

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

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

| | |
|--|-------|
| Počet podaných přihlášek | 152 |
| Počet přihlášených uchazečů | 139 |
| Počet uchazečů, kteří splnili podmínky přijetí | 81 |
| Počet uchazečů, kteří nesplnili podmínky přijetí | 58 |
| Počet uchazečů přijatých ke studiu, bez uvedení počtu uchazečů přijatých ke studiu až na základě výsledku přezkoumání původního rozhodnutí | 81 |
| Počet uchazečů přijatých celkem | 81 |
| Percentil pro přijetí | 15,00 |

Základní statistické charakteristiky

| | Informatika | Matematika | Celkem | |
|---|-------------|------------|--------|-------|
| Počet otázek | 30 | 25 | 55 | |
| Počet uchazečů, kteří se zúčastnili přijímací zkoušky | 120 | 120 | 87 | |
| Nejlepší možný výsledek | 30.00 | 25.00 | 55.00 | |
| Nejlepší skutečně dosažený výsledek | 24.25 | 21.25 | 40.25 | |
| Průměrný výsledek | 13.79 | 13.29 | 26.95 | |
| Medián | 14.00 | 13.50 | 27.25 | |
| Směrodatná odchylka | 5.13 | 3.87 | 7.88 | |
| | Percentil | | | |
| Decilové hranice výsledku * | 10 | 7.25 | 8.75 | 18.00 |
| | 20 | 9.25 | 10.25 | 21.50 |
| | 30 | 11.50 | 12.25 | 23.50 |
| | 40 | 12.50 | 13.00 | 26.25 |
| | 50 | 14.00 | 13.50 | 27.25 |
| | 60 | 15.25 | 14.50 | 29.25 |
| | 70 | 16.75 | 15.25 | 31.25 |
| | 80 | 18.75 | 16.75 | 33.50 |
| | 90 | 20.50 | 17.50 | 37.50 |

* 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 |
|------------------------------------|-----------------|--------------|
| | | 2 |

Počítačové sítě

- 1** Pro mezidoménové směrování
- ***A** se využívají Path Vector směrovací protokoly, které vznikly úpravou Distance Vector směrovacích protokolů a podporují rozhodování podle politik
 - B** se v dnešních sítích kvůli problémům s cykly využívají Distance Vector směrovací protokoly
 - C** se využívají Path Vector směrovací protokoly s metrikou počet hopů bez podpory směrovacích politik
 - D** se v dnešních sítích kvůli problémům s cykly využívají Link State směrovací protokoly
 - E** se v dnešních sítích pro řešení problémů se složitými topologiemi využívá kombinace Link State a Distance Vector
-

- 2** Multiplexing - technika sdílení dostupné přenosové kapacity:
- A** pro digitální i analogové signály využívá časový multiplexing (TDM)
 - ***B** pro digitální signály využívá časový multiplexing (TDM) a pro analogové signály využívá vlnový i frekvenční multiplexing (FDM a WDM)
 - C** využívá vlnový a frekvenční multiplexing (FDM a WDM) pouze pro digitální signály
 - D** pro digitální i analogové signály využívá vlnový i frekvenční multiplexing (FDM a WDM)
 - E** se již nevyužívá, protože současné sítě mají kapacit nadbytek
-

- 3** Metoda řízení přístupu k přenosovému médiu CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance)
- A** zabrání kolizím vysílání pouze aplikačních dat
 - B** sníží pravděpodobnost kolizí vysílání pouze protokolárních dat a nikoli aplikačních dat
 - C** zabrání kolizím vysílání pouze protokolárních příkazů
 - ***D** sníží pravděpodobnost kolizí vysílání
 - E** zabrání kolizím vysílání
-

