

Vytvoření diagramu tříd

Vytvoření diagramu tříd ze zadání úlohy je iterativní proces. Postupně jsou do něho zařazovány třídy, které tvoří model úlohy. Obecný postup vytváření tříd má fáze:

- Stanovení objektů a odvození tříd z nich.
- Stanovení atributů v třídách.
- Stanovení vztahů mezi třídami.
- Stanovení zodpovědností jednotlivých tříd (použitím CRC).
- Rozdělení zodpovědností do operací a atributů.
- Sestavení specifikací procesu - popis operací.

Stanovení objektů a odvození tříd

Diagram tříd je užitečný způsob, jak modelovat strukturu objektů v systému a vztahů mezi nimi. Když ale začínáme s návrhem nového programu (aplikací), třídy nejprve musíme stanovit. Zde můžeme začít tím, že nejprve najdeme, jaké by v systému mohly být objekty, a následně rozhodneme, jak z nich lze vytvořit třídy.

Objekty mohou být z různých kategorií. Pokud si uvědomíme, do jakých kategorií jednotlivé objekty patří, lze z toho odvodit, čím se máme při jejich analýze zabývat. Kategorie objektů mohou být:

- Lidé (zákazníci, zaměstnanci, studenti)
- Organizace (škola, knihovna, podnik)
- Fyzické předměty (kniha, kolo, výrobek)
- Konceptuální záležitosti (výpůjčka knihy, objednávka zákazníka, zasedací pořádek)

Pro stanovení, jaké by navrhovaný systém měl mít objekty, jsou různé techniky. Nelze ale říci, že by některá z nich nám poskytovala přímý návod k vytvoření objektů. Jsou spíše jen pomocný nástroj. Jedna taková užitečná metoda je stanovení objektů na základě podstatných jmen v zadání systému. Výhodné je zde mít zadání systému, které je úplné a zároveň stručné.

Stanovení objektů použitím analýzy podstatných jmen

- Vyhledáme v zadání systému všechna podstatná jména nebo fráze obsahující podstatná jména. Dostaneme poměrně dlouhý seznam možných kandidátů na objekty. U některých je na první pohled zřejmé, že je můžeme vyloučit, neboť nejsou vhodné, aby z nich byly odvozeny objekty. U některých se to zjistí až při hlubší úvaze.
- Vyloučíme nevhodné kandidáty podle kritérií, která budou popsána dále.

Příklad

Máme vytvořit systém pro půjčovnu kol. Uživatelské požadavky, co má systém obsahovat, jsou:

- P1. Seznam všech kol s jednotlivými údaji: číslo kola, typ kola, velikost, provedení, model, denní půjčovné, záloha při zapůjčení.
- P2. Seznam všech zákazníků s evidencí všech jejich současných a minulých zapůjčení.
- P3. Funkci počítající půjčovné pro dané kolo a stanovený počet dní.
- P4. Záznam obsahující jednotlivé údaje o zapůjčení: datum počátku zapůjčení, předpokládaná doba zapůjčení, který zákazník si kolo půjčuje a které kolo si půjčuje. Tento záznam by měl umožnit při vrácení kola uvedené údaje snadno nalézt.

- P5. Byl schopný vyhledat všechna zapůjčení kol, které zákazník má, aby bylo možné vystavit společnou stvrzenku za všechna zapůjčení, a ne účtovat každé zvlášť.
- P6. Byl schopný realizovat zapůjčení pro jednoho zákazníka nejen pro více kol, ale také jednotlivá kola na různě dlouhou dobu.
- P7. Měl funkci, která při vrácení kola zjistí, jak dlouho kolo bylo půjčeno, na jak dlouho byla zaplacená záloha a kolik je potřeba doplatit.
- P8. Uchoval pro další použití celkovou částku, která se má zaplatit a kolik bylo skutečně zaplacen.
- P9. Vytiskl stvrzenku pro zákazníka.
- P10. Umožnil zjistit stav každého kola, zda volné nebo je půjčeno nebo je v opravě.
- P11. Umožnil uložit doplňující údaje pro speciální kola.

