

**Zadání a řešení testu z informatiky a zpráva
o výsledcích přijímacího řízení do magisterského
navazujícího studia od podzimu 2017**

**Zpráva o výsledcích přijímacího řízení
do magisterského navazujícího studia od podzimu 2017**

Počet podaných přihlášek	389
Počet přihlášených uchazečů	358
Počet uchazečů, kteří splnili podmínky přijetí	243
Počet uchazečů, kteří nesplnili podmínky přijetí	116
Počet uchazečů přijatých ke studiu, bez uvedení počtu uchazečů přijatých ke studiu až na základě výsledku přezkoumání původního rozhodnutí	243
Počet uchazečů přijatých celkem	243
Percentil pro přijetí	15,00

Základní statistické charakteristiky

	Informatika	Matematika	Celkem	
Počet otázek	30	25	55	
Počet uchazečů, kteří se zúčastnili přijímací zkoušky	151	151	151	
Nejlepší možný výsledek	30.00	25.00	55.00	
Nejlepší skutečně dosažený výsledek	21.50	20.00	36.50	
Průměrný výsledek	7.91	7.83	15.74	
Medián	7.75	8.00	15.25	
Směrodatná odchylka	4.63	4.78	8.17	
	Percentil			
Decilové hranice výsledku *	10	2.25	1.50	4.75
	20	4.00	3.25	8.50
	30	5.75	5.00	11.50
	40	6.50	6.75	13.25
	50	7.75	8.00	15.25
	60	8.75	9.25	18.00
	70	10.00	10.50	21.00
	80	12.25	11.50	23.00
	90	13.25	14.50	25.75

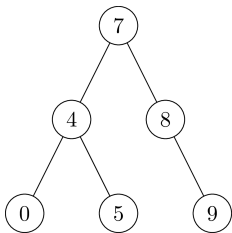
* Decilové hranice výsledku zkoušky vyjádřené d1, d2, d3, d4, d5, d6, d7, d8, d9 jsou hranice stanovené tak, že rozdělují uchazeče seřazené podle výsledku zkoušky do stejně velkých skupin, přičemž d5 je medián.

Přijímací zkouška - Informatika

Jméno a příjmení - pište do okénka	Číslo přihlášky	Číslo zadání
		24

Algoritmy a datové struktury

- 1** Uvažujme binární vyhledávací strom (BVS) s n prvky a následující čtyři operace: SEARCH (najde zadaný prvek v BVS), INSERT (vloží nový prvek do BVS), REMOVE (odstraní zadaný prvek z BVS) a ROOT (vrátí hodnotu kořenového prvku v BVS). Kolik z těchto čtyř operací má v *nejhorším* případě *lineární* časovou složitost?
- *A 3
 - B 2
 - C 4
 - D 1
 - E 0
- 2** Uvažujme orientovaný graf. Na tomto grafu spustíme prohledávání do hloubky (depth-first search); toto prohledávání přiřadí každému vrcholu v dvě čísla: $v.d$ je čas objevení v , $v.f$ je čas opuštění (uzavření, ukončení) v . Které z těchto tvrzení je obecně pravdivé?
- A Existuje-li cesta z vrcholu u do vrcholu v v grafu, pak $u.d < v.d$.
 - B Existuje-li cesta z vrcholu u do vrcholu v v grafu, pak $u.f > v.f$.
 - C Existuje-li cesta z vrcholu u do vrcholu v v grafu, pak $u.d > v.d$.
 - *D Žádné z ostatních tvrzení není obecně pravdivé.
 - E Existuje-li cesta z vrcholu u do vrcholu v v grafu, pak $u.f < v.f$.
- 3** Které z těchto tvrzení je pravdivé?
- A B-stromy jsou speciálním případem binárních vyhledávacích stromů.
 - B Časová složitost vkládání prvku do hašovací tabulky je v nejhorším případě v $\mathcal{O}(1)$.
 - *C Pro hledání zadaného prvku v binární haldě neexistuje algoritmus se sublineární časovou složitostí.
 - D Řadící algoritmus mergesort je stabilní, in situ (inplace, v místě) a má časovou složitost v $\mathcal{O}(n \log n)$.
 - E Funkce n^n roste asymptoticky stejně rychle jako funkce $n!$.



- 4** Uvažujme strom na obrázku výše. Kolika způsoby můžeme vrcholům tohoto stromu přiřadit červenou nebo černou barvu tak, aby výsledný strom byl korektním červenočerným stromem?
- *A Existují právě dva způsoby.
 - B Existuje právě osm způsobů.
 - C Existují právě čtyři způsoby.
 - D Existuje právě jeden způsob.
 - E Žádný takový způsob není.

