

Návrhové vzory

Tomáš Černý

Návrhové vzory (Design Patterns)

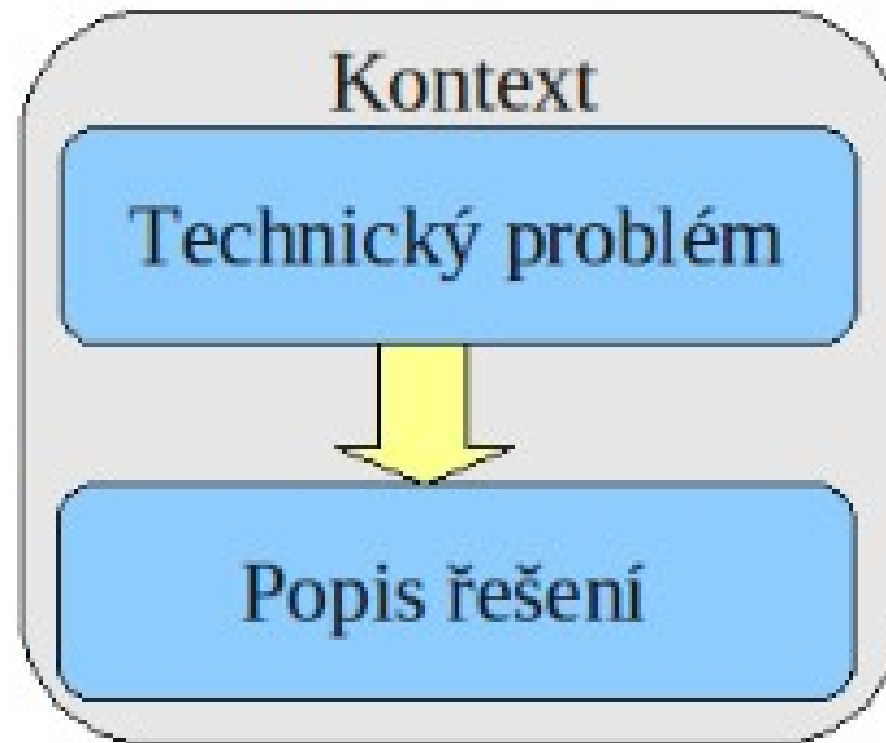
Naše poznatky při návrhu software jsou následující:

- Problémy při návrhu se opakují
- Mnoho problémů je stejnou strukturu řešení
- Přesné řešení záleží na kontextu
- Vývojář znalý návrhových vzorů může řešit nové problémy rychleji a lépe
- „Best practices“

Návrhové vzory (Design Patterns)

Na návrhové vzory lze též nahlížet jako na **trojici**, která vyjadřuje vztah mezi určitým

- 1.kontextem
- 2.problémem
- 3.řešením



Návrhové vzory

Jaké vzory znáte?

Návrhové vzory

Jaké vzory znáte?

```
List<String> aList = new ArrayList<String>();  
// Add Strings to alist  
for (Iterator<String> iter = aList.iterator();  
    iter.hasNext(); ) {  
    String s = iter.next();  
    // No downcasting required.  
    System.out.println(s);  
}
```

Návrhové vzory

Jaké vzory znáte?

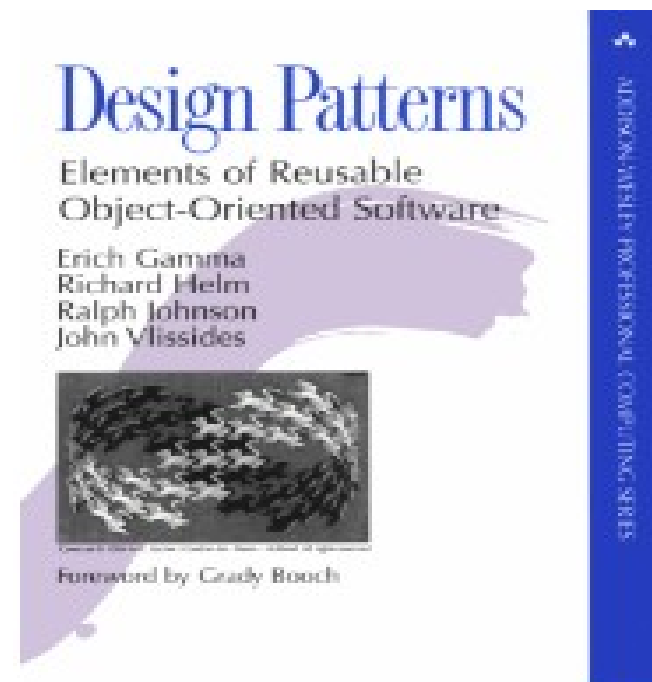
```
List<String> aList = new ArrayList<String>();  
// Add Strings to alist  
for (Iterator<String> iter = aList.iterator();  
    iter.hasNext(); ) {  
    String s = iter.next();  
    // No downcasting required.  
    System.out.println(s);  
}
```

Nebo skrytě

```
for (String s : aList) {  
    System.out.println(s);  
}
```

Návrhové vzory (Design Patterns)

- Mnoho knih o návrhových vzorech
- Gang of 4 neboli Gamma book
 - katalog vzorů pro návrh řešení studovaného problému



Návrhové vzory (Design Patterns)

Návrhový vzor zachycuje

- nejlepší dostupnou zkušenost z vývoje software
 - ve formě problém-řešení

Návrhový vzor se skládá z:

