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

Zpráva o výsledcích přijímacího řízení do magisterského navazujícího studia od jara 2014

Studium v českém jazyce

Počet podaných přihlášek	**176**
Počet přihlášených splňujících kriteria pro prominutí	27
Počet přihlášených splňujících podmínky pro přijetí	118
Počet přihlášených nesplňujících podmínky pro přijetí	58
Percentil pro přijetí	4,0 (odpovídá více než10 bodům celkem)
Čas na řešení testu z informatiky	90 minut
Čas na řešení testu z matematiky	75 minut

Základní statistické charakteristiky

Cronbachovo alfa 0,836

		Informatika	**Matematika**	**Celkem**
Počet otázek		30	25	55
Počet uchazečů, kteří se zúčastnili přijímací zkoušky		107	106	107
Nejlepší možný výsledek		30.00	25.00	55.00
Nejlepší skutečně dosažený výsledek		26.50	25.00	47.75
Průměrný výsledek		15.84	12.03	27.76
Medián		16.25	12.75	28.00
Směrodatná odchylka		5.34	5.01	9.32
	Percentil			
Decilové hranice výsledku *	10	8,7	5,5	14,5
	20	11,55	8	21,35
	30	13,75	9,75	24
	40	15,35	11,25	25,75
	50	16,25	12,75	28
	60	17,5	13,75	29,65
	70	19,3	14	32,9
	80	20,05	17,4	35,95
	90	21,25	19,7	38,75

* 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*
		146

Databázové systémy

---

1 Mějme relaci *zamestnanec(id, jmeno, prijmeni, datum_nastupu, datum_ukonceni, plat, id_nadrizeny)*, představující jednoduchou evidenci zaměstnanců. Co vrací následující SQL dotaz?

```
SELECT v.jmeno, v.prijmeni, SUM(z.plat)
FROM zamestnanec AS z, zamestnanec AS v
WHERE v.id = z.id_nadrizeny AND z.datum_ukonceni IS NOT NULL
GROUP BY v.id, v.jmeno, v.prijmeni
```

A Jména a příjmení pracovníků a celkové množství peněz vyplácených jako platy jejich přímých nadřízených, kteří ve firmě pracují.

***B** Jména a příjmení vedoucích pracovníků a celkové množství peněz vyplácených jako platy jejich přímých podřízených, kteří už ve firmě nepracují.

C Jména a příjmení vedoucích pracovníků a celkové množství peněz vyplácených jako platy jejich přímých podřízených, kteří ve firmě pracují.

D Dotaz je syntakticky špatně a vrátí chybu.

E Jména a příjmení pracovníků a celkové množství peněz vyplácených jako platy jejich přímých nadřízených, kteří už ve firmě nepracují.

---

2 Mějme následující relaci *zvire*, představující evidenci zvířat v zoologické zahradě:

ID_zvirete	**Jmeno**	**Rok_narozeni**	**Pohlavi**	**ID_matky**	**ID_otce**
1	Cora	1998	F	*NULL*	*NULL*
2	Umca	1998	M	*NULL*	*NULL*
3	Nanuk	2012	M	1	2
4	Kometa	2012	F	1	2

Jestliže se v zoologické zahradě narodí Nanukovi a Kometě v roce 2013 nové zvíře "Pepa", který z následujících výrazů relační algebry použijeme, abychom měli evidenci v pořádku?

A $zvire \leftarrow zvire + \{(5,'Pepa', 2013,'M', 4, 3)\}$

B $zvire \leftarrow (5,'Pepa', 2013,'M', 4, 3)$

C $zvire \leftarrow zvire + (5,'Pepa', 2013,'M', 4, 3)$

***D** $zvire \leftarrow \{(5,'Pepa', 2013,'M', 4, 3)\} \cup zvire$

E $zvire \leftarrow (5,'Pepa', 2013,'M', 4, 3) \cup zvire$

---

3 Která z následujících vlastností **není vyžadována** pro databázovou transakci?

A Atomicita - transakce se provede buď celá nebo se neprovede vůbec.

B Trvalost - změny provedené dokončenou transakcí jsou uloženy v databázi a nemohou být ztraceny.

***C** Dokončitelnost - každá transakce musí být dokončena v konečném čase.

D Konzistence - po provedení transakce zůstává databáze v konzistentním stavu.

E Izolovanost - operace prováděné uvnitř transakce jsou viditelné pouze pro tuto transakci.

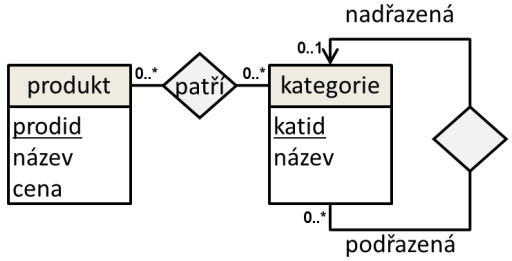
---

4 Jestliže návrh relací v dané databázi splňuje třetí normální formu, pak **nesmí** platit:

A Existuje atribut v některé relaci, který není tranzitivně závislý na primárním klíči dané relace.
B Všechny relace mají definován primární klíč.
C Existuje atribut v některé relaci, který je dále nedělitelný.
D Existuje atribut v některé relaci, který je součástí více superklíčů dané relace.
***E** Existuje atribut v některé relaci, který nezávisí na žádném kandidátním klíči dané relace.

