

Princípy tvorby softvéru

Implementácia, TDD

Robert Lukočka

lukotka@dcs.fmph.uniba.sk

www.dcs.fmph.uniba.sk/~lukotka

M-255

Implementácia

V rámci kurzu sme už pokryli veľa vecí, ktoré spadajú aj do fázy implementácie

- Dizajnové princípy
- DRY, Rule of 3
- Code smells, RefaktORIZÁCIE
- Písanie testov
- ...

Coding conventions

Code conventions sú dôležité lebo ([wiki](#)):

- 40%–80% ceny počas životného cyklu softvéru pripadá na údržbu.
- Takmer žiaden softvér nie je udržiavaný počas celého životného cyklu jedným autorom.
- Coding conventions zlepšujú čitateľnosť softvéru, umožňujú ostatným pracovať s kódom rýchlejšie a poriadnejšie.
- Zdrojový kód je váš produkt, a je dobré aby ste svoj produkt odovzdali upravený.

Coding conventions

Coding konvencie možno prispôbiť, upraviť podľa špecifických potrieb:

- Konvencie pre programovací jazyk
- Konvencie danej organizácie
- Konvencie pre projekt
- Konvencie pre komponent
- Konvencie pre unit

Čím nižšia úroveň, tým dôležitejšia je konvencia.

Coding conventions

- Comment conventions
- Indent style conventions
- Line length conventions
- Naming conventions
- Programming practices
- Programming principles
- Programming style conventions

Comment conventions

- Používať Docstringy (Python)/Javadocy, alebo nie?
- Ktoré komentáre sú povinné
- `/* ... */` vs `//` (C, C++)
- ...

Indent style conventions

- medzery alebo taby (medzery, s výnimkou Makefile)?
- koľko medzier?
- ako rozdeliť dlhé riadky (aritmetické výrazy, zoznamy parametrov, ...)
 - Je rozdelenie čitateľné?
 - Je rozdelenie stabilne vzhľadom k budúcim zmenám?
- Kam dať { } (C++), [Indentation style](#)
- ...

Line length conventions

79, 80, 99, 100 (Python), 180 (Mono), unlimited (Go), ...

- [Prehľad](#)

Naming conventions

Čo?

- premenná (lokálna/globálna), namespace, konštanta, package, trieda, objekt, metóda, funkcia, procedúra, ...

Ako?

- dĺžka
- veľke/male písmená
- veľke/male prvé písmeno
- skracovať dlhé názvy, spájať slová
- oddeľovanie slov:
 - CamelCase
 - snake_case
 - kebab-case

Naming conventions Java

- triedy - UpperCamelCase
- metódy - lowerCamelCase
- premenné - lowerCamelCase
- konštanty -
UPPER_CASE_SEPARATED_BY_UNDERSCORES

Naming conventions

S menami identifikátorov netreba byť stručný (najmä, keď máme k dispozícii dosť znakov na riadok). Historické Dôvody, používania krátkych identifikátoroviki:

- Obmedzenia veľmi starých linkerov.
- Editory bez autocomplete.
- Malé monitory, obrazovky (napr. max 80 znakov)
- Computer Science vyšla z matematiky, kde sú krátke mená premenných obvyklé.

Príklady - Python

- PEP 8,
- PEP 20 (Examples, More examples)

Príklady - C++, C

- Google C++ coding standards
- SEI CERT C++ Coding Standard - pravidla (Example)

Príklad - C++ prechádzanie poľa

- `for(auto prvok: pole)`
 - Normálne
- `for(auto it=pole.begin(); it<pole.end(); it++)`
 - Očakávame vkladanie, mazanie s poľa, skákanie komplikovanejšia ako možno dosiahnuť pomocou `continue`
- `for(unsigned int i=0; i<pole.size(); i++)`
 - Index je z nejakého dôvodu veľmi dôležitý

Linters

- pylint
- Cpplint

TDD 1

- 1 Pridaj test
- 2 Spusti test
- 3 Napíš kód
- 4 Spusti test
- 5 Refaktoruj
- 6 Spusti test

TDD 2

- 1 You are not allowed to write any production code unless it is to make a failing unit test pass.
- 2 You are not allowed to write any more of a unit test than is sufficient to fail; and compilation failures are failures.
- 3 You are not allowed to write any more production code than is sufficient to pass the one failing unit test.

Zdôrazňuje:

- YAGNI
- Rýchly feedback

Zdroje

- Coding conventions - wikipedia
- Naming conventions - wikipedia
- Python - PEP 8
- Python - PEP 20
- pylint
- Google C++ coding standards
- Cpplint
- SEI CERT C++ Coding Standard - pravidla