

1 Základní charakteristika projektu

Název projektu:

Návrh, implementace a dokumentace **dvou skriptů** v jazyce **Perl** a **Python 3**

Revize zadání: Jako každý dokument i toto zadání bude mít jistě pár chyb, a tudíž prosím o pomoc jich co nejvíce odhalit do konce února, kdy se stane zadání naprosto definitivní a neměnné. V žádném případě nepůjde o změny zásadní, ale spíše kosmetické, takže můžete začít okamžitě pracovat na řešení projektu.

2012-02-02: Vznik zadání a vytvoření konkrétních zadání úloh

2012-02-10: Zveřejnění zadání, registrace bude spuštěna koncem týdne

Rámcové zadání: Projekt je individuální a sestává se z naprogramování skriptů pro **dvě úlohy**, které si student vybírá z možných variant (viz sekce 2). První skript bude povinně implementován v jazyce **Perl** a druhý v jazyce **Python 3** (možnost použití technik OOP v kombinaci s dalšími programovacími paradigmaty).

Skripty jsou aplikace pro příkazovou řádku (tzv. filtry) se společným základem zadání (viz sekce 3). Každá úloha (viz sekce 7) má identifikátor ze tří písmen (např. CHA, CST, CSV, DKA, JSN, MKA, SYN, XQR, XTD).

2 Odevzdání

Forma odevzdání: Projekt se odevzdává individuálně prostřednictvím IS FIT v předmětu IPP (odevzdání emailem není možné, a to ani po termínu). Všechny odevzdané soubory k dané úloze budou zkomprimovány programem TAR+GZIP či ZIP do jediného archívu pojmenovaném `xlogin00-XYZ.zip` nebo `xlogin00-XYZ.tgz`, kde `xlogin00` je Váš login a `XYZ` je identifikátor úlohy implementované a dokumentované v daném archívu. Velikost každého archívu bude omezena informačním systémem (pravděpodobně na 1 MB). Archív nesmí obsahovat speciální či spustitelné soubory. Názvy všech souborů mohou obsahovat pouze písmena anglické abecedy, číslice, tečku, pomlčku a podtržítko. Archív nebude obsahovat žádné adresáře výjma adresářů pro přiložené povolené knihovny, které nejsou nainstalovány na serveru Merlin.

V rámci prvního termínu se ve WIS přihlásíte na variantu (dvě úlohy s určením konkrétních implementačních jazyků). Do prvního termínu budete také odevzdávat archív pro první úlohu v jazyce Perl. Několik dní po uzavření prvního termínu bude otevřen druhý termín pro odevzdání archívu pro druhou úlohu v jazyce Python 3.

Termíny odevzdání:

První úloha v pátek **6. dubna** 2012 do 23:59:59,

Druhá úloha v pátek **27. dubna** 2012 do 23:59:59.

Dodatečné informace a konzultace k projektu z IPP:

- Často kladené otázky k projektu z předmětu IPP, viz Wiki předmětu v informačním systému
- Fórum předmětu IPP pro ak. rok 2011/2012, témata IPP: **Projekt.***
- U cvičících zodpovědných za jednotlivé úlohy (viz zadání konkrétních úloh)
- Zbyněk Křivka: v konzultačních hodinách nebo po dohodě e-mailem (uvádějte předmět začínající "IPP:"), viz <http://www.fit.vutbr.cz/~krivka>
- Dušan Kolář: po dohodě e-mailem (uvádějte předmět začínající "IPP:"), viz <http://www.fit.vutbr.cz/~kolar> (jen v závažných případech)

Hodnocení:

Výše bodového hodnocení projektu v předmětu IPP je **celkem 20 bodů**.

Každý skript včetně dokumentace bude hodnocen **10 body**. Z toho bude dokumentace až za 2 body, avšak maximálně 30 % hodnocení skriptu (tedy v případě neodevzdání funkčního skriptu bude samotná dokumentace hodnocena 0 body). Body za úlohu se zaokrouhlují na jedno desetinné místo.

