

# Kvalita kódu

Y36SI3 - Realizace programových  
systémů

Ondřej Macek

# Osnova přednášky

- Základní kritéria pro kvalitu kódu
- Duplikace kódu a dat
- Code smells
- Refaktoring

# Kvalita kódu

- Co je to kvalita kódu?
- Proč se o ní snažit?

# Kvalita kódu - oblasti

- Čitelnost
- Udržovatelnost
- Testovatelnost
- Jednoduchost
- Malé HW nároky
- Dobrý error handling
- ....

# **ZÁKLADNÍ KRITÉRIA KVALITY KÓDU**

# Ortogonalita

- Vzájemná nezávislost modulů projektu
  - kolik modulů se změní při změně požadavku zákazníka?

# Coupling vs. Cohesion

- Loose (Low) coupling
- High cohesion

# Oddělení politiky od mechanismu

Separate policy from mechanism

- Vrstevnatá architektura
- Využití konfiguračních souborů



# Demeter law

„Don't talk to strangers“

Metoda objektu smí volat

- Jiné metody objektu
- Metody objektů které vlastní
- Metody objektů které vytvořila
- Metody objektů které jsou parametrem metody

**DUPLIKACE KÓDU A DAT**

# Duplikace kódu a dat

- Stejná data na dvou místech
- Stejný kód na dvou místech
- Příklad:
  - 10000 různých způsobů validace amerického Social Security Number ve vládních systémech

# Příčiny duplikace

- Ctrl-C, Ctrl-V
- Programovací prostředí zdánlivě nutí k duplikaci
  - „Evil Wizards“
- Programátor neví že data už jsou v projektu reprezentována (dělbá práce)
- Předčasná optimalizace
  - denormalizace databáze na začátku vývoje „kvůli výkonu“

# Jak se vyhnout duplikaci dat

- Normalizace databáze
- 1NF (1. normální forma)
  - sloupce obsahují skalární hodnoty
- 2NF (2. normální forma)
  - neklíčové sloupce závisí pouze na kompletním klíči, nikoli na části klíče
- 3NF (3. normální forma)
  - neklíčové sloupce závisí přímo na některém klíči (ne přes jiný neklíčový sloupec)
- BCNF (Boyce-Coddova normální forma)
  - pro každé dvě množiny sloupců  $A \rightarrow B$  a  $B$  není podmnožinou  $A \Rightarrow A$  je klíč nebo nadmnožina klíče

# Jak se vyhnout duplikaci kódu

Předcházení:

- Technický vedoucí SW týmu
- Komunikace mezi programátory
- Generátory kódu
- Frameworky

Oprava:

- Code Review = Statické testování kódu
- Refaktoring

# Duplikace kódu – přístupy k řešení

- Jeden zdroj pravdy
  - Model Driven Development
- DRY principle (Hunt, Thomas)
  - Don't repeat yourself
- Code Smell
  - Kód „smrdí“

# Model Driven Development

- MDD = „Jeden model jim všem vládne“



# Don't repeat yourself

- DRY principle
  - [Pragmatic Programmer]

*“Every piece of knowledge must have a single, unambiguous, authoritative representation within a system”*

**CODE SMELL**

# Code smell

- Reakcí na code smell je refaktoring
- Co všechno je code smell?

# Porozumění kódu

*A fool can write code that a computer can understand. Good programmers write code that humans can understand.*

*When you feel you have to write the comment, first try to refactor the code so that any comment becomes superfluous*

# Příklady code smell

- Duplikovaný kód
- Dlouhé metody
- Dlouhé třídy
- Dlouhé seznamy parametrů
- Proměnné třídy
- Shotgun surgery
- Feature Envy

# Příklady code smell 2

- Skupiny dat
- Posedlost primitivními typy
- Posedlost pro switch
- Zbytečné třídy
- Porušení Demeter law
- Prostředník
- ....

**REFAKTORING**

# Refaktoring

- Změna struktury kódu aniž by došlo ke změně funkce
- Dnes součást mnoha IDE
  - Které je nejlepší?



# Kdy refaktorovat?

- Při přidávání nové funkčnosti
- Při odstraňování bugu
- Při code review

Co říct manažerovi?

# Kdy nerefaktorovat?

- Když je lepší kód přepsat
- Když se blíží deadline

# Refaktorovací pravidla

Pravidlo =

- Motivace
- Mechanismus
- Viz literatura
- Často se objevují dvojice pravidel pro přechod tam a zpátky

# První a poslední Krok refektoringu

- Testování
  - Bez existujících testů nejsem schopen si ověřit/prokázat, že nedošlo ke změně kódu
  - Než začnu refaktorovat ověřím si, že mám testy
  - Když skončím s refaktoringem, tak spustím testy
- Refaktoring provádějte po malých krocích

# Extract method

- Část kódu má být spojena dohromady
  - Příliš dlouhá metoda
  - Potřebuji komentář abych něčemu porozuměl
- Jaká pravidla je třeba dodržet při extrakci?
- Jaký je vlastně mechanismus?
- Malé metody fungují jen tehdy jsou-li dobře pojmenované

**Opačné pravidlo:** Inline method

**Jaká další znáte?**

# Refakroting a výkon

- Refaktoring
  - zpomalí program
  - Vytvoří program, který se dá vylepšit
    - Většinou se program „zdrží“ na malém množství kódu

# Zajímavá literatura

**SKŘIVAN J.: *DATABÁZE A JAZYK SQL* / *INTERVAL.CZ***

<http://interval.cz/clanky/database-a-jazyk-sql/>

[cit. 12. 11. 2010]

**BOCK D.: *THE PAPERBOY, THE WALLET AND THE LAW OF DEMETER***

<http://www.ccs.neu.edu/research/demeter/demeter-method/LawOfDemeter/paper-boy/demeter.pdf>

[cit. 12. 11. 2010]

- Viz stránky předmětu