

Implementace

Ing. Jiří Mlejnek

Katedra softwarového inženýrství
Fakulta informačních technologií
České vysoké učení technické v Praze

© Jiří Mlejnek, 2013

jiri.mlejnek@fit.cvut.cz

Softwarové inženýrství BI-SI1
LS 2015/ Před. 8



Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti



Obsah

- Úvod
- Objektové paradigma
- Základní pravidla implementace

Úvod

- Cílem je dodat produkt který:
 - Funguje dle požadavků
 - Snadno se používá
 - Nutnost znát cílovou skupinu
 - Závisí na úrovni uživatelů
 - Předmět BI-TUR
 - Neobsahuje chyby
 - Funguje na všech cílových prostředích
 - Snadno se udržuje a rozšiřuje

Úvod

- Činnosti
 - Přepis scénářů případů užití do kódu
 - Za dodržení architektury a návrhových rozhodnutí
 - Neporušující dohodnutá pravidla
 - Návrh v malém
 - Ne vše je již rozhodnuto
 - Nová rozhodnutí by měla být v souladu s celkovou koncepcí

Úvod

- Shrnutí
 - Implementace je hlavní pracovní postup u softwarových projektů
 - Implementace je jediná činnost, která nemůže být vynechána

Úvod

Dotazy?

Objektové paradigma

- Systém je dekomponován na objekty, které spolu navzájem komunikují
- Každý objekt si sebou nese vlastní data a metody, které realizují zodpovědnost za vykonání určité funkcionality
- Data objektu mohou být chráněna před nepřípustnou modifikací nebo přístupem pomocí zapouzdření

Objektové paradigma

- Třída
 - Šablona, podle které se vytvářejí objekty
 - Třída definuje
 - Metadata atributů
 - Signatury metod
 - Implementace metod
 - Konstruktory, které slouží pro inicializaci atributů při vytváření objektu

Objektové paradigma

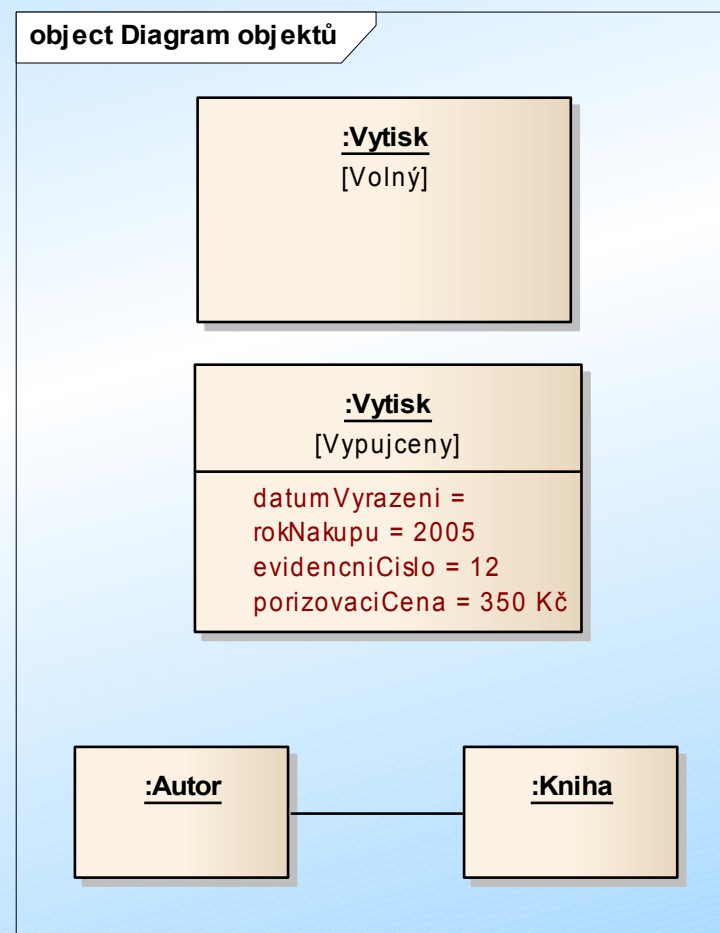
- Objekt
 - Hodnoty atributů objektu patří tomuto objektu (nejsou sdílené)
 - Metody objektu jsou společné pro všechny objekty dané třídy

