

**Zadání a řešení testu z informatiky a zpráva
o výsledcích přijímacího řízení do magisterského
navazujícího studia od jara 2018**

**Zpráva o výsledcích přijímacího řízení
do magisterského navazujícího studia od jara 2018**

Počet podaných přihlášek	117
Počet přihlášených uchazečů	104
Počet uchazečů, kteří splnili podmínky přijetí	53
Počet uchazečů, kteří nesplnili podmínky přijetí	51
Počet uchazečů přijatých ke studiu, bez uvedení počtu uchazečů přijatých ke studiu až na základě výsledku přezkoumání původního rozhodnutí	53
Počet uchazečů přijatých celkem	53
Percentil pro přijetí	15,00

Základní statistické charakteristiky

	Informatika	Matematika	Celkem	
Počet otázek	30	25	55	
Počet uchazečů, kteří se zúčastnili přijímací zkoušky	57	57	57	
Nejlepší možný výsledek	30.00	25.00	55.00	
Nejlepší skutečně dosažený výsledek	24.25	19.25	41.00	
Průměrný výsledek	13.08	9.46	22.54	
Medián	13.00	8.75	22.50	
Směrodatná odchylka	5.29	4.84	9.23	
	Percentil			
Decilové hranice výsledku *	10	6.50	3.00	11.50
	20	10.00	5.50	16.25
	30	11.25	7.50	18.25
	40	12.25	8.00	19.75
	50	13.00	8.75	22.50
	60	13.50	11.00	23.75
	70	15.25	11.75	27.25
	80	18.25	13.50	31.00
	90	20.25	16.75	35.25

* 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
		8

Algoritmy a datové struktury

- 1** 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 (tj. **platí ve všech grafech**)?
- A Existuje-li v grafu cesta z vrcholu u do vrcholu v , pak $u.d < v.d$.
 - B Existuje-li v grafu cesta z vrcholu u do vrcholu v , pak $u.f > v.f$.
 - *C Žádné z ostatních tvrzení není obecně pravdivé.
 - D Existuje-li v grafu cesta z vrcholu u do vrcholu v , pak $u.f < v.f$.
 - E Existuje-li v grafu cesta z vrcholu u do vrcholu v , pak $u.d > v.d$.
-
- 2** 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 2, 9, 12, 5, 1, 7
 - *B 9, 2, 12, 1, 5, 7
 - C 2, 9, 5, 12, 1, 7
 - D 9, 2, 12, 5, 1, 7
 - E 9, 2, 5, 12, 1, 7
-
- 3** Kterou z následujících podmínek **nemusí** splňovat korektní červenočerný strom?
- A Kořen je černý
 - *B Hodnota v kořeni je nejmenší hodnotou z celého stromu.
 - C Každá cesta z kořene do listu obsahuje stejný počet černých vrcholů.
 - D Každý červený vrchol má dva černé syny.
 - E Každý vrchol je buď červený nebo černý.
-
- 4** Uvažujme libovolný binární vyhledávací strom (BVS) s n prvky. Do které složitostní třídy patří operace INSERT (vlození prvku do BVS)?
- A $\mathcal{O}(\sqrt{n})$
 - B $\mathcal{O}(\log n)$
 - C $\mathcal{O}(1)$
 - D $\mathcal{O}(1/n)$
 - *E $\mathcal{O}(n)$
-
- 5** Které z těchto tvrzení je pravdivé?
- A Vkládání prvků do hašovací tabulky má **v nejhorším případě** složitost $\mathcal{O}(1)$.
 - B Datová struktura zásobník pracuje na principu FIFO (first in, first out).
 - C Logaritmus roste asymptoticky stejně rychle jako druhá odmocnina.
 - *D Totálně korektní algoritmus zaručeně skončí na všech vstupech, které splňují vstupní podmínku algoritmu.
 - E Mezi binární vyhledávací stromy patří binární halda.

Programování

- 6** Které tvrzení je obecně platné v běžných OOP jazycích typu C++, Java, C#?
- A** Pokud třída B dědí od třídy A, pak objekty třídy B mají přístup ke všem atributům (členským proměnným) třídy A.
 - B** Pojmy třída a objekt jsou totožné, rozlišují se jen z historických důvodů.
 - C** Pokud třída B dědí od třídy A, pak objekty třídy A mají přístup ke všem atributům (členským proměnným) třídy B.
 - D** Statické metody tříd se liší od ostatních metod tím, že mají přístup ke statickým atributům třídy.
 - *E** Je-li použita pozdní vazba (volání virtuálních metod), pak se o tom, která metoda se skutečně zavolá, rozhoduje až za běhu; tedy nikoli v době překladu.
-

