



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**WEBOVÝ PROHLÍŽEČ PANORAMATICKÝCH SNÍMKŮ**

WEB BROWSER OF PANORAMIC IMAGES

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**TOMÁŠ SLUNSKÝ**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Doc. Ing. MARTIN ČADÍK, Ph.D.**

**BRNO 2017**

## **Abstrakt**

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v českém (slovenském) jazyce.

## **Abstract**

Do tohoto odstavce bude zapsán výtah (abstrakt) práce v anglickém jazyce.

## **Klíčová slova**

Sem budou zapsána jednotlivá klíčová slova v českém (slovenském) jazyce, oddělená čárkami.

## **Keywords**

Sem budou zapsána jednotlivá klíčová slova v anglickém jazyce, oddělená čárkami.

## **Citace**

SLUNSKÝ, Tomáš. *Webový prohlížeč panoramatických snímků*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Čadík Martin.

# Webový prohlížeč panoramatických snímků

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana X... Další informace mi poskytli... Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Tomáš Slunský  
10. dubna 2017

## Poděkování

V této sekci je možno uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc (externí zadavatel, konzultant, apod.).

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Webové technologie prohlížeče</b>	<b>3</b>
2.1	HTML5 . . . . .	3
2.1.1	Video . . . . .	3
2.1.2	Plátno . . . . .	6
2.2	WebGL . . . . .	7
2.3	GLSL . . . . .	8
2.3.1	Vertex shader . . . . .	8
2.3.2	Fragment shader . . . . .	8
<b>3</b>	<b>Nikdy to nebude naprosto dokonalé</b>	<b>9</b>
<b>4</b>	<b>Typografické a jazykové zásady</b>	<b>10</b>
4.1	Co to je normovaná stránka? . . . . .	11
<b>5</b>	<b>Závěr</b>	<b>13</b>
	<b>Literatura</b>	<b>14</b>
	<b>Přílohy</b>	<b>15</b>
<b>A</b>	<b>Jak pracovat s touto šablonou</b>	<b>16</b>

# Kapitola 1

## Úvod

V současné době se technologie ve všech odvětvích stále posouvá a není tomu jinak ani u prohlížení fotek a videí. V současné době i takto nahrané informace pomocí digitálních zařízení, je trend dále posouvat a zvyšovat tak zážitek ze zaznamenané události. Díky tomuto trendu si už nevystačíme s “klasickými” přehrávači, popř. prohlížeči videí, nebo fotografií. Díky sférickým 360 stupňovým kamerám, je dnes možné zaznamenat video o srovnatelné velikosti jako u dnes běžného telefonu, ale s mnohem větším objemem informací, které se ale nedají srozumitelně přehrát v klasických přehrávačích. Ruku v ruce se sdílením takových dat, je velice výhodné takový prohlížeč nabídnout jako webovou verzi, aby i jiní uživatelé mohli svá takto natočená videa přehrávat, popř. sdílet.

Následující práce se věnuje implementaci webového prohlížeče panoramatických snímků a videí v různých modech, ve kterých je video interpretováno. Cílem je tedy navrhnout a zrealizovat řešení, aby uživatel po natočení videa dále nepotřeboval k přehrání např. sférického videa v režimu rybího oka specifický program, který by video musel nejprve překonvertovat. To stejné se týká i panoramatických snímků a videí v equirectangulárním modu.

Další inovací v prohlížení videí je přidání některých dat, které v běžném prohlížení není k dispozici. Např. informace o světových stranách ve vztahu k obrázku, či videu popřípadě údaje o velikosti náhledu v daném kontextu prohlížení až po dodatečná metadata v panoramatických obrázcích, které jsou vhodné např. k tomu, aby autor obrázku mohl popsat jeho zajímavé části popřípadě blíže popsat zachycenou scénu.