Za řešení je možné získat až 25 % bonusových bodů, kdy se Váš maximální bonusový zisk počítá z aktuálního základního bodového zisku (např. při získání 20 bodů může být bonus celkem až 5 bodů, při získání 16 bodů může být bonus až 4 body, hranice se zaokrouhluje nahoru). Takový bonus je možné získat pouze za obzvláště kvalitní a podařené řešení některého z registrovaných rozšíření (viz zadání jednotlivých úloh v sekci 7), kvalitativně nadprůměrnou účast na fóru projektu apod.

Skripty budou spouštěny příkazem: *interpret skript parametry*, kde *interpret* bude `perl` nebo `python3`, jméno *skript* závisí na dané úloze. Hodnocení funkčnosti bude zajišťovat automatizovaný nástroj. Kvalitu dokumentace, komentářů a strukturu zdrojového kódu budou hodnotit cvičící.

Podmínky pro opakující studenty ohledně případného uznání hodnocení loňského projektu najdete v často kladených otázkách na Wiki stránkách předmětu.

Způsob odevzdání: Skript a dokumentace budou součástí řešení každé úlohy. Každá úloha bude odevzdána ve zvláštním archívu (viz forma odevzdání, která je popsána na začátku této sekce). Skript bude umístěn v kořenovém adresáři odevzdávaného systému souborů a odtud se bude i spouštět.

Pokud tyto pokyny nebudou dodrženy, projekt se považuje za neodevzdáný! Dále je třeba odevzdat zadanou část projektu v daném termínu (viz výše). Pokud tomu tak nebude, je projekt opět považován za neodevzdáný, nebo minimálně se dají očekávat významné bodové srážky (cca bod za každou započatou hodinu zpoždění). Stejně tak pokud se bude jednat o plagiátorství jakéhokoli druhu, je projekt ohodnocen 0 body, nebude udělen zápočet a bude zváženo podání návrhu na disciplinární řízení.

Projekt je hodnocen jako neodevzdaný (shrnutí), pokud:

- nebude doručen včas (viz termíny odevzdání jednotlivých úloh)
- nebude doručen v předepsaném formátu (viz požadavky na odevzdání)
- nebude možné interpretovat skript tak, jak je uvedeno v zadání (např. je nutný další zásah do skriptu, ruční nastavení přístupových cest, apod.)
- nebude skript vykazovat ani základní funkčnost požadovanou zadáním
- půjde o plagiát v jakékoliv podobě¹

Pokud student nezachová kterékoliv z výše uvedených pravidel, zadavatelé projektu mají právo jej hodnotit nulou.

Pokud máte jakékoliv dotazy, problémy, či nejasnosti ohledně tohoto projektu, tak po přečtení často kladených otázek a fóra předmětu (využívejte i možnosti hledání na fóru), neváhejte napsat na fórum (u obecného problému, jenž se potencionálně týká i vašich kolegů) či kontaktovat zodpovědného cvičícího (viz zadání úlohy), garanta projektu (v případě individuálního problému, e-mail: krivka@fit.vutbr.cz), nebo nouzově garanta předmětu (e-mail: kolar@fit.vutbr.cz), kdy do předmětu uveďte na začátek řetězec "IPP:". Na problémy zjištěné v řádu hodin až jednotek dní před termínem odevzdání nebude brán zřetel, začněte proto projekt řešit s dostatečným předstihem.

Pokud se při vytváření projektu dopustíte jakéhokoliv prohřešku proti akademické morálce (např. plagiátorství), Váš přestupek bude ohlášen garantovi předmětu a může vyústit ve vyloučení z akademické obce a tím pádem z celého VUT.