- 7** Uvažujme následující program.
- ```
function fun(integer n)
begin
 result = 1
 while n != 0
 n = n - 1
 result = result * 2
 end while
 return result
end
```
- Předpokládejme, že parametr  $n$  je vždy nezáporný. Jakou funkci počítá funkce `fun`? Tj., jaký je vztah mezi výslednou hodnotou proměnné `result` a počáteční hodnotou parametru  $n$ ?
- A**  $n$
  - B**  $2 * n$
  - C**  $n^2$
  - D** 1
  - \*E**  $2^n$
- 

- 8** Které z následujících tvrzení je **nepravdivé**?
- \*A** Rekurzivní volání funkcí je typicky implementováno pomocí fronty.
  - B** Líná vyhodnocovací strategie ve funkcionálním programování umožňuje pracovat s nekonečnými datovými strukturami.
  - C** Při volání referencí (odkazem) se změna parametru uvnitř funkce projeví změnou parametru (toho, se kterým byla funkce volána) i navenek.
  - D** Při volání hodnotou se změna parametru uvnitř funkce navenek nijak neprojeví.
  - E** V čistě funkcionálních jazycích nemají funkce žádné vedlejší efekty.
-

**9** Uvažujme následující program. Funkce `print` vypíše zadané číslo následované koncem řádku.

```
function foo(integer n)
begin
 if n <= 1 then
 return n

 return foo(n - 1) + foo(n - 2)
end

program main()
begin
 print foo(10)
end
Co vypíše tento program?
```

- A 10
- B 17
- C 13
- D Program bude cyklit a nikdy neskončí.
- \*E 55

**10** Která z následujících tří tvrzení I, II a III jsou pravdivá (v běžných jazycích typu C++, Java, C#). Vyberte možnost, která obsahuje **všechna pravdivá** tvrzení a žádná nepravdivá.

- I. Lokální proměnné funkcí jsou alokovány na haldě.
- II. Volání funkcí jsou implementována pomocí zásobníku.
- III. Je-li zachycena výjimka (v bloku `catch`), může být znovu vyvolána (použitím `throw`).

- A I, II
- B III
- C I, II, III
- D I, III
- \*E II, III

## Databázové systémy

**11** Aplikace pro elektronický obchod obsahuje neprázdnou relaci *produkt(id, název, množství, cena, kategorie)*, jejímž primárním klíčem je atribut *id*. Co vrací následující SQL dotaz?

```
SELECT kategorie, SUM(cena * množství) FROM produkt WHERE název LIKE 'LG %' GROUP BY kategorie;
```

- A Syntaktickou chybu, protože současné použití `WHERE` a `GROUP BY` se vylučuje.
- B Kategorii a celkovou hodnotu produktů pro každou kategorii obchodu.
- \*C Pro každou kategorii obchodu vrací její název a celkovou hodnotu produktů s názvem začínajícím 'LG '.
- D Kategorii a celkovou hodnotu produktů pro každou kategorii obchodu, jejíž název začíná 'LG '.
- E První kategorii (vzhledem k lexikografickému uspořádání) a celkovou cenu produktů začínající na 'LG '.

**12** Databázový spouštěč (trigger) je:

- A procedura, která se volá **automaticky vždy**, když je z databáze vymazán záznam, např. kvůli úspoře místa stlačením dat (COMPACT DATA).
- B technologie vzdáleného volání procedur v distribuovaných systémech a v databázových systémech se nepoužívají.
- C funkce, která je vždy volaná jednou za zadaný interval (typicky jednou denně), která obsluhuje sloupce s typem sériové datum (SERIAL DATE).
- D procedura, která se spouští na vyžádání uživatele, např. pro vysypání koše (PURGE RECYCLEBIN).
- \*E kód, který se zavolá, pokud nastane pro něj definovaná událost, např. vložení nového záznamu do tabulky.

**13** Mezi integritní omezení používané v databázových systémech **nepatří**:

- A doménové omezení
- B uživatelem definované omezení
- \*C omezení sekundárního klíče
- D referenční omezení
- E omezení zakazující nedefinované hodnoty

**14** Máme relaci  $r(A, B, C, D, E, F, G, H, I)$  a následující množinu funkčních závislostí:

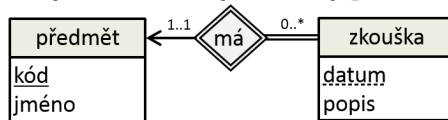
$A \rightarrow B, C$   
 $A, D \rightarrow E, F, G$   
 $E \rightarrow D$   
 $E \rightarrow I$   
 $G \rightarrow H$

Který z následujících rozkladů je bezztrátový, zachovává funkční závislosti a splňuje druhou normální formu (2NF)?