## Kapitola 2

# Webové technologie prohlížeče

### 2.1 HTML5

Jazyk HTML (HyperText Markup Language) je značkovací jazyk určený pro popis webových stránek. Vychází z univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). V současné době aktuální verzi jazyka HTML je jeho již pátá verze - HTML5. Nová verze HTML přináší zásadní vylepšení, nové funkce a možnosti, které jsou nezbytné pro návrh a implementaci webového prohlížeče.

Pro návrh samotného programu je nezbytná podpora multimédií – tedy audio a video, v neposlední řadě také plátno canvas sloužící pro práci s grafikou.

#### 2.1.1 Video

Dříve nebylo možné do webových stránek vložit video tak jako dnes. K tomuto účelu bylo využíváno různých zásuvných modulů třetích stran a do webových stránek se video vkládalo např. jako objekt. Nejvíce rozšířeným přehrávačem videí a tedy jakási náhrada za podporu videí, kterou tehdy HTML nemělo, se v širším spektru stal Adobe Flash, který funkci přehrávače plní v menší míře až doposud, avšak je již zastíněn efektivnějším řešením, a to právě HTML5.

Nový prvek v HTML5 vytvořený k tomuto účelu je `<video>`. Byl navržen tak, aby mohl být použit bez detekčních skriptů na stránce. V elementu videa je možné nastavit více souborů s videem a dle podpory si daný prohlížeč vybere jim podporované video. V případě, že by prohlížeč prvek videa nepodporoval, bude jej ignorovat. Nastavení více zdrojů videa s odlišnými kodeky lze pomocí elementu `<source>` uvnitř páru elementů `<video>...</video>`. Jakmile prohlížeč narazí na `<video>`, podívá se, zdali je přítomen atribut `src`, v opačném případě začne procházet jeden element `<source>` po druhém a bude hledat právě takový, který umí přehrát.

Aby bylo možné ovládat video dynamicky pomocí kláves nebo myši, bude nutné využít DOM (Document Object Model). Jedná se o stromovou strukturu dokumentu, kterou si prohlížeč sestavuje po načtení webové stránky. Některé značky se v DOM vytvoří, aniž by byly ve zdrojovém kódu zapsány. Obecně platí, že pomocí objektového modelu je díky javascriptu možné tuto stromovou strukturu dále upravovat a rozšiřovat.

Další velice důležitou částí je načítání videa a jeho ověření, zdali je již video připraveno k přehrání. K ověření dostupnosti videa lze použít jeden ze síťových stavů elementu pomocí metody `networkState()`, popřípadě přímo zjišťovat připravenost videa pomocí metody `readyState()`, vracející jeden z následujících stavů, podle kterých se můžeme dále při přehrávání řídit a přizpůsobit tomu běh programu. Jednotlivé konstanty nabývají hodnot od 0 do 4 a dle hodnot rozlišujeme:

- **HAVE\_NOTHING** (ekvivalentní hodnotě 0)
  - nastane v situaci kdy zdrojové video není dostupné, nebo žádná data pro aktuální přehrávanou pozici
  - koresponduje také s návratovou hodnotou síťové metody `networkState()`, když její návratová hodnota je rovna 0, což odpovídá konstantě `NETWORK_EMPTY`
- **HAVE\_METADATA**
  - nějaké data se již podařilo získat a zdroj je tedy považován jako dostupný, ale zatím ještě není dostatek dat, aby bylo možné začít s přehráváním.
- **HAVE\_CURRENT\_DATA**
  - data pro bezprostřední začátek přehrávání jsou připravena, ale video ještě není načteno dál za tuto pozici. Přehrávání může dostat do stavu `HAVE_METADATA`, nebo následující data videa již nejsou k dispozici.
- **HAVE\_FUTURE\_DATA**
  - data pro bezprostřední začátek přehrávání jsou připravena
- **HAVE\_ENOUGH\_DATA**

Důležité atributy pro práci s videem jsou uvedeny níže.

**src** - jedná se o atribut, do kterého se definuje zdrojové adresa videa jako URL, které se má zobrazit. Používá se zejména v situaci, kdy je video jen v jedné verzi.

