

**Zadání a řešení testu z informatiky a zpráva
o výsledcích přijímacího řízení do magisterského
navazujícího studia od jara 2015**

**Zpráva o výsledcích přijímacího řízení
do magisterského navazujícího studia od jara 2015**

Studium v českém jazyce

Počet podaných přihlášek	173
Počet přihlášených uchazečů	162
Počet uchazečů, kteří splnili podmínky přijetí	93
Počet uchazečů, kteří nesplnili podmínky přijetí	69
Počet uchazečů přijatých ke studiu, bez uvedení počtu uchazečů přijatých ke studiu až na základě výsledku přezkoumání původního rozhodnutí	93
Počet uchazečů přijatých celkem	93
Percentil pro přijetí	9,00 (odpovídá celkem alespoň 19 bodům)

Základní statistické charakteristiky

	Informatika	Matematika	Celkem	
Počet otázek	30	25	55	
Počet uchazečů, kteří se zúčastnili přijímací zkoušky	107	107	107	
Nejlepší možný výsledek	30.00	25.00	55.00	
Nejlepší skutečně dosažený výsledek	28.75	25	52.5	
Průměrný výsledek	17.03	16.24	33.27	
Medián	17.75	17.5	35.0	
Směrodatná odchylka	5.82	5.1	9.51	
	Percentil			
Decilové hranice výsledku *	10	8.9	9.3	20.15
	20	12.1	12.8	26.35
	30	15.15	14.5	29.95
	40	16.35	16.0	32.95
	50	17.75	17.5	35.0
	60	19.4	18.4	36.65
	70	20.25	19.25	39.0
	80	22.2	20.25	40.75
	90	23.85	21.9	43.6

* 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
		14

Algoritmizace a datové struktury

- 1** Co je platný argument pro použití hašovacích funkcí v databázích?
- A Umožňují uložit velké množství položek s následným vyhledáním s časovou složitostí $O(\log(n))$.
 - B Zajišťují automatické odstranění duplicitních položek.
 - *C Umožňují vkládat položky s časovou složitostí $O(1)$ (pokud nedojde ke kolizi).
 - D Hašovací funkce se v databázích typicky nepoužívají.
 - E Poskytují neztrátovou kompresi vkládaných položek s pamětovou složitostí $O(1)$.
-

- 2** Předpokládejme existenci oboustranně spojovaného seznamu prvků (double-linked list), u kterého máme uložený ukazatel pouze na první prvek. Délka seznamu je n . Které z uvedených tvrzení platí?
- A Časová složitost odstranění již nalezeného prvku ze seznamu je $\theta(n)$.
 - *B Časová složitost získání posledního prvku v seznamu je $\theta(n)$.
 - C Časová složitost nalezení zadaného prvku je $\theta(\log(n))$.
 - D Časová složitost vložení nového prvku za již nalezený existující prvek je $\theta(n)$.
 - E Paměťová režie pro realizaci spojovaného seznamu je stejná jako pro pole prvků uložených v paměti kontinuálně za sebou.
-

- 3** Pro datovou strukturu známou jako prioritní fronta platí:
- A naposledy vložený prvek bude při vyčítání vždy odebrán jako první
 - B naposledy vložený prvek bude při vyčítání vždy odebrán jako poslední
 - *C prvek s nejvyšší přiřazenou prioritou bude při odebírání získán jako první
 - D prvky musí být do struktury vkládány seřazené dle priority
 - E prvky ve struktuře jsou vždy řazeny dle pořadí vkládání
-

- 4** Pro datovou strukturu známou jako AVL strom (samovyvažující se binární vyhledávací strom) platí:
- A obsahuje nejvýše jeden cyklus, kořen, uzly a listy
 - B jeho výška je vždy lineární vzhledem k počtu uzlů
 - C má-li $(n-1)$ uzlů, pak obsahuje právě n hran
 - D operace vyhledávání v AVL stromu s n uzly má časovou složitost $O(\log(\log(n)))$
 - *E operace vložení nového uzlu do AVL stromu má složitost $O(\log(n))$
-