Pozn: relace  $r$  má dva kandidátní klíče  $\{A, D\}$  a  $\{A, E\}$ .

- A  $r_1(A, B, C, D, E)$  a  $r_2(E, F, G, H, I)$
- B  $r_1(A, D, E, F, G)$ ,  $r_2(A, B, C)$  a  $r_3(E, D, I, G, H)$ .
- C  $r_1(A, D, E, F, G, H, I)$  a  $r_2(A, B, C)$ .
- \*D  $r_1(A, D, E, F, G, H)$ ,  $r_2(A, B, C)$  a  $r_3(E, I)$ .
- E Relaci nelze převést do druhé normální formy.

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



- A předmět(kód, jméno, datum), zkouška(datum, popis)
- B předmět(kód, jméno), zkouška(datum, popis), má(kód, datum)
- \*C předmět(kód, jméno), zkouška(kód, datum, popis)
- D předmět(kód, jméno), zkouška(datum, popis)
- E předmět(kód, jméno), zkouška(datum, popis)

## Počítačové sítě

**16** TCP/IP síťový model je tvořen těmito vrstvami:

- A Vrstva fyzická, vrstva datového spoje, síťová vrstva, transportní vrstva.
- B Vrstva fyzická, vrstva datového spoje, síťová vrstva, transportní vrstva a aplikační vrstva.
- C Vrstva fyzická, vrstva datového spoje, síťová vrstva, transportní vrstva, relační vrstva a prezentační vrstva.
- D Vrstva přístupu k síti/médiu, síťová vrstva, transportní vrstva, relační vrstva a aplikační vrstva.
- \*E Vrstva přístupu k síti/médiu, síťová vrstva, transportní vrstva a aplikační vrstva.

**17** Spojované sítě (sítě s přepínáním okruhů) lze charakterizovat tímto tvrzením:

- A Jsou charakteristické sofistikovanými směrovacími mechanismy, nejpoužívanější protokoly jsou Border Gateway Protocol (BGP) a Intermediate System to Intermediate System (IS-IS).
- B Datové jednotky (pakety) jsou směrovány na vnitřních prvcích sítě dle cílové adresy uvedené v hlavě.
- \*C Mezi komunikujícími uzly je před začátkem přenosu ustaveno spojení (okruh), které je udržováno během celé komunikace, např. analogové telefonní sítě.
- D Jsou charakteristické sofistikovanými směrovacími mechanismy, nejpoužívanější protokoly jsou Multicast Open Shortest Path First (MOSPF) a Distance Vector Multicast Routing Protocol (DVMRP).
- E Okruhy mezi komunikujícími uzly vznikají pouze na úrovni lokálních sítí.

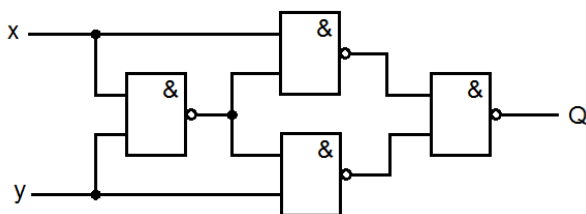
- 18** Stanovení norem/standardů popisujících nejrůznější akce, činnosti, formy či způsoby komunikace, atp. jsou realizovány:
- A** Výlučně de jure – standardy vypracované a schválené oficiálním mezinárodním nebo národním normalizačním orgánem.
  - B** Prostřednictvím zákonů dle země realizace.
  - C** Výlučně de facto – technická řešení, která se svým úspěchem na trhu prosadila do té míry, že jsou akceptována většinou výrobců jako příklad hodný následování.
  - \*D** De facto – technická řešení, která se svým úspěchem na trhu prosadila do té míry, že jsou akceptována většinou výrobců jako příklad hodný následování; a de jure – standardy vypracované a schválené oficiálním mezinárodním nebo národním normalizačním orgánem.
  - E** Formou komunitních orgánů a zveřejnění jejich rozhodnutí.