Nyní vyhledáme v požadavcích podstatná a fráze s podstatnými jmény (pokud se stejné podstatné jméno nebo fráze vyskytuje v požadavcích vícekrát, je pro přehlednost zvýrazněn jen první výskyt):

- J1. Seznam všech kol s jednotlivými údaji: číslo kola, typ kola, velikost, provedení, model, denní půjčovné, záloha při zapůjčení.
- J2. Seznam všech zákazníků s evidencí všech jejich současných a minulých zapůjčení.
- J3. Funkci počítající půjčovného pro dané kolo a stanovený počet dní.
- J4. Záznam obsahující jednotlivé údaje o zapůjčení: datum počátku zapůjčení, předpokládaná doba zapůjčení, který zákazník si kolo půjčuje a které kolo si půjčuje. Tento záznam by měl umožnit uvedené údaje snadno nalézt při vrácení kola.
- J5. Byl schopný vyhledat všechna zapůjčení kol, které zákazník má, aby bylo možné vystavit společnou stvrzenku za všechna zapůjčení, a ne účtovat každé zvlášť.
- J6. Byl schopný realizovat zapůjčení pro jednoho zákazníka nejen pro více kol, ale také jednotlivá kola na různě dlouhou dobu.
- J7. Funkci, která při vrácení kola zjistí, jak dlouho kolo bylo půjčeno, na jak dlouho byla zaplacená záloha a kolik je potřeba doplatit.
- J8. Uchoval pro další použití celkovou částku, která se má zaplatit a kolik bylo skutečně zaplacen.
- J9. Vytiskl stvrzenku pro zákazníka.
- J10. Umožnil zjistit stav každého kola, zda volné nebo je půjčeno nebo je v opravě.
- J11. Umožnil uložit doplňující údaje pro speciální kola.

Vyhledáním podstatných jmen a frází s nimi v požadavcích na systém dostáváme seznam:

- seznam kol
- jednotlivé údaje o kolech: číslo kola, typ, velikost, provedení, model, denní půjčovné, záloha při zapůjčení
- seznam zákazníků
- současná s minulá zapůjčení
- kolo

- počet dní
- jednotlivé údaje o zapůjčení: datum počátku zapůjčení, předpokládaná doba zapůjčení
- zákazník
- stvrzenka
- různě dlouhá doba
- vrácení kola
- různě dlouho
- celková částka
- stav každého kola
- doplňující údaje pro speciální kola

Nyní následuje analýza jednotlivých položek v seznamu, abychom zjistili, které je potřeba vyloučit, protože nejsou vhodné pro vytvoření objektů. Vyloučíme položky, které jsou:

- **Atributy.** U některých podstatných jmen je na první pohled zřejmé, že označují spíše atribut než objekt. Takovými jsou *číslo kola*, *typ*, *velikost*, *provedení*, *model*, *denní půjčovné* a *záloha při zapůjčení*. Zapůjčení naopak vypadá jako možný objekt, přičemž *datum počátku zapůjčení* a *předpokládaná doba zapůjčení* jsou patrně jeho atributy. Dále jako atributy se zdají být *počet dní*, *různě dlouhá doba*, *celková částka* a *doplňující údaje pro speciální kola*. Jsou-li pochybnosti, zda podstatné jméno reprezentuje objekt, může se zeptat, jaké bude mít atributy a zda bude mít nějaké operace.
- **Redundantní.** Občas se stejný koncept objeví v textu v různých podobách. *Současná a minulá zapůjčení* a *zapůjčení* patrně bude vést na stejný typ objektu. *Různě dlouhá doba*, *počet dní*, *předpokládaná doba* mají stejný význam – všechny označují dobu trvání zapůjčení (navíc jsou to atributy a ne objekty).
- **Příliš neurčité.** Pokud nedokážeme přesně určit, co je podstatným jménem myšleno a co je s ním spojeno, zřejmě z něho nebude dobrý objekt. Z tohoto důvodu odmítneme *vrácení kola* jako objekt. Vrácení kola je událost. Údaje, které souvisí s vrácením kola, lze uložit společně s údaji o zapůjčení.
- **Příliš fyzicky svázané se vstupy a výstupy.** Tím označujeme něco, co sice skutečně v reálném světě existuje, ale je produktem systému nebo vstupem dat do systému. Například účtenka sice existuje, ale je jen výstupem systému vytvořeným z uložených údajů. Účtenku z tohoto důvodu jako objekt zamítneme. Navíc ji lze považovat za redundantní, neboť údaje uvedené na ní máme někde již uloženy nebo jsou vypočítány z již uložených údajů. *Seznam kol* je rovněž zapotřebí zvážit. Máme-li pro každé kolo objekt, který ho reprezentuje, můžeme z těchto objektů vygenerovat jejich seznam. Proto *kolo* je dobrý kandidát na objekt, *seznam* bychom mohli zamítnout. Totéž platí pro frázi *seznam zákazníků* a podstatné jméno *zákazník*.
- **Asociace.** *Zapůjčení* jsme doposud modelovali jako asociativní vztah mezi třídami *Zákazník* a *Kolo*. Není zde ale jisté, zda je to opravdu jen asociativní vztah nebo je *zapůjčení* objekt. Obecný pravidlo pro posouzení takových případů je, že s takovým vztahem jsou spojeny určité údaje (a zde jsou), pak ho pravděpodobně budeme modelovat objektem.
- **Je mimo oblast systému.** Tím se myslí, že zamítneme cokoli, co je mimo předpokládané použití systému. Například u navrhovaného systému do něho nespadá výpočet mezd, účetnictví a další.

- **Je ve skutečnosti operace nebo událost.** Toto kritérium může být matoucí a vyžaduje hlubší zamyšlení. Jestliže kandidát na objekt není spojen s žádnými daty, může být modelován jen jako operace začleněná do některé třídy. Je-li naopak kandidát na objekt spojen s určitými údaji, modelujeme ho jako objekt. Například zapůjčení kola je událost, s níž jsou spojeny údaje, jako jsou datum počátku zapůjčení a doba zapůjčení, a budeme ji tedy modelovat jako objekt.
- **Reprezentuje celý systém.** Není rozumné, abychom měli jeden objekt, který by reprezentoval celý systém. Normální je, aby systém byl rozdělen do více samostatných objektů.

Po uplatnění uvedených zásad nám zůstávají kandidáti na objekty:

- kolo
- zákazník
- zapůjčení kola
- speciální kolo

Z těchto objektů můžeme začít sestavovat diagram tříd.

Analýza podstatných jmen je sice dobrý začátek, ale často nestačí k vytvoření úplného souboru tříd. Některé objekty se v požadavcích na systém nemusí objevit reprezentované podstatnými jmény. Je to záležitost dalších úvah.

Vraťme se k našemu rozhodnutí, že zapůjčení bude reprezentováno jako objekt. Učinili jsme ho, protože se zapůjčením jsou asociované určité údaje. Jsou ale další důvody, proč zapůjčení modelovat jako třídu. Pokud bychom tuto třídu neměli, museli bychom údaje o zapůjčení buďto uložit do třídy *Zákazník* nebo do třídy *Kolo*. Uvážíme-li, že zákazník může mít zapůjčeno více kol, znamená to mít v této třídě zajištěno, aby se dalo uložit více údajů o zapůjčení, což patrně znamená použít pro ně datovou strukturu určenou pro uložení více údajů. To je nevýhodné. U třídy *Kolo* nastává obdobná situace, protože požadujeme nejen uložení údajů o současném zapůjčení, ale i uchování údajů o předchozích zapůjčení.

