

MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY



Korekce dle typografických pravidel a jejich výuka

BAKALÁŘSKÁ PRÁCE

Jakub Fiala

Brno, jaro 2011

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Vedoucí práce: **Mgr. Tomáš Obšíváč**

Poděkování

Poděkování patří vedoucímu práce Mgr. Tomáši Obšívačovi za jeho vedení, rodině za trpělivost a pomoc s testováním webové použitelnosti. Také Bc. Lucii Tokárové, která přispěla odbornými typografickými radami, Sylvii Huškové, Kristýně Řezníčkové a Mgr. Lence Chloupkové za korekturu Slovníku a online testů. Největší poděkování patří Michalu Valovi a jeho pomoci s Korektorem a Bc. Pavle Pinkasové, která se podílela na korektuře bakalářské práce.

Shrnutí

Teoretická část bakalářské práce se věnuje dvěma základním tématům. První sleduje odlišnosti v aplikaci nejednotných typografických pravidel v různém prostředí. Druhá je věnována rozboru vlastního programu, jenž má za úkol opravovat chybnou interpunkci.

Praktickou součástí práce je webová prezentace Litéra. Tato webová aplikace má dvě stěžejní části, již zmíněný program s názvem Korektor a textovou oporu Slovník s pravidly typografie a online testy.

Klíčová slova

Typografie, interpunkce, korektura, typografická pravidla, pravidla českého pravopisu, Ústav pro jazyk český, Litéra, Korektor, Slovník, online testy, syntaktická analýza

Obsah

1	ÚVOD	7
2	ZÁKOUTÍ TYPOGRAFIE	8
2.1	ZDROJE PRAVIDEL	8
2.1.1	<i>Typografická pravidla</i>	9
2.1.2	<i>Webová typografická pravidla</i>	9
2.1.3	<i>Internetová jazyková příručka ÚJČ</i>	9
2.1.4	<i>Pravidla českého pravopisu</i>	9
2.1.5	<i>Česká technická norma</i>	9
2.2	FORMY TEXTOVÉHO VÝSTUPU	11
2.2.1	<i>Formální text</i>	11
2.2.2	<i>Text určený pro web</i>	11
2.2.3	<i>Běžná elektronická komunikace</i>	11
2.2.4	<i>Tvůrčí práce s textem</i>	12
2.3	DOSTUPNOST SPECIÁLNÍCH ZNAKŮ.....	12
2.3.1	<i>Náhrada podobným znakem</i>	12
2.3.2	<i>Mapování klávesnice</i>	12
2.3.3	<i>Využití klávesových zkratk</i>	13
2.4	INFORMACE O TYPOGRAFII NA WEBU	14
3	LITÉRA	15
3.1	KOREKTOR	16
3.1.1	<i>Typologie chyb</i>	16
3.1.2	<i>Ruční vyhodnocení</i>	17
3.1.3	<i>Pozitivní předpoklady</i>	17
3.1.4	<i>Potvrzení volby</i>	17
3.2	UKÁZKA FUNKCE KOREKTORU.....	17
3.3	SLOVNÍK.....	20
3.3.1	<i>Výběr znaků</i>	20
3.3.2	<i>Stránka se znakem</i>	21
3.3.3	<i>Kapitoly teoretické části</i>	22
3.3.4	<i>Online test</i>	23
4	DOKUMENTACE KOREKTORU	25
4.1	ROZBOR PROGRAMU	25
4.1.1	<i>Vlastnosti</i>	25
4.1.2	<i>Přednosti</i>	26
4.1.3	<i>Nedostatky</i>	27
4.1.4	<i>Chyby při korekci</i>	27
4.2	SYNTAKTICKÁ ANALÝZA	28
4.2.1	<i>Fáze syntaktické analýzy</i>	28
4.2.2	<i>Ukázka syntaktické analýzy znaménka pro násobení</i>	30
4.3	ZDROJOVÝ KÓD	30
4.3.1	<i>Soubory se zdrojovým kódem</i>	31
4.3.2	<i>Použité programovací jazyky</i>	32
4.3.3	<i>Regulární výrazy (RE)</i>	32
4.3.4	<i>Priorita regulárních výrazů</i>	33
5	ZÁVĚR	34
	LITERATURA A POUŽITÉ ZDROJE	35
	PŘÍLOHA 1: ZDROJOVÉ KÓDY LITÉRY	37
	PŘÍLOHA 2: UKÁZKA STRÁNKY SLOVNÍKU	39

1 Úvod

Obsahová či gramatická správnost psaného textu je dnes již samozřejmostí. Obecně je očekávána také od elektronických či webových dokumentů. Zde však vstupuje do hry další důležitý aspekt a tím je typografická správnost. Přesto, že se tato forma stala dnes již zcela běžnou a nevyhnutelnou, nevešla zatím typografická pravidla pro její psaní do obecného povědomí, jako tomu je například u gramatiky.

Přesto, že nelze typografická pravidla označit za nedostupná, jejich znalost a dodržování je spíše výjimkou. Dnešní uspěchaná společnost však vyžaduje především okamžitý a nekomplikovaný výsledek. Tomu rozsáhlost typografických pravidel a velké množství jejich zdrojů zrovna neodpovídá.

Pokud však uživatel pravidla přeci jen zná, ještě stále nemá typografie vyhráno. Do cesty se jí staví další komplikace v podobě složitosti klávesových zkratk, způsobené omezenými technickými možnostmi běžné počítačové klávesnice. Jejich zápis od uživatele vyžaduje další znalosti.

Litéra je program, který dodržování typografických pravidel běžnému uživateli zpřístupňuje. Vyžaduje od něj přitom jen nenáročnou operaci a malý časový vklad. Slovník může případné zájemce o znalostech typografických pravidel vzdělávat. Korektor je schopen typografické chyby opravit a udělat tak za uživatele složitou práci.

V případě webových textů je však situace o něco komplikovanější. Mnohé prohlížeče svou technickou nedokonalostí zobrazení typograficky korektních textů neumožňuje. Nezbývá než doufat, že i tvůrci webových prohlížečů přestanou brát význam typografické správnosti na lehkou váhu.

2 Zákoutí typografie

Mluvit správně česky – doufejme – je samozřejmostí. Chyby v psaném slově jsou častou neřestí. A interpunkční znaménka? Holé neštěstí. Ale také umění. Pravidla jejich psaní nemáme dostatečně v obecném povědomí, mimo jiné proto, že na rozdíl od gramatických jevů nejsou součástí školní výuky.

Typografie, do které psaní nepísmenných a nečíselných znaků spadá, je náročná z několika důvodů. Existuje více zdrojů částečně odlišných pravidel (2.1), jejich použití se navíc liší v závislosti na formě textového výstupu (2.2). Zápis některých speciálních znaků na klávesnici (2.3) je pro běžného uživatele příliš složitý. Navíc dochází k záměrnému porušování pravidel, které při své práci využívají grafici v zájmu uměleckého vyjádření.

Velké množství zdrojů komplikuje uživateli orientaci v pravidlech a problematizuje volbu vhodné varianty. Situaci ztěžuje také fakt, že uživatel nemá možnost dohledat informace o nejvhodnějším výběru pravidel pro daný text. Náročnost zápisu speciálních znaků způsobuje, že je lidé nepoužívají, popřípadě nahrazují jinými. Vystavení uživatele odborným grafickým výstupům, které ale porušují pravidla v zájmu umělecké kvality, způsobuje osvojování chybného použití některých interpunkčních symbolů.

2.1 Zdroje pravidel

Pro úpravu elektronických textů existuje několik zdrojů pravidel, které si v jistých situacích vzájemně **protiřečí**. S odkazem na tyto zdroje lze tentýž dokument zpracovat **různě**, avšak vždy **korektně**. Výčet zdrojů pravidel ovlivňujících interpunkci je následující:

1. Typografická pravidla (*TP*)
2. Webová typografická pravidla (*WTP*)
3. Internetová jazyková příručka Ústavu pro jazyk český (*ÚJČ*)
4. Pravidla českého pravopisu (*PČP*)
5. ČSN 01 6910: Úprava písemností zpracovaných textovými editory (*ČSN*)

2.1.1 Typografická pravidla

Typografická pravidla ustanovili typografové s ohledem na umělecké vyjádření psaného projevu. Jejich pravidla jsou vhodná pro psaní textů, jež kladou důraz na precizní vizuální výstup spojený s tvůrčím přístupem k textu. Taková tvorba není pro běžného uživatele účelná. *TP* lze využít převážně ve specializovaných programech.

2.1.2 Webová typografická pravidla

Omezené technické možnosti webového prostředí vyžadují zjednodušení Typografických pravidel. Úpravy, které vešly v platnost, nejsou v kompletním znění sepsány v jednom zdroji, ale lze je dohledat po částech na různých místech. *WTP* jsou určeny pro potřeby uživatelů webových stránek a zohledňují technické možnosti různých prohlížečů.