- 4** Přenos dat v rámci protokolu UDP je realizován tak, že
- A** aplikace předává bloky dat, které UDP opatřuje hlavičkou, čísluje a předává síťovému protokolu
 - B** aplikace předává bloky dat, které UDP opatřuje hlavičkou, čísluje a jejich čísla odvozuje od náhodně stanoveného začátku, ke kterému přičítá jedničku za každý připravený blok
 - C** aplikace předává proud bytů, které UDP segmentuje, čísluje a předává síťovému protokolu
 - ***D** aplikace předává bloky dat, které UDP opatřuje hlavičkou a předává síťovému protokolu
 - E** aplikace předává proud bytů, které UDP segmentuje, čísluje a jejich čísla odvozuje od náhodně stanoveného začátku, ke kterému přičítá pořadové číslo bytu, kterým paket začíná
-

- 5** Jmenný prostor internetu (Domain Name Space)
- A** je plochý a za jeho činnost odpovídá centrální jmenná autorita
 - B** je řešen za pomoci tzv. host souborů v lokálních sítích
 - C** má hierarchické uspořádání s maximálním počtem úrovní 3
 - ***D** má hierarchické uspořádání s maximálním počtem úrovní 128 a slouží pro překlad doménových jmen na IP adresy a zpět
 - E** má hierarchické uspořádání a slouží pro překlad doménových jmen na fyzické adresy (MAC) a zpět
-

Programování

- 6** Rozhodněte, které z uvedených tvrzení je v běžných OOP jazycích (C++, Java, C#) obecně platné:
- A** Pokud třída B dědí od třídy A, pak má třída A přístup ke všem atributům třídy B, které byly deklarovány jako public nebo protected.
 - B** Žádné z ostatních uvedených tvrzení není platné.
 - C** Pokud je metoda třídy statická, znamená to, že má na rozdíl od ostatních metod přístup ke statickým atributům třídy.
 - D** Rozdíl mezi třídou a objektem je v tom, že objekty se alokují na haldě, zatímco třídy se alokují na zásobníku.
 - *E** Při pozdní vazbě (late binding; volání virtuální metody) se o tom, která metoda se přesně zavolá, rozhoduje až za běhu.
-

- 7** Které z následujících tvrzení **neplatí**?
- *A** Příkladem jazyka logického programování je Haskell.
 - B** Normální vyhodnocovací strategie ve funkcionálním programování umožňuje pracovat s nekonečnými seznamy.
 - C** Při volání odkazem se změna hodnoty parametru uvnitř funkce projeví i navenek.
 - D** Do paradigmatu deklarativního programování lze zahrnout funkcionální i logické programování.
 - E** Je-li funkce bez vedlejších efektů zavolána se stejnými parametry, dá vždy stejný výsledek.
-

- 8** Rozhodněte, která z uvedených tvrzení I, II a III jsou pravdivá (pro běžné jazyky typu C++, Java, C#). Vyberte takovou možnost, která vyjadřuje právě všechna pravdivá tvrzení.

I. Volání funkcí je implementováno pomocí zásobníku, na který se ukládá informace o tom, kam se po skončení funkce vrátit.

II. Parametry funkcí se funkcím předávají pomocí haldy.

III. Je-li výjimka zachycena (v bloku catch), je možno ji opětovně vyhodit (pomocí throw).

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

- 9**
- ```
read(n)
sum = 0
XXX {
 sum = sum + (n mod 10)
 n = n div 10
}
print sum
```

Nechť mod představuje zbytek po dělení a div je celočíselné dělení. Předpokládejme, že na vstupu výše uvedeného pseudokódu je přirozené číslo  $n$ . Co musíme doplnit na místo označené "XXX", aby výše uvedený pseudokód dal na výstup ciferný součet čísla  $n$ ?

- \*A** Žádná z ostatních uvedených možností není správná.
  - B** while sum < 0
  - C** while n > sum
  - D** while n < sum
  - E** while n == 0
-

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

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

Předpokládejte, že funkce print vždy ukončuje výstup koncem řádku. Kolik řádků bude mít výstup tohoto programu (tj. kolikrát se provede funkce print)?

- A Uvedený kód bude cyklit a nezastaví.
- B 9
- \*C 37
- D 11
- E Žádná z ostatních odpovědí není správná.