**autoplay** - jedná se atribut typu `bool`. V případě, že je `true`, spustí se přehrávání média automaticky. Uvedení názvu atributu do elementu samotného je již dostačující informace o tom, co se bude dít s videem. Tedy bude tato hodnota typu `bool` považována jako `true`, jinak `false`

**loop** - jedná se atribut typu `bool`, který začne po dosažení konce videa s přehráváním videa opět od začátku.

**controls** - jedná se opět o atribut typu `bool`, který říká, aby prohlížeč použil vlastní ovládací prvky pro video. Ovládací prvky se dají také vytvořit a přizpůsobit velice snadno díky DOMu.

**preload** - tímto říkáme, jakou část videa by měl webový prohlížeč načíst ihned po načtení stránky. Tento atribut nabývá jednou ze tří hodnot:

- **none** - nebude načítat nic
- **metadata** - načte pouze metadata daného videa
- **auto** - sám zvolí, co přesně udělá

## **Track**

Pokud by video mělo více jazyků, je možné k videu přidat více jazykových stop, k tomuto účelu slouží element `<track>`.



### 2.1.2 Plátno

Jak již bylo avizováno výše, plátno, neboli element `<canvas>` slouží v HTML5 pro vykreslení grafů, herní grafiky, obrazů bitmap apod. Díky elementu `<canvas>` lze vykreslovat i náročnější grafické objekty za pomoci WebGL. V HTML5 slouží tedy především k vykreslení 2D prvků pomocí Javascriptu.

Plátno je bezpochyby nejdůležitější částí pro realizaci celé práce. Bude pro vyobrazení používat především `<video>`, z kterého bude číst data a využije tedy element video jak zdroj. Díky DOM pak dokážeme s objektem snadno manipulovat. Díky elementu `<canvas>` jsme tedy schopni získat kontext Javascriptového API - WebGL a díky němu schopni pracovat na úrovni grafické karty.

## 2.2 WebGL

Sem webgl

## **2.3 GLSL**

GLSL úvod

### **2.3.1 Vertex shader**

Vertex shader

### **2.3.2 Fragment shader**

Fragment shader

## Kapitola 3

# Nikdy to nebude naprosto dokonalé

Když jsme už napsali vše, o čem jsme přemýšleli, uděláme si den nebo dva dny volna a pak si přečteme sami rukopis znovu. Uděláme ještě poslední úpravy a skončíme. Jsme si vědomi toho, že vždy zůstane něco nedokončeno, vždy existuje lepší způsob, jak něco vysvětlit, ale každá etapa úprav musí být konečná.

## Kapitola 4

# Typografické a jazykové zásady

Při tisku odborného textu typu *technická zpráva* (anglicky *technical report*), ke kterému patří například i text kvalifikačních prací, se často volí formát A4 a často se tiskne pouze po jedné straně papíru. V takovém případě volte levý okraj všech stránek o něco větší než pravý – v tomto místě budou papíry svázány a technologie vazby si tento požadavek vynucuje. Při vazbě s pevným hřbetem by se levý okraj měl dělat o něco širší pro tlusté svazky, protože se stránky budou hůře rozevírat a levý okraj se tak bude oku méně odhalovat.

Horní a spodní okraj volte stejně veliký, případně potištěnou část posuňte mírně nahoru (horní okraj menší než dolní). Počítejte s tím, že při vazbě budou okraje mírně oříznuty.

Pro sazbu na stránku formátu A4 je vhodné používat pro základní text písmo stupně (velikosti) 11 bodů. Volte šířku sazby 15 až 16 centimetrů a výšku 22 až 23 centimetrů (včetně případných hlaviček a patiček). Proklad mezi řádky se volí 120 procent stupně použitého základního písma, což je optimální hodnota pro rychlost čtení souvislého textu. V případě použití systému LaTeX ponecháme implicitní nastavení. Při psaní kvalifikační práce se řiďte příslušnými závaznými požadavky.