2.1.3 Internetová jazyková příručka ÚJČ

Jedním ze zdrojů dostupných online, který popisuje pravidla interpunkce, je Internetová jazyková příručka¹. Tu provozuje Ústav pro jazyk český spadající pod Akademii věd České republiky. Online příručka rozšiřuje pravidla Českého pravopisu o doporučení *ÚJČ*. Doporučení lze chápat jako přípustné varianty zápisu znamének.

2.1.4 Pravidla českého pravopisu

Vydavatelem Pravidel českého pravopisu je opět Ústav pro jazyk český, jenž má za úkol výzkum českého jazyka. *PČP* je publikace uvádějící pevná pravidla pro psaní textu, která se opírá se o český pravopis. Typografie včetně interpunkce nejsou hlavním tématem Pravidel, ale základní informace o psaní textu v nich běžný uživatel nalezne.

2.1.5 Česká technická norma

Česká technická norma – *ČSN 01 6910* popisuje pravidla pro úpravu písemností zpracovaných textovými editory. Na seznamu výše uvedených pravidel má své místo především proto, že běžnému uživateli poskytuje mimo jiné informace,

¹ <http://prirucka.ujc.cas.cz/> – Internetová jazyková příručka (ÚJČ AV ČR, v. v. i.).

jak lze interpunkční znaménka zapsat prostřednictvím základních znaků na klávesnici bez použití klávesových zkratk.

Příklad: Výpustka v nedokončené větě (zámlka, aposiopese)

Z popisu jednotlivých zdrojů pravidel je patrné, že se od sebe liší především tím, na co kladou důraz. Je tedy zřejmé, že se vyskytují případy, kdy se pravidla jednotlivých zdrojů navzájem vylučují. Jeden příklad za všechny:

Výpustka v nedokončené větě, na jejím konci a začátku. V tomto případě se jedná o tzv. *zámlku*, popř. *aposiopesi*, což značí nedokončenou nebo přerývanou řeč (existuje také *výpustka/elipsa* ve funkci nahrazení věty nebo odstavce, který je úmyslně vypuštěn – v tom případě se ale řídí jinými pravidly).

1. **WTP, ČSN** (přiražení k souvisejícímu znaku, z druhé strany mezera)
2. **ÚJČ, PČP** (vždy pouze mezera za – podobně jako u tečky, čárky, ...)
3. **TP** (zúžená mezera u souvisejícího znaku, obyčejná z druhé strany)

1. Krásný člověk ten, jehož dokonalost tvoří spousta...
...malých chyb.
2. Krásný člověk ten, jehož dokonalost tvoří spousta...
...malých chyb.
3. Krásný člověk ten, jehož dokonalost tvoří spousta...
...malých chyb.

2.2 Formy textového výstupu

Lze stanovit čtyři základní formy textového výstupu. Jedná se o nejpoužívanější textové formy, se kterými se uživatel běžně setkává.

1. Formální text
2. Text určený pro web
3. Běžná elektronická komunikace
4. Tvůrčí práce s textem

2.2.1 Formální text

Formálním textem se rozumí každý text, který podléhá všeobecně platným požadavkům. Jeho formu určují pravidla zažitá v dané společnosti. Proto je tato kategorie velmi obsáhlá – může jít například o odbornou či vědeckou publikaci, diplomovou práci, novinový článek, esej či žádost o práci. Zpravidla se jedná o texty, které jsou dále prezentovány, a proto je požadavek na korektnost zásadní. K dosažení požadované úrovně textu je vhodné řídit se *TP*.

2.2.2 Text určený pro web

Text určený pro web se od formálního liší v jednom zásadním ohledu. Přesto, že se jedná o dokument dále prezentovaný, pravidla pro jeho psaní jsou volnější. To je dáno jednak omezenými technickými možnostmi webu, jednak benevolencí vůči textům uveřejňovaným na internetu. Se zažitou tolerancí k internetovým textům počítají *WTP*. Proto je lze pro psaní webových dokumentů doporučit.

2.2.3 Běžná elektronická komunikace

Do kategorie běžné elektronické komunikace lze taktéž zahrnout širokou paletu textů. Mezi nejpoužívanější prostředky takové komunikace patří e-mail, diskuzní fórum, chat, komentáře a jiné. V případě těchto elektronických textů lze jen stěží hovořit o dodržování typografických pravidel. Pokud však lze některá z nich doporučit, potom jistě *ČSN*, a to především pro jejich přístupnost a snadnější aplikaci.

2.2.4 Tvůrčí práce s textem

V kontextu typografických pravidel je za tvůrčí text považována taková forma výstupu, která s textem pracuje kreativně na grafické úrovni. Často se s tím setkáme u plakátů, letáků, brožur či promo materiálů. Při tvorbě může docházet k úmyslnému porušování pravidel. Základem pro tento materiál by však stále měly být *TP*.

2.3 Dostupnost speciálních znaků

Počet interpunkčních znaků přístupných na klávesnici jedním stiskem je málo. Zhruba dvě třetiny symbolů je nutné psát klávesovými zkratkami². V případě matematických symbolů jde téměř o sto procent z nich.

Některé složitě dostupné znaky se postupem času dostávají do širšího povědomí, což je způsobeno rozvojem internetu a také důkladnějším grafickým znázorněním přímo na klávesách. Zápis *hranaté závorky, svorky, násobení, dělení, zavináče* a několika dalších se tak stává o něco jednodušší. Oproti tomu například *pomlčka, výpusťka, české uvozovky* a většina matematických symbolů je nadále zdrojem problémů.

2.3.1 Náhrada podobným znakem

Nejčastějším, ale ve výsledku chybným způsobem, je náhrada znaku jiným vizuálně podobným dostupným přímo na klávesnici. Například *české uvozovky* bývají nahrazovány *strojovými* (rovnými programátorskými), *pomlčky spojovníkem*, *násobení písmenem „x“*, *dělení dvojtečkou* nebo znak *kříže* znaménkem *plus* (ČSN a PČP některé tyto ústupky toleruje).

2.3.2 Mapování klávesnice

Možnost takzvaného „mapování klávesnice“ je pro veřejnost téměř neznámá. Jedná se o proces, ve kterém si uživatel sám může nastavit rozložení kláves na klávesnici. Zvolený symbol (včetně běžně nedostupných) lze uložit pod libovolnou klávesu. Nový symbol ten původní zcela nahradí.

² Vyvolání speciálního symbolu při stisku kombinace více kláves.

2.3.3 Využití klávesových zkratek

Veškeré symboly (také alfanumerické) lze vyjádřit číselnou kombinací. Na internetu existuje množství zdrojů, které je uvádí. Aby počítač správně zobrazil požadovaný symbol, místo číselné kombinace je nutné předem stisknout klávesu levý [Alt]³. Podmínkou správné klávesové zkratky je zápis čísel na numerické části klávesnice (vpravo), nikoli alfanumerické.

Kromě číselných klávesových zkratek lze využít také nečíselné. Zápis speciálních symbolů je pomocí nich kratší, protože využívá vždy jen jednu klávesu v alfanumerické části. Vyvolávají se pravým tlačítkem [Alt] v kombinaci s příslušnou klávesou. Navíc většina českých klávesnic uvádí grafický popis daných symbolů – například *zavináč* bývá znázorněn na klávěse „v“.

Příklad: Číselná klávesová zkratka pomlčky

Číselné vyjádření *pomlčky* je 0150 (včetně nuly na první pozici). Klávesová zkratka pro *pomlčku* se zapíše následovně:

- Stisknutí levé klávesy [Alt]
- Stisknutí čísel 0150 (numerická část klávesnice)
- Puštění klávesy [Alt]

V informatice se taková zkratka označuje jako „Levý Alt + 0150“, přičemž znaménko plus znázorňuje držení klávesy [Alt] při zápisu následných číslic.

Příklad: Nečíselná klávesová zkratka zavináče

Nečíselné vyjádření *zavináče* je písmeno „v“. Klávesová zkratka pro *zavináč* se zapíše následovně:

- Stisknutí pravé klávesy [Alt]
- Stisknutí písmene „v“ (bez uvozovek)
- Puštění klávesy [Alt]

V informatice se taková zkratka označuje jako „Pravý Alt + V“.

³ Hranaté závorky značí klávesu, jejich obsah pak název.

2.4 Informace o typografii na webu

Přesto, že článků o typografii nalezneme na internetu nespočet, nelze se při výzkumu opírat o cizojazyčné zdroje, protože každý jazyk má své typografické zákonitosti. Pokud se zaměříme na prameny věnující se české typografii, zjistíme, že kvantita převyšuje kvalitu.

Za důvěryhodné lze považovat mimo jiné typomil.com⁴ a [Dero](http://typografie.dero.name/)⁵, které jsou stručné a zaměřují se především na *TP* a *WTP*. Dále Internetová jazyková příručka opírající se o *ÚJČ*. Dokument *ČSN* je k dispozici za poplatek. *Interval*⁶ a *Root*⁷ pak jen shrnují obecná pravidla.