- 5** Uvažujme hašovací tabulku s lineárním sondováním (linear probing) a hašovací funkcí $h(x) = (5 \cdot x + 3) \bmod 7$. Začínáme s prázdnou hašovací tabulkou. V jakém pořadí musíme do tabulky vložit hodnoty 1, 2, 5, 7, 9, 12 tak, aby obsah výsledné tabulky byl následující: 2, 12, 1, 5, 7, prázdná položka, 9?
- *A 9, 2, 12, 1, 5, 7
 - B 2, 9, 5, 12, 1, 7
 - C 9, 2, 12, 5, 1, 7
 - D 2, 9, 12, 5, 1, 7
 - E 9, 2, 5, 12, 1, 7

Počítačové systémy

- 6** Jedna paměťová buňka DRAM, která slouží pro uložení jednoho bitu, se skládá z:
- A dvou tranzistorů a jednoho kondenzátoru.
 - B jednoho bistabilního klopného obvodu.
 - C jednoho tranzistoru a jednoho rezistoru.
 - D jednoho kondenzátoru.
 - *E jednoho tranzistoru a jednoho kondenzátoru.
-
- 7** Které číslo ve dvojkovém doplňkovém kódu je ekvivalentem čísla vyjádřeného v desítkové soustavě jako -88?
- A 11011000
 - *B 10101000
 - C 11101010
 - D 10100111
 - E 10001010
-
- 8** V průběhu převodu logické (virtuální) adresy na adresu fyzickou se v chráněném režimu u desktopových procesorů Intel (a kompatibilních) **nevyužívá**:
- A stránková tabulka.
 - B lineární adresa.
 - C stránkový adresář.
 - *D jednotka DMA (Direct Memory Access).
 - E tabulka deskriptorů.
-
- 9** Které z následujících tvrzení o multitaskingu v operačních systémech **není** pravdivé:
- A Preemptivní multitasking zahrnuje použití mechanismu přerušení.
 - *B Kooperativní multitasking umožňuje počítačovému systému spolehlivěji garantovat, že každému procesu bude procesor přidělen na stálé časové kvantum.
 - C Preemptivní multitasking je způsob multitaskingu, při kterém jádro operačního systému může iniciovat přepnutí kontextu tak, aby bylo vyhověno prioritám plánovací politiky.
 - D Kooperativní multitasking je způsob multitaskingu, při kterém operační systém nikdy neinicuje přepnutí kontextu z běžícího procesu na jiný proces.
 - E Kooperativní multitasking je způsob multitaskingu, při kterém procesy dobrovolně přenechávají řízení periodicky nebo v době nečinnosti tak, aby více aplikací mohlo běžet současně.
-
- 10** Necht' je dán logický obvod multiplexor se 16 *datovými* vstupy. Jaký je *celkový* počet jeho vstupů?
- A 24
 - B 32
 - *C 20
 - D 18
 - E 16
-

Programování

- 11** Které tvrzení je obecně platné v běžných OOP jazycích (C++, Java, C#)?
- A Pokud třída B dědí od třídy A, pak třída B může přistupovat ke všem atributům (členským proměnným) třídy A.
 - *B Žádné z ostatních tvrzení není pravdivé.
 - C Rozdíl mezi třídou (class) a objektem (object) je v tom, že třídy jsou alokovány na haldě, zatímco objekty jsou alokovány na zásobníku.
 - D Je-li použita pozdní vazba (volání virtuálních metod), pak o tom, která metoda se skutečně zavolá, rozhoduje překladač v době překladu.
 - E Statické metody (členské funkce) mohou být virtuální.
-

- 12** Která z následujících tří tvrzení I, II a III jsou pravdivá (v běžných jazycích typu C++, Java, C#)? Vyberte možnost, která obsahuje právě všechna pravdivá tvrzení (a žádná nepravdivá).
- I. Lokální proměnné funkcí jsou alokovány na haldě. Při opuštění funkce jsou automaticky dealokovány.
- II. Volání funkcí jsou implementována pomocí zásobníku a tento zásobník obsahuje návratové hodnoty funkcí.
- III. Je-li zachycena výjimka (v bloku catch), může být znovu vyhozena (použitím throw).
- A** I, II, III
B II, III
***C** III
D I, II
E I, III
-

- 13** Uvažujme následující program. Funkce print vypíše zadané číslo následované koncem řádku.

```
function foo(integer n)
begin
    print n
    if n > 0 then
        foo(n-1)
        print n
        foo(n-1)
        print n
    end if
end

program main()
begin
    foo(3)
end
```

Kolik řádků tento program vypíše?