Stupeň písma u nadpisů různé úrovně volíme podle standardních typografických pravidel. Pro všechny uvedené druhy nadpisů se obvykle používá polotučné nebo tučné písmo (jednotně buď všude polotučné nebo všude tučné). Proklad se volí tak, aby se následující text běžných odstavců sázel pokud možno na *pevný rejstřík*, to znamená jakoby na linky s předem definovanou a pevnou roztečí.

Uspořádání jednotlivých částí textu musí být přehledné a logické. Je třeba odlišit názvy kapitol a podkapitol – píšeme je malými písmeny kromě velkých začátečních písmen. U jednotlivých odstavců textu odsazujeme první řádek odstavce asi o jeden až dva čtverčíky (vždy o stejnou, předem zvolenou hodnotu), tedy přibližně o dvě šířky velkého písmene M základního textu. Poslední řádek předchozího odstavce a první řádek následujícího odstavce se v takovém případě neoddělují svislou mezerou. Proklad mezi těmito řádky je stejný jako proklad mezi řádky uvnitř odstavce.

Při vkládání obrázků volte jejich rozměry tak, aby nepřesáhly oblast, do které se tiskne text (tj. okraje textu ze všech stran). Pro velké obrázky vyčleňte samostatnou stránku. Obrázky nebo tabulky o rozměrech větších než A4 umístěte do písemné zprávy formou skládanky vřité do přílohy nebo vložené do záložek na zadní desce.

Obrázky i tabulky musí být pořadově očíslovány. Číslování se volí buď průběžné v rámci celého textu, nebo – což bývá praktičtější – průběžné v rámci kapitoly. V druhém případě se číslo tabulky nebo obrázku skládá z čísla kapitoly a čísla obrázku/tabulky v rámci kapitoly – čísla jsou oddělena tečkou. Čísla podkapitol nemají na číslování obrázků a tabulek žádný vliv.

Tabulky a obrázky používají své vlastní, nezávislé číselné řady. Z toho vyplývá, že v odkazech uvnitř textu musíme kromě čísla udát i informaci o tom, zda se jedná o obrázek či tabulku (například “... viz tabulka 2.7 ...”). Dodržování této zásady je ostatně velmi přirozené.

Pro odkazy na stránky, na čísla kapitol a podkapitol, na čísla obrázků a tabulek a v dalších podobných příkladech využíváme speciálních prostředků DTP programu, které zajistí vygenerování správného čísla i v případě, že se text posune díky změnám samotného textu nebo díky úpravě parametrů sazby. Příkladem takového prostředku v systému LaTeX je odkaz na číslo odpovídající umístění značky v textu, například návěští (`\ref{navesti}`) – podle umístění návěští se bude jednat o číslo kapitoly, podkapitoly, obrázku, tabulky nebo podobného číslovaného prvku), na stránku, která obsahuje danou značku (`\pageref{navesti}`), nebo na literární odkaz (`\cite{identifikator}`).

Rovnice, na které se budeme v textu odvolávat, opatříme pořadovými čísly při pravém okraji příslušného řádku. Tato pořadová čísla se píší v kulatých závorkách. Číslování rovnic může být průběžné v textu nebo v jednotlivých kapitolách.

Jste-li na pochybách při sazbě matematického textu, snažte se dodržet způsob sazby definovaný systémem LaTeX. Obsahuje-li vaše práce velké množství matematických formulí, doporučujeme dát přednost použití systému LaTeX.

Mezeru neděláme tam, kde se spojují číslice s písmeny v jedno slovo nebo v jeden znak – například *25krát*.

Členicí (interpunkční) znaménka tečka, čárka, středník, dvojtečka, otazník a vykřičník, jakož i uzavírací závorky a uvozovky se přimykají k předcházejícímu slovu bez mezery. Mezera se dělá až za nimi. To se ovšem netýká desetinné čárky (nebo desetinné tečky). Otevírací závorka a přední uvozovky se přimykají k následujícímu slovu a mezera se vynechává před nimi – (takto) a “takto”.