Informace o typografii jsou však v těchto webových příručkách kusé, navíc nenabízí uživateli možnost korekce textu. Program *Litéra*, který je praktickou součástí této práce, soustřeďuje dostupné informace na jedno místo, popřípadě uvádí odkazy na doplňující teorii. Zároveň poskytuje uživateli možnost automatické opravy textu.

⁴ www.typomil.com/ – písmo, typografie a sazba na webu.

⁵ <http://typografie.dero.name/> – Typografie – skrytá tvář webdesignu.

⁶ www.interval.cz/ – Webdesign, programování, nástroje, magazín, hotová řešení, eShop.

⁷ www.root.cz/ – informace nejen ze světa Linuxu.

3 Litéra



Webovou příručku Litéra tvoří dvě části – typografický **Korektor** českých textů *online* a podrobný **Slovník** znaků s typografickými pravidly. Server je koncipován tak, aby naplnil požadavky běžných uživatelů, jež potřebují bez námahy opravit vlastní texty. Nasytí i ty, kteří mají zájem studovat jednotlivá pravidla.

Až na ojedinělé výjimky se Korektor i Slovník opírají výhradně o *WTP*. Slovník uživatele upozorňuje na ostatní normy a doporučení (*TP*, *ÚJČ*, *PČP* a *ČSN* – viz kapitola **2.1**) formou poznámek pod čarou. Z uvedených norem oba programy čerpají pouze v případech, které autor programu považoval za vhodnější. Tím se Litéra liší od ostatních serverů, které vychází jen z jednoho zdroje.

3.1 Korektor



Vývoj Korektoru připomínal mávnutí ptačích křídel, kdy na počátku nezáživnou teorii o ručním nahrazování nevhodných znaků vystřídala velká myšlenka následovaná sestřičkou v reálnější podobě. Automatické korektury včetně textů zdrojového kódu⁸ HTML⁹ je prozatím přílišné sousto. Při vývoji programu citlivého na kontext písmen, čísel a interpunkce je důležitá preciznost, což by zohledňování dalších prvků nadměrně ztěžovalo. Svou vyšší potenciální chybovostí by při uvedení do širšího povědomí Litéra důvěru nezískala.

Korektor je tedy intuitivní program, který opravuje typografické chyby v elektronických textech. Uživateli umožňuje vepsat nebo vložit vlastní text do textového pole. Po té stiskne tlačítko pro vykonání opravy a dostává okamžitě vyhodnocení.

3.1.1 Typologie chyb

Program není natolik inteligentní, aby vždy rozpoznal kontext¹⁰ jednotlivých interpunkčních znamének – ty se mohou zapisovat různými způsoby¹¹. Mnohé chyby jsou tak zkrátka neopravitelné (nebo je nelze rozpoznat). Pro zjednodušení analýzy vyhodnocení a transparentnost programu byly typy chyb pojmenovány na tzv. „bílé“, „žluté“ a „červené“.

- **Bílé chyby**

Bílé chyby jsou takové výrazy, které jsou s jistotou zapsány chybně a mají jediný správný zápis. Korektor je po vyhledání automaticky opraví.

Například mezera za otevírací závorkou je bílou chybou. Program ji ihned maže.

- **Žluté chyby (problém rozpoznatelnosti)**

Žluté chyby nemusí být chybami v pravém slova smyslu. Program dokáže lokalizovat výraz, který lze zapsat v různých kontextech více způsoby.

⁸ Zápis počítačového programu v programovacím jazyce – oproti běžnému textu obsahuje speciální symboly a příkazy pro provádění různých činností.

⁹ HTML je „značkový jazyk“, který umožňuje zobrazovat data ve webovém prohlížeči.

¹⁰ Kontextem se rozumí pozice okolních souvisejících alfanumerických znaků.

¹¹ Které výrazy mohou měnit svou podobu vzhledem ke kontextu popisuje Slovník znaků. Kapitola Syntaktická analýza vylicuje důvody, proč nelze některé chyby opravovat.

Například výraz „6.6“ může být správný zápis kapitoly, ale také chybný zápis data (chybí mezera za tečkou). Chybu nelze strojem rozpoznat.

- **Červené chyby (problém opravitelnosti)**

Také červené chyby Korektor neopravuje. Program s jistotou pozná, že výraz lze zapsat různými způsoby, ale nalezený není určitě správně.

Například spojovník s mezerami z obou stran je chybou. Nelze ji však opravit z důvodu možného dvojího zápisu – bez mezer, nebo s pravostrannou mezerou.

3.1.2 Ruční vyhodnocení

Veškeré obarvené výrazy (žluté a červené chyby) poskytují dle kontextu nabídku s možnostmi oprav. Dodatečné ruční úpravy jsou tedy (téměř vždy) nutností, díky nabídkám s praktickými příklady však stále proveditelné laikem. Uživatel si vybere takový příklad, který se typově podobá výrazu v jeho textu. Podle něj je pak provedena oprava.

3.1.3 Pozitivní předpoklady

Při korektuře se často vyskytují žluté chyby způsobené množstvím pravidel pro jednotlivé znaky. V takovém případě se využívá tzv. *pozitivní předpoklad*, který uživateli přednostně zaškrtně tu volbu, která podle Korektoru odpovídá záměru uživatele. Nabídka alternativ tedy není statická – generuje se s ohledem na zápis původního výrazu. Upřednostňována je ta množina příkladů, která souhlasí s textem uživatele.

3.1.4 Potvrzení volby

Po rozhodnutí uživatele, že označený výraz spadá do určité množiny příkladů, uživatel potvrzuje volbu zatržením. Příslušné opravy se provedou samy a obarvení chyby se mění na zelené.

3.2 Ukázka funkce Korektoru

Příklad: Dvojtečka bez mezer

Uživatel napsal následující text (vyjádření sportovního skóre):

- Kometa porazila Budějice 3:2.

Protože se jedná o *žlutou chybu* (dvojtečku lze psát třemi způsoby), výraz je pouze zvýrazněn a uživateli pak poskytnuta nabídka možností zápisu. Díky *pozitivnímu předpokladu* je upřednostňována třetí odrážka. Nabídka vypadá následovně:

1. Uvození věty, výčtu
Př.: Aymary se dělí na: ■pruhované, proužkované, ...
2. Citace
Př.: FIALA, Jakub. Litéra ■: ■Typografický korektor
3. **Čas s uvedením vteřin, skóre**
Př.: V čase 42:15 vyrovnal na 2:2

Původní text souhlasí s posledním výčtem a je tedy správně. Uživatel výběr 3) potvrzuje zatržením, výraz zůstává bez opravy a označení *žluté chyby* mizí.

Příklad: Dvojtečka s mezerami

Uživatel napsal následující text (vyjádření sportovního skóre):

- Kometa porazila Budějice 3 ■: ■2.

Opět *žlutá chyba*. Z důvodu *pozitivního předpokladu* je ale zvýrazněna druhá možnost (mezery kolem dvojtečky) s ohledem na výraz uživatele.

1. Uvození věty, výčtu
Př.: Aymary se dělí na: ■pruhované, proužkované, ...
2. **Citace**
Př.: FIALA, Jakub. Litéra ■: ■Typografický korektor
3. **Čas s uvedením vteřin, skóre**
Př.: V čase 42:15 vyrovnal na 2:2

Protože se však dle kontextu jasně jedná o sportovní skóre, výraz je špatně a uživatel si vybírá výčet 2). Po té výběr potvrzuje zatržením a počítač následně odmazává přebytečné okolní mezery. *Žlutá chyba* mizí.

Vložte/vepište vlastní text

Plus (s chybami)

Pak jsme začali probírat znaménko+, tedy „plus“. Prý se může objevit ve funkci sčítání (5+2), „a více“ (5 +) a také vyjadřuje

Zkorigovaný text

počet varování: 8

příklad „5+“, ale
ných+ 5. Dostal
lyž zkusila i 5+
zkou 2 + a 3+).

Plus (s chybami)

Pak jsme začali probírat znaménko +, tedy „plus“. Prý se může objevit ve funkci sčítání (5 + 2), „a více“ (5+) a také vyjadřuje kladnou hodnotu: +5. Peťa na tabuli rozepsal příklad „5+“, ale celý znechucen, skončil. Přitom venku bylo ‚krásných‘ +5. Dostal za to 5 + (pět lépe). Alča pak vypočítala x +5. Když zkusila i 5+ y, bylo hůř. Nakonec vyfasovala dvě známky (hezkou 2 + a 3+).

Potenciální chyba

Plus

Matematický příklad

$$5 + 10 = 15$$

Plus ve funkci slova „a více“

Vstup lidem 18+

Potvrď volbu

Podrobnosti o znaménku [Plus](#)

Obrázek 4.1: Korekce textu a uživatelská volba

Ukázka vhodného textu pro zkoušku funkcí Korektoru

Pak jsme začali probírat znaménko+, tedy „plus“. Prý se může objevit ve funkci sčítání (5+2), „a více“ (5 +) a také vyjadřuje kladnou hodnotu: + 5 . Peťa na tabuli rozepsal příklad „5+“, ale celý znechucen, skončil. Přitom venku bylo ‚krásných‘+ 5. Dostal za to 5 + (pět lépe). Alča pak vypočítala x +5. Když zkusila i 5+ y, bylo hůř. Nakonec vyfasovala dvě známky (hezkou 2 + a 3+).