- 19** Datagram při cestě k cíli prochází různými sítěmi a ne všechny sítě mohou přenášet data stejné velikosti. Pokud je datagram větší než hodnota MTU (Maximum Transfer Unit) přiřazená k výstupní lince, přes kterou má být datagram směrován, dojde:
- A** k zahození datagramu bez dalších aktivit.
  - B** pro sítě s IPv6 k zahození datagramu a vygenerování ICMP zprávy "Packet too big" a pro sítě s IPv4 k fragmentaci datagramu, tj. rozdělení na menší části s nově vygenerovanými hlavičkami, ve kterých jsou údaje v polích "Total Length" a TTL údaje potřebné k rekonstrukci původního datagramu.
  - \*C** pro sítě s IPv6 k zahození datagramu a vygenerování ICMP zprávy "Packet too big" a pro sítě s IPv4 k fragmentaci datagramu, tj. rozdělení na menší části s nově vygenerovanými hlavičkami, ve kterých jsou v polích Identification, Flags a Offset údaje potřebné k rekonstrukci původního datagramu.
  - D** pro sítě s IPv6 k fragmentaci datagramu, tj. rozdělení na menší části s nově vygenerovanými hlavičkami, ve kterých jsou údaje Identification, Flags a Offset potřebné k rekonstrukci původního datagramu.
  - E** pro sítě s IPv4 k zahození datagramu a vygenerování ICMP zprávy "Packet too big".

- 20** Kolik různých sítí a rozhraní lze adresovat IP adresami třídy A (tj. maska 255.0.0.0)?
- A**  $2^5$  sítí a 256 rozhraní.
  - \*B**  $2^7$  sítí a  $2^{24}$  rozhraní.
  - C** Nelze určit, protože se jedná o multicastové adresy.
  - D**  $2^{24}$  sítí a  $2^7$  rozhraní.
  - E** Nelze určit, protože se jedná se o rezervovaný prostor, který se nevyužívá.

## Počítačové systémy

- 21** Je dán následující logický obvod:



Pro hodnotu Q na jeho výstupu platí:

- \*A**  $Q = x \text{ XOR } y$
- B**  $Q = x \text{ NOXOR } y$ , neboli  $Q = \text{NOT}(x \text{ XOR } y)$  nebo také  $Q = x \text{ XNOR } y$
- C**  $Q = x \text{ NOR } y$
- D** žádná z uvedených variant
- E**  $Q = x \text{ OR } y$

- 22** Celočíselný datový typ se znaménkem, jehož hodnoty jsou zobrazeny ve dvojkovém doplňkovém kódu a v operační paměti uloženy na 8 bitech, má rozsah:
- A -127 až 127
  - B -128 až 128
  - \*C -128 až 127
  - D -255 až 256
  - E -127 až 128
- 

- 23** Které z následujících tvrzení **neplatí** pro symetrický multiprocessing (SMP)?
- A Kterýkoliv proces může běžet na kterémkoliv procesoru.
  - B Celý systém je řízen jedním operačním systémem.
  - C Je podporován většinou soudobých operačních systémů.
  - D Zároveň může běžet více procesů, aniž by došlo ke snížení výkonu.
  - \*E Výpočetní výkon celého systému roste lineárně s rostoucím počtem procesorů.
- 

- 24** Která z následujících tvrzení (týkajících se cache pamětí) jsou u soudobých procesorů pravdivá? Vyberte možnost, která obsahuje **všechna pravdivá** tvrzení (a žádná nepravdivá).
- I. L1 cache paměť je rozdělena na část pro instrukce a část pro data.
  - II. Cache paměti procesorů jsou realizovány jako paměti typu SDRAM.
  - III. Cache paměti procesorů jsou realizovány jako paměti typu SRAM.
  - IV. Každé jádro procesoru obsahuje svou vlastní L3 cache paměť.
  - V. Cache paměti procesorů pracují jako asociativní paměti.
- A I, II, V
  - B III, IV
  - C II, IV, V
  - D II, IV
  - \*E I, III, V
- 

- 25** Které z následujících tvrzení **není** pravdivé pro Direct Memory Access (DMA)?
- \*A Použití DMA nemůže vést k problémům s paměťovou koherencí.
  - B DMA nahrazuje programovaný vstup/výstup při velkých přesunech dat.
  - C Při použití DMA dochází k výměně bloků mezi vstupním/výstupním zařízením a fyzickým adresovým prostorem.
  - D Při přenosu prostřednictvím DMA se obchází procesor.
  - E Systém využívající DMA požaduje speciální DMA řadič.
- 

## Softwarové inženýrství

---

- 26** Které z uvedených nefunkčních atributů patří mezi základní požadavky na kvalitu software (je třeba jim věnovat pozornost u největšího počtu aplikací)?
- A udržovatelnost, spolehlivost, testovatelnost, přenositelnost
  - \*B výkonnost, udržovatelnost, spolehlivost, bezpečnost
  - C spolehlivost, bezpečnost, přenositelnost, důvěrnost
  - D výkonnost, udržovatelnost, testovatelnost, přenositelnost
  - E bezpečnost, výkonnost, testovatelnost, důvěrnost
-



**27** Přiřadte následujícím popisům softwarových architektur správná pojmenování:

**Architektura X** označuje principy členění aplikací do vzájemně spolupracujících vrstev, kdy sousedící vrstvy spolupracují přes definovaná rozhraní, a mohou proto být jejich implementace měněny, aniž by to mělo dopad na funkčnost celé aplikace.