Pro spojovací a rozdělovací čárku a pomlčku nepoužíváme stejný znak. Pro pomlčku je vyhrazen jiný znak (delší). V systému TeX (LaTeX) se spojovací čárka zapisuje jako jeden znak “pomlčka” (například “Brno–město”), pro sázení textu ve smyslu intervalu nebo dvojic, soupeřů a podobně se ve zdrojovém textu používá dvojice znaků “pomlčka” (například “zápas Sparta – Slavie”; “cena 23–25 korun”), pro výrazné oddělení části věty, pro výrazné oddělení vložené věty, pro vyjádření nevyslovené myšlenky a v dalších situacích (viz Pravidla českého pravopisu) se používá nejdelší typ pomlčky, která se ve zdrojovém textu zapisuje jako trojice znaků “pomlčka” (například “Další pojem — jakkoliv se může zdát nevýznamný — bude neformálně definován v následujícím odstavci.”). Při sazbě matematického mínus se při sazbě používá rovněž odlišný znak. V systému TeX je ve zdrojovém textu zapsán jako normální mínus (tj. znak “pomlčka”). Sazba v matematickém prostředí, kdy se vzoreček uzavírá mezi dolary, zajistí vygenerování správného výstupu.

Lomítko se píše bez mezer. Například školní rok 2008/2009.

Pravidla pro psaní zkratk jsou uvedena v Pravidlech českého pravopisu [1]. I z jiných důvodů je vhodné, abyste tuto knihu měli po ruce.

## 4.1 Co to je normovaná stránka?

Pojem *normovaná stránka* se vztahuje k posuzování objemu práce, nikoliv k počtu vytištěných listů. Z historického hlediska jde o počet stránek rukopisu, který se psal psacím strojem na speciální předtištěné formuláře při dodržení průměrné délky řádku 60 znaků a při 30 řádcích na stránku rukopisu. Vzhledem k zápisu korekturních značek se používalo řádkování 2 (ob jeden řádek). Tyto údaje (počet znaků na řádek, počet řádků a proklad

mezi nimi) se nijak nevztahují ke konečnému vytištěnému výsledku. Používají se pouze pro posouzení rozsahu. Jednou normovanou stránkou se tedy rozumí  $60 \cdot 30 = 1800$  znaků. Obrázky zařazené do textu se započítávají do rozsahu písemné práce odhadem jako množství textu, které by ve výsledném dokumentu potisklo stejně velkou plochu.

Orientační rozsah práce v normostranách lze v programu Microsoft Word zjistit pomocí funkce *Počet slov* v menu *Nástroje*, když hodnotu *Znaky (včetně mezer)* vydělíte konstantou 1800. Do rozsahu práce se započítává pouze text uvedený v jádru práce. Části jako abstrakt, klíčová slova, prohlášení, obsah, literatura nebo přílohy se do rozsahu práce nepočítají. Je proto nutné nejdříve označit jádro práce a teprve pak si nechat spočítat počet znaků. Přibližný rozsah obrázků odhadnete ručně. Podobně lze postupovat i při použití OpenOffice. Při použití systému LaTeX pro sazbu je situace trochu složitější. Pro hrubý odhad počtu normostran lze využít součet velikostí zdrojových souborů práce podělený konstantou cca 2000 (normálně bychom dělili konstantou 1800, jenže ve zdrojových souborech jsou i vyznačovací příkazy, které se do rozsahu nepočítají). Pro přesnější odhad lze pak vyextrahovat holý text z PDF (např. metodou cut-and-paste nebo *Save as Text...*) a jeho velikost podělit konstantou 1800.

## Kapitola 5

### Závěr

Závěrečná kapitola obsahuje zhodnocení dosažených výsledků se zvlášť vyznačeným vlastním přínosem studenta. Povinně se zde objeví i zhodnocení z pohledu dalšího vývoje projektu, student uvede náměty vycházející ze zkušeností s řešeným projektem a uvede rovněž návaznosti na právě dokončené projekty.