3.3 Slovník



Původně se Slovník věnoval pouze webové typografii, čímž opomíjel ostatní zdroje pravidel. Zároveň uváděl informace, které odbíhaly od jeho hlavní náplně, jako například informace o citování a zpracování elektronických dokumentů nebo zohlednění slovenské typografie. Kromě teorie o správném psaní formálních dokumentů měla kapitola ambice nabízet praktické ukázky a šablony předem naformátovaných souborů s texty. Zmíněné kapitoly by však přesahovaly rámec projektu a proto nebyly naplněny.

Návrh provázané struktury s přehlednou sestavou informací o typografických pravidlech na jednom místě si vyžádal několikanásobnou reorganizaci textů Slovníku. Některé zajímavosti byly vyřazeny, praktické ukázky vyzdviženy a dodatečné podkapitoly potlačeny. Čtenář získává okamžitý přehled o nejpodstatnějších informacích aniž by odcházelo k znejistění množstvím. Přesto existují výjimečné znaky (například kapitola o *pomlčce*), které svými rozsáhlými pravidly překračují mnohostranné „bestsellery“...

Slovník lze rozdělit do tří pomyslných kapitol. Obsahuje podrobný seznam nejčastěji používaných nepísmenných a nečíselných znaků (3.3.1) doplněných výčtem pravidel jejich užití (3.3.2). Ta jsou doplněna o praktické ukázky. Druhým oddílem jsou rozšiřující informace s náhledem do teorie (3.3.3). Další sekci je online test (3.3.4), který ověřuje znalosti získané z předchozích částí. Tato složka Slovníku je zcela jedinečnou záležitostí v oblasti české typografie. Internet nenabízí žádnou jinou formu e-learningu zaměřenou na nauku o písmu.

3.3.1 Výběr znaků

Úvodní stránka nabízí výběr jednotlivých znaků, které jsou dostupné také přes podrobný rejstřík. V něm se odkazy duplikují, protože názvů pro konkrétní interpunkční znaménka je často několik. Běžný uživatel tak snáze najde hledaný symbol (například znak „|“ je dostupný přes klíčová slova „absolutní hodnota“, „rovná závorka“ a „svislítko“).

Kromě interpunkce jsou součástí seznamu matematické a jiné symboly, používané mj. na webu.

- **Seznam interpunkce**

Pevná mezera, tečka, čárka, středník, dvojtečka, výpustka, otazník, vykřičník, české, francouzské, anglické a strojové uvozovky (dvojitě a jednoduché), apostrof, lomítko, závorky (kulaté, hranaté, složené, ostré), pomlčka, dlouhá pomlčka, spojovník.

- **Seznam matematických a jiných symbolů**

Plus, mínus, násobení, dělení, matematické lomítko, absolutní hodnota, větší, menší, větší nebo rovno, menší nebo rovno, rovnost, nerovnost, identita, plus mínus, přibližně rovno, vlnka, stupeň, minuta, vteřina, procento, promile, programátorská mocnina, odmocnina, integrál, nekonečno, suma, celkový součin, hvězdička, křížek, ampersand, paragraf, mřížka, copyright, registrovaná značka, obchodní značka, Euro, Libra, Dolar, zavináč, obrácené lomítko, podtržítka, středová tečka, šipky.

3.3.2 Stránka se znakem

Jak již bylo zmíněno v úvodu kapitoly, Litéra se opírá převážně o WTP. Tyto jsou rozepsána na stránkách jednotlivých znaků a považují se za výchozí. Pokud je tomu jinak (stává se zřídka), v poznámce pod čarou se o tom objevuje dodatečná informace¹². V případě, že znak lze podle výkladu dalších pravidel zapsat i jiným způsobem, vede odkaz v poznámce pod čarou na stránku věnovanou znakům s nejednotným výkladem pravidel (teoretická část).

Poznámky mají za úkol shlukovat dodatečné informace, které by jinak stránky zahlcovaly. Patří sem také klávesové zkratky špatně dostupných znaků a zmínky o jejich podobě¹³. Pomocí odkazu pro skrytí teorie lze zobrazit pouze praktické příklady, takže nauka pravidel pro zápis interpunkce se stává o něco jednodušší a přehlednější. Každý znak obsahuje praktické příklady zápisu, ale i častých chyb.

Horní menu nabízí pohyb mezi jednotlivými symboly a jejich pravidly a obsahuje odkazy na teorii a online test.

¹² Příkladem může být pomlčka ve funkci „od do, až, versus, z do, a“ u víceslovného výrazu – za výchozí pravidla jsou považována ta z ÚJČ.

¹³ Např. pomlčka, dlouhá pomlčka, spojovník a mínus jsou podobné znaky.

Interpunkce

·	,	;	:	...	
?	!	”	“	,	‘
»	«	>	<	"	'
,	”	/	([{
<	—	—	—		

Matematické a jiné symboly

Další odkazy

Otazník a vykřičník

[skrjji/ukaž teorii](#)

Otazník se používá v tázacích větách, vykřičník při důrazu. Obě znaménka větu ukončují, Tečka se za nimi nepíše (což vyjadřuje samotný puntík pod oběma znaky). Čárka se z důvodu zřetelnosti větné stavby může v ojedinělých případech za otazník psát. Ke znaménku se tiskne **bez mezer**.

1 Psaní mezer

Pro oba znaky platí stejná pravidla. Běžně se přiráží k poslednímu znaku a po nich **následuje mezer**. V případě vyjádření domněnky či údivu je lze klást do Závorky uprostřed věty. Mezeru ruší pravidlo závorky, dále viz Priorita.

Př. 1

1. Otázka
2. Důraz
3. Vyjádření domněnky

1. A zítra bude jak? Zítra bude hůř.
2. Kolego, neblbni! To jsi nezvládl.
3. Vstával už okolo desáté ráno (?).

Obrázek 4.2: Stránka se znaky otazník a vykřičník

3.3.3 Kapitoly teoretické části

- Zdroje uváděných pravidel

Pro běžného uživatele je tato úvodní kapitola pouze informativní. Pojednává o sestavě použitých pravidel, jejich upřednostňování a důvodech rozličnosti. V závěru jsou uvedeny citace s odkazy na pravidla.

- Rozdíly v pravidlech (TP, ÚJČ, ...)

Zatímco předchozí část na rozdíly pouze poukazovala, tato se jim věnuje podrobněji i s užitím praktických příkladů. Jsou v ní popsány veškeré situace, kde se uplatnění zápisu znaků s ohledem na zdroj pravidel liší.

- Priorita pravidel sousedních znaků

Stránky věnované konkrétním znakům obsahují soubor pravidel pro jejich zápis (viz 3.3.2). Často však dochází k situaci, kdy se v textu vyskytují dvě a více interpunkčních znamének vedle sebe. Jejich pravidla se tak mohou dostat do konfliktu. K jeho vyřešení slouží kapitola o prioritě uplatňovaných pravidel.

- **Řádkový zlom**

Typografie se věnuje také dělení výrazů na konci řádku. Slovník sice uživateli pravidla pro Řádkový zlom objasňuje, ale Korektor je z technických důvodů zohlednit nedokáže.

- **Číselné údaje (data, čas, peněžní částky)**

Kapitola shrnuje pravidla pro interpunkční znaménka sousedící s číselnými údaji. Věnuje se také rozdílům ve vyjádření podstatného a přídavného jména stojícího vedle číslovek. Na závěr poukazuje na nejčastější chyby v zápisu.

- **Podoba některých znaků**

Z důvodu podoby některých znaků a symbolů bylo přínosné zařadit také informativní stránku s praktickými ukázkami těchto znamének. Jejich záměna, která je v běžných textech častá, je totiž chybou.

- **Zkratky (značky, tituly, hodnosti)**

Pro úplnost poskytovaných služeb je součástí Litéry také sekce s výčtem nejčastěji používaných zkratk.

- **Tabulka hůře dostupných znaků**

Kapitola s tabulkou špatně dostupných znaků je nutným doplňkem teoretické části. Běžný uživatel může přečíst přešel typografických knih, ale pokud je nebude umět zapsat na klávesnici, nebude jeho studium k užitku. Tabulka mu má práci s klávesovými znaky usnadnit.

3.3.4 Online test

Online test pracuje na principu zaškrtování (ne)správných vět s příklady interpunkčních znamének a teorie.