Varianty projektu: Projekt se skládá z několika úloh (viz sekce 7), z nichž si student vybírá dvojici úloh (první implementuje v Perlu, druhou v Pythonu 3) z následujících povolených variant:

1. CHA+XQR (C Header Analysis v Perlu, XML Query v Pythonu 3)
2. XTD+CST (XML2DDL v Perlu, C Stats v Pythonu 3)
3. JSN+SYN (JSON2XML v Perlu, Zvýraznění syntaxe v Pythonu 3)
4. CSV+SYN (CSV2XML v Perlu, Zvýraznění syntaxe v Pythonu 3)
5. JSN+MKA (JSON2XML v Perlu, Minimalizace konečného automatu v Pythonu 3)
6. CSV+DKA (CSV2XML v Perlu, Determinizace konečného automatu v Pythonu 3)
7. SYN+CHA (Zvýraznění syntaxe v Perlu, C Header Analysis v Pythonu 3)
8. CST+XTD (C Stats v Perlu, XML2DDL v Pythonu 3)
9. SYN+XTD (Zvýraznění syntaxe v Perlu, XML2DDL v Pythonu 3)
10. XQR+CHA (XML Query v Perlu, C Header Analysis v Pythonu 3)
11. XTD+DKA (XML2DDL v Perlu, Determinizace konečného automatu v Pythonu 3)

¹Mimo jiné je za plagiát považováno i dílo obsahující shody pouze v částech s jiným dílem, kde není prokazatelně stejný autor (včetně převzetí zdrojových kódů z veřejných zdrojů). Například převzetí cizího kódu, změna jmen identifikátorů a komentářů je jednoznačně plagiátorstvím. Stejně tak krádež zdrojového kódu je považována za důvod k zahájení disciplinárního řízení.

3 Společný základ zadání všech úloh

1. Implementujte v daném programovacím jazyce programové vybavení (skript) pro řešení přidělené úlohy.
2. Toto programové vybavení doplňte o vhodnou implementační dokumentaci.

3.1 Dokumentace

Implementační dokumentace (dále jen dokumentace) musí být stručným a uceleným průvodcem **vašeho způsobu řešení** zadané úlohy. Bude vytvořena ve formátu **PDF**. Jakékoliv jiné formáty dokumentace než PDF budou ignorovány, což povede ke ztrátě bodů za dokumentaci. Dokumentaci je možné psát buď česky (s diakritikou, formálně čistě), nebo anglicky (formálně čistě). Dokumentace může být doplněna o graf pro použitý konečný automat, pravidla Vámi vytvořené gramatiky nebo popis jiných speciálních technik a algoritmů. Nicméně **nesmí obsahovat ani částečnou kopii zadání**.

Dokumentace bude dále popisovat způsob a Váš specifický postup řešení (např. řešení sporných případů nedostatečně upřesněných zadáním, konkrétní řešení rozšíření) a celkovou filozofii návrhu. **Rozsah** celého dokumentu je minimálně jedna celá strana a maximálně dvě strany A4 (10-bodové písmo Times New Roman a případně Courier pro identifikátory a skutečně krátké úryvky kódu; nevkládejte žádnou zvláštní úvodní stranu, obsah ani závěr) pro řešení každé úlohy. V rozumné míře je vhodné používat nadpisy první a druhé úrovně (12-bodové a 11-bodové písmo Times New Roman) pro vytvoření logické struktury dokumentace.

Nadpis a hlavička dokumentace bude na prvních třech řádcích obsahovat:

```
Dokumentace úlohy %XYZ%: %task% v %language% do IPP 2011/2012
Jméno a příjmení: %name_surname%
Login: %xlogin00%
```

kde %XYZ% je identifikátor vaší úlohy, %task% jméno vaší úlohy, %language% implementační jazyk dané úlohy (Perl nebo Python 3), %name_surname% vaše jméno a příjmení a váš login %xlogin00%.

Dokumentace bude v hlavním adresáři odevzdaného archívu a pojmenována XYZ-doc.pdf, kde XYZ je identifikátor úlohy.

