Návrhový model tříd I.

PV167 Projekt z obj. návrhu IS

B. Bühnová

25. dubna 2011

Diagram tříd - opakování

Diagram tříd poskytuje statický pohled na třídy systému, jejich atributy, operace a vzájemné vztahy.
UML určuje pouze vizuální syntaxi diagramu tříd, postup tvorby včetně rozdělení na analytickou a návrhovou fázi určují metodiky.

Analytický model tříd:

- Modeluje obchodní doménu systému - typy objektů a vztahy mezi nimi.
- Snaha o zachování přehlednosti a jednoduchosti bez zanášení implementačních detailů.

Návrhový model tříd:

- Rozšiřuje analytický model tříd o implementační třídy a detaily.
- Často obsahuje až 10x více tříd.

Návrhové třídy

Návrhová třída je na rozdíl od analytické třídy na takové úrovni abstrakce, že ji lze snadno implementovat.

Dobrá návrhová třída nese tyto vlastnosti:

- Jednoduchost - obsahuje jednoduché nedělitelné metody
- Soudržnost - vnitřní soudržnost a min. vazeb na jiné třídy
- Spolupráce - třída neexistuje ve vzduchoprázdnu
- Dobré pojmenování - název třídy by měl odrážet její účel
- Je úplná a dostačující - obsahuje vše podstatné pro implementaci a nic navíc (všechny metody odráží její účel)
- Veřejné metody správně definují funkce poskytované ostatním klientům

Přechod od analytického modelu tříd k návrhovému (1)

Možný postup:

1. Nalezení návrhových tříd
2. Doplnění detailů návrhových tříd (atributy, metody)
3. Doplnění rozhraní, šablon, vnořených tříd, stereotypů
4. Doplnění chybějících relací
5. Upřesnění analytických relací
6. Aplikace návrhových vzorů

Přechod od analytického modelu tříd k návrhovému (2)

1. **Nalezení návrhových tříd**
   - Z obchodní domény - upřesněním analytických tříd (rozklad na více tříd, doplnění implementačních detailů)
   - Z domény řešení - třídy vyžadované použitou technologií (třídy pro práci s GUI, databází, komponentami)

2. **Doplnění detailů návrhových tříd**
   - Atributy (včetně viditelnosti a typu)
   - Metody (včetně viditelnosti, argumentů a typu návratové hodnoty)
     - Metody vzniklé rozložením analytických operací
     - Konstruktory a destruktory
     - **get** a **set** metody
     - Implementační metody

Přechod od analytického modelu tříd k návrhovému (3)

Upřesnění atributů a metod:

Plná specifikace atributu se skládá z prvků:

- viditelnost (+, -, #, ~), název, násobnost, typ, počáteční hodnota, platnost (instance, třídy/static)
- Příklady: "-pocet: int = 0", "#adresa[3]: String"

Plná specifikace metody se skládá z prvků:

- viditelnost (+, -, #, ~), název, argumenty, typ návratové hodnoty, platnost (instance, třídy/static)
- Příklad: "+getStavUctu(datum: Date): float"

Přechod od analytického modelu tříd k návrhovému (4)

3. Doplnění rozhraní, šablon, vnořených tříd, stereotypů

- Rozhraní jsou důležité prvky umožňující zvýšit flexibilitu aplikace odprostěním od přímych vazeb na konkrtétní třídy.
- Šablony a vnořené třídy by se měly používat jen tehdy, když je podporuje implementační jazyk.

4. Doplnění chybějících relací

- Analogicky jako ve fázi analýzy

Přechod od analytického modelu tříd k návrhovému (5)

5. Upřesnění analytických relací

- Upřesnění vybraných asociací do relací typu agregace nebo kompozice.
- Stanovení řiditelnosti asociací
- Implementace obousměrných asociací (rozklad na dvě jednosměrné)
- Revize relací typu 1:1 (zda ponechat jako kompozici, sloučit nebo vložit jako atribut)
- Implementace relací typu M:N (rozkladem přes vazební třídu)
- Implementace asociačních tříd

6. Aplikace návrhových vzorů

- Praxí ověřená řešení často se vyskytujících problémů