Otázky mají vyzkoušet znalosti uživatele získané z jednotlivých kapitol Slovníku. Lze zaškrtovat více odpovědí najednou, tedy také všechny, ale i žádnou. Správnost volby je možné pod příkladem ihned kontrolovat. Obsahuje-li odpověď alespoň jednu chybu, rámeček se rozsvítí červeně. Změnu kombinace odpovědí i jejich kontrolu lze vyvolat opakovaně, pro efektivnost testu je však doporučeno provádět opravu maximálně jedenkrát na jednom příkladu. Test nabízí na konci stránky vyhodnocení odpovědí a zobrazení řešení.

Test vychází z pravidel Slovníku, vyjma kapitoly Rozdíly v pravidlech (není-li uvedeno jinak).

1 Interpunkční znaménka

Správně může být vždy 0 až n odpovědí.

Tečka, čárka

Př. 1 Zaškrtněte správné věty:

10. dubna 4057 se v čase 5.15 zhroutí vesmír.
- Stáhni si verzi 3.1 a projdi kapitolu 1.5.
- Viděl jsem Earl's Court, Ealing aj. Jednou se tam vrátím.

Zkontroluj odpovědi:

Tečka, čárka, středník

Př. 2 Zaškrtněte správné věty:

- 300.000.00 je jeho roční plat (tři sta tisíc celých).
- Vytiskni faktury ze dne 03.07.2008.
- Vyšla mu desetinná čísla 3,15, 4,12, 7,71 a 8,5.
- Nápověda pro jméno města nad Ponávkou je T.....

Zkontroluj odpovědi:

Datum

Př. 3 Zaškrtněte správné tvary dat:

1. 6. 2007.
1. X. 2007.
- 2007-31-10.
- '97.
03. 08. 2006 (strojové datum).
13. 10. 2006.

Otazník a vykřičník

Př. 5 Zaškrtněte správné věty:

- Tak to jsi teda uhodl! Už se tu neukazuj.
Vykřičník s mezerou za
- Na rok do Zimbabwe ??? Zkus radši Madeiru.
Správný tvar: otazník bez mezery před
Na rok do Zimbabwe??? Zkus radši Madeiru.
- A co na to její rodiče? ! Museli se zbláznit.
Správný tvar: otazníky a vykřičníky se seskupují bez mezer
A co na to její rodiče?! Museli se zbláznit.

Zkontroluj odpovědi (správně = 1): odpovědi jste špatně.

Uvozovky a závorky

Př. 6 Zaškrtněte správné věty:

- Bolí mě hlava (dnes nic nebude) je oblíbenou větou.
Správný tvar: obsah závorek bez mezer, vnější text s mezerami
Bolí mě hlava (dnes nic nebude) je oblíbenou větou.
- Znamená „tais toi“ [te tua] mlč?
Obsah uvozovek i závorek bez mezer, vnější text s mezerami
- Za toho „renaulta“ s čárkou dostaneš veřejné práce.
Správný tvar: obsah uvozovek bez mezer, vnější text s mezerami
Za toho „renaulta“ s čárkou dostaneš veřejné práce.
- „Antressmaline“ je loď třídy M (1 a 2)“ řkala věta v knize.
Obsah uvozovek i závorek bez mezer, okolní text s mezerami; seskupení více druhů uvozovek nebo i závorek bez mezer

Zkontroluj odpovědi (správně = 2, 4): odpovědi jste správně.

Uvozovky

Př. 7 Zaškrtněte věty, které lze použít v české publikaci:

- Jeho „smeč“ by doběhla i želva se zlomenou nohou.
České uvozovky lze použít

Obrázek 4.3: Výchozí a vyplněný online test

4 Dokumentace Korektoru

Vývoj Korektoru si vyžádal *syntaktickou analýzu* (4.2). Při programování bylo potřeba dbát, kromě zdrojového kódu (4.3), také na *použitelnost*, tedy na intuitivní ovládání programu, jeho jednoduchost a přehlednost.

4.1 Rozbor programu

Pojem *použitelnost* je samostatný obor informatiky, který bývá při vývoji menších nebo nekomerčních programů podceňován. Korektor je určen široké veřejnosti, jeho užívání by měl tedy zvládat běžný uživatel internetu, e-mailu a popřípadě textového editoru *Microsoft Word* nebo *Open Office*. Korektor na *použitelnost* dbá.

4.1.1 Vlastnosti

Prostor pro text

Při práci s textem a opravami je důležitý dostatečný prostor. Navržené vstupní a výstupní pole je tedy poměrně velké. Po uživateli se vyžaduje pouze vlastní text, který je následně opravován. To je spolu s nápovědou zdůrazněno na vstupní stránce Litéry. Samotný Korektor tak neobsahuje další doprovodné texty, které by mohly působit rušivě. Z důvodu vzniku možných chyb při korekci je před prvním použitím uživatel vyzván k přečtení zmíněné nápovědy. Předejde se tak možným nepřesnostem.

Barevné chyby

Dělení na *žluté* a *červené chyby* se může zdát jako nepřehledné a zpočátku uživatele nutí přemýšlet nad barevným označením textu. Žlutá a červená barva byly vybrány, aby asociovaly „varování“ (*žlutá*) a „pozor, chyba“ (*červená*). Toto specifikum programu je jako jediné popsáno přímo pod zkorigovaným textem a je viditelné vždy. Text je doplněn o stručné vysvětlení přístupu k chybám a způsobu vyvolání formuláře s nabídkou oprav.

Odlišná barva tedy ihned ukazuje, kde jsou v textu buď potenciální, nebo jisté chyby, čímž uživatele částečně učí správnosti textu a vizuálně nabádá

k opravě. Kromě barvy jsou chyby zvýrazněné velikostně, protože nemalé množství interpunkčních znaků je špatně viditelné.

Formulář s nabídkou oprav

Formulář s nabídkou oprav, který se zobrazí po kliknutí na chybu, prošel největším počtem změn. Předat uživateli informace o opravě je stěžejním úkolem programu. Je důležité nezatěžovat ho teorií o mezerách a pravidlech, ale zároveň je nutné vybrat z množství informací ty podstatné. I tady se ukazuje, jak byla *syntaktická analýza* důležitá. Podle kontextu lze často určit, zda uživateli nabídnout v možnostech oprav všechny dostupné typy zápisu znaménka nebo jen možnosti související s jeho textem. Například *plus* lze zapisovat třemi způsoby, ale po analýze okolního textu stačí, aby byly nabídnuty jen dva z nich. Tím se snižuje riziko případné špatné opravy a zvyšuje přehlednost. Protože by z typů zápisu symbolů nemuselo být dostatečně jasné, který případ je správný, každý je doplněn praktickým příkladem, což poskytuje uživateli vítanou zpětnou vazbu.

Zařazení *pozitivního předpokladu* bylo původně zamýšleno ve formě uvedení pravděpodobného správného výrazu na prvním místě. Tím by se ale u formulářů vyvolaných stejným znakem měnilo pořadí nabídky oprav, což by mohlo být pro uživatele matoucí. Nastavení programu bylo nakonec pozměněno tak, aby pořadí zůstalo zachováno, ale potenciální správný výraz byl upřednostněn – formou implicitního zatržení. U *červených chyb* se logicky takové předpoklady neuplatňují, předem není zatržen žádný výraz.

Zvýrazňování

Po korekci vloženého textu může zvýraznění právě opravované chyby zaniknout vedle barevného označených zbývajících chyb. Z toho důvodu je nutné odlišné obarvení. Po opravě se zvýraznění mění na zelenou barvu. Tím uživatel získává přehled, jaké chyby již opravil. Zeleně označené výrazy zůstávají i nadále aktivní a uživatel dostává možnost opakovaného úkonu. Tuto funkci lze využít například v případě, že napoprvé omylem zvolí špatnou opravu.

4.1.2 Přednosti

Kromě automatických oprav interpunkce (seznam rozpoznatelných znaků viz **3.3.1**), zvýraznění neopravitelných výrazů a schopnosti pozitivního předpokladu dokáže Korektor splňovat i jiné úkoly. Například automaticky doplňuje *pevné*

*mezery*¹⁴ na místa, kde jsou zapotřebí, a opravuje chybný zápis velikosti písmen u většiny titulů a hodnotí. Nabízí náhradu vizuálně podobného, ale nevhodného znaku a vypisuje počet chyb.

4.1.3 Nedostatky

Program dokáže vkládat *pevné mezery* pouze na místa sousedící s interpunkcí, nikoli však mezi dvě slova nebo čísla (mimo některé ustálené zkratky, tituly a hodnoty). Existují tituly a hodnoty, které jsou nad rámec oprav Korektoru, což však není jeho hlavním úkolem. Také se nedokáže vypořádat se specifickými pravidly pro *spojovník* a *lomítko* na zlomu řádku.

Chyby ve znacích *plus*, *minus*, *násobení*, *čárka* a *pevná mezera* nelze zvýraznit, pokud se bezprostředně po sobě nachází deset a více chyb ve stejném znaménku, v jehož okolí je na obou stranách buď číslo, nebo písmeno (například 5+5+5+5+...). Důvody jsou rozepsány v kapitole 4.3.3.