Dále musíme vyřešit, kam uložit údaje o platbách. Ty bychom mohli uložit do třídy *Zákazník*. Vzhledem k tomu, že požadujeme možnost společné platby za více zapůjčení, a navíc požadujeme uložení všech dosavadních účtů pro jejich případnou revizi, jeví se účelné vytvořit pro platby samostatnou třídu.

Stanovení atributů

Řada atributů se objevuje jako podstatné jméno v zadání úlohy. Někdy je ale obtížné rozhodnout, kam je dát. Zde lze použít jednoduché pravidlo, že atribut by měl patřit do té třídy, v jejíž popisu se objevuje.

Třída *Kolo*. Zatím jsme stanovili, že tato třída bude mít atributy *číslo kola*, *typ*, *velikost*, *provedení*, *model*, *denní půjčovné*, *záloha při zapůjčení*. Jeden z požadavků na systém je, aby bylo možné zjistit stav kola, tj. zda je volné nebo půjčeno nebo na opravě. Proto zavedeme další atribut *dostupnost*.

Třída *SpeciálníKolo*. Tato třída bude mít stejné atributy jako třída *Kolo* a navíc k tomu další atributy vztahující se ke speciálním kolům (tandemová apod.). Zavedeme pro ně další atribut *speciální typ*.

Třída *Zapůjčení*. Již jsme stanovili, že bude mít atributy *datum počátku zapůjčení* a *předpokládaná doba zapůjčení*. K tomu ještě přidáme atribut, ve kterém je *datum vrácení kola*. Pokud je kolo vráceno později, než byla doba, na kterou bylo zapůjčení, znamená to další

platbu za pozdní vrácení. Pro tento účel zavedeme atribut *pozdní vrácení*. Je-li kolo vráceno poškozené, znamená to, že budou účtovány předpokládané náklady na opravu. Pro ni zavedeme atribut *náhrada za poškození*.

Třída *Zákazník*. V této třídě budou uloženy základní údaje o zákazníkovi – *jméno*, *adresa*, *telefon*. Navíc je výhodné si zavést jednoznačné *číslo zákazníka*.

Třída *Platba*. V ní si uložíme *datum platby* a *zaplacenou částku*. Zaplacená částka se skládá z půjčového a náhrad za pozdní vrácení a poškození. Při placení je částka snížena o velikost zálohy nebo je část zálohy vrácena, pokud je půjčové nižší než záloha. Proto si ještě v této třídě uchováme *velikost zálohy*.

Stanovení vztahů mezi třídami

Asociativnost a násobnost. Protože objekt třídy *Zapůjčení* má v sobě uloženy údaje o zapůjčení kola, je viditelně asociativní vztah mezi třídami *Zapůjčení* a *Kolo*. Zbývá nám stanovení násobnosti. Podle údajů uložených v objektu třídy *Zapůjčení* je zřejmé, že ty jsou jen pro jedno kolo. Kolo naopak může být zapůjčeno vícekrát nebo i vůbec ne. Proto vztah tříd *Zapůjčení* a *Kolo* bude 0..* : 1.

Zákazník si může půjčit více kol. Vztah mezi třídami *Zákazník* a *Zapůjčení* bude tedy 1 : 1..*.

Zákazník může mít více plateb. Vztah mezi třídami *Zákazník* a *Platba* bude opět 1 : 1..*.

Zobecnění a dědičnost. Z diagramu je zřejmé, že třída *Speciální kolo* má všechny atributy, které jsou ve třídě *Kolo*. Nabízí se zde využití dědičnosti. Tím se stává dále zbytečný atribut *Speciální typ*. To, že jde o speciální kolo, nyní udává samotná třída. Navíc třída děděním z třídy *Kolo* rovněž dědí všechny vztahy mezi třídou *Kolo* a ostatními třídami.