V rámci dokumentace bude hodnoceno i komentování zdrojového kódu skriptu (minimálně každá funkce a modul by měly mít svůj komentář o jejich funkci a parametrech; u složitějších funkcí dokumentujte i omezení na vstupy či výstupy). Každý skript (i případný pomocný) musí v řádkovém komentáři na třetím řádku² zdrojového textu obsahovat zakomentovaný dvojtečkou oddělený identifikátor úlohy a login autora skriptu, čímž stvrzujete výhradní autorství daného skriptu. Například (v jazyce Perl i Python 3):

```
#XQR:xnovak99
```

3.2 Programová část

Zadání úloh vyžadují implementaci textového filtru nebo jiné aplikace příkazové řádky³, která má vstupní parametry a je definováno, jakým způsobem manipuluje se vstupy a výstupy. Skript nesmí spouštět žádné další procesy či dokonce jiné příkazy operačního systému. Veškeré chybové hlásky,

²Z důvodu že v Pythonu může být na druhém řádku informace o kódování zdrojového textu.

³konzolové aplikace

varování a ladicí výpisy směřujte pouze na standardní chybový výstup, jinak pravděpodobně nedodržíte zadání kvůli modifikaci definovaných výstupů (ať již do externích souborů nebo do standardního výstupu). Jestliže proběhne činnost skriptu bez chyb, vrací se návratová hodnota 0 (nula). Jestliže došlo k nějaké chybě, vrací se výstupní návratová hodnota větší jak nula. Některé typy chyb mají závazné chybové návratové hodnoty:

1. špatný formát parametrů skriptu nebo byla použita zakázaná kombinace parametrů;
2. neexistující zadaný vstupní soubor nebo chyba otevření zadaného vstupního souboru pro čtení;
3. chyba při pokusu o otevření zadaného výstupního souboru pro zápis;
4. chybný formát vstupního souboru (pokud nebyla bezchybovost vstupu zaručena v konkrétní úloze nebo nebyla specifikována jiná návratová hodnota pro konkrétní chybu v konkrétní úloze);
10. - 99. návratové kódy chyb specifických pro jednotlivé úlohy.
100. - 255. ostatní typy chyb, které nejsou postihnuty žádnou z předchozích variant.

Pokud zadání konkrétní úlohy nestanoví jinak, tak veškeré vstupy a výstupy jsou v kódování UTF-8. Implicitní nastavení **locale na serveru Merlin bude pro účely projektu z IPP změněno**⁴ na `LC_ALL=cs.CZ.UTF-8`. V případě XML souborů není třeba uvažovat deklarace jmenných prostorů. Všechna vstupní XML budou v tomto smyslu nekolizní a nebudou jmenné prostory využívat. Nicméně deklarace se v XML souboru vyskytnout může a tu potom ignorujte. Je-li výstupem soubor ve formátu XML, tak bude většinou porovnáván s referenčními výsledky pomocí nástroje A7Soft JExamXML⁵.

Každý skript se bude jmenovat podle identifikátoru úlohy převedeného na malá písmena a příponu⁶ bude mít podle daného skriptovacího jazyka dle platných zvyklostí (Perl (`.pl`), Python 3 (`.py`)). Vyhodnocení skriptů bude prováděno na serveru Merlin s aktuálními verzemi interpretů (dne 9. 2. 2012 byl na tomto serveru nainstalován perl verze 5.8.8 a python3⁷ verze 3.2.2).

K řešení lze využít standardně předinstalované knihovny obou jazykových prostředí na serveru Merlin. V případě využití jiné knihovny kromě knihovny podporující načítání formátů XML, popř. CSV či JSON je třeba konzultovat s garantem projektu (především z důvodu, aby se řešení projektu použitím vhodné knihovny nestalo zcela triviálním).

Každý skript bude pracovat s těmito společnými parametry:

- `--help` vypíše na standardní výstup nápovědu skriptu, kterou lze převzít z tohoto zadání (lze odstranit diakritiku, případně přeložit do angličtiny dle zvoleného jazyka dokumentace). Tento parametr nelze kombinovat s žádným dalším parametrem, jinak skript ukončete s chybou.
- `--input=filename` zadaný vstupní soubor může být zadán relativní cestou⁸ nebo absolutní cestou; v případě, že by název či cesta obsahovala mezeru nebo uvozovky, tak musí být uvedena celá cesta i se jménem souboru v uvozovkách (uvozovky ve jméně souboru není třeba uvažovat); chybí-li tento parametr, tak je uvažován standardní vstup.