Vstupní pole pro vkládání textů k opravě neumí zpracovat horní a dolní indexy (podobně jako obyčejné textové editory) – ty jsou převedeny na běžný text.

4.1.4 Chyby při korekci

Kromě pravděpodobných chyb způsobených lidským faktorem může způsobovat program chyby v následujících příkladech:

- „rodová alternativa“ (specifické přiřazené závorky)
 - výrazy typu „čekal(a)“ nebo „(o)tiskneme vám vše“ jsou poškozeny vloženými mezerami vně *závorek*
- tečky v názvu uskupení
 - výrazy bez numerických znaků – například „skupina B.R.N.K“ – je poškozen vloženými mezerami za *tečkami*
- internetové odkazy
 - poškození vloženými mezerami za *dvojtečkou*, *tečkou* a *ampersandem*
- koncovky souborů – například. výraz „.html“ (mezera před *tečkou*)
 - poškození smazanou mezerou před *tečkou* a vloženou za ní

¹⁴ Pevná (také „nedělitelná“) mezera zakazuje dělení řádku mezi sousedními výrazy.

- tečky skrývající znaky
 - výrazy jsou poškozeny převodem na *výpustku* (ta se používá v nedokončených větách a při úmyslném vypuštění delšího textu)

4.2 Syntaktická analýza

Nedílnou součástí pro naprogramování Korektoru byla *syntaktická analýza*. Bez ní by psaní kódu nebylo možné. Analýzou se rozumí zjišťování, jaké typy alfa-numerickech a jiných znaků se mohou kolem jednotlivých interpunkčních znamének vyskytovat. Precizní diagnostika podobných situací umožňuje omezit *žluté* a *červené chyby* a nahradit je *bílými*. Jinými slovy lze tedy důkladněji rozlišovat kontext a podle něj chyby opravovat strojem. Tím se eliminuje dodatečný uživatelský zásah a zároveň snižuje chybovost. Kromě vlivu na *barevné chyby* zjednodušuje *syntaktická analýza* následný zápis *regulárních výrazů*, což jsou počítačové příkazy pro vyhledání řetězce znaků a následné nahrazení jiným řetězcem. Specifikace těchto výrazů, které jsou popsány v kapitole **4.3.3**, navíc zjednodušuje formuláře s výběrem oprav.

Interpunkční znaménka, jejich pravidla, priorita pravidel sousedních znaků a samotný kontext vytváří velké množství modelových situací. V závislosti na nich narůstá složitost programového kódu. Počet hypotetických případů je natolik velký, že není v možnostech Korektoru zahrnout všechny. Nedokonalost programu je tak závislá na **nezahrnutí extrémních situací**, ale také úmyslném **zjednodušování** netypických pravidel. To může ojediněle v korektním textu zadaném uživatelem vyvolat následnou chybu. Případy jsou však natolik zanedbatelné a řádně **okomentované**, že jejich přítomnost nezpůsobuje vážné problémy. Pokud by se totiž braly v potaz, program by ztrácel na účinnosti oprav v závislosti na nově vzniklých *žlutých* a *červených chybách*. Nevýhody v chybovosti Korektoru jsou ve výsledku převáženy výhodami automatických oprav.

4.2.1 Fáze syntaktické analýzy

Popis analýzy všech znaků (čítá dohromady **64**), které Litéra zohledňuje, by vydal na celou knihu. Zahájena byla na znaménku *plus*. Pojednání o něm mělo rozsah přes deset stran formátu A5. Postup se ukázal jako neefektivní a rozsahově příliš náročný. Během práce se tedy měnil přístup k analýze znamének a hle-

dala se co možná nejjednodušší varianta popisu těch dalších. V první fázi byly ustanoveny typy znaků a jejich zkratky:

- Písmenný (P) a nepísmenný (NP)
- Číselný (C) a nečíselný (NC)
- Interpunkční (I) a neinterpunkční (NI)
- Mezera () – značena podtržítkem
- Konec řádku (Z)

Z důvodu složitosti problému dvou sousedících interpunkčních znamének byly tyto, NP, NČ a NI analyzovány až v úplném závěru. V okolí interpunkce se tedy zprvu hodnotily pouze písmena nebo čísla a popřípadě mezera. Z toho vyplývá **šestnáct** možností typu okolní interpunkce (příklad se znaménkem *plus*):

- P+_P, P_+P, P+_P, P+P (písmena z obou stran)
- P+_C, P_+C, P+_C, P+C (kombinace písmena a čísla)
- C+_P, C_+P, C+_P, C+P (kombinace čísla a písmena)
- C+_C, C_+C, C+_C, C+C (čísla z obou stran)

Na tyto typy okolí byly aplikovány *WTP* ze Slovníku a následně napsány *regulární výrazy* (viz **4.3.3**). Každý interpunkční znak nepotřebuje vlastní kód, mnoho jich lze sjednotit. Tím došlo k výraznému zjednodušení práce. Po naprogramování všech výrazů přišla na řadu fáze, ve které byla do okolí zahrnuta interpunkce. Z důvodu její složitosti – lze ji dělit například na uzavírací, otevírací, matematickou, ... – bylo nutné zkoumat jednotlivé znaky podrobněji. Zpřesněním *regulárních výrazů* přestaly být z důvodu přílišné obecnosti zkratky I a NI použitelné.

Po naprogramování interpunkce bylo využito možnosti sjednotit více znaků do jednoho příkazu – například symbol *rovnosti* a *nerovnosti* se řídí stejnými pravidly.

4.2.2 Ukázka syntaktické analýzy znaménka pro násobení

	1	2	3	4
A	P_x_P	P_x_C	C_x_P	C_x_C
B	P_xP	P_xC	C_xP	C_xC
C	Px_P	PxC	Cx_P	CxC
D	PxP	PxC	CxP	CxC

Následující data čerpají z WTP:

1. Začíná-li výraz písmenem (například „ $x \times 3$ “ nebo „ $x \times y$ “), znak pro násobení musí být odsazen mezerami z obou stran. Výrazy 1A–D, 2A–D jsou *bílou chybou* – chybí-li mezery, jsou doplněny.
2. Sudý počet mezer (tedy žádná mezera nebo mezera z obou stran) u výrazů začínajících a končících číslem (například „ 5×7 “ – matematický příklad; „ 3×10^{15} “ – mantisa s exponentem) je *žlutou chybou* v 4A, 4D.
3. Výraz začínající číslem následovaným jednou nebo žádnou mezerou a končící mezerou před písmenem (například „ $5 \times y$ “ – matematický příklad; „ $5 \times$ spadl“ – násobek v běžné větě) je *žlutou chybou* v 3A, 3C.
4. Výskyt písmene za násobkem bez mezery – 3B, 3D – je převeden na *žlutou chybu* bodu 3.
5. Lichý počet mezer (tedy mezera pouze na jedné straně) u výrazů začínajících a končících číslem je *červenou chybou* v 4B, 4C. Výrazem lze vyjádřit buď matematický příklad, nebo zápis mantisy a exponentu.

4.3 Zdrojový kód

Během programování se postupně měnila struktura zdrojového kódu Korektoru. Úvodní *syntaktická analýza* usnadnila první návrh funkcí programu a zároveň určila směr dalšího vývoje. Bylo jisté, že Korektor bude schopen přijmout uživatelský text a ten s využitím stěžejní části kódu – *regulárních výrazů* (viz kapitola 4.3.3) – v některých případech opravit, v dalších pak pouze zvýraznit. *Žluté chy-*

by byly rozšířeny na červené a program tak mohl vypisovat počet oprav a varování. Prvotní návrh splnil očekávání a následoval další krok.

Bylo nutné analyzovat napsaný kód a upravit jej do finální přehledné a komentované formy. Důkladný přístup zjednodušuje budoucí práci s dodatečnými úpravami, které vždy a jakýkoli program degradují. Zdrojový kód byl rozdělen do čtyř samostatných souborů, z nichž každý má svoji úlohu (viz **4.3.1**). Indikace počtu chyb musela být zrušena (počet varování zůstal), z důvodu návrhu *regulárních výrazů* a jejich priority.

Úsměvná je skutečnost, jak je interpunkce zrádná. Některá znaménka jsou typická složitostí zápisu, u některých je tato komplikovanost skrytá. Při prvním i druhém návrhu kódu Korektoru bylo vybráno vždy jedno zkušební znaménko, které program dokázal opravovat. Oba výběry ale doplatily na zjevně nedostačnou *syntaktickou analýzu*, protože se vyskytlo mnoho alternativ zápisu. Nakonec přesně taková zátěžová zkouška je ideálním začátkem pro vývoj aplikace. Pokud by byly složitější problémy řešeny až ke konci, zpětné úpravy předchozího by opět narušily přehlednost a kompaktnost programu.

