

ICP: zadání týmového projektu

Toto je draft (2018-02-21) -- Posílejte připomínky

Název projektu

Aplikace BlockEditor

Základní zadání (je téměř shodné s předmětem IJA)

Navrhněte a implementujte aplikaci pro návrh a editaci blokových schémat.

Poznámka: Zadání definuje podstatné vlastnosti aplikace, které musí být splněny. Předpokládá se, že detaily řešení si doplní řešitelské týmy.

Týmy budou dvoučlenné, výjimečně může být tým i tříčlenný a bude řešit rozšířené zadání (viz níže). Jeden člen týmu je vedoucí a ten projekt odevzdá podle pokynů uvedených níže.

Specifikace požadavků

1. Základní požadavky
 - aplikace umožní vytvářet, editovat, ukládat a načítat bloková schémata
 - každé schéma má svůj jedinečný název
 - vytvořená schémata lze uložit a opětovně načíst
 - schéma je složeno z bloků a propojů mezi bloky
2. Bloky
 - každý blok má definované vstupní a výstupní porty
 - s každým portem je spojen typ, který je reprezentován množinou dat v podobě dvojice název->hodnota; hodnota bude vždy typu double
 - bloky je možné spojit pouze mezi výstupním a vstupním portem
 - každý blok obsahuje výpočet (vzorce), které transformují hodnoty ze vstupních portů na hodnoty výstupních portů
3. Propojení mezi bloky
 - systém kontroluje kompatibilitu vstupního a výstupního portu propoje (stejný typ dat)
 - typ dat je přiřazen propoji automaticky podle spojených portů
4. Výpočet
 - po sestavení (načtení) schématu je možné provést výpočet
 - systém detekuje cykly v schématu; pokud jsou v schématu cykly, nelze provést výpočet
 - systém požádá o vyplnění dat vstupních portů, která nejsou napojena a poté postupně provádí výpočty jednotlivých bloků podle definovaných vzorců v každém bloku
 - při výpočtu se vždy zvýrazní blok, který je právě přepočítáván
 - výpočet lze krokovat (jeden krok = přepočet jednoho bloku)
5. Další podmínky
 - najetím myši nad propoj se zobrazí aktuální stav dat
 - zvažte způsob jednoduchého rozšiřování systému o nové bloky a data

Rozšíření pro případný tříčlenný tým

1. Je možné současně pracovat na více schématech.

2. Složené bloky
 - každé schéma může mít vstupní a výstupní porty a tím tvořit jeden složený blok
 - vytvořené schéma může být součástí jiného schématu jako složený blok; v takovém případě je zobrazen pouze jako blok s příslušnými vstupy a výstupy, lze zobrazit interní schéma tohoto složeného bloku
3. Realizujte způsob jednoduchého rozšiřování systému o nové bloky a data.

Součást odevzdání

- připravte předem alespoň 5 různých bloků a 3 různé typy (množiny) dat

Doporučení

- zamyslete se nad použitím vhodných návrhových vzorů

Další požadavky

- Zadání definuje podstatné vlastnosti aplikace, které musí být splněny. Předpokládá se, že details řešení si doplní řešitelské týmy.
- Při návrhu postupujte tak, aby výsledný program byl dobře použitelný. Hodnotit se bude nejen to, zda splňuje výše uvedené požadavky, ale také JAK je splňuje.
- Návrh musí oddělit vlastní model aplikace od GUI (navrhněte vhodné rozhraní).
- Návrh programu zdokumentujte ve zdrojovém kódu ve formátu pro program "doxygen".
- Používejte vhodné návrhové vzory (Design Patterns) - viz literatura.
- Pro vytváření menu, dialogů a dalších komponent grafického uživatelského rozhraní (GUI) použijte toolkit **Qt minimálně ve verzi 5.5** (nainstalováno v učebnách a na serveru "merlin" v adresáři /usr/local/share/Qt-5.5.1/). Pokud pro vývoj použijete verzi novější/starší než je na učebnách, předpokládám, že zajistíte zpětnou/dopřednou kompatibilitu vašich programů. (V případě problémů, počítejte s nutností předvedení překladu/sestavení/běhu aplikace na vašem notebooku. Toto se nedoporučuje.)