Objektové paradigma

- Objektový diagram - UML
 - Patří do skupiny diagramů struktur
 - Usnadňuje pochopení diagramu tříd
 - Snímek aktuálního stavu systému v daný časový okamžik
 - Objekt – instance třídy
 - Spojení objektů – instance asociace
- EA: UML Structural - Object

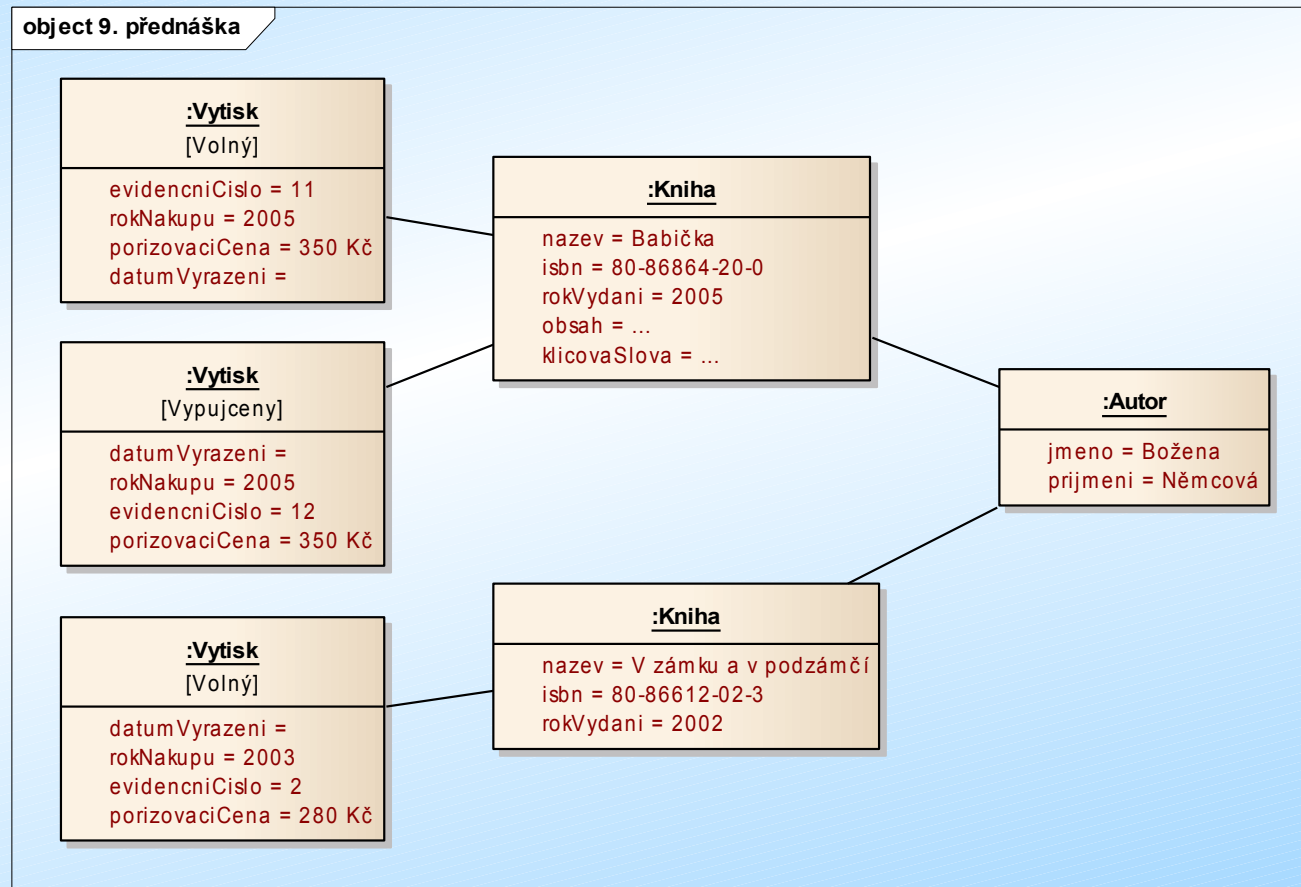
Objektové paradigma

- Objektový diagram
 - Objekt
 - Stav objektu
 - Hodnoty atributů
 - Spojení objektů



Objektové paradigma

- Diagram objektů



Objektové paradigma

- Zapouzdření
 - Objekt je krabice, která drží stav objektu uvnitř sebe sama
 - Skrývání informací = implementační detaily jsou schovány za množinou veřejných metod

Objektové paradigma

- Dědičnost
 - Umožňuje při definici třídy uvést odkaz na jinou třídu, jejíž vlastnosti se přidají do nové třídy
 - Vlastnosti dědičnosti
 - Atributy a metody z nadtřídy jsou vloženy do podtřídy
 - Metody podtřídy mohou překrýt metody nadtřídy
 - Třída může dědit z více tříd – ne ve všech jazycích

Objektové paradigma

- Abstraktní třída
 - Atributy jsou povoleny
 - Metody jsou povoleny a mohou být abstraktní
 - Konstruktory jsou povoleny, ale nelze je použít k vytváření instancí
