

**Zadání a řešení testu z informatiky a zpráva
o výsledcích přijímacího řízení do magisterského
navazujícího studia od podzimu 2014**

**Zpráva o výsledcích přijímacího řízení
do magisterského navazujícího studia od podzimu 2014**

Studium v českém jazyce

Počet podaných přihlášek	524
Počet přihlášených uchazečů	473
Počet uchazečů, kteří splnili podmínky přijetí	397
Počet uchazečů, kteří nesplnili podmínky přijetí	76
Počet uchazečů přijatých ke studiu, bez uvedení počtu uchazečů přijatých ke studiu až na základě výsledku přezkoumání původního rozhodnutí	397
Počet uchazečů přijatých celkem	397
Percentil pro přijetí	uspěli všichni uchazeči, kteří se zúčastnili přijímací zkoušky

Základní statistické charakteristiky

	Informatika	Matematika	Celkem	
Počet otázek	30	25	55	
Počet uchazečů, kteří se zúčastnili přijímací zkoušky	264	264	264	
Nejlepší možný výsledek	30.00	25.00	55.00	
Nejlepší skutečně dosažený výsledek	27.5	25	51.25	
Průměrný výsledek	16.00	13.68	29.69	
Medián	16.75	14.0	30.5	
Směrodatná odchylka	4.96	5.02	8.41	
	Percentil			
Decilové hranice výsledku *	10	9.33	7.0	18.65
	20	11.75	9.65	23.75
	30	13.25	11.0	26.0
	40	15.25	13.0	27.6
	50	16.75	14.0	30.5
	60	17.7	14.95	32.4
	70	18.75	16.25	34.03
	80	20.25	17.85	36.75
	90	22.50	20.0	39.68

* 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.

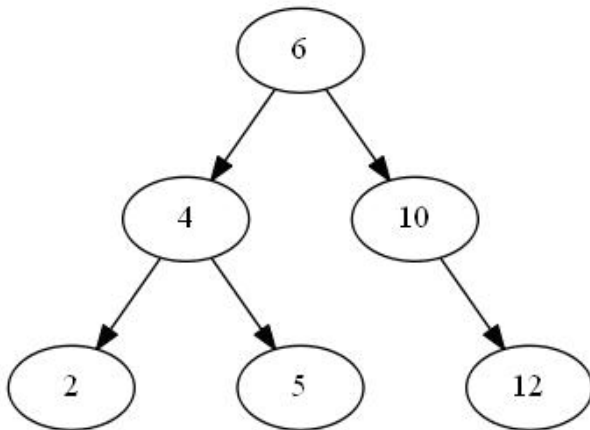
Přijímací zkouška - Informatika

Jméno a příjmení - pište do okénka	Číslo přihlášky	Číslo zadání
		14

Algoritmizace a datové struktury

- 1** Která z následujících datových struktur je nejvhodnější (z hlediska časové i paměťové složitosti) pro implementaci vyhodnocení aritmetického výrazu zapsaného v postfixové notaci?
- A ohodnocený graf
 - B prioritní fronta
 - C množina
 - *D zásobník
 - E hašovací tabulka
-

2



Předpokládejte zobrazený binární vyhledávací strom. Která z následujících sekvencí hodnot mohla vést k vytvoření uvedeného binárního stromu za předpokladu, že byl strom na počátku prázdný a během vkládání jsme strom nevyvažovali?

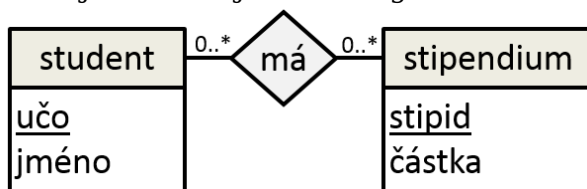
- A 6, 10, 2, 12, 4, 5
 - B 6, 12, 5, 10, 4, 2
 - C 6, 5, 12, 4, 10, 2
 - *D 6, 4, 5, 10, 2, 12
 - E 6, 4, 2, 12, 10, 5
-

