

Priezvisko:	tlačeným písmom
Meno:	

1b	
2b	
3b	

Skúška trvá 60 minút.

Odpovede na otázky 1–9 vpíšte do tabuľky vľavo. Odpovede na otázky 10 a 11 píšete na prídavný list, ktorý ste dostali.

V otázkach s výberom odpovede je len jedna možnosť správna a hodnotia sa len odpovede v tabuľke bez hodnotenia postupu. Správna odpoveď má hodnotu vyznačenú v otázke.

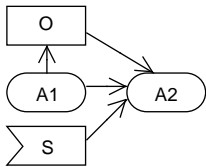
V prípade opravy jasne vyznačte odpoveď, ktorá platí. Nesprávna, nejednoznačná alebo neúplná odpoveď v otázkach 1–9 má hodnotu 0 bodov. Len nepoškodený list bude akceptovaný.

1	
2	
3	
4	
5	
6	
7	
8	
9	

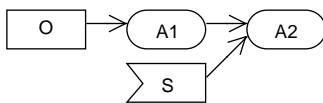
1. (1 b) V jazyku UML uvedenie atribútu v triede (s výnimkou atribútu primitívneho typu) predstavuje formu

- (a) oddelenia záležitostí
- (b) generalizácie
- (c) abstrakcie
- (d) špecializácie
- (e) agregácie

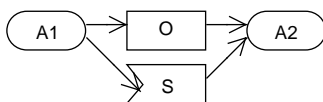
2. (3 b) Po akcií A1 sa do akcie A2 pošle objekt O, pričom sa akcia A2 aktivuje až po prijatí signalu S. Presne túto situáciu znázorňuje obrázok



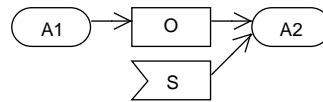
(a)



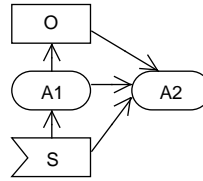
(b)



(c)



(d)



(e)

3. (2 b) Pri modelovaní prípadov použitia podľa Cockburna pred podmienenými krokmi sa uprednostňujú

- (a) pomocné toky
- (b) alternatívne toky
- (c) vzťahy dedenia medzi prípadmi použitia
- (d) ošetrenia v predpokladoch prípadu použitia
- (e) ošetrenia v dôsledkoch prípadu použitia

4. (3 b) V algebraickej špecifikácii grafického objektu okrem iných boli identifikované operácie zistenia farby (*color*) a vyplnenia (*fill*):

$$color : GObject \rightarrow Color$$

$$fill : GObject \times Color \rightarrow GObject$$

Axióma, ktorá hovorí, že po vyplnení grafický objekt má mať zadanú farbu výplne, znie:

- (a)  $\forall o : GObject, c : color \bullet color(fill(o, c)) = c$
- (b)  $\forall o : GObject \exists c : color \bullet color(o) = c$
- (c)  $\exists o : GObject, c : color \bullet fill(o, c) \Rightarrow color(o) = c$
- (d)  $\forall o : GObject, c : color \bullet fill(o, c) \Rightarrow color(o) = c$
- (e)  $\exists o : GObject, c : color \bullet color(fill(o, c)) = c$

5. (2 b) Podľa nasledujúcej schémy v jazyku Z:

$$\frac{Op \quad \exists S \quad a? : M \quad b! : M}{a? \notin W \quad b! = a?}$$

Posledný riadok vyjadruje:

- (a) predpoklad
- (b) invariant
- (c) aj predpoklad, aj dôsledok
- (d) nič z uvedeného
- (e) dôsledok

6. (1 b) Prístup DCI sa zameriava na

- (a) zachovanie prípadov použitia v kóde
- (b) využitie aspektovo-orientovaného programovania
- (c) tvorbu modelov v UML pred napísaním kódu
- (d) použitie rolí namiesto tried
- (e) použitie procedurálneho programovania

7. (2 b) Je možné realizovať sprístupnenie viacerých prípadov použitia prostredníctvom rôznych formulárov GUI?