 - Podtřída musí definovat implementace všech abstraktních metod nebo musí být také abstraktní

Objektové paradigma

- Polymorfizmus
 - Při volání metody je implementace metody určena typem objektu, ne typem proměnné

Objektové paradigma

Dotazy?

Základní pravidla

- Správný návrh metody
 - Základní stavební blok
 - Dodržovat princip vysoké soudržnosti
 - Prázdný řádek v těle metody
 - Komentář v těle metody
 - Pojmenování
 - Srozumitelné
 - Samo popisné

Základní pravidla

- Správný návrh metody
 - Délka těla metody
 - Čím delší, tím vyšší chybovost
 - Jedna obrazovka
 - Max. 100-200 řádků

Základní pravidla

- Vyhnout se duplikacím
 - Kódu
 - Zapouzdření funkčnosti do metody
 - Dat
 - Správný návrh způsobu ukládání dat
 - Používání pojmenovaných konstant

Základní pravidla

- Programování proti rozhraní
 - Signatura
 - Název metody
 - Parametry
 - Návrátový typ
 - Výjimky
 - Kontrakt
 - Povolené hodnoty parametrů
 - Kontrola za běhu programu
 - Definován pomocí dokumentačních komentářů

Základní pravidla

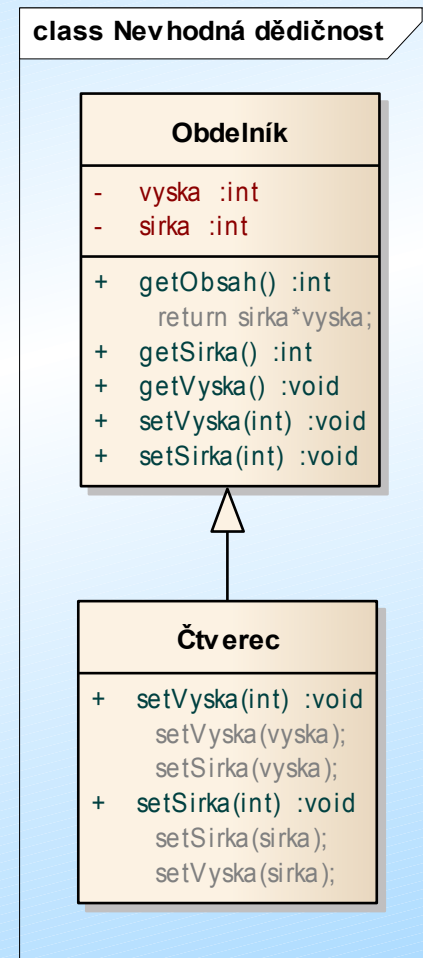
- Programování proti rozhraní
 - Rozhraní všude tam, kde jsou očekávány změny nebo rozšíření
 - Zveřejněné rozhraní je „neměnné“.