- A** Program bude cyklit a nikdy neskončí.
B 7
C 30
D Žádná z ostatních odpovědí není pravdivá.
***E** 29
-

- 14** Uvažujme následující program. Operátor mod počítá zbytek po dělení (modulo) a operátor div počítá celočíselné dělení.

```
read(n)
sum = 0
k = n
XXX {
    sum = sum + (n mod 10)
    n = n div 10
}
print sum
```

Na vstupu je zadáno kladné celé číslo. Toto číslo je načteno do proměnné n . Který z následujících kusů kódu můžeme do programu napsat místo „XXX“, aby program počítal součet desítkových cifer zadaného čísla?

- A** Žádná z ostatních odpovědí není pravdivá.
B while n == k
***C** for i = 0 to k
D for i = 1 to 10
E while n > sum
-

15 Které z následujících tvrzení je **nepravdivé**?

- *A Haskell není čistě funkcionální jazyk, protože umožňuje operace vstupu a výstupu (I/O).
- B V čistě funkcionálních jazycích nemohou mít funkce vedlejší efekty.
- C Tail-rekurzivní funkci je vždy možno přepsat iterativním způsobem.
- D Rekurzivní funkci je vždy možno přepsat iterativním způsobem.
- E Líná vyhodnocovací strategie ve funkcionálním programování umožňuje pracovat s nekonečnými datovými strukturami.

Počítačové sítě

16 TCP protokol poskytuje zabezpečený transport dat včetně kontroly toku dat a zahlcení sítě. Zpráva je dělena do paketů, které jsou číslovány. Pořadové číslo každého paketu se odvozuje

- A od náhodně stanoveného začátku, ke kterému se přičítá jednička za každý odeslaný paket tvořící zprávu.
- B od nuly, k níž se přičítá pořadové číslo toho bytu ve zprávě, kterým paket začíná.
- C od náhodně stanoveného začátku, ke kterému se přičítá pořadové číslo toho bitu ve zprávě, kterým paket začíná.
- *D od náhodně stanoveného začátku, ke kterému se přičítá pořadové číslo toho bytu ve zprávě, kterým paket začíná.
- E od náhodně stanoveného začátku, ke kterému se přičítá pořadové číslo toho bytu ve zprávě, kterým paket končí.

17 Elektronická pošta

- A používá protokol IMAP4 k doručení zprávy od lokálního poštovního serveru k cílovému poštovnímu serveru a protokol POP3 pro stažení nebo přečtení zprávy.
- B používá protokol POP3 nebo IMAP4 jak pro doručení zprávy lokálnímu poštovnímu serveru, tak i pro stažení nebo přečtení zprávy.
- C používá protokol POP3 nebo IMAP4 k doručení zprávy od lokálního poštovního serveru k cílovému poštovnímu serveru.
- D používá protokol POP3 k doručení zprávy od lokálního poštovního serveru k cílovému poštovnímu serveru a protokol SMTP pro stažení nebo přečtení zprávy.
- *E používá protokol SMTP k doručení zprávy lokálnímu poštovnímu serveru a od lokálního poštovního serveru k cílovému poštovnímu serveru, ale protokol POP3 nebo IMAP4 pro stažení nebo přečtení zprávy.

18 Při všesměrovém bezdrátovém vysílání se vysílaná energie šíří od vysílače všemi směry. Přijímaná energie přijímačem

- *A se snižuje kvadraticky s rostoucí vzdáleností přijímače od vysílače.
- B se snižuje exponenciálně s rostoucí vzdáleností přijímače od vysílače.
- C je shodná s vysílanou energií bez ohledu na vzdálenost od vysílače.
- D se snižuje lineárně s rostoucí vzdáleností přijímače od vysílače.
- E se snižuje kubicky s rostoucí vzdáleností přijímače od vysílače.

19 Samoopravné kódy používané při přenosu dat slouží přijímači k

- *A detekování velmi častých chyb vzniklých během přenosu dat a opravení některých z nich.
- B detekování téměř všech chyb vzniklých během přenosu dat.
- C informování vysílače o chybách vzniklých během přenosu dat.
- D informování vysílače o chybách vzniklých během přenosu dat a vyžádání zopakování přenosu chybně přenesených dat.
- E detekování a opravení všech chyb vzniklých během přenosu dat.