- 3** Která z uvedených struktur je nejvhodnější na uložení dat v tabulce symbolů (např. pro definované proměnné, funkce, ...) vytvářených během překladač programu překladačem?
- A zřetězený seznam
 - B prioritní fronta
 - C množina
 - D zásobník
 - *E hašovací tabulka
-

- 4** V případě, že použijeme hašovací funkci F pro implementaci hašovací tabulky, může docházet ke kolizím. Jeden ze způsobů řešení kolize je násobné hašování – pokud dojde ke kolizi s dříve vloženou položkou, hašujeme kolidující adresu znovu a nový výsledek použijeme jako adresu, na kterou se pokusíme uložit vkládanou položku. Které z uvedených tvrzení je pravdivé?
- A** Časová složitost vložení libovolné položky dle zadaného klíče je $O(1)$ a všechny položky se podaří vždy vložit.
- B** Násobné hašování řeší problém kolizí, při opakovaném hašování nemůže ke kolizi dojít.
- *C** V případě, že hledaná položka existuje, je časová složitost nalezení položky dle zadaného klíče $O(n)$.
- D** Násobné hašování je, z hlediska možnosti kdy lze vložit do hašovací tabulky novou položku, ekvivalentní řešení kolizí pomocí zřetězeného seznamu kolidujících položek.
- E** Násobné hašování se stejnou funkcí F nelze využít, je nutné použít pro každou iteraci hašování jinou funkci F_i .
-
- 5** Které z uvedených tvrzení platí pro verzovací nástroje (SVN, Git, Mercury, ...)? Vyberte možnost, ve které jsou všechna uvedená tvrzení pravdivá.
- *A** Verzování kódu umožňuje práci více vývojářů souběžně nad jedním projektem, umožňuje návrat zpět na libovolnou verzi, umožňuje vytváření separátních vývojových větví s možností jejich začlenění zpět do hlavního projektu, umožňuje zaslání aktualizovaných souborů (Commit).
- B** Verzování kódu umožňuje práci pouze jedinému (ne několika) vývojáři nad daným projektem, umožňuje návrat zpět na libovolnou verzi, poskytuje zálohování kódu mimo adresář se soubory, umožňuje vytváření separátních vývojových větví s možností jejich začlenění zpět do hlavního projektu, umožňuje stažení nových verzí souborů (Update).
- C** Verzování kódu umožňuje práci více vývojářů souběžně nad jedním projektem, umožňuje stažení aktuální verze (head), umožňuje vytváření separátních vývojových větví, které ale nelze začlenit zpět do hlavního projektu, umožňuje zaslání aktualizovaných souborů (Commit).
- D** Verzování kódu umožňuje práci více vývojářů souběžně nad jedním projektem, umožňuje vytváření separátních vývojových větví s možností jejich začlenění zpět do hlavního projektu, umožňuje stažení nových verzí souborů (Update), automaticky maže starší verze souborů na serveru poté, co si je všichni vývojáři stáhli pomocí Update.
- E** Verzování zdrojového kódu se nepoužívá, typicky se verzují pouze výsledné binární soubory vzniklé při překladu. Verzování umožňuje návrat zpět na libovolnou starší verzi, umožňuje vytváření separátních vývojových větví s možností jejich začlenění zpět do hlavního projektu, umožňuje stažení nových verzí souborů (Update).

Databázové systémy

- 6** Uvažujte následující E-R diagram:



Které z tvrzení je pravdivé?

- A** Student může mít maximálně jedno stipendium.
- B** Stipendium může být uděleno maximálně jednomu studentovi.
- C** Každé stipendium musí být vždy uděleno alespoň jednomu studentovi.
- *D** Student může mít více stipendií.
- E** Každý student musí mít alespoň jedno stipendium.

