

Zápočtový test 2.2.2009 – Model verifier

Obecné požadavky:

- 3hod čistého času
- po třech hodinách se můžete rozhodnout, zda chcete aplikaci předvést a nebo odejít. Kdo se rozhodne počkat na předvedení, nesmí již aplikaci dále upravovat!
- žádný internet
- slidy z přednášky a literatura v papírové formě jsou povolené

Zadání:

Cílem testu je naprogramovat aplikaci, která rozhodne zda daný model odpovídá všem pravidlům zadaného popisu meta-modelu.

Meta-model – popis tříd a vztahů mezi nima.

Model – instanciuje třídy, které jsou definované meta-modelem. Musí brát v úvahu omezení (atributy tříd, vztahy mezi nima, ...), které meta-model specifikuje.

Specifikace meta-modelu

Třída

```
class A [extends NazevTridy]
  attribute Int|String <name>
end
```

– extends znamená, že třída A obsahuje všechny atributy třídy *NazevTridy* a také dědí všechny asociace – viz níže.

- nastavení atributu v modelu
 - `a.setName(3)`
 - `a.setNazev("zbla")`

Asociace (vztahy mezi třídami)

A --<nazev-asociace>--> B 1:1

- říká, že jedna instance třídy A obsahuje odkaz na právě jednu instanci třídy B
- tj. `a.setNazevAsociace(b)`

A --<nazev-asociace>--> B 1:N

- říká, že jedna třída A obsahuje odkaz na libovolný počet instancí třídy B
- tj. `a.getNazevAsociace().add(b)`

A <--<nazev-asociace>--> B 1:1

- říká, že jedna třída A obsahuje odkaz na právě jednu instanci třídy B a naopak
- tj. `a.setNazevAsociace(b)`, ale také `b.setNazevAsociace(a)`

A <--<nazev-asociace>--> B 1:N

- obousměrná asociace, třída A se odkazuje na libovolný počet instancí třídy B (0..*), a instance

třídy B obsahují odkaz na A

- tj. `a.getNazevAsociace().add(b)`, ale také `b.setNazevAsociace(a)`

A <--<nazev-asociace>-->B N:M

- obecná relace mezi instancemi třídy A, B. A se může odkazovat na libovolný počet instancí třídy B a naopak
- tj. `a.getNazevAsociace().add(b)` a také `b.getNazevAsociace().add(a)`

Specifikace modelu

Popisuje postup instanciaci modelu. Jeden příkaz na řádek.

Např.

```
a = new A
b = new B
a.setAttribute1(13)
a.setAssociation(b)
```

Poznámka: Každý řádek, který začíná znakem # musí být ignorován, jelikož se jedná o komentář.

Volání programu:

program.exe <meta-model> <model>

Vstup:

- meta-model
 - v textovém souboru, můžete předpokládat, že bude korektně zadaný
- model
 - v textovém souboru, můžete předpokládat, že bude korektně zadaný (syntakticky)

Výstup:

- “OK” pokud model splňuje podmínky meta-modelu
 - jinak popis **první** nalezené nesrovnalosti (LINE je číslo řádku s chybou)
 - “LINE: Neexistující atribut XXX”
 - “LINE: Neexistující asociace XXX”
 - “LINE: XXX není instancí třídy YYY”
-

Příklad

Metamodel

```
class Clovek
    attribute String jmeno
    attribute Int vek
end

class Rodic extends Clovek
end
```

```

class Dite extends Clovek
end

Rodic --deti--> Dite 1:N

class Otec extends Rodic
end

class Matka extends Rodic
end

Otec <--partner--> Matka 1:1

```

Model

```

otec = new Otec
otec.setJmeno("Adam")
otec.setVek(30)

matka = new Matka
matka.setJmeno("Eva")
matka.setVek(26)

otec.setPartner(matka)
matka.setPartner(otec)

dite1 = new Dite
dite1.setJmeno("Ester")
dite1.setVek(3)

otec.getDeti().add(dite1)
matka.getDeti().add(dite1)

```

Vystup: OK

Když přidáme:

```
otec.setZamestnani("ITak")
```

Vystup:

“X: Neexistující atribut Zamestnani na trídě Otec”

Když přidáme:

```
otec.getPredchoziManzelky().add(matka)
```

Vystup:

“X: Neexistující asociace PredchoziManzelky na trídě Otec”

Když přidáme:

```
dite2.setJmeno("Ivan")
```

Vystup:

“X: Neexistující instance Dite2”

Když přidáme:

```
otec.getDeti().add(matka)
```

Vystup:

“X: matka není instancí třídy Dite”