- (a) áno, ale len s použitím aspektovo-orientovaného programovania
- (b) áno, ak používajú rozdielne ovládacie prvky formulárov
- (c) nie
- (d) áno, toto je obvyklé
- (e) áno, ale je to nežiaduce

8. (3 b) V systéme na podporu tvorby návrhov projektov každý návrh projektu je najprv predmetom úprav, pričom vznikajú nové verzie návrhu projektu. Návrh projektu môže byť predložený na schválenie, výsledkom čoho môže byť schválenie, zamietnutie alebo vrátenie na úpravu. Po schválení návrh projektu už nie je možné upravovať ani znovu predkladať. Korektný stavový diagram návrhu projektu s minimálnym počtom uzlov, ktorý by vyhovoval tomuto opisu, by okrem uzlov počiatočného a koncového stavu mal obsahovať nasledujúce uzly:

- (a) Upravovaný, Predložený, Vrátený, Schválený a Zamietnutý
- (b) Upravovaný, Nová verzia, Predložený, Vrátený, Schválený a Zamietnutý
- (c) Upravovaný, Predložený, Schválený a Zamietnutý
- (d) Predložený a Vrátený, pričom počiatočný stav bude označený ako Upravovaný a koncový stav ako Schválený/Zamietnutý
- (e) Predložený, pričom počiatočný stav bude označený ako Upravovaný a koncový stav ako Schválený/Zamietnutý
- (f) Nová verzia, Predložený a Vrátený, pričom počiatočný stav bude označený ako Upravovaný a koncový stav ako Schválený/Zamietnutý

9. (3 b) Udalosť je v osobnom plánovači reprezentovaná triedou *Event*. Operácia tejto triedy *nastav()* umožňuje nastaviť parametre udalosti iníciaľne a tiež aj ich následnú zmenu:

```
setTime(start : Time, end : Time)
```

kde *Time* predstavuje dátum a čas. Pritom nie je povolené meniť začiatok a koniec udalosti, ktorá prebieha alebo skončila, čo je vyjadrené zodpovedajúcim ohraničením v OCL:

context Event::

```
setTime(start : Time, end : Time) : void
```

```
pre TimeChange:
```

```
currentTime.before(start) and
```

```
currentTime.before(end)
```

kde operácia *before()* triedy *Time* vracia *true*, ak je daný dátum a čas pred dátumom a časom poskytnutým ako parameter, a inak vracia *false*. Premenná *currentTime* je typu *Time* a obsahuje aktuálny čas.

Trieda *ExternalEvent* dedí od triedy *Event*. Časové údaje tohto typu udalosti sa po iníciaľnom nastavení, pred ktorým majú hodnotu *null*, nesmú meniť. Dá sa toto doplniť do existujúceho modelu ako podmienka v jazyku OCL?

- (a) áno, ale vzťah dedenia nebude korektný, lebo predpoklad bude zoslabený
- (b) áno, ale vzťah dedenia nebude korektný, lebo predpoklad bude zosilnený
- (c) áno, so zachovaním korektnosti dedenia
- (d) nie
- (e) áno, ale vzťah dedenia nebude korektný, lebo predpoklad bude *null*

10. (4 b) Nástroj na kompresiu údajov vykonáva kompresiu jedným z troch algoritmov podľa nastavení. Vyjadrite adekvátne štruktúru v UML diagramom komponentov. Aký zásah do tejto štruktúry vyžaduje pridanie ďalšieho algoritmu kompresie? (Model nastavení nie je predmetom otázky.)

11. (6 b) Aplikácia umožňuje poslať SMS jednému alebo viacerým príjemcom. Po odoslaní sa SMS uloží do zoznamu odoslaných správ. Odoslanie každej SMS je spoplatnené sadzbou podľa balíka služieb, ktorý používateľ má. Identifikujte a pomenujte dva hlavné prípady použitia, ktoré vyplývajú z tohto opisu, a uveďte ich v krokoch. Zabezpečte pritom, aby vyjadrenie odoslania SMS bolo zbavené závislosti od spoplatňovania. Nakreslite zodpovedajúci diagram prípadov použitia.

30 b

1 e

2 d

3 b

4 a

5 e

6 a

7 d

8 c

9 b

10 Odpoveďou je obvyklá schéma zastrešenia komponentov rozhraním. Klientske komponenty potom využívajú rozhranie. Kompresia je naznačená príslušnou operáciou.

11 Dva hlavné prípady použitia by sa dali nazvať *Odošli SMS* a *Spoplatni odoslanie SMS* (ten druhý by pritom mohol byť koncipovaný širšie na sústredenie spoplatnenia všetkých poskytovaných služieb). Prípad použitia *Spoplatni odoslanie SMS* rozširuje (extend) *Odošli SMS*.