- 5** Kterou z uvedených datových struktur typicky využívá implementace vstupně/výstupních proudů (I/O) ve standardních knihovnách běžných programovacích jazyků? Příkladem takového proudu je např. standardní vstup používaný pro přenos znaků mezi klávesnicí a programem nebo standardní výstup pro přenos mezi programem a obrazovkou.
- A zřetězený seznam
 - B prioritní fronta
 - *C fronta
 - D zásobník
 - E hašovací tabulka
-

Počítačové systémy

- 6** Které číslo ve dvojkové soustavě je ekvivalentem čísla vyjádřeného v šestnáctkové (hexadecimální) soustavě jako A96B?
- *A 1010 1001 0110 1011
 - B 1011 0110 1001 1010
 - C 1101 0110 1001 0101
 - D 1011 1001 0110 1010
 - E 43371
-

- 7** Mezi prostředky IPC (inter-process communication) v OS patří:
- *A semafor
 - B dálnice (highway)
 - C race condition
 - D uváznutí (deadlock)
 - E výpadek stránky (page fault)
-

- 8** Jaký je maximální počet procesů, které se zároveň mohou nacházet v kritické sekci:
- A 0
 - *B 1
 - C 2
 - D 4
 - E 8
-

- 9** Příkladem blokového zařízení v operačním systému je:
- *A pevný disk
 - B myš
 - C terminál
 - D klávesnice
 - E konzole
-

- 10** Instrukce assembleru "JZ" a "JNZ" u procesorů rodiny Intel x86 provádí:
- *A podmíněný skok
 - B nepodmíněný skok
 - C naplnění registru Z
 - D návrat z podprogramu
 - E vstup, resp. výstup z V/V brány
-

Programování

11 integer Function foo(integer x, integer y, integer& z)

```
begin
  x = y
  y = z
  z = x
  return x + y + z
```

end

```
program main()
```

```
begin
```

```
  integer a = 1
```

```
  integer b = 2
```

```
  integer c = 3
```

```
  b = foo(a, b, c)
```

```
  c = foo(a, b, c)
```

```
  print "a = ", a, ", b = ", b, ", c = ", c
```

```
end
```

Předpokládejte, že funkce foo je volána hodnotou argumentů pro proměnné x a y a referencí pro proměnnou z. Jaký bude výstup programu po vytištění hodnot a, b, c?

- *A a = 1, b = 7, c = 16
 - B a = 1, b = 7, c = 10
 - C a = 1, b = 3, c = 2
 - D a = 1, b = 3, c = 16
 - E a = 6, b = 3, c = 10
-

12 // PSEUDOKÓD 1

```
res = 1
for i = 1 to n {
  res = res * i
}
print res
```

// PSEUDOKÓD 2

```
res = 1
i = n
do {
  res = res * i
  <MODIFIKACE>
} while ( <PODMÍNKA> )
print res
```

Pro výše uvedené pseudokódy platí, že všechny proměnné jsou typu integer (celé číslo se znaménkem) a $n \geq 1$. Která z uvedených možností může být použita pro nahrazení částí <MODIFIKACE> a <PODMÍNKA> tak, aby oba pseudokódy vypsaly stejnou hodnotu res?

- A <MODIFIKACE> nahradit za $i = i + 1$; <PODMÍNKA> nahradit za $i \geq 1$
 - B <MODIFIKACE> nahradit za $i = i + 1$; <PODMÍNKA> nahradit za $i > 1$
 - C <MODIFIKACE> nahradit za $i = i * i$; <PODMÍNKA> nahradit za $i > 0$
 - *D <MODIFIKACE> nahradit za $i = i - 1$; <PODMÍNKA> nahradit za $i \geq 2$
 - E <MODIFIKACE> nahradit za $i = i - 1$; <PODMÍNKA> nahradit za $i < n$
-

- 13** Rozhodněte, která z tvrzení I., II. a III. jsou pravdivá (pro běžné jazyky typu C++, Java, C#). Následně vyberte takovou možnost, která obsahuje všechna nalezená pravdivá tvrzení a neobsahuje žádná nepravdivá tvrzení.
- I. Lokální proměnné funkce jsou uloženy na zásobníku.
 - II. Pokud je paměť dynamicky alokována pomocí operátoru new, je jí přiřazeno umístění na haldě.
 - III. Lokální proměnná na zásobníku může obsahovat ukazatel na dynamicky alokovanou paměť na haldě.
- A I.
 - B II.
 - C I. a II.
 - *D I. a II. a III.
 - E II. a III.