- 7** Předpokládejme, že relační databáze obsahuje tabulku *zamestnanec* (#ID, jmeno, plat), tabulku *projekt* (#ID, jmeno) a tabulku *pracuje_na* (#ID_zamestnanec, #ID_projekt). Primární klíče jsou označeny symbolem #, atributy ID_zamestnanec a ID_projekt v tabulce *pracuje_na* jsou cizí klíče do tabulky *zamestnanec* a *projekt*.
Jaký výsledek vrátí následující SQL dotaz:

```
SELECT jmeno
FROM projekt, pracuje_na
WHERE id_projekt=id AND jmeno LIKE '%oracle%'
GROUP BY id,jmeno HAVING count(*)>=5;
```

- *A** Jména projektů, které obsahují ve svém jméně podřetězec 'oracle' a na nichž pracuje alespoň 5 zaměstnanců.
B Jména projektů, které obsahují ve svém jméně podřetězec 'oracle' a na nichž nepracuje žádný zaměstnanec.
C Jména zaměstnanců, kteří pracují alespoň na 5 projektech, které obsahují ve svém jméně podřetězec 'oracle'.
D Počet projektů, které obsahují ve svém jméně podřetězec 'oracle'.
E Jména všech projektů, na nichž pracuje alespoň 5 zaměstnanců.

- 8** Vyberte pravdivé tvrzení o kandidátním klíči (KK):

- A** Pro jednu entitní množinu existuje unikátní KK.
B KK je maximální počet závislých atributů.
***C** KK je super klíč.
D KK má vždy minimálně dva atributy.
E KK je libovolná podmnožina atributů.

- 9** První normální forma (1NF) vyžaduje:

- *A** aby všechny atributy byly atomické
B aby pro všechny atributy platilo pravidlo ACID
C aby všechny závislosti byly atomické
D aby všechny normální formy byly atomické
E aby všechny závislosti byly atomické a aby zároveň platila i druhá normální forma (2NF)

- 10** Jazyk SQL:

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

Počítačové systémy

- 11** Které číslo v desítkové soustavě je ekvivalentem čísla vyjádřeného v šestnáctkové (hexadecimální) soustavě jako 2FE?

- *A** 766
B 667
C 777
D 666
E 101111110

12 DMA (direct memory access) slouží k:

- *A přenosu dat mezi operační pamětí a vstupně/výstupním zařízením bez kopírování dat do registrů procesoru
- B přenosu dat mezi operační pamětí a vstupně/výstupním zařízením s využitím kopírování dat do registrů procesoru
- C přenosu dat mezi operační pamětí a cache procesoru bez kopírování dat do registrů procesoru
- D přenosu dat mezi operační pamětí a cache procesoru s využitím kopírování dat do registrů procesoru
- E k přenosu dat mezi jednotlivými registry procesoru

13 V osmibitové reprezentaci se ve dvojkovém doplňkovém kódu dekadické číslo -11 zapíše jako:

- *A 11110101
- B 11110100
- C 00001011
- D 00001100
- E 00001101

14 V operačním systému UNIX se pro vytvoření nového procesu využívá systémové volání:

- *A fork
- B execve
- C pthread_create
- D ptrace
- E _exit

15 Necht v časovém okamžiku 0 vznikly požadavky na procesy P1, P2 a P3. Proces P1 potřebuje pro svůj běh 8 časových jednotek CPU, proces P2 potřebuje 3 jednotky a proces P3 potřebuje 5 jednotek. Pokud máme k dispozici jeden procesor a plánování CPU je prováděno algoritmem SJF (shortest job first) s předbíráním, jaký proces bude mít k dispozici procesor v časovém okamžiku 6 jednotek CPU (od okamžiku 0)?

- A P1
- B P2
- *C P3
- D P1 a zároveň P2
- E žádný z těchto procesů

Programování

16 Předpokládejte dotazníkové šetření s 32 různými otázkami. Povolené odpovědi jsou pouze ANO / NE. Výsledky pro konkrétní osobu jsou uloženy jako bitové pole o délce 32 bitů, bit 1 na pozici k odpovídá odpovědi ANO na k. otázku, 0 odpovídá odpovědi NE. Bitové pole je uloženo v datovém typu neznaménkové celé číslo (uint32). Programátor má za úkol napsat program, který dostane jako vstup jednorozměrné pole bitových polí (uint32 odpovedi[X]) o celkové délce X, kde X je počet odpovídajících osob. Program má co nejrychleji na architektuře x86 vyhodnotit, zda pro každou z položených otázek byl počet odpovědí pro tuto otázku ANO sudý. Vyhodnocení proběhne pro všechny otázky a vrátí OK v případě, že sudý počet odpovědí ANO byl pro každou z 32 otázek, jinak vrátí NOK. Která z uvedených odpovědí je správná?