⁴Změna prostředí je nezbytná, aby bylo možné používat a správně zpracovávat parametry příkazové řádky v UTF-8. Pro správnou funkčnost je třeba mít na UTF-8 nastavenou i znakovou sadu konzole (např. u programu PuTTY v kategorii *Window.Translation* nastavíte *Remote character set* na UTF-8). Pro změnu ovlivňující aktuální sezení lze využít příkaz `export LC_ALL=cs.CZ.UTF-8`

⁵Parametry nastavení nástroje A7Soft JExamXML při vyhodnocování budou zveřejněny na fóru k projektu.

⁶Pomocné skripty nebo knihovny mohou mít jinou příponu. Např. `.pm` pro Perl.

⁷Upozornění: Na serveru Merlin je třeba dodržet testování příkazem `python3`, protože pouhým `python` se spouští stará nekompatibilní verze!

⁸Relativní cesta nebude obsahovat zástupný symbol `~`.

- `--output=filename` zadaný výstupní soubor může být zadán relativní cestou nebo absolutní cestou; v případě, že by název či cesta obsahovala mezeru, tak musí být uvedena cesta i se jménem souboru v uvozovkách (uvozovky ve jméně souboru není třeba uvažovat); chybí-li tento parametr, tak je výstup přeměřován na standardní výstup. Existuje-li již výstupní soubor, bude bez varování přepsán, jinak bude vytvořen nový soubor.

Kombinovatelné parametry skriptů jsou odděleny alespoň jedním bílým znakem a mohou být uváděny v libovolném pořadí, pokud nebude řečeno jinak. Zakázané kombinace parametrů ukončí skript s chybou. Pokud to nekoliduje se zadáním (např. se zakázanou kombinací parametrů) a domluvíte-li se se zodpovědným cvičícím dané úlohy, je možno implementovat i další parametry skriptů nad rámec zadání. Není-li u úlohy specifikováno jinak, tak se vstupní parametry nesmí opakovat! Například nepovolujeme zadat dvakrát stejný nebo i různý soubor parametrem `--input`.

Dle konvencí unixových systémů budou v některých úlohách zástupné zkrácené (s jednou pomlčkou) a dlouhé parametry (se dvěma pomlčkami), které lze se zachováním sémantiky zaměňovat (tzv. alias parametry).

Reference:

- Extensible Markup Language (XML) 1.0. W3C. World Wide Web Consortium [online]. 5. vydání. 26. 11. 2008 [cit. 2012-02-02]. Dostupné z: <http://www.w3.org/TR/xml/>
- A7Soft JExamXML is a java based command line XML diff tool for comparing and merging XML documents. c2012 [cit. 2012-02-07]. Dostupné z: <http://www.a7soft.com/jexamxml.html>

4 Registrovaná rozšíření

V případě implementace některých registrovaných rozšíření za bonusové body bude odevzdaný archiv obsahovat soubor `rozsireni`, ve kterém uvedete na každém řádku identifikátor jednoho implementovaného rozšíření⁹ (řádky jsou ukončeny znakem LF). V průběhu řešení mohou být zaregistrována nová rozšíření úlohy za bonusové body (viz fórum předmětu IPP). Cvičícím můžete přes fórum zasílat návrhy na nová netriviální rozšíření, která byste chtěli navíc implementovat. Cvičící rozhodnou o přijetí/nepřijetí rozšíření a hodnocení rozšíření dle jeho náročnosti včetně přiřazení unikátního identifikátoru. Implementovaná rozšíření neidentifikovaná v souboru `rozsireni` nebudou hodnocena.

5 Nejčastější chyby v minulých letech

Tato sekce popisuje nejčastější chyby (neseřazeno) studentů v projektech z minulých let, kterým se lze snadno vyhnout.

