

## Metódy a prostriedky špecifikácie

doc. Ing. Valentino Vranič, PhD., ÚISI FIIT STU

Skúška – riadny termín – 21. december 2011

A

Priezvisko:	
Meno:	

1b	
2b	
3b	

Skúška trvá 60 minút.

V otázkach 1–13 je len jedna možnosť správna. Vyznačte svoju odpoveď krížikom do tabuľky. Odpovede na otázky 14 a 15 píšete na čistý list, ktorý ste dostali.

V prípade opravy jasne vyznačte odpoveď, ktorá platí. Každá správna odpoveď má hodnotu vyznačenú v otázke. Nesprávna odpoveď, vyznačenie viac odpovedí alebo nejednoznačné vyznačenie nebude hodnotené. Postup riešenia sa pre otázky 1–13 nehodnotí. Poškodený list nebude uznaný.

	a	b	c	d	e
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					

### 1. (1 b) Závislosti medzi balíkmi v UML

- (a) netreba uvádzať, ak sú uvedené závislosti medzi ich prvkami
- (b) reflektujú závislosti medzi ich prvkami
- (c) treba uvádzať, len ak majú opačný smer ako závislosti medzi ich prvkami
- (d) nie sú povolené
- (e) musia mať stereotyp «package»

### 2. (1 b) Aby výraz v predikátovej časti schémy v jazyku Z vyjadroval dôsledok, musí

- (a) obsahovať aspoň jednu premennú bez apostrofu
- (b) označený kľúčovým slovom post
- (c) obsahovať len premenné bez apostrofu
- (d) obsahovať aspoň jednu premennú s apostrofom
- (e) obsahovať len premenné s apostrofom

### 3. (1 b) Jedným z mechanizmov vyjadrenia variability v modelovaní vlastností sú

- (a) spoločné vlastnosti
- (b) alternatívne vlastnosti
- (c) podvlastnosti
- (d) body spájania
- (e) koncepty

### 4. (1 b) Funkcionálna dekompozícia je pri modelovaní prípadov použitia

- (a) dôležitá
- (b) nevyhnutná
- (c) nepodstatná
- (d) aplikovateľná
- (e) nevhodná

### 5. (1 b) Použitie vzťahu generalizácie-spezializácie medzi účastníkmi na modelovanie používateľských práv je

- (a) možné, ale nepostačuje
- (b) možné, ale nie je odporúčané
- (c) nekorektné
- (d) nevyhnutné
- (e) odporúčané

### 6. (2 b) Informačný systém galérie by okrem iného mal slúžiť na evidenciu všetkých umeleckých diel v galérii. Na základe tejto požiadavky by ste na toto v jazyku Z použili

- (a) množinu
- (b) pole
- (c) postupnosť
- (d) schému
- (e) vymenovaný typ

### 7. (2 b) Jeden z rozdielov medzi Cockburnovou a Jacobsonovou notáciou modelovania prípadov použitia spočíva v tom, že

- (a) jedna podporuje alternatívne toky, kým iná nie
- (b) v jednej sa kroky čísloujú, kým v inej nie
- (c) v jednej sa uprednostňuje väzba extend, kým v druhej väzba include
- (d) v jednej sa body rozšírenia vyjadrujú priamo, kým v inej nie
- (e) v jednej je možná generalizácia prípadov použitia, kým v inej nie

### 8. (2 b) V metóde OOram

- (a) objekt je rola
- (b) rola pozostáva z viacerých objektov
- (c) objekt pozostáva z viacerých rolí
- (d) objekt hrá práve jednu rolu
- (e) objekt môže hrať viac rolí

### 9. (2 b) Pri implementácii vzťahu extend aspektovo-orientovaným spôsobom v jazyku AspectJ je vhodné

- (a) pre rozširujúci prípad použitia použiť medzitypovú deklaráciu
- (b) pre rozširujúci prípad použitia použiť videnie
- (c) pre rozširovaný prípad použitia použiť ako videnie
- (d) pre rozširovaný prípad použitia použiť medzitypovú deklaráciu
- (e) aj pre rozširujúci, aj pre rozširovaný prípad použitia použiť videnie

### 10. (2 b) Obojsmerná závislosť medzi balíkmi je

- (a) vítaná vlastnosť
- (b) neriešiteľný problém
- (c) tolerovateľná pri menších balíkoch
- (d) jav bez zvláštneho významu
- (e) riešiteľný problém

11. (3 b) Grafický objekt môže byť viditeľný alebo nie, od čoho závisia ďalšie operácie nad ním. V algebraickej špecifikácii by ste túto skutočnosť modelovali

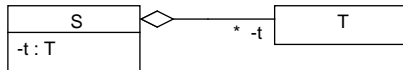
- (a) pridaním atribútu *viditelny* : *Boolean*
- (b) pridaním aj atribútu *viditelny* : *Boolean*, aj operácie *jeViditelny* : *GObject* → *Boolean*
- (c) pridaním operácie *jeViditelny* : *Boolean*
- (d) pridaním operácie *jeViditelny* : *GObject* → *Boolean*
- (e) pridaním aj atribútu *viditelny* : *Boolean*, aj operácie *jeViditelny* : *Boolean*

12. (3 b) Prípad použitia *A* je na úrovni používateľského rozhrania realizovaný formulárom *FormX*. Podobne, prípad použitia *B* je realizovaný formulárom *FormY*. Formulár *FormX* obsahuje tlačidlo, ktorým je možné vyvolať formulár *FormY*. Na základe tohto sa

- (a) nedá povedať, či je medzi *A* a *B* nejaký vzťah
- (b) dá povedať, že medzi *A* a *B* je závislosť, ale nedá sa určiť jej smer
- (c) dá povedať, že *A* zahŕňa *B*
- (d) dá povedať, že *B* zahŕňa *A*
- (e) dá povedať, že *B* rozširuje *A*

13. (3 b) V diagrame na obrázku 1 treba

- (a) pridať násobnosť k atribútu *t* – [\*]
- (b) vypustiť atribút *t*
- (c) vypustiť atribút *t* alebo vzťah medzi *S* a *T*
- (d) vypustiť vzťah medzi *S* a *T*
- (e) zmeniť kosoštvorec na vyplnený



Obr. 1: Diagram k otázke 13.

