Pokud citujeme webovou stránku nebo jiný zdroj, který se může měnit v čase, pak uvedeme zdroj [online] a doplníme informaci o datum a čas vzniku stránky (je-li uveden) a datum a čas, kdy byl dokument citován.

Podrobněji je problematika evoluce zpracována v [1] a [2]. Citovat můžeme i více [3, 4] děl najednou.

Podrobněji je problematika evoluce zpracována v `\cite{goldberg}` a `\cite{marik}`. Citovat můžeme i více `\cite{kriz, mcgrenerere}` děl najednou.

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

Reference

- ```
\begin{thebibliography}{}
\bibitem{goldberg}Goldberg, D. :
Genetic Algorithms in Search,
Optimization and Machine
Learning, Addison-Wesley
Longman Publishing, 1989,
ISBN 0201157675.
\bibitem{marik}Mařík V. a kol. :
Umělá inteligence 3, Academia,
2001, ISBN 8020004726.
\bibitem{kriz}Kriz, S., Hegarty, M.:
Top-down and bottom-up
influences on learning from ani-
mations. Int. J. Hum.-Comput. Stud. 65, 11,
2007, 911-930.
\bibitem{mcgrenere}McGrenere, J.,
Moore, G. Are we all in the same
"bloat"? In: Proceedings of Gra-
phics Interface, Lawrence Erlbaum Associ-
ates, Montreal, Quebec, Canada. 2000, 187-
196.
\bibitem{jerabkova}
Jeřábková, P.: Komunistická strana propa-
guje totéž, za co soud zrušil KSM [online],
dostupné z http://zpravy.idnes.cz/
2008-12-05, 13:11 [citováno 2008-19-05,
12:22].
\end{thebibliography}
```
- [1] Goldberg, D. : Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing, 1989, ISBN 0201157675.
- [2] Mařík V. a kol. : Umělá inteligence 3, Academia, 2001, ISBN 8020004726.
- [3] Kriz, S., Hegarty, M.: Top-down and bottom-up influences on learning from animations. Int. J. Hum.-Comput. Stud. 65, 11, 2007, 911-930.
- [4] McGrenere, J., Moore, G. Are we all in the same "bloat"? In: Proceedings of Graphics Interface, Lawrence Erlbaum Associates, Montreal, Quebec, Canada. 2000, 187-196.
- [5] Jeřábková, P.: Komunistická strana propaguje totéž, za co soud zrušil KSM [online], dostupné z <http://zpravy.idnes.cz/2008-12-05, 13:11> [citováno 2008-19-05, 12:22].



Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

## 6 Jak na prezentaci

Prezentace je vhodným doplňkem vystoupení na obhajobě. Na jedné straně vám poslouží jako vodítko při vaší řeči a zefektivní její průběh, na druhé straně se bude přítomným, pro které prezentujete, lépe vaše vystoupení sledovat. Prezentace je vhodným prostředníkem mezi přednášejícím a posluchači. U bakalářské práce by se prezentace měla zaměřit na její klíčové části a to zejména:

- cíle práce
- způsob řešení
- dosažené výsledky

Nyní uvedeme souhrn doporučení pro úspěšnou prezentaci.

1. Texty v prezentaci musí být stručné. Měly by být formulovány heslovitě a vyjadřovat klíčkové pojmy vzhledem k aktuálnímu tématu. Prezentace není určena ke čtení souvislého textu.

### Průvodce studiem

Platí pravidlo 5 x 5. Každý snímek by měl obsahovat maximálně pět bodů a každý bod maximálně pět slov.

2. Prezentace je dobrý sluha, ale špatný pán. Přednášející musí znát strukturu prezentace a musí být tím kdo prezentuje — prezentace je doplňkem vystoupení. Je zcela nevhodné pokud přednášející jednotlivé body čte a ještě horší je, pokud je nijak dále nerozvede. Texty v prezentaci slouží jako témata, která přednášející dále rozebere.
3. Čas běží při prezentaci rychleji. Vystoupení je dobré si předem připravit. Obhajoba je časově omezená a pokud si řeč předem připravíte, můžete ušetřit mnoho času. Formulace budou stručnější a výstižnější. Pozor však, abyste vystoupení neproměnili ve „recitaci na školní besídce“. Přebytek informací nevyřešíte zrychlením své mluvy. Osobnost a verbální projev přednášejícího jsou velmi důležité.
4. Používejte bezpatková písma. Bezpatková písma (viz obrázek 1) jsou na větší vzdálenost lépe čitelná, působí lehčím dojmem než písma patková. Naopak, u souvislého textu se častěji používají písma patková. Například tento učební text je sázen písmem Palatino. Příklad patkových písem uvádíme na obrázku 2.

## Bezpatkové písmo Helvetica Bezpatkové písmo Lucida Grande

Obrázek 1: Bezpatková písma

## Patkové písmo Palatino Patkové písmo Times New Roman

Obrázek 2: Patková písma

### **Průvodce studiem**

Bezpatková písma (označovaná též jako jako sans-serifová) jsou v běžném tištěném textu obvykle hůře čitelná. Je-li ovšem text malý nebo naopak velký, pak jsou bezpatková písma mnohem vhodnější než patková.