## Počítačové systémy

11 Mikrojádro operačního systému typicky **nezabezpečuje**:

- A správu paměti
- B správu procesorů
- \*C správu souborového systému
- D správu přerušení
- E komunikaci mezi procesy předáváním zpráv

12 Pomocí kterého z následujících logických obvodů lze realizovat převod paralelního přenosu informací na přenos sériový?

- A komparátoru
- B kodéru
- C demultiplexoru
- \*D multiplexoru
- E dekodéru

13 Translation Look-aside Buffer (TLB) slouží:

- A k ukládání předvybraných instrukcí
- B ke zvýšení efektivity procesu segmentace
- C jako úložiště instrukcí, které budou vykonávány mimo pořadí
- \*D ke zvýšení efektivity procesu stránkování
- E jako pomocná paměť pro implementaci předvídání větvení

14 Způsob uložení čísel v operační paměti počítače, při kterém se na paměťové místo s nejnižší adresou uloží nejméně významný byte a za něj se ukládají ostatní byty až po nejvíce významný byte se nazývá:

- A big endian
- B reversed endian
- \*C little endian
- D straight endian
- E small endian

- 15** Které číslo v šestnáctkové soustavě je ekvivalentem čísla vyjádřeného v osmičkové soustavě jako 23456?
- \*A 272E
  - B 272F
  - C 172E
  - D 172F
  - E 271E

---

## Databázové systémy

---

- 16** Mějme jednoduchou evidenci zaměstnanců v relaci *zamestnanec(id, jmeno, plat, id\_nadrizeny)*. Atribut *id* je primární klíč relace, atribut *id\_nadrizeny* je cizí klíč, který se odkazuje na atribut *id* relace *zamestnanec*. Uvažujte následující dotaz v relační algebře:

$$\pi_{a.jmeno, zamestnanec.jmeno}(\sigma_{a.id\_nadrizeny=zamestnanec.id}(\rho_a(zamestnanec)))$$

Které z následujících tvrzení je správné?

- A Dotaz je syntakticky správně, ale nevrací smysluplný výsledek.
- B Dotaz je syntakticky správně, ale při jeho vykonávání dojde k zacyklení a bude vrácena chyba příliš dlouho trvajícího výpočtu.
- \*C Dotaz je syntakticky špatně.
- D Dotaz je syntakticky správně, ale vždy vrátí prázdný výsledek.
- E Dotaz je syntakticky správně a vrací smysluplný výsledek.

- 
- 17** Jazyk SQL:

- A vychází z jazyka XML
- B je založen na evolučních programovacích metodách
- C patří mezi imperativní jazyky
- D funguje spolehlivě, jen pokud vždy zadáme superklíč
- \*E patří mezi deklarativní jazyky

- 
- 18** Navrhněte E-R model pro systém, který eviduje bankovní účty a zákazníky banky. Chceme, aby systém splňoval následující požadavky:

- \* Zákazník má dispoziční práva k libovolnému počtu účtů.
- \* S účtem může disponovat libovolný počet zákazníků.
- \* U zákazníka evidujeme rodné číslo, jméno a bydliště.
- \* U účtu evidujeme číslo účtu a aktuální zůstatek.

Výsledný ERD obsahuje právě:

- A Tři entitní množiny: účet, zákazník a dispoziční právo; tři binární vztahy účet-zákazník, zákazník-dispoziční právo a dispoziční právo-účet, všechny s násobností 1-1.
  - \*B Dvě entitní množiny: účet a zákazník; jeden binární vztah účet-zákazník s násobností m-n.
  - C Tři entitní množiny: účet, zákazník a dispoziční právo; jeden ternární vztah účet-zákazník-dispoziční právo s násobností 1-1-1.
  - D Tři entitní množiny: účet, zákazník a dispoziční právo; žádný vztah.
  - E Dvě entitní množiny: účet a zákazník; binární vztah účet-zákazník s násobností 1-1.
-

**19** Uvažujte následující relace z databáze autopůjčovny:

*auto*(SPZ, znacka, rok\_vyroby),  
*zapujceni*(ID\_pujcky, SPZ, datum\_od, datum\_do).