- 14** Rozhodněte, které z uvedených tvrzení **není** v běžných OOP jazycích (C++, Java, C#) platné:
- A Blok, ve kterém se provádí zachytávání výjimky, je obalen klauzulí try {}.
 - B Pokud není výjimka obsloužena v aktuální funkci, tak se propaguje v zásobníku volání o úroveň výš do volající funkce.
 - *C Po obsloužení výjimky se pokračuje v kódu na řádku následujícím po řádku, ve kterém došlo k vyvolání výjimky.
 - D Standardní knihovna obsahuje několik předdefinovaných tříd výjimek, programátor může dodefinovat své další.
 - E Nezachycená výjimka způsobí ukončení programu.

- 15** Rozhodněte, které z uvedených tvrzení **není** v běžných OOP jazycích (C++, Java, C#) platné:
- A Předpokládejme, že třída Y je potomek třídy X. Objekt typu X nelze přetypovat na objekt typu Y.
 - B Předpokládejme, že třída Y je potomek třídy X. Objekt typu Y lze vždy přetypovat na objekt typu X.
 - C Předpokládejme, že třída Y je potomek třídy X a třída Z je potomek třídy Y. Objekt typu Z lze vždy přetypovat na objekt typu X.
 - *D Předpokládejme, že třída Y je potomek třídy X a třída Z je potomek třídy X a není potomek třídy Y. Objekt typu Z lze vždy přetypovat na objekt typu Y.
 - E Předpokládejme, že třída X je rozhraní (Java - interface, C++ abstraktní třída). Z třídy X nelze přímo vytvářet instance, lze však vytvářet potomky nebo rozhraní implementovat.

Počítačové sítě

- 16** Nejvyšší dosažitelná rychlost přenosu dat v b/s přenosovým kanálem je daná
- A frekvencí procesoru vysílajícího počítače
 - B frekvencí použitého nosného signálu
 - C kódováním přenášené informace do prvků nosného signálu
 - *D rozsahem frekvencí signálů přenositelných přenosovým kanálem a poměrem vysílané a šumové energie v přenosovém kanálu
 - E minimem z frekvencí procesorů přijímajícího a vysílajícího počítače

- 17** Spojovaný komunikační protokol zaručuje
- A doručení alespoň 95 procent vyslaných dat
 - *B doručení všech vyslaných dat v originálním pořadí
 - C doručení všech vyslaných dat ve stanoveném časovém intervalu
 - D doručení všech vyslaných dat v originálním pořadí ve stanoveném časovém intervalu
 - E doručení všech vyslaných dat v originálním pořadí v časovém intervalu stanoveném kvalitou spoje

18 URL (Uniform Resource Locator) se skládá z následujících částí (v uvedeném pořadí):

- A metoda (protokol), adresa uzlu, port (nepovinně)
- B metoda (protokol), adresa uzlu
- C metoda (protokol), adresa uzlu, autentizace, port
- *D metoda (protokol), adresa uzlu, port (nepovinně), cesta a další parametry (nepovinně)
- E metoda (protokol), adresa uzlu, port (nepovinně), cesta a autentizace

19 Komunikační model P2P (Peer to Peer) lze charakterizovat takto:

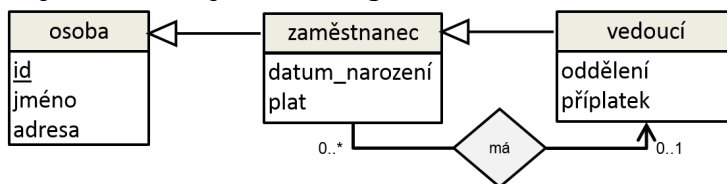
- *A přímá komunikace klientů, poskytování a využívání zdrojů
- B sdílení zdrojů; bez přímé komunikace klientů
- C centralizace zdrojů, klient zahajuje a uzavírá komunikaci
- D decentralizace zdrojů, zdroj zahajuje a uzavírá komunikaci
- E centralizace zdrojů, zdroj zahajuje a klient uzavírá komunikaci