# Literatura

- [1] Hlavsa, Z.; aj.: *Pravidla českého pravopisu*. Academia, 2005, ISBN 80-200-1327-X.
- [2] Rábová, Z.; Hanáček, P.; Peringer, P.; aj.: *Užitečné rady pro psaní odborného textu*. FIT VUT v Brně, Listopad 2008, [Online; navštíveno 12.05.2015].  
URL [http://www.fit.vutbr.cz/info/statnice/psani\\_textu.html](http://www.fit.vutbr.cz/info/statnice/psani_textu.html)
- [3] Rybička, J.: *L<sup>A</sup>T<sub>E</sub>X pro začátečníky*. Konvoj, 1999, ISBN 80-85615-77-0.

# Přílohy

# Příloha A

## Jak pracovat s touto šablonou

V této kapitole je uveden popis jednotlivých částí šablony, po kterém následuje stručný návod, jak s touto šablonou pracovat.

Jedná se o přechodnou verzi šablony. Nová verze bude zveřejněna do konce roku 2016 a bude navíc obsahovat nové pokyny ke správnému využití šablony, závazné pokyny k vypracování bakalářských a diplomových prací (rekapitulace pokynů, které jsou dostupné na webu) a nezávazná doporučení od vybraných vedoucích. Jediné soubory, které se v nové verzi změní, budou `projekt-01-kapitoly-chapters.tex` a `projekt-30-prilohy-appendices.tex`, jejichž obsah každý student vymaže a nahradí vlastním. Šablonu lze tedy bez problémů využít i v současné verzi.

### Popis částí šablony

Po rozbalení šablony naleznete následující soubory a adresáře:

**bib-styles** Styly literatury (viz níže).

**obrazky-figures** Adresář pro Vaše obrázky. Nyní obsahuje `placeholder.pdf` (tzv. TODO obrázek, který lze použít jako pomůcku při tvorbě technické zprávy), který se s prací neodevzdává. Název adresáře je vhodné zkrátit, aby byl jen ve zvoleném jazyce.

**template-fig** Obrázky šablony (znak VUT).

**fitthesis.cls** Šablona (definice vzhledu).

**Makefile** Makefile pro překlad, počítání normostran, sbalení apod. (viz níže).

**projekt-01-kapitoly-chapters.tex** Soubor pro Váš text (obsah nahradte).

**projekt-20-literatura-bibliography.bib** Seznam literatury (viz níže).

**projekt-30-prilohy-appendices.tex** Soubor pro přílohy (obsah nahradte).

**projekt.tex** Hlavní soubor práce – definice formálních částí.

Výchozí styl literatury (`czechiso`) je od Ing. Martínka, přičemž anglická verze (`englishiso`) je jeho překladem s drobnými modifikacemi. Oproti normě jsou v něm určité odlišnosti, ale na FIT je dlouhodobě akceptován. Alternativně můžete využít styl od Ing. Radima Loskota

nebo od Ing. Radka Pyšného<sup>1</sup>. Alternativní styly obsahují určitá vylepšení, ale zatím nebyly řádně otestovány větším množstvím uživatelů. Lze je považovat za beta verze pro zájemce, kteří svoji práci chtějí mít dokonalou do detailů a neváhají si nastudovat detaily správného formátování citací, aby si mohli ověřit, že je vysázený výsledek v pořádku.

Makefile kromě překladu do PDF nabízí i další funkce:

- přejmenování souborů (viz níže),
- počítání normostran,
- spuštění vlny pro doplnění nezlomitelných mezer,
- sbalení výsledku pro odeslání vedoucímu ke kontrole (zkontrolujte, zda sbalí všechny Vámi přidané soubory, a případně doplňte).

Nezapomeňte, že vlna neřeší všechny nezlomitelné mezery. Vždy je třeba manuální kontrola, zda na konci řádku nezůstalo něco nevhodného – viz Internetová jazyková příručka<sup>2</sup>.