4.3.1 Soubory se zdrojovým kódem

index.php

Úvodní stránka Korektoru s výstupem programu. Obsahuje HTML kód pro zobrazení ovládacích prvků a zprostředkovává vizuální část.

korektor_class.php

Soubor obsahuje základní třídu Korektoru, dále seznam pravidel, rozhraní pro vložení pravidel a potřebné výstupy. Provádí úpravy textu podle definovaných pravidel v souboru *rules.php*. Úprava je vyvolána odesláním formuláře.

rule_class.php

Soubor obsahuje základní třídu pro pravidlo „Rule“, třídy jednotlivých pravidel „RegWhite“, „RegYellow“, „RegRed“ (*bílé a barevné chyby*) a třídu pro formulář s opravami.

rules.php

Soubor s *regulárními výrazy*. „Srdce“ celého Korektoru, které přímo provádí veškerá nahrazení a návrhy oprav.

4.3.2 Použité programovací jazyky

Při programování Litéry byly použity programovací jazyky PHP a JavaScript. Na PHP jsou postavené základy programu, objekty a *regulární výrazy*. JavaScript umožňuje obarvování výrazů a doplňující funkce jako zobrazování a skrývání textů a formulářů. Pro zobrazení na webu je využíván značkovací jazyk HTML a CSS, který zabezpečuje stylování dokumentu.

4.3.3 Regulární výrazy (RE)

Regulární výraz slouží k vyhledání požadovaného výrazu, který následně nahradí jiným. RE použité v Litěře se dají slovně popsat takto:

- Najdi výraz „x“
- Nahraď nalezený výraz výrazem „y“

Na uvedeném principu pracují všechny *bílé chyby*. Například pro opravu nechtěné mezery za otevírací závorkou bude použit RE „*Najdi řetězec ,otevírací závorka, za kterou následuje mezera‘ a tento řetězec nahraď (pouze) otevírací závorkou.*“. Princip RE tedy chápe mazání jako „nahrazení“.

Žluté a *červené chyby* fungují na podobné bázi, rozdíl od *bílých* je v tom, že po provedení funkce „*Najdi*“ není provedena funkce „*Nahraď*“, ale „*Obarvi*“. Nahrazení provádí uživatel ručně za pomoci formuláře s nabídkou oprav.

Pro zlepšení přehlednosti složitějších RE je zapotřebí předem provést dostatečnou *syntaktickou analýzu* a brát v úvahu jejich *prioritu zápisu*. Preciznost této činnosti způsobuje přesnější rozpoznání kontextu znaku, a tedy konkretizuje případné chyby. Následkem jsou přesnější formuláře s návrhy oprav bez nesouvisících příkladů. Tím se snižuje případná chybovost při uživatelském zásahu.

Problém RE spočívá v jejich aplikaci na sousední chyby. Jak bylo zmíněno výše, jeho úkolem je „vyhledat a nahradit“. Při dohledání výrazu jej celý takzvaně *fixuje* – pracuje pouze s ním a po té pokračuje dále až za *fixovaným výrazem*. Má-li za úkol vyhledat a opravit součet dvou čísel bez mezer, *fixuje* C+C. Při součtu tří čísel (C+C+C) *nachází* pouze první výskyt (C+C+C), z důvodu *fixace*

druhý přeskakuje. Proto se u některých znamének v Korektoru provádí specifikace RE, které tím ale ztrácí univerzálnost zápisu. Důsledkem je neschopnost opravit více než deset sousedních chyb konkrétního znaku. Jedná se však o vysoce extrémní případ.

4.3.4 Priorita regulárních výrazů

Na prioritě, tedy pořadí uvedení RE ve zdrojovém kódu, záleží z důvodu možného zápisu více interpunkčních znaků vedle sebe. Například pokud by bylo pravidlo o mezerách vně *závorek* aplikováno až po té, co bylo provedeno smazání mezer před *čárkami*, Korektor by zanášel do uživatelských textů chyby. V souboru *rules.php* jsou znaky seřazeny do typově podobných skupin závislých na prioritě. Později uvedený RE má vždy vyšší prioritu než dříve uvedený.

Nejprve se převádějí takzvané „*bílé znaky*“ (mezery, tabulátory, běžně neviditelné znaky konce řádků a jiné), po té matematické a speciální symboly. Později přichází na řadu znaky s vyšší prioritou – klasická interpunkce. Nejprve je ošetřeno vnější okolí párových znaků (*závorky* a *uvozovky*), poté následuje interpunkce dělicí věty slova nebo výrazy (*výpustka*, *apostrof*, *lomítko*, *pomlčka*, ...) a interpunkce pro větnou stavbu (*tečka*, *čárka*, *středník*). Teprve tehdy přichází na řadu vnitřní okolí párových znaků, což mimo jiné opravuje chybně zanořené *závorky*. Na závěr jsou chybně zapsané znaky a výrazy opraveny na správné (*strojové uvozovky*, špatný zápis titulu nebo hodnoti) a výčet *regulárních výrazů* uzavírá vkládání *pevných mezer*.

Korektor zpracovává nejprve *bílé chyby*, až po té přichází *barevné*. Uživatel tak vstupuje do kompletně zpracovaného textu – po jeho zásahu již nejsou programem prováděny žádné další kroky.

5 Závěr

Webová aplikace Litéra je navržena tak, aby bylo možné dodatečně upravovat její podobu, funkci a obsah v závislosti na měnících se trendech v oblasti typografie. Pokud tedy dojde ke změně typografických pravidel, bude možné program jednoduše inovovat. To z Litéry činí flexibilní a moderní program.

Po uveřejnění na internetu bude aplikace vystavena požadavkům případných uživatelů. Autor programu je připraven komentáře k jeho funkčnosti zohlednit a návrhy na zlepšení jednotlivých funkcí sledovat. Na základě nich pak dále program rozvíjet.

V současné době je program připraven ke spuštění. Přesto jsou již nyní známy možnosti, jak ho lze ještě vylepšit. Obměny budou prováděny průběžně bez nutnosti vyřadit program z provozu. Příkladem může být zamýšlená detailnější specifikace *regulárních výrazů*.

Literatura a použité zdroje

- BERAN, Vladimír, et al. *Aktualizovaný typografický manuál*. Praha : Kafka Design, 1999. 107 s.
- PECINA, Martin. *Typomil - písmo a typografie* [online]. 2004 [cit. 2009-07-17]. Dostupné z WWW: <<http://typomil.com/>>.
- POLAKOVIČ, Jaroslav. *Typografie – skrytá tvář webdesignu* [online]. 2007 [cit. 2009-07-17]. Dostupné z WWW: <<http://typografie.dero.name/>>.
- Ústav pro jazyk český. *Internetová jazyková příručka* [online]. 2008 [cit. 2009-07-17]. Dostupné z WWW: <<http://prirucka.ujc.cas.cz/>>.
- Lingea. *Pravidla českého pravopisu*. 2. vyd. Brno : Lingea, 2010. 895 s. ISBN 978-80-87062-88-3.
- ČSN 01 6910. *Úprava písemností zpracovaných textovými editory : Česká technická norma*. Praha : Fortuna, 2007. 48 s.
- ÚNMZ. *Úřad pro technickou normalizaci, metrologii a státní zkušebnictví* [online]. 2010 [cit. 2010-11-28]. Dostupné z WWW: <<http://www.unmz.cz>>.
- digitalwaffle. *LiveJournal : digitalwaffle: When Is A Quote Not A Quote?* [online]. 23. 9. 2003 [cit. 2010-11-20]. Dostupné z WWW: <<http://digitalwaffle.livejournal.com/2496.html>>.
- KORPELA, Jukka K. *Unicode Explained : Internationalize Documents, Programs and Web Sites* [online]. First edition. United States of America : O'Reilly Media, 2006 [cit. 2011-05-20]. Dostupné z WWW: <http://books.google.cz/books?id=PcWU2yxc8WkC&pg=PA385&lpg=PA385&dq=apl+quote+inch&source=bl&ots=TyvYBkiPWC&sig=6zU6AgRwOkD3_1-yZRmUmcm-L8k&hl=cs&ei=xTHoTLarN4jsOaXQoKAK&sa=X&oi=book_result&ct=result&resnum=1&ved=0CBkQ6AEwAA#v=onepage&q=apl%20quote%20inch&f=false>. ISBN 0-596-10121-X.
- diskuze.jak psát web.cz. *Diskusní fórum o webdesignu* [online]. 2004 [cit. 2010-07-01]. Dostupné z WWW: <<http://diskuse.jakpsatweb.cz/>>.
- VÍTA, Martin. *Interval.cz : Webdesignerovy poznámky z typografie* [online]. 2004 [cit. 2009-08-11]. Dostupné z WWW: <<http://interval.cz/clanky/webdesignerovy-poznamky-z-typografie/>>.

- *PHP : Hypertext Preprocessor* [online].
2001, 13 May 2011 [cit. 2010-10-12]. Dostupné z WWW:
<<http://cz.php.net>>.
- *Root.cz. Root.cz : Regulární výrazy v příkladech* [online].
2003 [cit. 2010-09-23]. Dostupné z WWW:
<<http://www.root.cz/clanky/regularni-vyrazy-v-prikladech/>>.
- *DAŘENA, František. Perl – regulární výrazy* [online].
2004 [cit. 2010-09-23]. Dostupné z WWW:
<<https://akela.mendelu.cz/~darena/Perl/regexp.html>>.

