Výzkumná zpráva:
Doporučení na programovací styl

MPO FV 10202: Databáze materiálových mikrostruktur pro aditivní výrobu

Vypracoval: M. Doškář

Poslední editace: 09/2020

Obsah

Stručná charakteristika úkolu

Pro usnadnění orientace ve zdrojovém kódu a pro jeho snadnější aktualizaci a rozšíření byla vypracována doporučení týkající se programovacího stylu. Nejedná se o pevně závazná pravidla, nicméně jejich dodržování je důrazně doporučováno, pakliže neexistuje relevantní důvod pro jejich porušení. Tato doporučení se týkají pouze zdrojových kódů psaných v jazyce C++ a následují vhodné praktiky doporučované velkými softwarovými firmami i předními členy komise zajišťujícími další rozvoj programovacího jazyka C++.

Doporučení následují C++ Core Guidelines od autora jazyka C++ Bjarne Stroustrupa dostupná na https://github.com/isocpp/CppCoreGuidelines a přebírají některé prvky z Google C++ Style Guide (https://google.github.io/styleguide/cppguide.html). Podobné myšlenky lze dále najít i u jiných programovacích jazyků, např. v doporučení Zen of Python.

Doporučení

- Kód by měl být psán v posledním standardu programovacího jazyka C++, v době psaní tedy C++17, a v maximální možné míře využívat všech jeho vlastností (např. `std::filesystem`, `std::optional`, auto deduction, for-range loops). Napsaný kód by měl být plně přenositelný napříč platformami.
- Zdrojový kód musí být psán v anglickém jazyce (jména proměnných, funkcí a metod, komentáře)
- Komentáře by měly být stručné a výstižné. Optimálně by záměr programátora měl být dokumentován výstižnými názvy proměnných, funkcí a metod a použitými algoritmy ze standardní knihovny.
- Název proměnných by měl být ve formátu camelCase, tj. s malým počátečním písmenem. Názvy tříd by měly být ve formě varianty CamelCase formátu s velkým počátečním písmenem, tzv. PascalCase.
- Názvy metod a funkcí by měly být výstižné a dostatečně popisné. V případě víceslovných názvů by měla být jednotlivá slova oddělená podtržítkem.
- Pořadí jednotlivých částí kódu by se mělo řídit doporučením Google C++ Style Guide.
    - V hlavičkovém souboru by první měly být importovány standardní knihovny, následované importem externích knihoven, a nakonec knihovnami vyvinutými pro danou aplikaci. V jednotlivých částech by měly být knihovny řazeny abecedně.
    - V definici třídy by měly být uvedeny prvně veřejné metody a data, následované chráněnými a soukromými metodami a daty. V rámci každé části s definovaným přístupem by měly první být definice, následované daty, a nakonec by měly být uvedeny metody.
- Funkce a metody by měly mít pouze vstupní parametry (případně vstupně-výstupní). V případě více výstupů funkce či metody je doporučeno využívat objekty třídy `std::pair` či `std::tuple`.
- Data do funkcí a metod by měla být v základu předávána pomocí referencí. V případě jednoduchých datových typů je možno předávat hodnotou.
- Primárně by měly být všechny vstupní parametry a proměnné definovány jako konstantní. Nepřítomnost konstantního kvalifikátoru indikuje, že bude daná proměnná dále v kódu měněna.
- Skutečné konstanty, jejichž hodnota je známá již při kompilaci, by měly být kvalifikovány jako `constexpr`.
- Seznam dostupných variant/nastavení by měl být implementován s využitím třídy `enum`.

- Při deklaraci proměnných by měla být primárně využívána automatická dedukce, jelikož zvyšuje nejen přehlednost kódu, ale především zajišťuje správnou volbu datového typu. Delší odůvodnění je možné nalézt v přednášce Herba Suttera z CppCon2017.
- Proměnné by měly být deklarovány co nejblíže jejich použití.
- Pro inicializaci proměnných by měla být použita syntaxe se složenými závorkami (aby se odlišila inicializace od jiných příkazů).
- Pro cyklus přes datové kontejnery by měly být primárně využívány algoritmy ze standardní knihovny (případně `for-range` cykly) namísto základního `for` cyklu. Algoritmy lépe dokumentují záměr programátora a umožňují snadné využití jejich paralelních verzí.
- Pro správu dynamicky alokované paměti musí být použity tzv. chytré ukazatele ze standardní knihovny std::memory. Běžné ukazatele na datové proměnné nesmí být používány pro správu dynamicky alokované paměti.