20 Síťový (komunikační) protokol řídí komunikaci mezi dvěma či více HW/SW komponentami počítačové sítě a definuje:

- *A strukturu (formát) zasílaných zpráv, pořadí zpráv a význam každé sekvence bitů zprávy
- B autentizaci a autorizaci zprávy
- C strukturu (formát) zasílaných zpráv a šířku pásma
- D autentizaci zprávy a rezervaci komunikačních prostředků
- E autentizaci zprávy a pořadí zpráv

Databázové systémy

21 Mějme následující E-R diagram:



Zvolte tvrzení, které **není** pravdivé:

- A O každém vedoucím evidujeme jméno, adresu, datum narození a plat včetně příplatku.
- *B Každý zaměstnanec má právě jednoho vedoucího.
- C Každý vedoucí je jednoznačně identifikován ID a má na starosti nejvýše jedno oddělení.
- D Vedoucí nemusí mít na svém oddělení zaměstnance.
- E Vedoucí může mít svého vedoucího.

22 Mějme databázi evidující zvířata v zoo, která obsahuje relaci *zvire*(*id_zvirete*, *jméno*, *id_druhu*, *rok_narození*, *id_matky*, *id_otce*) a relaci *druh*(*id_druhu*, *název*, *popis*, *ohrožení*). Atributy *id_zvirete* a *id_druhu* jsou celočíselné primární klíče příslušných relací. V relaci *zvire* je atribut *id_druhu* cizí klíč do relace *druh* a atributy *id_matky* a *id_otce* jsou cizí klíče do relace *zvire*. Uvažujte následující dotaz v relační algebře:

$$\sigma_{zvire.id_matky=druh.id_druhu}(zvire \times druh)$$

Které z následujících tvrzení je správné?

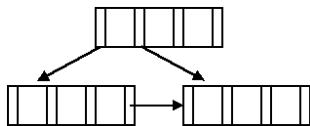
- A Dotaz je syntakticky správně (lze jej vyhodnotit) a vrací smysluplný výsledek.
- *B Dotaz je syntakticky správně, ale nevrací smysluplný výsledek.
- C Dotaz je syntakticky správně, ale při jeho vykonávání dojde k zacyklení a bude vrácena chyba příliš dlouho trvajícího výpočtu.
- D Dotaz je syntakticky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

- 23** Mějme v databázi relaci osoba(id, jmeno, prijmeni, datum_narozeni), ve které existuje alespoň jeden záznam. Datový typ atributu id je celé číslo, datové typy atributů jmeno a prijmeni jsou řetězce maximální délky 200 znaků a datum_narozeni má datový typ datum. Atribut id tvoří primární klíč relace. Co se stane po spuštění následujícího příkazu?

```
INSERT INTO osoba (SELECT id, prijmeni, jmeno, datum_narozeni FROM osoba)
```

- A Nestane se nic - databáze neohlásí chybu, ale obsah tabulky se nezmění.
- B Databáze ohlásí chybu, protože neodpovídají datové typy vkládaných dat.
- *C Databáze ohlásí chybu, protože by se vložil duplicitní záznam.
- D Počet řádků v tabulce osoba se zdvojnásobí - budou znovu vloženy všechny osoby, ale jejich jména se stanou příjmením a příjmení jmény.
- E Databáze ohlásí chybu, protože příkaz není syntakticky správně.