Protože toolkity jsou přenositelné na více platform, měly by vaše programy fungovat minimálně na Linuxu i na Windows. (Nejen Java umožňuje psaní přenositelných aplikací.) Je jedno na které platformě proběhne vývoj aplikací, jen dbejte na pravidla psaní přenositelných programů.

(Testovat se bude na Linuxu = server merlin, ale Makefile nesmí být závislý na konkrétním adresáři s lokální instalací toolkitu.)

Na vytvoření GUI dialogů a menu můžete použít interaktivní editory/generátory.

- Použijte C++14 (nebo C++17). (Pokud použijete knihovny z Boost, tak použité soubory přibalte. Existuje nástroj [BCP](#) pro výběr podmnožiny Boost.).
- Programy budou vytvořeny zadáním příkazu "make" v adresáři s rozbalenými zdrojovými kódy projektu (Pozor - bude se automaticky překládat). Musíte vytvořit soubor Makefile, který nesmí obsahovat absolutní cesty k souborům. Výsledný program se bude jmenovat "blockeditor". Pro testovací účely půjde vše spustit použitím "make run".
- Součástí projektu bude programová dokumentace vytvořená programem Doxygen (konfigurační soubor nastavte na HTML výstup se zahrnutím zdrojových textů: SOURCE_BROWSER = YES). Dokumentace se vygeneruje automaticky (= nezabalovat do archivu) při zadání "make doxygen" a bude uložena v podadresáři doc.

Poznámky k odevzdávání

- Termín odevzdání je uveden v IS.
- Řešení zabalené ve formátu .zip (nebo .tar.gz) odevzdá pouze vedoucí týmu do IS a to s dostatečným předstihem před uzávěrkou. (Nikdo nebude psát, že neodevzdal, protože *** a posílat to poštou.)

Jméno odevzdaného souboru musí mít tvar: xname01-xname02.zip, kde xname01 je jméno vedoucího.

- **POZOR:** Archiv může mít maximálně 2MB, aby mohl být vložen do IS.
- Odevzdaný archiv obsahuje minimálně tuto adresářovou strukturu:

```
./src/*      - zdrojové texty
./examples/* - příklady (pro testování)
./doc/       - dokumentace (bude později generován: "make doxygen")
./README.txt - základní přehled o co jde, autoři, poznámky, ...
./Makefile   - zavolá rekurzivně make na src/Makefile, případně
               na generování dokumentace atd
```

- Neodevzdávejte nic, co lze vytvořit pomocí "make" (spustitelné programy ani dokumentaci, kterou lze vygenerovat použitím "make doxygen").
Poznámka: Příkaz "make clean" všechny produkty překladu smaže, "make pack" vytvoří archiv pro odevzdání - zkontrolujte, zda obsahuje vše potřebné rozbalením v prázdném adresáři a překladem/spuštěním.
- Všechny zdrojové soubory musí obsahovat na začátku komentář se jmény autorů daného modulu, popisem obsahu modulu, atd. Z komentářů by také mělo být patrné, které části byly napsány ručně a které vygenerovány automaticky.

Poznámky k hodnocení

- Projekt je hodnocen na stupnici 0-100 procent/bodů.
- Po ohodnocení **může** dojít k přepočtu bodů na jednotlivé členy týmu podle návrhu vedoucího týmu, který je zakódován do jména odevzdaného souboru (formát jména pak je: xname01-xname02-AA-BB.zip součet AA+BB=100, **POZOR: pro rovnoměrné rozdělení AA=BB=50% není třeba uvádět "-50-50" a je to preferovaná možnost**).
- Hodnocení zahrnuje kvalitu OO návrhu, kvalitu implementace v C++, dodržování stylu psaní programu (odsazování, kvalita komentářů, vhodné identifikátory) a především funkčnost programu. Pokud budou ve zdrojových textech chybět podstatné komentáře (zvláště jména autorů), bude projekt hodnocen 0b.
- **POZOR:**
Pokud nepůjde přeložit příkazem make = 0b nebo předvedete na svém notebooku a vysvětlíte kde byl problém,
pokud nebudou komentáře se jmény autorů = 0b.
Zcela nevhodně formátovaný a nečitelný kód = 0b.
Plagiáty atd. = 0b + návštěva u disciplinární komise.

Poslední modifikace: 21. February 2018

Pokud naleznete na této stránce chybu, oznamte to dopisem na adresu peringer AT fit.vutbr.cz