Příloha 1: Zdrojové kódy Litéry

- Zdrojové kódy:
https://is.muni.cz/auth/th/172528/fi_b/Litera.zip
- Litéra online (Korektor a Slovník):
<http://litera.odemne.com>

Ukázky zdrojového kódu

- Žlutá chyba – rule_class.php

```
/**
 * Žluté pravidlo dědí od pravidla bílého, obsahuje navíc obarvení
 **/
class RegYellow extends RegWhite{
    protected $paint; // co obarvovat
    protected static $errId=1; // id chyby - identifikace s formulářem pro opravy
    protected $idRule; // id pravidla - identifikace opravného formuláře

    /**
     * konstruktor získá hledaný text (zděděná vlastnost), co obarvit a id pravidla
     **/
    public function __construct($from,$paint,$idRule) {
        $this->from=$from; //zděděná vlastnost bílé chyby
        $this->paint=$paint;
        $this->id=RegWhite::$counter++; //zděděná vlastnost bílé chyby
        $this->idRule=$idRule;
    }

    /**
     * provedení regulárního výrazu nad daným textem
     **/
    public function execute(&$text) {
        $count=0; //počet varování vynulujeme
        $text=preg_replace_callback($this->from,array(&$this,'paint'),$text,-1,$count);
        // provedeme regulární výraz nad textem pomocí callback funkce, kterou voláme metodu
        paint
        return $count; //vracíme počet varování(= obarvení)
    }

    /**
     * nález podle $from obarví podle $paint
     **/
    protected function paint($matches) {
        // vrací nalezenou část, kterou zčásti obarví. $errId slouží k identifikaci
        s opravným formulářem. Používá se dále v javascriptu, kam se zde zapíše do
        proměnné idChyba
        // $errId se dále inkrementuje pro další obarvení
        return preg_replace($this->paint,
            '\3<a onclick="idChyba='.RegYellow::$errId.';ukaz(\'z\'.'.$this->idRule.'\'); this.className =\'yellowfocus\' id="chyba'.RegYellow::$errId++.'"
            class="yellow" title="Potenciální chyba">\1\4</a>\2',
            $matches[0]);
    }
}
```

- **Úprava textu, počet varování – korektor_class.php**

```

/* provedení úpravy textu */
private function execute() {
    $this->output=$this->text;    // do výstupu se uloží odeslaný text. Pravidla
                                // se aplikují na něj

    foreach($this->rules as $r) // první průchod pole z důvodu oddělení pořadí
                                // aplikace jednotlivých pravidel
        $this->count+=$r->execute($this->output,"rules");
                                // aplikace bílých pravidel. do $count se přičítá
                                // počet varování (u bílé chyby vždy 0)
    $this->output=htmlspecialchars($this->output);    // po aplikaci bílých chyb
                                                        // se přepíší klíčové HTML
                                                        // znaky na entity

    //druhý a třetí průchod postupně aplikuje žlutá a červená pravidla
    foreach($this->rules as $r)
        $this->count+=$r->execute($this->output,"rulesY");

    foreach($this->rules as $r)
        $this->count+=$r->execute($this->output,"rulesR");
}
/* vrací počet varovných chyb po úpravách */
public function getCount() {
    return $this->count;
}
/* vrací text vstupního formuláře */
public function getForm() {
    return $this->form;
}
}

```

- **Regulární výrazy pro znaménko minus – rules.php**

```

$minus=new Rule();    // minus
$minus->addWhite('/(\\pL)\\-/u','\\1 -');    // vloží mezeru do P-
$minus->addWhite('(/[0-9])[ ]?\\-[ ]?([.,;?!-])/u','\\1-\\2');    // C- a C - na C-
$minus->addWhite('(/[0-9])[ ]?\\-[ ]?([„»»])/u','\\1- \\2');    // C- a C - na C-

// vloží mezery do výrazu C-C, C -C, C- C (K1 12):

$minus->addWhite('(/[0-9])[ ]?\\-[ ]?([$€£-0-9])[ ]?\\-[ ]?([$€£-0-9])[ ]?\\-[ ]?([$€£-0-9])/u','\\1 - \\2 - \\3 - \\4');
$minus->addWhite('(/[0-9])[ ]?\\-[ ]?([$€£-0-9])[ ]?\\-[ ]?([$€£-0-9])/u','\\1 - \\2 - \\3');
$minus->addWhite('(/[0-9])[ ]?\\-[ ]?([$€£-0-9])/u','\\1 - \\2');
$minus->addYellow('(/[0-9][ ]?\\-[ ]?[^„»»$€£-0-9]/u','/([ ]?\\-)/u');
// Ž chyba výrazu C - NC, C- NC
$minus->addYellow('(/[0-9][ ]?\\-[ ^ .,;?!-0-9]/u','/([ ]?\\-)/u');
// Ž chyba výrazu C - NC, C- NC

$minus2=new Rule();    // minus - doplňující pravidla pro vyjádření záporné hodnoty
$minus2->addWhite('/'"[$€£%]"[ ]?\\-[ ]?([0-9])/u','\\1 - \\2');    // -C a - C na - C
$minus2->addWhite('/'"\\«<:--+×»"[ ]?\\-[ ]?([0-9])/u','\\1 -\\2');    // -C a - C na -C
$minus2->addWhite('/'"([+×·])"[ ]?\\-[ ]?/u','\\1 -');    // mat. znak vedle minus
$minus2->addWhite('/'([0-9])[ ]?([×·])[ ]-/u','\\1 \\2 -');    // vloží levou mezeru v C× I
                                                    // (mat. vedle minus)

$minus2->addWhite('/'(\\<)/[ ]-/u','\\1-');    // maže mezeru v < -
$minus2->addYellow('/(\\pL)[ ]-[ ]?([0-9]|(\\pL))/u','/(-[ ]?)/u');    // Ž chyba: P -
$minus2->addYellow('/'"\\«<:--+×·"[$€£%0-9][ ]\\-[ ]?([0-9])/u','/(-[ ]?)/u');
// Ž chyba: NC -C, NC - C
$minus2->addYellow('/'[^ <0-9]\\-[ ]?([0-9])/u','/(-[ ]?)/u');    // Ž chyba: NC -C, NC - C

$minus3=new Rule();    // minus - doplňující pravidla pro C-P
$minus3->addYellow('/[0-9][ ]?-( [ ]?| ?)(\\pL)/u','/([ ]?-[ ]?)/u');
// C- a C - na C- (kolize)

```

V ukázce lze vidět tři základní proměnné: \$minus1, \$minus2 a \$minus3. Více variant vzniká z důvodu existence tří typů formuláře pro opravu znaku minus. AddWhite, addYellow a addRed značí obarvení/opravy výrazu.

Příloha 2: Ukázka stránky Slovníku

Otazník a vykřičník

Otazník se používá v tázacích větách, vykřičník při důrazu. Obě znaménka větu ukončují, Tečka se za nimi nepíše (což vyjadřuje samotný puntík pod oběma znaky). Čárka se z důvodu zřetelnosti větné stavby může v ojedinělých případech za otazník psát. Ke znaménku se tiskne bez mezery.

1 Psaní mezer

Pro oba znaky platí stejná pravidla. Běžně se přiřazuje k poslednímu znaku a po nich následuje mezera. V případě vyjádření domněnky či údivu je lze klást do Závorky uprostřed věty. Mezeru ruší pravidlo závorek, dále viz Priorita.

Př. 1

1. Otázka
 2. Důraz
 3. Vyjádření domněnky
1. A zítra bude jak? Zítra bude hůř.
 2. Kolego, neblbni! To jsi nezvládl.
 3. Vstával už okolo desáté ráno (?).

2 Expresivní vyjádření

Jedná se o ztrojování znaků pro vyjádření velkého údivu/důrazu. Úzus a slušnost doporučuje používat spíše jen jeden znak (už samotný vykřičník budí dojem křiku na druhé). V ojedinělých případech – např. na reklamních poutačích – lze tolerovat trojici otazníků/vykřičníků, jiné a vyšší počty vzbuzují neohrabanost sdělujícího.

Př. 2

- Cestu kolem světa za 80 dní, to jsi se zbláznil???

3 Kombinace znaků

Chceme-li vyjádřit údiv i důraz zároveň, lze znaky kombinovat. Mezera se mezi ně nevkládá. V jakém pořadí je zapisovat není přesně dané, v celém dokumentu by se ale mělo postupovat stejně.

Př. 3

- Cestu kolem světa za 80 dní?!

4 Časté chyby

Př. 4

1. Nevhodná mezera
2. Přílišná fantazie (velké množství a vícenásobná kombinace)
 1. Cestu kolem světa za 80 dní? Blázne.
 2. Zbláznil ses????!?!?