5. Na velikosti záleží. Vezměte v úvahu v jaké vzdálenosti budou posluchači a jak velká je projekční plocha, aby vaše prezentace byla dobře čitelná. Obecně řečeno, použitá písma by měly mít velikost alespoň 20 bodů.
6. Respektujte tok textu a informací. V euroamerickém kulturním prostředí se čte zleva-doprava a shora-dolů. Tomu by také měla odpovídat struktura snímku prezentace. Oko při rychlé orientaci v textu preferuje směr vertikální před horizontálním. Struktura snímku také musí respektovat tok informací od přenášejícího. To znamená, že řeč by měla postupovat po jednotlivých bodech. Klíčovou informaci v rámci snímku je vhodné uvést na jeho konci — postupujeme jakoby od argumentů k tvrzení.
7. Tučně a barevně. Chcete-li nějaké slovo zvýraznit je vhodné použít tučný řez nebo jinou barvu. Není dobré používat kurzívu, protože působí neostře a hůř se čte. U barev je třeba dbát na střídmost a čitelnost. Může se stát, že sál nebude dostatečně zatemněný a při použití tlumených barev bude vaše prezentace nečitelná. Barvy musí být syté. Čitelnost napomáhá kontrast: světlé pozadí — tmavý text nebo naopak.
8. Animace. Využití animací a přechodových efektů je potřeba volit s rozvahou. Použití musí být vždy funkční. Animace a přechodové efekty nikdy nepoužíváme na okrasu. Nepoužívejte ke stejnému účelu různé efekty.
9. Obrázky. Výklad mohou velmi efektivně doplnit grafy, obrázky, schémata a podobě. Grafické objekty obvykle zaberou značnou část snímku a není vhodné je doplňovat větším množstvím textu.
10. Šablony. Aplikace pro tvorbu prezentací obsahují předpřipravené šablony. Vyberte si takovou, která se vizuálně hodí k tomu co chcete prezentovat (je rozdíl, pokud ukazujeme známým prezentaci o vaší dovolené nebo obhajujete bakalářskou práci) a respektujte její strukturu. Každá šablona má předdefinované typy snímků (například nadpis bez textu, nadpis a text, nadpis + text vlevo + obrázek vpravo ...).

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

## A Vzor titulních stran

|                                               |                |
|-----------------------------------------------|----------------|
| UNIVERZITA PALACKÉHO V OLOMOUCI               |                |
| PŘÍRODOVĚDECKÁ FAKULTA<br>KATEDRA INFORMATIKY |                |
| BAKALÁŘSKÁ PRÁCE                              |                |
| rok                                           | Jméno Příjmení |

|                                               |                |
|-----------------------------------------------|----------------|
| UNIVERZITA PALACKÉHO V OLOMOUCI               |                |
| PŘÍRODOVĚDECKÁ FAKULTA<br>KATEDRA INFORMATIKY |                |
| Název bakalářské práce                        |                |
| rok                                           | Jméno Příjmení |

|                                                             |                     |
|-------------------------------------------------------------|---------------------|
| Prohlašuji, že jsem bakalářskou práci vypracoval samostatně |                     |
| datum                                                       | vlastnoruční podpis |

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

Děkuji .... (tato část je nepovinná)

#### **Anotace**

*V rozsahu jednoho odstavce uveďte popis své bakalářské práce, čemu je práce věnována, co v ní čtenář nalezne. Anotace by neměla přesáhnout 10 řádků.*

Obsah

Seznam obrázků

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

## Seznam tabulek

*Nyní následuje text samotné práce, kapitoly a podkapitoly se číslují, a to do úrovně 3 (např. 3.1.2.)  
Pro psaní textu použijte zásadně patkový font (Palatino, popř. Times, nebo jiný vhodný font),  
zdrojové kódy, názvy souborů, internetové odkazy sdějte strojopisným řezem, například fontem  
Courier (například [www.inf.upol.cz](http://www.inf.upol.cz)).*

## Příloha 1 (pouze pokud máte přílohy)

## Závěr

*Zde stručně zhodnoťte svou práci, co jste se naučili, jaké jsou její přínosy, komu bude sloužit, je-li  
nasazena v praxi a případně jaký bude její další vývoj atd.*

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

### Seznam literatury

- [1] Smith, John. User and program. Publisher, City, 1990.
- [2] Kovář, Jan. Jak programovat. Nakladatelství Město, 1990.

*Na vnitřní stranu zadní obálky vlepíte CD v papírovém přebalu s textem práce ve formátu PDF, zdrojovými kódy a samotnou aplikaci včetně instalátoru, pokud jej obsahuje. Je vhodné připojit soubor readme.txt s popisem obsahu CD*

Toto je pracovní a nedokončená verze učebního textu Průvodce bakalářskou prací.

## Reference

- [1] Boldiš, P. : Bibliografické citace dokumentů podle ČSN ISO 690 a ČSN ISO 690-2 [online], dostupné z [www.boldis.cz/citace/citace2.pdf](http://www.boldis.cz/citace/citace2.pdf) [citováno 2008-19-05, 18:49].
- [2] Dostál M. : Základy tvorby uživatelského rozhraní, elektronický učební text, 2008.
- [3] Page-Jones, M. : Základy objektově orientovaného návrhu v UML, Grada, 2001.
- [4] Rybička, J. :  $\text{\LaTeX}$  pro začátečníky, 3 vydání, Brno:Konvoj a CSTUG, 2003, ISBN 80-7302-049-1.
- [5] Parlament ČR : Zákon 121/2000 Sb. Zákon o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).