**Pozor na číslování stránek!** Pokud má obsah 2 strany a na 2. jsou jen „Přílohy“ a „Seznam příloh“ (ale žádná příloha tam není), z nějakého důvodu se posune číslování stránek o 1 (obsah „nesedí“). Stejný efekt má, když je na 2. či 3. stránce obsahu jen „Literatura“ a je možné, že tohoto problému lze dosáhnout i jinak. Řešení je několik (od úpravy obsahu, přes nastavení počítadla až po sofistikovanější metody). **Před odevzdáním proto vždy přezkontrolujte číslování stran!**

## Doporučený postup práce se šablonou

1. **Zkontrolujte, zda máte aktuální verzi šablony.** Máte-li šablonu z předchozího roku, na stránkách fakulty již může být novější verze šablony s aktualizovanými informacemi, opravenými chybami apod.
2. **Zvolte si jazyk,** ve kterém budete psát svoji technickou zprávu (česky, slovensky nebo anglicky) a svoji volbu konzultujte s vedoucím práce (nebyla-li dohodnuta předem). Pokud Vámi zvoleným jazykem technické zprávy není čeština, nastavte příslušný parametr šablony v souboru projekt.tex (např.: `documentclass[english]{fitthesis}`) a přeložte prohlášení a poděkování do angličtiny či slovenštiny.
3. **Přejmenujte soubory.** Po rozbalení je v šabloně soubor projekt.tex. Pokud jej přeložíte, vznikne PDF s technickou zprávou pojmenované projekt.pdf. Když vedoucímu více studentů pošle projekt.pdf ke kontrole, musí je pracně přejmenovávat. Proto je vždy vhodné tento soubor přejmenovat tak, aby obsahoval Váš login a (případně zkrácené) téma práce. Vyhněte se však použití mezer, diakritiky a speciálních znaků. Vhodný název tedy může být např.: „xlogin00-Cisteni-a-extrakce-textu.tex“. K přejmenování můžete využít i přiložený Makefile:

```
make rename NAME=xlogin00-Cisteni-a-extrakce-textu
```

---

<sup>1</sup>BP Ing. Radka Pyšného <http://www.fit.vutbr.cz/study/DP/BP.php?id=7848>

<sup>2</sup>Internetová jazyková příručka <http://prirucka.ujc.cas.cz/?id=880>

4. Vyplňte požadované položky v souboru, který byl původně pojmenován `projekt.tex`, tedy typ, rok (odevzdání), název práce, svoje jméno, ústav (dle zadání), tituly a jméno vedoucího, abstrakt, klíčová slova a další formální náležitosti.
5. Nahraďte obsah souborů s kapitolami práce, literaturou a přílohami obsahem svojí technické zprávy. Jednotlivé přílohy či kapitoly práce může být výhodné uložit do samostatných souborů – rozhodnete-li se pro toto řešení, je doporučeno zachovat konvenci pro názvy souborů, přičemž za číslem bude následovat název kapitoly.
6. Nepotřebujete-li přílohy, zakomentujte příslušnou část v `projekt.tex` a příslušný soubor vyprázdníte či smažete. Nesnažte se prosím vymyslet nějakou neúčelnou přílohu jen proto, aby daný soubor bylo čím naplnit. Vhodnou přílohou může být obsah přiloženého paměťového média.
7. Nascanované zadání uložte do souboru `zadani.pdf` a povolte jeho vložení do práce parametrem šablony v `projekt.tex` (`\documentclass[zadani]{fitthesis}`).
8. Nechcete-li odkazy tisknout barevně (tedy červený obsah – bez konzultace s vedoucím nedoporučuji), budete pro tisk vytvářet druhé PDF s tím, že nastavíte parametr šablony pro tisk: (`\documentclass[zadani,print]{fitthesis}`). Barevné logo se nesmí tisknout černobíle!
9. Vzor desek, do kterých bude práce vyvázána, si vygenerujte v informačním systému fakulty u zadání. Pro disertační práci lze zapnout parametrem v šabloně (více naleznete v souboru `fitthesis.cls`).
10. Nezapomeňte, že zdrojové soubory i (obě verze) PDF musíte odevzdat na CD či jiném médiu přiloženém k technické zprávě.