Atribut *SPZ* relace *zapujceni* je cizí klíč, který odkazuje atribut *SPZ* v relaci *auto*. Který z následujících výrazů zjistí datum, kdy bylo naposledy vypůjčeno auto s *SPZ* 'BZS 1447'?

- \*A SELECT MAX(datum\_od) FROM auto NATURAL INNER JOIN zapujceni WHERE SPZ = 'BZS 1447'
- B SELECT MIN(datum\_od) FROM auto INNER JOIN zapujceni USING (SPZ) WHERE SPZ = 'BZS 1447'
- C SELECT COUNT(\*) FROM zapujceni WHERE SPZ = 'BZS 1447'
- D SELECT MIN(datum\_od) FROM zapujceni WHERE SPZ = 'BZS 1447'
- E SELECT DISTINCT datum\_od FROM zapujceni WHERE SPZ = 'BZS 1447'

**20** Jestliže návrh relací v dané databázi splňuje druhou normální formu (2NF), pak **nesmí** platit:

- A Existuje atribut v některé relaci, který je dále nedělitelný.
- \*B Existuje atribut v některé relaci, který závisí na části primárního klíče dané relace.
- C Všechny relace mají definován primární klíč.
- D Existuje atribut v některé relaci, který je součástí více superklíčů dané relace.
- E Existuje atribut v některé relaci, který je tranzitivně závislý na primárním klíči jiné relace.

## Algoritmizace a datové struktury

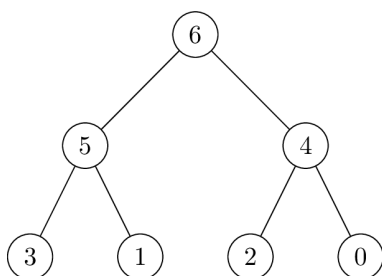
**21** Mějme binární vyhledávací strom (BVS), který obsahuje  $n$  prvků. Máme k dispozici pouze ukazatel na kořenový prvek stromu. Které z uvedených tvrzení platí?

- \*A Nalezení zadaného prvku v BVS má časovou složitost v  $O(n)$ .
- B Nalezení minimálního prvku v BVS má časovou složitost v  $O(1)$ .
- C Nalezení minimálního prvku v BVS má časovou složitost v  $O(\log n)$ .
- D Odebrání minimálního prvku v BVS má časovou složitost v  $O(1)$ .
- E Odebrání minimálního prvku v BVS má časovou složitost v  $O(\log n)$ .

**22** Uvažujme orientovaný graf a dva jeho vrcholy  $u$  a  $v$  takové, že z  $u$  do  $v$  vede cesta, zatímco z  $v$  do  $u$  cesta nevede. Které z následujících tvrzení je vždy pravdivé?

- A Při prohledávání grafu do šířky bude vrchol  $v$  poprvé navštíven dříve než vrchol  $u$ .
- B Při prohledávání grafu do hloubky bude vrchol  $v$  poprvé navštíven dříve než vrchol  $u$ .
- C Při prohledávání grafu do šířky bude vrchol  $u$  poprvé navštíven dříve než vrchol  $v$ .
- \*D Ani jedno ze zbylých tvrzení není vždy pravdivé.
- E Při prohledávání grafu do hloubky bude vrchol  $u$  poprvé navštíven dříve než vrchol  $v$ .

**23**



V jakém pořadí bychom museli vkládat prvky do prázdné (maximové) binární haldy, aby výsledkem byla binární halda uvedená na obrázku?

- A 6, 4, 1, 5, 3, 2, 0
- B 5, 4, 2, 6, 1, 3, 0
- \*C 3, 6, 4, 5, 1, 2, 0
- D 4, 6, 3, 5, 0, 2, 1
- E 4, 5, 1, 6, 2, 3, 0

**24** Které z následujících tvrzení je pravdivé?

- A Mezi binární vyhledávací stromy patří AVL stromy a binární haldy.
- B Vstupní podmínka algoritmu popisuje právě ty vstupy, pro které platí invariant algoritmu.
- \*C Vložení prvku do binární haldy má časovou složitost v  $O(n)$ .
- D Datová struktura binární vyhledávací strom je vhodná pro procházení grafů.
- E Funkce  $n \cdot \log(n)$  roste asymptoticky stejně rychle jako funkce  $\log(n)$ .