1. Nedodržování návratové hodnoty. Studenti si často pletli standardní výstup a návratovou hodnotu vykonání skriptu. Automatické opravující skripty validují obojí.
2. Laxní až vágní výstupy. Hlavním hodnotícím prostředkem je automatický překlad, spouštění nad mnoha testy a na výsledcích založené hodnocení. Pokud jsou vaše výsledky automaticky neporovnatelné s referenčními, nebudou vám body přiděleny.

⁹Identifikátory rozšíření jsou uvedeny u konkrétního rozšíření tučně (viz zadání jednotlivých úloh).

3. Pokud Váš skript neumí otevřít a načíst soubor ze zadané relativní (např. začínající předponou „./“) či absolutní cesty a uložit výsledek do zadané relativní nebo absolutní cesty, nelze Vaši úlohu vůbec hodnotit. Některé úlohy jsou více testovány se zadáním relativních cest a některé zase s využitím absolutních cest, takže vždy podporujte všechny způsoby načítání dat (především ze souborů), abyste se vyvarovali zbytečně velkým bodovým srážkám.
4. Projekt byl nefunkční kvůli neotestování na serveru Merlin. Většina studentů vytváří poslední verzi projektu na poslední chvíli před odevzdáním a neprovede pak otestování na referenčním serveru Merlin, kde bude také probíhat opravování. Dodatečné zásahy do kódu projektu jsou však přísně penalizovány včetně těch celkem banálních jako změna kódování, protože testování a ladění výsledné aplikace je také součástí tohoto projektu.
5. Studenti podcenili své znalosti daného implementačního jazyka a jeho povolených knihoven.
6. Nepochopení lexikálních pravidel jazyka Python. Ve zdrojovém textu nelze mixovat odsazení příkazů pomocí mezer a pomocí tabulátorů. Doporučujeme zvolit vhodný editor, který vám umožní tyto problémy ohlídat a všechny tabulátory nahrazovat mezerami.
7. Nekvalitní dokumentace. Dokumentaci čteme podstatně pozorněji než potom zdrojový kód, který pouze projdeme pro ohodnocení přehlednosti a základní struktury.

6 Obecné slovo závěrem

Na stránkách předmětu IPP a v příspěvcích na fóru předmětu IPP se budou v prvních týdnech řešení projektu objevovat doporučení a návody na vhodné řešení zadání. Tato doporučení však nebudou závazná a jejich nedodržení je vhodné obhájit v dokumentaci projektu.

Projekt z předmětu IPP si klade za cíl vás seznámit se dvěma skriptovacími jazyky a možnostmi kombinování různých programovacích paradigmat. Dále bychom rádi studenty přesvědčili o vhodnosti různých jazyků pro různé úlohy. Formou diskuzí na fóru potom budeme částečně simulovat komunikaci Vás návrhářů a programátorů se zadavatelem či zákazníkem.

V samotném závěru bych chtěl zdůraznit, že všechny úlohy budou opravovány automatizovaně a tudíž je nezbytné stoprocentní splnění formálních požadavků jak na formáty vstupů a výstupů, tak i na samotné transformace. Pro usnadnění precizního pochopení zadání bude zveřejněna základní sada příkladů vstupů a výstupů jednotlivých skriptů. Je ale především na Vás samotných, abyste si vytvořili i vlastní rozsáhlé balíky testů.

7 Zadání jednotlivých úloh

Zadání jednotlivých úloh jsou nově umístěna ve zvláštních souborech:

1. cha.pdf: C Header Analysis
2. cst.pdf: C Stats
3. csv.pdf: CSV2XML
4. dka.pdf: Determinizace konečného automatu
5. jsn.pdf: JSON2XML

- 6. mka.pdf: Minimalizace konečného automatu
- 7. syn.pdf: Zvýraznění syntaxe
- 8. xqr.pdf: XML Query
- 9. xtd.pdf: XML2DDL

Opakující studenti POZOR!!! Kromě zcela nových úloh bylo aktualizováno a často i pozměněno zadání úloh starých. Na vaše omyly ohledně implementace zadání z minulého roku nebude brán žádný zřetel.