14. (6 b) Druh výrobku je v elektronickom obchode reprezentovaný triedou *Vyrobok*. Operácia tejto triedy

`nastavCenu(cena : Mena, zvyhodnenaCena : Mena)`

nastavuje bežnú a zvyhodnenú cenu výrobku. Zvýhodnená cena musí byť nižšia od bežnej ceny, čo je vyjadrené zodpovedajúcim ohraničením v OCL:

```

context Vyrobok :: nastavCenu(cena : Mena,
    zvyhodnenaCena : Mena) : void
pre zvyhodnenie:
    zvyhodnenaCena < cena
  
```

Obchod príležitostne predáva aj výrobky, ktoré nemajú zvyhodnenú cenu. Od triedy *Vyrobok* je preto odvodená trieda *SpecialnyVyrobok*. Dá sa a ako dosiahnuť, aby sa na triedu *SpecialnyVyrobok* nevzťahovalo uvedené ohraničenie? Vysvetlenie ozrejmite príkladom použitia.

15. (5 b) Komponenty *C*, *D* a *E* poskytujú operácie *hop()* a *kop()* predpísané rozhraním *P*. Komponenty *X*, *Y* a *Z* volajú operácie *hop()* a *kop()* a poskytujú operáciu *m()* predpísanú rozhraním *Q*. Komponent *W* volá operáciu *m()*. Vyjadrite príslušnú štruktúru zodpovedajúcim diagramom v jazyku UML a vysvetlite ju. Dbajte o flexibilitu riešenia.

35 b

1 b

2 d

3 b

4 e

5 c

6 a

7 d

8 e

9 b

10 e

11 d

12 a

13 c

14

Uvedené ohraňenie predstavuje predpoklad operácie *nastavCenu()*. V odvodenom type je potrebné tento predpoklad vynechať, čo predstavuje jeho zoslabenie. Zoslabenie predpokladu nenarušuje vzťah typ-podtyp, takže je to možné urobiť. Príslušné ohraňenie v OCL by vyzeralo takto:

```
context SpecialnyVyrobok :: nastavCenu (
    cena : Mena,
    zvyhodnenaCena : Mena) : void
pre zvyhodnenie:
```

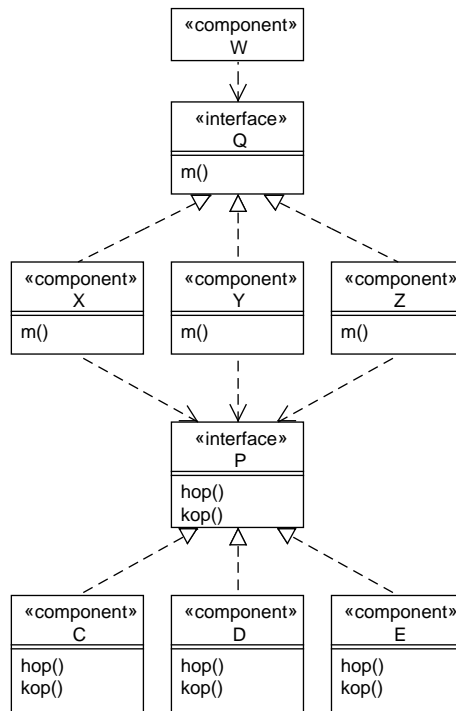
Klientskemu kódu, ktorý by napríklad pristupoval k inštanciám triedy *SpecialnyVyrobok* cez rozhranie ich nadtypu *Vyrobok* a hromadne v slučke nastavoval ceny, by takáto relaxácia predpokladu nespôsobilala problémy:

```
for (Vyrobok vyrobok : zoznamVyrobkov)
    nastavCenu(vyrobok.getCena(),
        vyrobok.getCena() * 0.9);
```

kde predpokladáme existenciu operácie *getCena()*, ktorá vracia aktuálnu cenu výrobku.

Na druhej strane, uvedené zoslabenie by spôsobilo aj to, že by v inštanciách triedy *SpecialnyVyrobok* bolo možné nastaviť vyššiu zvyhodnenú cenu než je bežná cena. Toto vlastne predstavuje zoslabenie dôsledku – hoci aj implicitného (bez vyjadrenia v OCL) – že by zvyhodnená cena mala byť nižšia než bežná cena. Klientsky kód, ktorý by napríklad počítal priemerné zvyhodnenie, by v takomto prípade dal mylný výsledok. Toto však predmetom otázky bezprostredne nebolo.

15



Komponenty C, D a E realizujú rozhranie P, ktoré predpisuje operácie *hop()* a *kop()*. Komponenty X, Y a Z používajú rozhranie P, čím nevzniká ich priama závislosť od komponentov C, D a E.

Podobne komponenty X a Y a Z realizujú rozhranie Q, ktoré predpisuje operáciu *m()*. Komponent W používa rozhranie Q, čím zase nevzniká jeho priama závislosť od komponentov X, Y a Z.