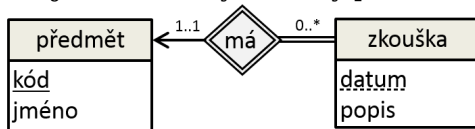
20 K přístupu na konkrétní službu cílového uzlu v síti Internet je nutné minimálně znát:

- A číslo portu.
- B IP adresu.
- C MAC adresu.
- *D IP adresu a číslo portu.
- E MAC adresu, IP adresu a číslo portu.

Databázové systémy

- 21** Relace je v Boyce-Coddově normální formě (BCNF), pokud
- A všechny atributy jsou atomické a je splněna třetí normální forma (3NF).
 - B všechny atributy jsou atomické, nezávislé, konzistentní a trvalé.
 - *C levá strana každé netriviální funkční závislosti je superklíčem.
 - D každý atribut, který není funkčně závislý na superklíči, je součástí alespoň jednoho kandidátního klíče.
 - E všechny atributy, které jsou částí libovolného cizího klíče, jsou atomické.

- 22** Vyberte správnou variantu konverze níže zobrazeného E-R diagramu do relačního modelu. Podtržením jsou označeny atributy primárních klíčů, čárkovaným podtržením jsou označeny parciální klíče.



- A $\text{předmět}(\underline{\text{kód}}, \text{jméno}, \text{datum})$, $\text{zkouška}(\underline{\text{datum}}, \text{popis})$;
- B $\text{předmět}(\underline{\text{kód}}, \text{jméno})$, $\text{zkouška}(\underline{\text{datum}}, \text{popis})$;
- *C $\text{předmět}(\underline{\text{kód}}, \text{jméno})$, $\text{zkouška}(\underline{\text{kód}}, \underline{\text{datum}}, \text{popis})$;
- D $\text{předmět}(\underline{\text{kód}}, \text{jméno})$, $\text{zkouška}(\underline{\text{datum}}, \text{popis})$, $\text{má}(\underline{\text{kód}}, \underline{\text{datum}})$;
- E $\text{předmět}(\underline{\text{kód}}, \text{jméno})$, $\text{zkouška}(\underline{\text{datum}}, \text{popis})$;

- 23** Vyberte správnou posloupnost stavů, kterými může projít databázová transakce.

- A aktivní → chybná → částečně potvrzená → zrušená
- *B aktivní → částečně potvrzená → chybná → zrušená
- C aktivní → částečně potvrzená → částečně zrušená → dokončená
- D aktivní → částečně potvrzená → chybná → potvrzená
- E aktivní → potvrzená → zrušená

- 24** Uvažujte relace *zaměstnanec* s relačním schématem *zaměstnanec*(*zam_id*, *jméno*, *plat*) a *projekt* s relačním schématem *projekt*(*proj_id*, *zam_id*, *téma*, *délka*), kde *projekt.zam_id* je cizím klíčem do relace *zaměstnanec*. Předpokládejte, že obě relace obsahují tisíce záznamů a že existuje samostatný index pro každý atribut.

Který z následujících výrazů relační algebry je nejvýkonnější transformací SQL příkazu:

```
SELECT plat * délka FROM zaměstnanec NATURAL INNER JOIN projekt
WHERE jméno = 'Joe' AND téma = 'X';
```

- A $\pi_{\text{plat} * \text{délka}}(\sigma_{\text{jméno} = \text{'Joe'}}(\sigma_{\text{téma} = \text{'X'}}(\text{projekt} \bowtie \text{zaměstnanec})))$
- B $\pi_{\text{plat} * \text{délka}}(\sigma_{\text{téma} = \text{'X'}}(\sigma_{\text{jméno} = \text{'Joe'}}(\text{zaměstnanec}) \bowtie \text{projekt}))$
- C $\pi_{\text{plat} * \text{délka}}(\sigma_{\text{jméno} = \text{'Joe'}} \wedge \text{téma} = \text{'X'}}(\text{projekt} \bowtie \text{zaměstnanec}))$
- *D $\pi_{\text{plat} * \text{délka}}(\sigma_{\text{jméno} = \text{'Joe'}}(\text{zaměstnanec}) \bowtie \sigma_{\text{téma} = \text{'X'}}(\text{projekt}))$
- E $\pi_{\text{plat} * \text{délka}}(\sigma_{\text{jméno} = \text{'Joe'}}(\text{zaměstnanec} \bowtie \sigma_{\text{téma} = \text{'X'}}(\text{projekt})))$

- 25** Aplikace pro elektronický obchod obsahuje neprázdnou relaci *produkt*(*id*, *jméno*, *množství*, *cena*, *kategorie*), jejímž primárním klíčem je atribut *id*.