- charakteristiky problému
- charakteristiky řešení
- množiny transformačních doporučení

Návrhové vzory (Design Patterns)

Řeší strukturní problémy jako

- Abstrakci
- Zapouzdření
- Schování informací
- Rozdělení zájmů / účasti
- Coupling & cohesion – párování a provázanost
 - menší párování je lepší
 - větší provázanost (jasnost, soudržnost) je lepší
- Rozdělení rozhraní od implementace
- Jeden odkazový bod
- Rozděl a panuj

Návrhové vzory (Design Patterns)

Řeší nefunkční problémy jako:

- Zaměnitelnost
- Spolupráci
- Efektivnost
- Spolehlivost
- Testovatelnost
- Znovu použití

Typy vzorů

Vzory požadavků

- Popisuje množinu požadavků pro množinu aplikací
 - check-in check-out pattern
 - systém knihovny
 - půjčovna automatická
 - video systémy

Typy vzorů

Vzory požadavků

- Popisuje množinu požadavků pro množinu aplikací
 - check-in check-out pattern
 - systém knihovny
 - půjčovna automatická
 - video systémy

Programovací styly

- Popisuje specifické řešení pro programovací jazyk
 - Reflexe pro Javu
 - Co jsou to Reflexe?

Typy vzorů

Architektonické vzory

- Popis architektur
 - Broker pattern
 - distribuované systémy, kde pozice souboru a služby je transparentní.
 - CORBA
 - **Model View Controller (GUI)**
 - Pipe-and-Filter (Staré překladače)
 - Vícevrstvé architektury (TCP/IP, J2EE)

Typy vzorů

Architektonické vzory

- Popis architektur
 - Broker pattern
 - distribuované systémy, kde pozice souboru a služby je transparentní.
 - Model View Controller
 - Pipe-and-Filter
 - Vícevrstvé architektury

Návrhové vzory

- Popisuje návrhová řešení v menším měřítku
 - Gang of Four

Poznatek

Vzory adresují opakující se vývojové **problémy**, vyskytující se určitých **kontextech** a podávají **řešení** problému.

Vzor zahrnuje nejlepší praktiky „best practices“ v softwarovém vývoji.

Tyto vzory též slouží ke **komunikaci** mezi vývojáři.

Jaký je rozdíl mezi architekturou, vzorem a frameworkem?

1.Architektura ???

2.Vzory ???

3.Frameworky ???

Jaký je rozdíl mezi architekturou, vzorem a frameworkem?

1.**Architektura** modeluje softwarovou strukturu na nejvyšší úrovni a podává pohled na systém jako celek. Architektura používá až několik různých vzorů v různých komponentách.

2.**Vzory** jsou menší či místní architektury či architektonické komponenty.

3.**Frameworky** jsou částečně kompletní softwarové systémy, které se mohou zaměřovat na konkrétní typ aplikace. Tyto systémy jsou pak upraveny na míru kompletováním nedokončených komponent.

Rozdělení GoF vzorů

- Tvorba nových instancí a tříd
- Strukturní vzory
- Vzory chování.

Rozdělení GoF vzorů

- Jaké že jsou? Ty tři? :D

Rozdělení GoF vzorů

Vzory pro tvorbu instancí a tříd jsou:

- Factory method
- Abstract Factory
- Prototype
- Builder
- Singleton
- ..

Rozdělení GoF vzorů

Strukturní vzory jsou:

- **Adapter**
- **Bridge**
- Composite
- Decorator
- **Facade**
- Flyweight
- Proxy
- ..

Rozdělení GoF vzorů

Vzory chování:

- Interpreter
- Template Method
- Chain of responsibility
- Command
- **Iterator**
- Mediator
- Observer
- State
- Strategy
- Visitor
- ...

GoF - Strukturní vzory

Hlavní myšlenkou je rozdělení rozhraní a implementace.

Composite: struktura pro budování rekurzivních agregací

Adapter: Překlad adaptující rozhraní servru a klienta

Bridge: Abstrakce pro spojení jedné či více implementací

Decorator: transparentní rozšíření objektu

Facade: zjednodušuje rozhraní a tedy přístup do podsystemu

Flyweight: Mnoho objektů s efektivním sdílením

Proxy: Jeden objekt aproximuje jiný

GoF - Strukturní vzory

Hlavní myšlenkou je ???

GoF - Vzory chování

- **Chain of responsibility:** požadavek vyslán do zodpovědné obsluhy
- **Observer:** one-to-many závislost mezi objekty, tak že změna stavu jednoho způsobí oznámení všem ostatním
- **Command:** volání rutiny jako objekt
- **Interpreter:** jazykový překlad pro menší gramatiky
- **Iterator:** k nasbíraným elementům přistupujeme sekvenčně
- **Mediator:** koordinuje interakce mezi přiřazenými objekty

GoF - Vzory chování

Memento: snapshot zachytí a obnoví soukromé stavy objektů

State: objekt jehož chování závisí na jeho stavu

Strategy: abstrakce pro výběr jedné ze strategií

Template Method: algoritmus s kroky dodanými odvozenou třídou

Visitor: operace aplikovány na elementy heterogenní objektové struktury

Vzory pro tvorbu instancí a tříd

- **Factory method**
- **Abstract Factory**
- **Prototype**
- **Builder**
- **Singleton** – jedna instance

Odkazy

Kniha

Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides, November 2000, ISBN 0-20-63361-2

Přednášky

- Baylor, Dr. Song
- MFF UK, Dr. Zavoral