- 24** Mějme následující B⁺-strom:



Zvolte nejmenší datovou množinu, kterou lze organizovat v tomto B⁺-stromu:

- A {2, 3}
- B {2, 3, 5}
- *C {2, 3, 5, 7}
- D {2, 3, 5, 7, 11}
- E {2, 3, 5, 7, 11, 13}

- 25** Mějme v databázi relaci data(A,B,C,D), která obsahuje velké množství záznamů. Jestliže vytvoříme index na atributu B, rychlost provádění operací SELECT, INSERT a DELETE nad touto relací:

- A zůstane beze změny
- *B rychlost operací SELECT (pokud porovnávají atribut B) bude zvýšena, změna rychlosti operace DELETE závisí na podmínce mazání a rychlost operace INSERT bude snížena
- C rychlost operací SELECT (pokud porovnávají atribut B) bude zvýšena, rychlost operací DELETE a INSERT zůstane beze změny
- D rychlost operací SELECT (pokud porovnávají atribut B) zůstane beze změny, změna rychlosti operace DELETE závisí na podmínce mazání a rychlost operace INSERT bude zvýšena
- E rychlost operací SELECT (pokud porovnávají atribut B) zůstane beze změny, rychlost operací DELETE a INSERT zůstane také beze změny

Softwarové inženýrství

- 26** Jako softwarový inženýr volíte strategii verifikace a validace pro dosud nedokončený projekt. Dáte přednost testování, statické analýze kódu či kombinaci obojího a proč? Zvolte nejvhodnější variantu:
- A Použijí testování, protože je snáz aplikovatelné na nedokončené projekty než na dokončené.
 - B Použijí statickou analýzu kódu, protože je levnější a efektivnější než testování.
 - C Použijí testování, protože statická analýza kódu mezi metody verifikace a validace nepatří.
 - D Použijí statickou analýzu kódu, protože testování mezi metody verifikace a validace nepatří.
 - *E Použijí kombinaci testování a statické analýzy kódu, protože každá z těchto technik dokáže pokrýt jiný typ chyb a tím se vhodně doplňují.

- 27** Který z následujících požadavků na webovou aplikaci je nefunkčním požadavkem?
- A Po kliknutí na odkaz "Vyhledej" se zobrazí stránka s výsledky vyhledávání.
 - B Po kliknutí na odkaz "Vyhledej" se zobrazí stránka s výsledky vyhledávání bez chybových hlášení.
 - C Po kliknutí na odkaz "Vyhledej" se barva odkazu změní z modré na červenou.
 - *D** Po kliknutí na odkaz "Vyhledej" se stránka s výsledky vyhledávání zobrazí do 2 sekund.
 - E Po kliknutí na odkaz "Vyhledej" stránka zešedne, uprostřed bude text "Vyhledávám..." a pak se se zobrazí výsledky.
-
- 28** Jednou z aktivit objektové i strukturované analýzy systémů je identifikace rolí, ve kterých vystupují fyzické osoby reálného světa (aktér v UML diagramu případů užití u objektové analýzy, terminátor v kontextovém diagramu u strukturované analýzy). Které z následujících tvrzení o vztahu rolí a osob reálného světa je obecně pravdivé (napříč různými systémy a scénáři jejich používání)?
- *A** Je možné, aby v jedné roli vystupovalo více osob; je možné, aby jedna osoba vystupovala ve více rolích.
 - B Je možné, aby v jedné roli vystupovalo více osob; jedna osoba může vystupovat v nejvýše jedné roli.
 - C V jedné roli může vystupovat nejvýše jedna osoba; je možné, aby jedna osoba vystupovala ve více rolích.
 - D V jedné roli může vystupovat nejvýše jedna osoba; jedna osoba může vystupovat v nejvýše jedné roli.
 - E Žádné z uvedených.
-
- 29** Co se v kontextu vývoje informačních systémů rozumí pod pojmem Model-View-Controller? Vyberte nejvýstižnější z následujících zařazení:
- A Testovací technika
 - B Iterativní metodika vývoje prototypu
 - C Metodika určující postup modelování v jazyce UML
 - *D** Architektonický vzor
 - E Metoda validace požadavků na systém
-
- 30** Ve kterém z následujících modelů vývojového procesu je kladen největší důraz na správnost a kompletnost požadavků před započítím implementace?
- *A** Vodopádový
 - B Spirálový
 - C Agilní
 - D Iterativní
 - E Inkrementální
-

Tato strana je prázdná.