## Pokyny pro oboustranný tisk

- Zapíná se parametrem šablony: `\documentclass[twoside]{fitthesis}`
- Po vytištění oboustranného listu zkontrolujte, zda je při prosvícení sazební obrazec na obou stranách na stejné pozici. Méně kvalitní tiskárny s duplexní jednotkou mají často posun o 1–3 mm. Toto může být u některých tiskáren řešitelné tak, že vytisknete nejprve liché stránky, pak je dáte do stejného zásobníku a vytisknete sudé.
- Za titulním listem, obsahem, literaturou, úvodním listem příloh, seznamem příloh a případnými dalšími seznamy je třeba nechat volnou stránku, aby následující část začínala na liché stránce (`\cleardoublepage`).
- Konečný výsledek je nutné pečlivě přezkontrolovat.

## Užitečné nástroje

Následující seznam není výčtem všech využitelných nástrojů. Máte-li vyzkoušený osvědčený nástroj, neváhejte jej využít. Pokud však nevíte, který nástroj si zvolit, můžete zvážit některý z následujících:

**MikTeX** L<sup>A</sup>T<sub>E</sub>X pro Windows – distribuce s jednoduchou instalací a vynikající automatizací stahování balíčků.

**TeXstudio** Přenositelné opensource GUI pro  $\text{\LaTeX}$ . Ctrl+klik umožňuje přepínat mezi zdrojovým textem a PDF. Má integrovanou kontrolu pravopisu, zvýraznění syntaxe apod. Pro jeho využití je nejprve potřeba nainstalovat MikTeX.

**JabRef** Pěkný a jednoduchý program v Javě pro správu souborů s bibliografií (literaturou). Není potřeba se nic učit – poskytuje jednoduché okno a formulář pro editaci položek.

**InkScape** Přenositelný opensource editor vektorové grafiky (SVG i PDF). Vynikající nástroj pro tvorbu obrázků do odborného textu. Jeho ovládnutí je obtížnější, ale výsledky stojí za to.

**GIT** Vynikající pro týmovou spolupráci na projektech, ale může výrazně pomoci i jednomu autorovi. Umožňuje jednoduché verzování, zálohování a přenášení mezi více počítači.

**Overleaf** Online nástroj pro  $\text{\LaTeX}$ . Přímě zobrazuje náhled a umožňuje jednoduchou spolupráci (vedoucí může průběžně sledovat psaní práce), vyhledávání ve zdrojovém textu kliknutím do PDF, kontrolu pravopisu apod. Zdarma jej však lze využít pouze s určitými omezeními (někomu stačí na disertaci, jiný na ně může narazit i při psaní bakalářské práce) a pro dlouhé texty je pomalejší.

## Užitečné balíčky pro $\text{\LaTeX}$

Studenti při sazbě textu často řeší stejné problémy. Některé z nich lze vyřešit následujícími balíčky pro  $\text{\LaTeX}$ :

- `amsmath` – rozšířené možnosti sazby rovnic,
- `float`, `afterpage`, `placeins` – úprava umístění obrázků,
- `fancyvrb`, `alltt` – úpravy vlastností prostředí Verbatim,
- `makecell` – rozšíření možností tabulek,
- `pdflscape`, `rotating` – natočení stránky o 90 stupňů (pro obrázek či tabulku),
- `hyphenat` – úpravy dělení slov,
- `picture`, `epic`, `eepic` – přímé kreslení obrázků.

Některé balíčky jsou využity přímo v šabloně (v dolní části souboru `fitthesis.cls`). Nahlédnutí do jejich dokumentace může být rovněž užitečné.

Sloupec tabulky zarovnaný vlevo s pevnou šířkou je v šabloně definovaný „L“ (používá se jako „p“).