- A Programátor bude potřebovat provést alespoň 32 * X operací porovnání.
- B Programátor bude potřebovat provést alespoň 32 * (X-1) operací součtu a 32 operací porovnání.
- C Programátor bude potřebovat provést alespoň 32 operací xor a (X-1) operací porovnání.
- D Programátor bude potřebovat provést alespoň X-1 operací součtu a 32 operací porovnání.
- *E Programátor bude potřebovat provést alespoň X-1 operací xor a 1 operaci porovnání.

```
17 integer foo(integer x, integer y, integer z)
begin
  if (x < 10) then y = 40
  z = x - y
  return y - z
end

program main()
begin
  integer a, b, c, d
  a = 10
  b = 20
  c = 30
  d = 40
  d = foo(a, b, c)
  print a, b, c, d
end
```

Předpokládejte, že funkce foo() využívá volání pomocí reference. Jaký bude výsledek pro uvedený pseudokód? (Pořadí proměnných vypsanych v možnostech je a b c d)

- A 10 20 30 30
- *B** 10 20 -10 30
- C 10 40 -30 70
- D 10 20 30 70
- E 10 20 30 40

```
18 // PSEUDOKÓD 1
sum = 0
for i = 1 to n {
  sum = sum + i
}
print sum

// PSEUDOKÓD 2
sum = 0
i = <iniciální_hodnota>
do {
  sum = sum + i
  i = i + 1
} while ( <podmínka> )
print sum
```

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 <iniciální_hodnota> a <podmínka> tak, aby oba pseudokódy vypsaly stejnou hodnotu sum?

- A <iniciální_hodnota> = 0; <podmínka> = $i < n$
- B <iniciální_hodnota> = 0; <podmínka> = $i < n - 1$
- *C** <iniciální_hodnota> = 0; <podmínka> = $i < n + 1$
- D <iniciální_hodnota> = 1; <podmínka> = $i < n - 1$
- E <iniciální_hodnota> = 1; <podmínka> = $i < n$

- 19** Rozhodněte, které z uvedených tvrzení je v běžných OOP jazycích (C++, Java, C#) platné:
- *A Pokud je metoda třídy virtuální, lze její implementaci změnit v potomcích této třídy.
 - B Pokud je metoda třídy virtuální, nelze změnit její implementaci v potomcích této třídy.
 - C Režie při zavolání virtuální metody je typicky nižší než při zavolání metody volané včasnou vazbou (nevirtuální).
 - D Virtuální metodu lze v potomcích změnit na metodu volanou včasnou vazbou.
 - E Pokud metoda není virtuální v dané třídě X, tak ji v potomcích třídy X nelze na virtuální předeklarovat.

- 20** Rozhodněte, která z uvedených možností (pro běžné jazyky typu C++, Java, C#) obsahuje právě všechna pravdivá tvrzení?
- I. Lokální proměnné funkce uložené na zásobníku jsou automaticky odstraněny v případě ukončení funkce, ve které byly vytvořeny.
 - II. Pokud je paměť dynamicky alokovaná na haldě, zaniká v případě, že je provedeno její explicitní uvolnění (jazyky bez garbage collection) nebo není paměť referencovaná (jazyky s garbage collection).
 - III. Lokální proměnná na zásobníku nemůže 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.

Počítačové sítě

- 21** V oblasti telekomunikací a počítačových sítí je multiplexování metoda, kterou se více signálů spojuje do jednoho signálu po sdíleném médiu. Platí:
- A Ve frekvenční doméně lze multiplexovat analogové i digitální signály.
 - B Ve frekvenční doméně lze multiplexovat pouze digitální signály.
 - *C Ve frekvenční doméně lze multiplexovat pouze analogové signály.
 - D V časové doméně nelze multiplexovat digitální signály.
 - E V časové doméně nelze multiplexovat analogové signály.

