

Závazné pokyny pro projekty FLP 2013/2014

Peter Matula
email: `imatula@fit.vutbr.cz`

Martin Hyrš
email: `ihyrs@fit.vutbr.cz`

20. února 2014

1 Obecné informace

Součástí předmětu FLP jsou dva projekty zaměřené na praktické použití funkcionálních a logických jazyků.

	Projekt 1. (funkcionální)	Projekt 2. (logický)
Implementační jazyk:	Haskell	Prolog
Datum odevzdání:	20.04.2014	04.05.2014
Standardní hodnocení:	12 bodů	8 bodů
Hodnotící:	Matula	Hyrš

(počty bodů jsou uvedeny pro jednoho řešitele).

Projekt budou řešit čtyřčlenné týmy. Týmy s jiným počtem členů nejsou přípustné. Jeden z členů týmu je vedoucím. Registrace týmu se provádí v informačním systému fakulty (dále jen IS) pomocí zvláštního termínu.

2 Hodnocení

Za funkcionální projekt si tým rozdělí při perfektním splnění požadavků zadání $4 \times 12 = 48$ bodů, za logický $4 \times 8 = 32$ bodů (při dodržení požadavků v části 4.1). Za inovativní přístup či obzvláště povedené řešení je možné získat navíc prémiové body.

Za každý projekt může jednotlivec získat maximálně 125% standardního hodnocení, tedy nelze přesáhnout hranici 15 bodů u funkcionálního a 10 bodů

u logického projektu. Absolutní maximum získatelné za projekty je tedy 25 bodů.

3 Práce na projektu

Každému týmu bude vytvořen repozitář na školním serveru za účelem efektivní práce v týmu a kvůli řešení sporů ohledně autorství a rozdělení bodů. Repozitář bude vždy obsahovat veškerá aktuální data.

Změny bude vždy přidávat jejich autor a budou patřičně popsány. Všichni členové týmu si budou udržovat přehled o práci ostatních, k čemuž je nástroj pro správu verzí velmi vhodný. V případě selhání spolupráce se tím obrátí na opravujícího, se kterým nastálou situaci vyřeší.

4 Odevzdávání

Odevzdávání probíhá výhradně prostřednictvím IS do datového skladu předmětu FLP. Za celý tým odevzdá každý projekt pouze vedoucí.

Všechny odevzdané soubory budou zkomprimovány do jediného archivu zabaleného metodou ZIP či TAR+GZIP, který bude pojmenován podle vzoru

`f1p-{fun|log}-login.{tgz|zip}`

Například při odevzdání funkcionálního projektu vedoucím s loginem `xnovak00` zabalené programem ZIP bude odevzdán soubor

`f1p-fun-xnovak00.zip`

Archiv nesmí obsahovat zbytečné soubory (například spustitelné).

Doporučujeme, aby byla k projektu přiložena **stručná** dokumentace, ideálně jako krátký textový soubor `README`. Rovněž je záhodno pečlivě komentovat zdrojové kódy.

Je možné, nikoliv však nutné, projekt osobně předvést opravujícímu (na cvičení nebo konzultaci) – v takovém případě se lze dohodnout na dřívějším ohodnocení projektu.

Oba projekty je třeba odevzdat v příslušném termínu (vizte výše). Pokud tomu tak nebude, je část považována za neodevzdanou.

V případě plagiátorství jakéhokoliv druhu bude projekt hodnocený nula body, navíc celému týmu nebude udělen zápočet a bude zváženo zahájení disciplinárního řízení. Pozor rovněž na autoplagiátorství!

4.1 Rozdelení bodů

Odevzdaný archiv bude **povinně** obsahovat textový soubor `rozdeleni`, ve kterém bude zohledněn přínos jednotlivých členů týmu. Na každém řádku bude uveden login jednoho člena týmu, bez mezery následován dvojtečkou a požadovaným celočíselným počtem procent bodů (bez uvedení znaku '%'). Každý řádek (i poslední) je poté ihned ukončen jedním znakem LF (ASCII hodnota 10). Jeden řádek tohoto souboru tedy bude vypadat například takto:

```
xnovak00:25
```

Soubor musí obsahovat všechny členy týmu. Součet všech procent musí být roven stu. Maximální podíl jednoho člena týmu je 40%, minimální 15%. Pokud některý z členů týmu bude pracovat méně, je třeba tuto skutečnost oznámit do dvou týdnů před odevzdáním. V takovém případě je po souhlasu opravujícího možné zadat i menší dohodnutý podíl.

V případě chybného rozdelení budou body děleny rovnoměrně.

5 Implementace

Referenčním strojem je server `merlin` bez jakýchkoliv speciálních nastavení. Proto doporučujeme projekt před odevzdáním na referenčním stroji vyzkoušet. Pokud projekt nebude fungovat na referenčním stroji, ztrácí tým právo na reklamací výsledků.

Součástí řešení bude soubor `Makefile` (projekty budou překládány pomocí `gmake`), ve kterém lze nastavit případné parametry překladu (viz `info gmake`). Pokud soubor pro sestavení cílového programu nebude obsažen, nebo se na jeho základě nepodaří sestavit cílový program, nebude projekt hodnocen. Cílový program pojmenujte loginem vedoucího týmu.

Úvod **všech** zdrojových textů musí obsahovat název projektu, přihlašovací jména a jména všech autorů.

Pokud není řečeno jinak, vytvořte program ve formě konzolové aplikace, není požadováno grafické uživatelské rozhraní.

5.1 Funkcionální projekt

Funkcionální část bude vypracována v jazyce Haskell. Pro opravování bude použit překladač GHC (Glasgow Haskell Compiler) s dalšími nástroji a knihovnami, které jsou součástí Haskell Platform nainstalované na serveru `merlin`. Pro vývoj doporučujeme použít právě tuto verzi Haskell Platform.

Zvažte možnost dodržovat formát komentářů pro nástroj `haddock`, kterým je pak možné automaticky vygenerovat programovou dokumentaci.

Konkrétní požadavky budou specifikovány v příslušné variantě zadání.

5.2 Logický projekt

Logická část bude vypracována v jazyce Prolog. Pro opravování bude použit SWI-Prolog ve verzi nainstalované na serveru `merlin` – pro vývoj doporučujeme právě tuto stabilní verzi.

Konkrétní požadavky budou specifikovány v příslušné variantě zadání.

6 Konzultace

Budou probíhat osobní formou na cvičení a po předchozí domluvě emailem. Cvičící může rovněž v rámci svých časových možností poskytovat konzultace prostřednictvím fóra předmětu v IS, které ovšem prioritně slouží pro komunikaci mezi studenty. Dále lze využít fakultní elektronickou poštu.

Konzultace k projektům poskytují pouze Ing. Matula (Haskell) a Ing. Hyrš (Prolog), neobtěžujte jimi ostatní pedagogy.

7 Uznávání projektů

Student žádá o uznání projektu z minulých let prostřednictvím příslušného termínu v IS (v rámci registrace týmů). Musí splňovat veškeré uvedené podmínky.

Uznávání projektů je zcela v kompetenci doc. Koláře.