Který z uvedených SQL příkazů vrátí stejný výsledek jako:

```
SELECT AVG(cena * množství) FROM produkt GROUP BY id;
```

- A `SELECT AVG(cena * množství) FROM produkt;`
- B `SELECT SUM(cena * množství) / COUNT(*) FROM produkt;`
- C `SELECT AVG(cena) * AVG(množství) FROM produkt;`
- D `SELECT SUM(cena) * SUM(množství) / COUNT(id) FROM produkt;`
- *E `SELECT cena * množství FROM produkt;`

Softwarové inženýrství

26 Která z uvedených technik nepatří mezi techniky testování software?

- A beta testování (beta testing)
- B regresní testování (regression testing)
- *C "velrybí" testování (whale testing)
- D testování jednotek (unit testing)
- E akceptační testování (acceptance testing)

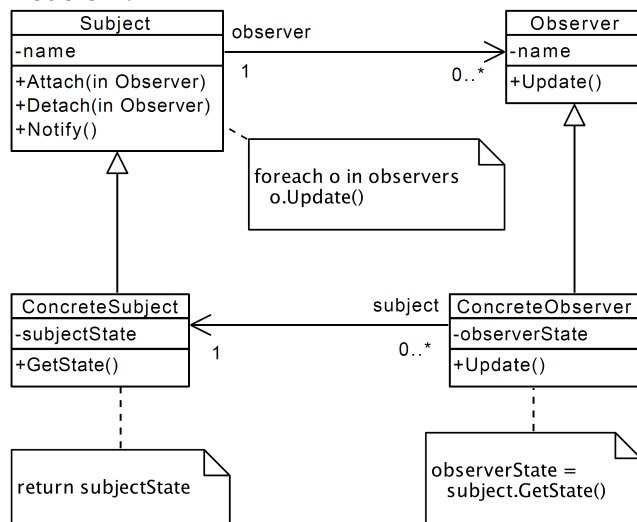
27 Která tvrzení popisují účel softwarových metrik při vývoji software?

- I. Usnadnit řízení softwarového projektu.
- II. Usnadnit určení aktuálního stavu softwarového projektu.
- III. Usnadnit řízení rizik spojených se softwarovým vývojem.
- IV. Usnadnit určení kvality softwarového produktu.

Zvolte správnou možnost:

- A Pouze II. a IV., nikoli I. a III.
- B Pouze III. a IV., nikoli I. a II.
- C Pouze I., II. a III., nikoli IV.
- D Pouze I. a III., nikoli II. a IV.
- *E Všechny I., II., III., IV.

28 Uvažujte model s UML diagramem tříd na následujícím obrázku. Které tvrzení je v souladu s tímto modelem?



- A Instance třídy ConcreteSubject nemůže mít referenci na žádnou instanci jiné třídy.
- B Instance třídy Subject má referenci na právě jednu instanci třídy Observer.
- C Instance třídy ConcreteObserver může mít referenci na nula nebo více instancí třídy ConcreteSubject.
- *D Instance třídy ConcreteSubject může mít referenci na nula nebo více instancí třídy Observer.
- E Instance třídy Observer má referenci na právě jednu instanci třídy Subject.

29 Který z diagramů jazyka Unified Modelling Language (UML) zvolíte jako nejvhodnější pro modelování interakce mezi objekty systému?

- A Diagram tříd (class diagram)
- B Paralelní diagram (parallel diagram)
- C Entitně-relační diagram (entity-relationship diagram)
- *D Sekvenční diagram (sequence diagram)
- E Diagram datových toků (data flow diagram)

-
- 30** Který z následujících popisů nejlépe charakterizuje servisně orientovanou architekturu (SOA)?
- A** SOA je architektonický styl, který doporučuje skládat složité aplikace z jasně vymezených vrstev obsahujících související služby.
 - B** SOA je přístup, který definuje procesní kroky v rámci zákaznické podpory.
 - C** SOA je aplikační architektonický styl, používaný pro systémy zákaznické podpory.
 - *D** SOA je architektonický styl, který doporučuje skládat složité aplikace ze skupiny na sobě nezávislých komponent poskytujících služby.
 - E** SOA je přístup, který definuje procesní kroky servisních operací v systému (ve fázi údržby).
-

Tato strana je prázdná.