- 22** Která z následujících položek je korektní adresa protokolu IPv6:

- *A 2607:f0d0:1002:51::4
- B 2607:f0g0:1002:51::4
- C 2607:f0d0:1002:51:::4
- D 2607:f0d0:1002:0051:0000:0000:0000:0000:0004
- E 2607.f0d0.1002.51.4

- 23** Mějme počítač A s IP adresou 10.10.0.10 a maskou sítě 255.255.255.0 a počítač B s IP adresou 10.0.10.10 a maskou sítě 255.255.0.0. Oba počítače jsou připojené ke svému routeru (směrovači) pomocí metalického UTP spoje. Jestliže chce počítač A poslat paket počítači B:
- *A musí ho poslat na bránu sítě 10.10.0.0
 - B musí ho poslat na bránu sítě 10.0.0.0
 - C posílá ho přímo počítači B
 - D nejprve si zjistí pomocí ARP protokolu hardwarovou adresu síťové karty (MAC) počítače B a pak pošle paket s touto MAC adresou
 - E paket nelze poslat, protože masky sítě nejsou kompatibilní

- 24** Z hlediska řízení přístupu k bezdrátovému přenosovému médiumu použitému pro šíření elektromagnetického záření je toto médium ekvivalentní fyzické topologii sítě (tvořené vodivými přenosovými médii) typu:
- A kruh
 - *B sběrnice
 - C strom
 - D hvězda
 - E úplné propojení (peer-to-peer)
-
- 25** Nejvyšší teoreticky dosažitelná rychlost přenosu dat v přenosovém médiumu (vyjádřená v b/s) je nejvíce omezena
- A taktovací frekvencí procesoru vysílajícího počítače
 - B frekvencí nosného signálu
 - C kódováním přenášené informace do prvků signálu
 - *D poměrem vysílané a šumové energie a rozsahem frekvencí signálů přenesitelných přenosovým médiumem
 - E minimem z taktovacích frekvencí procesorů přijímajícího a vysílajícího počítače

Softwarové inženýrství

- 26** Které z následujících tvrzení o funkčních a nefunkčních požadavcích na software je pravdivé?
- A Nefunkční požadavky popisují, jaké procedury by systém neměl provádět.
 - B Mezi funkční požadavky patří spolehlivost.
 - C Rozdělení požadavků na funkční a nefunkční se využívá k označení nekonzistencí v požadavcích.
 - D Funkční požadavky musí být specifikovány dříve než nefunkční.
 - *E Mezi nefunkční požadavky patří bezpečnost, výkonnost a testovatelnost.
-
- 27** Která z následujících aktivit by **neměla** být součástí refaktoringu?
- A snížení složitosti kódu
 - B zvýšení přehlednosti kódu
 - C změna architektury systému
 - D rozložení složitých tříd či funkcí na více jednodušších
 - *E implementace nových uživatelských funkcí
-
- 28** Jakou testovací techniku zvolíte na ověření, zda změna v otestovaném systému nevnesla do systému nové chyby?
- A alfa testování (alpha testing)
 - B beta testování (beta testing)
 - *C regresní testování (regression testing)
 - D akceptační testování (acceptance testing)
 - E testování jednotek (unit testing)
-
- 29** Které z následujících tvrzení o návrhových vzorech **není** pravdivé?
- A Návrhové vzory popisují řešení častých problémů v kontextu návrhu softwarových systémů.
 - B Návrhový vzor není hotovým návrhem, ale spíše předlohou návrhu pro mnoho různých situací.
 - C Objektově orientované návrhové vzory typicky zobrazují vztahy a interakce mezi třídami či objekty.
 - *D Návrhové vzory se v první řadě zaměřují na řešení problémů s bezpečností software.
 - E Příkladem návrhového vzoru je vzor Singleton, který pomáhá zajistit existenci maximálně jedné instance určité třídy.

30 Který z následujících termínů **nemá** (či má nejmenší) spojitost s agilním vývojem software?

- A SCRUM
 - B extrémní programování
 - *C Yourdonova analýza
 - D test driven development
 - E lean software development
-