Základní pravidla

- Pozor na dědičnost
 - Měla by splňovat Liskov substitution princip:
 - pokud instanci nadtřídy nahradíme instancí podtřídy, pak kód, který očekává instanci nadtřídy, by měl fungovat i s instancí podtřídy
 - Rozhodně bychom neměli používat dědičnost jenom proto, že obě třídy mají určitou množinu společných atributů

Základní pravidla

- Nevhodné použití dědičnosti



Základní pravidla

- Nevhodné použití dědičnosti

```
Obdelnik obdelnik = new Ctverec();  
obdelnik.setSirka(2);  
obdelnik.setVyska(3);  
obsah = obdelnik.getObsah();
```

- Jaký bude výsledek?

Základní pravidla

- Upřednostnění skládání před dědičností
 - Dědičnost je velmi silná vazba
 - Většina jazyků umožňuje použít pouze jednoduchou dědičnost
 - Vícenásobná dědičnost většinou není vhodným řešením
 - Většinou porušuje princip vysoké soudržnosti

Základní pravidla

- Upřednostnění skládání před dědičností
 - Nevhodné použití dědičnosti
 - Zřetězený seznam a Fronta
 - Bod a Čtverec

Základní pravidla

- Dodržování konvencí psaní kódu (štábní kultura)
 - Pojmenování tříd, metod, parametrů
 - Velikost písmen, oddělování slov
 - Srozumitelnost použitých názvů
 - Komentáře
 - Odsazování
 - atd..

Základní pravidla

- Dodržování konvencí psaní kódu (štábní kultura)
 - Jmenné konvence zdrojových kódů:
 - <http://www.oracle.com/technetwork/java/codeconv-138413.html>
 - <http://msdn.microsoft.com/en-us/library/0b283bse%28v=vs.71%29.aspx>
 - <http://msdn.microsoft.com/en-us/library/ff926074.aspx>
 - <http://pear.php.net/manual/en/standards.php>

Základní pravidla

- Psát dobře čitelný kód
 - Co dělá tento řádek kódu?

```
float z = 50+y(x.x());
```

Základní pravidla

- Psát dobře čitelný kód

```
float z = 50+y(x.x());
```

nebo

```
public static final float SPRAVNI_POPLATEK=50;
```

```
...
```

```
int delkaZpozdeni = vypujcka.delkaZpozdeni();
```

```
float pokuta = pokutaZaZpozdeni(delkaZpozdeni);
```

```
float celkem=SPRAVNI_POPLATEK+pokuta;
```

Základní pravidla

- Nepodřizovat návrh snaze zlepšit efektivitu
 - Není vhodné optimalizovat něco, o čem nevíme, zda to skutečně bude program zdržovat
 - Je mnohem důležitější připravit kód na možné budoucí změny

Základní pravidla

- Konfiguraci do externích souborů
 - Pro usnadnění procesu nasazení v různých prostředích
- Popisy GUI do externích souborů
 - Umožňuje provedení lokalizace
 - Souvisí s duplikací dat

Základní pravidla

- Literatura

- Pecinovský, R. Návrhové vzory. ISBN 978-80-251-1582-4
- McConnell, S. Code Complete: A Practical Handbook of Software Construction, Second Edition. ISBN 978-0735619678

Základní pravidla

Dotazy?

Děkuji za pozornost.