**Architektura Y** označuje sadu principů pro skládání složitých aplikací a jiných systémů ze skupiny na sobě nezávislých komponent poskytujících služby.

**Architektura Z** rozděluje datový/aplikační model aplikace, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent tak, že modifikace kterékoli z nich má jen minimální vliv na ostatní.

- A X = vícevrstvá architektura, Y = klient-server architektura, Z = model-view-controller architektura
- B X = vícevrstvá architektura, Y = servisně orientovaná architektura, Z = komponentová architektura
- C X = pipe and filter architektura, Y = servisně orientovaná architektura, Z = vícevrstvá architektura
- D X = pipe and filter architektura, Y = komponentová architektura, Z = vícevrstvá architektura
- \*E X = vícevrstvá architektura, Y = servisně orientovaná architektura, Z = model-view-controller architektura

**28** Uvažujme následující tvrzení o automatizovaném a manuálním testování. Vyberte možnost, která obsahuje **všechna pravdivá** tvrzení a žádná nepravdivá.

I. Pro nově vyvíjenou funkcionalitu, pro niž zatím není zcela vyjasněna cílová podoba, je vhodnější zvolit manuální testování.

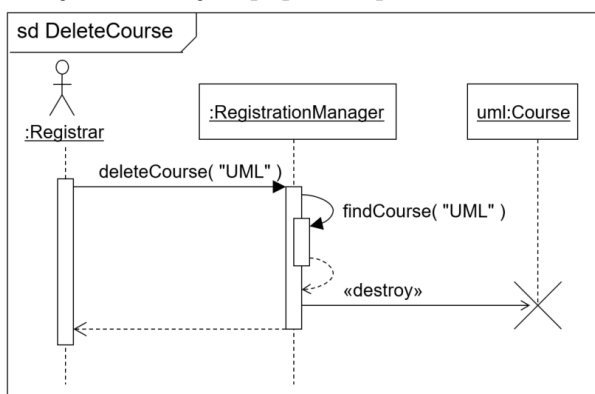
II. Regresní testování je vhodnější provádět automatizovaně (než manuálně).

III. Automatizované testování by mělo být použito v maximální míře, protože vždy vede k vyšší kvalitě systému za nižší cenu.

IV. Výkonnostní a zátěžové testy je vhodnější provádět manuálně.

- A II, III, IV
- B I, IV
- C I
- D II, III
- \*E I, II

**29** Který z uvedených popisů odpovídá UML sekvenčnímu diagramu na obrázku?



- A V rámci metody deleteCourse() třída RegistrationManager sama na sobě zavolá statickou metodu findCourse(), která vyvolá smazání objektu uml typu Course.
- B V rámci modelovaného scénáře se po sobě (sekvenčně) provedou tři metody: deleteCourse(), findCourse(), destroy().
- C V rámci modelovaného scénáře se po sobě (sekvenčně) provedou dvě metody: deleteCourse(), findCourse(). Tím se stane nepotřebným objekt uml třídy Course, který se smaže.
- D V rámci metody deleteCourse() instance třídy RegistrationManager sama na sobě zavolá metodu findCourse(), jejíž součástí je smazání objektu Course.
- \*E V rámci metody deleteCourse() instance třídy RegistrationManager sama na sobě zavolá metodu findCourse() a následně vyvolá smazání objektu uml typu Course.

**30** Které z následujících tvrzení **není** pravdivé?

- \*A** Strukturovaná analýza na rozdíl od objektové analýzy nabízí širší portfolio diagramů pro zakreslení jednotlivých pohledů na systém (jako například stavový diagram či diagram komponent).
  - B** Objektová analýza nahlíží na systém jako na kolekci objektů, které charakterizuje jejich typ (třída), stav (atributy) a chování (operace/metody).
  - C** Strukturovaná analýza člení systémy na malé a vhodně definované aktivity, pro něž definuje pořadí a předávaná data.
  - D** Objektová analýza na rozdíl od strukturované analýzy kombinuje data i funkce uvnitř jednotlivých klíčových elementů modelu, kterými jsou objekty/třídy.
  - E** Strukturovaná analýza využívá hierarchickou dekompozici procesů odehrávajících se v systému, které modeluje pomocí diagramu datových toků.
-