

Princípy tvorby softvéru

Robert Lukočka

lukotka@dcs.fmph.uniba.sk
www.dcs.fmph.uniba.sk/~lukotka
M-255

SOLID

- 1 Bob Martin: Bob Martin SOLID Principles of Object Oriented and Agile Design

SOLID

- 1 Single responsibility principle
- 2 Open-closed principle
- 3 Liskov substitution principle
- 4 Interface segregation principle
- 5 Dependency inversion principle

Open-closed principle

Nástroje:

- 1 Inheritance (nepreferované)
- 2 Composition
 - Strategy pattern (pozor na zavedenie cirkulárnych dependencií)

How to do composition and not create a dependency?

- dependency injection (pri vytvorení, metódou)
- factory method

ISP and DIP

- 1 DIP otáča dependencie, robí kód menej rigidný a fragilný
- 2 Package zavisia od interfaceov nie od iných packageov
- 3 ISP - Čosi ako SRP pre interfacý, znižuje dopad zmien.

Dizajn

Komplexita väčšiny softveru spočíva v tom, že sa jedná o veľké množstvo jednoduchých úloh.

Dizajn

Dobré vlastnosti (ktoré sú však často v protiklade)

- 1 Spĺňa požiadavky
- 2 Ľahko pochopiteľný, ľahká navigácia v systéme
- 3 Ľahko modifikovateľný
- 4 Efektívny

Modifikovateľnosť

Pre dosiahnutie ľahkej modifikovateľnosti, môže byť potrebné odchyliť sa od konceptuálneho modelu.

- 1 YAGNI (Nečakajte, že viete predpokladať zmeny požiadaviek)
- 2 DRY
- 3 Rule of 3
- 4 RefaktORIZÁCIE

Modifikovateľnosť

Doménový model je často vhodné zovšeobecniť

- 1 Priama aplikácia doménového modelu môže viesť k "Anemic domain model" - kopec objektov bez zmysluplného správania.
- 2 Objekty sú o správaní, nie o atribútoch.
- 3 Objekt by mal mať netriviálnu metódu
- 4 Strategy pattern - Aj metódy môžu byť atribúty

Modifikovateľnosť

- 1 Menej dependencií pomáha modifikovateľnosti
- 2 Information expert - funkcionálna má byť v objekte ktorý má relevantné informácie
- 3 Information expert sa môže biť so SRP.