**25** Předpokládejme hašovací tabulku (hashtable) s řešením kolizí pomocí lineárního sondování (linear probing). Velikost hašovací tabulky je 7, hašovací funkce je  $h(x) = (3 \cdot x + 5) \bmod 7$ . Na začátku výpočtu je tabulka prázdná. V jakém pořadí bychom museli do tabulky vkládat prvky 1, 2, 5, 7, 9, 12, aby byl na konci její obsah takovýto: 12, 5, 1, prázdné pole, 2, 9, 7?

- A 12, 5, 1, 2, 9, 7
- B 2, 9, 7, 5, 12, 1
- \*C 2, 9, 7, 12, 5, 1
- D 9, 2, 7, 12, 5, 1
- E 12, 1, 5, 2, 9, 7

## Softwarové inženýrství

**26** Jako softwarový inženýr máte zvolit vhodnou techniku testování pro ověření, že spojení již otestovaných částí (dílkých komponent) systému nevnese do systému chyby. Která z uvedených technik je za tímto účelem nejvhodnější?

- A A/B testování
- B Testování jednotek
- C Regresní testování
- \*D Integrovaní testování
- E Alfa testování

**27** Který z následujících výčtů nejlépe charakterizuje změny v modelu tříd SW systému (za použití UML diagramu tříd) při přechodu od analýzy k návrhu?

- A nárůst počtu tříd, odebrání nepotřebných atributů, přidání implementačních detailů
- B nárůst velikosti tříd (počtu metod a atributů), dopřesnění atributů, přidání implementačních detailů
- C nárůst velikosti tříd (počtu metod a atributů), odebrání nepotřebných atributů, rozpracování stávajících implementačních detailů
- \*D nárůst počtu tříd, dopřesnění atributů, přidání implementačních detailů
- E odebrání nepotřebných tříd, dopřesnění atributů, rozpracování stávajících implementačních detailů

**28** Jako softwarový inženýr máte zvolit vhodnou notaci pro popis persistentních dat v procedurálně programovaném systému. Dáte přednost UML diagramu tříd nebo entitně relačnímu diagramu (ERD) a proč?

- A Zvolím ERD, protože UML diagram tříd nelze pro popis struktury persistentních dat použít.
- B Zvolím UML diagram tříd, protože ERD je pro popis struktury persistentních dat nevhodný.
- \*C Zvolím ERD, protože se lépe hodí pro popis dat uložených v relační databázi, která bude v tomto případě vhodnější než objektová databáze.
- D Zvolím UML diagram tříd, protože jím lze modelovat dědičnost, která bude v této situaci potřebná.
- E Zvolím UML diagram tříd, protože nabízí silnější prostředky pro popis primárních a cizích klíčů, což je u persistentních dat podstatné.

**29** Která z následujících charakteristik nepatří mezi principy agilního vývoje?

- A Reakce na změnu před dodržováním plánu
- \*B Návrh architektury před nestabilitou struktury kódu
- C Spolupráce se zákazníkem před sjednáváním smluv
- D Fungující software před obsáhlou dokumentací
- E Lidé a jejich spolupráce před procesy a nástroji



- 30** Který z následujících požadavků na webovou aplikaci je nefunkčním požadavkem?
- A** Po kliknutí na tlačítko "Stáhnout" se automaticky zahájí stahování do výchozího adresáře, či chybové hlášení, pokud takový adresář není nastaven.
  - B** Po kliknutí na tlačítko "Stáhnout" se zobrazí dialogové okno s dotazem, zda se má skutečně zahájit stahování.
  - \*C** Po kliknutí na tlačítko "Stáhnout" se automaticky zahájí stahování (či zobrazí chybová hláška) nejpozději do 2 sekund.
  - D** Po najetí na tlačítko "Stáhnout" se v "mouse over" nápovědě zobrazí informace o funkci tlačítka.
  - E** Po najetí na tlačítko "Stáhnout" se barva tlačítka změní ze zelené na červenou.
-