---

5 Pro zadaný E-R model určete, co popisuje:



A Model popisuje databázi produktů zařazených do kategorií. Každý produkt lze zařadit do několika kategorií, produkt musí být zařazen alespoň do jedné kategorie. Kategorie lze hierarchicky strukturovat tak, že každá kategorie může mít maximálně jednu nadřazenou kategorii.
B Model popisuje databázi produktů zařazených do kategorií. Každý produkt lze zařadit do jedné kategorie, produkt nemusí být zařazen do žádné kategorie. Kategorie lze hierarchicky strukturovat tak, že každá kategorie může patřit do několika nadřazených kategorií.
C Model popisuje databázi produktů zařazených do kategorií. Každý produkt lze zařadit do několika kategorií, produkt musí být zařazen alespoň do jedné kategorie. Kategorie lze hierarchicky strukturovat tak, že každá kategorie může patřit do několika nadřazených kategorií.
***D** Model popisuje databázi produktů zařazených do kategorií. Každý produkt lze zařadit do několika kategorií, produkt nemusí být zařazen do žádné kategorie. Kategorie lze hierarchicky strukturovat tak, že každá kategorie může mít maximálně jednu nadřazenou kategorii.
E Model popisuje databázi produktů zařazených do kategorií. Každý produkt lze zařadit do jedné kategorie, produkt nemusí být zařazen do žádné kategorie. Kategorie lze hierarchicky strukturovat tak, že každá kategorie může mít maximálně jednu nadřazenou kategorii.

Algoritmizace a datové struktury

---

6 Předpokládejme existenci jednostranně spojovaného seznamu prvků, u kterého máme uložený pouze ukazatel LAST na poslední vkládaný prvek. Délka seznamu je n. Které z uvedených tvrzení platí?

A Časová složitost získání prvního prvku v seznamu je O(1).
B Paměťová režie pro realizaci jednostranně spojovaného seznamu je typicky nižší než pro pole prvků uložených kontinuálně za sebou.
C Časová složitost nalezení prvku se zadanou hodnotou je O(1).
D Tento seznam je vhodný pro reprezentaci struktury typu FIFO.
***E** Časová složitost nalezení prvku se zadanou hodnotou je O(n).

---

7 Pro datovou strukturu známou jako kompletní binární strom platí, že každý uzel s výjimkou listů má právě dva potomky. Kolik interních uzlů (tj. uzlů mimo listy) je v kompletním binárním stromě obsahujícím 1024 listů?

A 1024
B 1025
***C** 1023
D 5512
E 2048

8 Která z uvedených datových struktur je nejvhodnější pro implementaci kolekce položek v případě, že požadujeme splnění těchto tří charakteristik? 1) Jednotlivé položky jsou vkládány a odebírány v LIFO pořadí. 2) Předem není určen limit na počet položek v kolekci hodnot. 3) Velikost jedné položky je výrazně větší než paměťový prostor nutný pro uložení ukazatele.

A Asociativní pole s ukazatelem jako klíčem.
B Pole hodnot sekvenčně uložených v paměti.
***C** Jednostranně spojovaný seznam s uchováním ukazatele na poslední vkládaný prvek.
D Binární strom.
E Oboustranně spojovaný seznam s uchováním ukazatele na jeho začátek i konec.

---

9 Která z uvedených datových struktur je nejvhodnější pro implementaci kolekce položek v případě, že požadujeme splnění těchto tří charakteristik? 1) Přístup k hodnotě jednotlivých položek dle zadaného klíče je v třídě časové složitosti O(1). 2) Předem je limitován počet položek v kolekci hodnot. 3) Velikost klíče není předem omezená.

A Pole hodnot sekvenčně uložených v paměti obsahující dvojici klíč a hodnota.
B Binární vyhledávací strom obsahující v uzlu dvojici klíč a hodnota.
C Jednostranně spojovaný seznam obsahující dvojici klíč a hodnota s uchováním ukazatele na poslední vkládaný prvek.
***D** Asociativní pole s vhodně zvolenou hašovací funkcí mapující klíč na hodnotu.
E Oboustranně spojovaný seznam obsahující dvojici klíč a hodnota s uchováním ukazatele na jeho začátek i konec.

---

10 Předpokládejte, že známe propustnosti linek mezi přímo propojenými síťovými prvky v rámci komunikační sítě. Síť nemusí spojovat každý síťový prvek s každým. Síť si můžeme představit jako neorientovaný graf s hranami ohodnocenými propustnostmi linek. Na vstupu dostaneme uzel Z. Úkolem je nalézt pro každý ze zbývajících uzlů grafu cestu z uzlu Z s nejvyšší propustností. Propustnost cesty je propustnost minimálně ohodnocené hrany cesty. Která z uvedených možností je optimálním řešením z hlediska časové náročnosti?

A Vyzkoušení všech možných cest v grafu síťových prvků hrubou silou se současným výpočtem propustnosti takové cesty.
B Hledání cesty v grafu síťových prvků s největším součtem propustností jednotlivých linek pomocí Euklidova algoritmu.
C Hledání minimální kostry grafu síťových prvků. Nejvýhodnější cesta pro přenos dat vede po nalezené kostře grafu.
D Procházení grafu síťových prvků do hloubky s ukládáním propustnosti aktuální cesty.
***E** Pomocí modifikovaného Dijkstrova algoritmu, kde jako metriku vzdálenosti uzlů v grafu považujeme minimální propustnost linky obsažené v dosud nalezené cestě a tuto hodnotu se snaží maximalizovat.

Počítačové sítě

---

11 Služba Domain Name Service (DNS) se v prostředí internetu používá:

***A** pro překlad numerických adres počítačů na symbolické názvy nebo naopak
B pro překlad numerických adres počítačů v internetu na symbolické názvy, opačný směr překladu není možný
C pro inicializaci kořenových a národních doménových prostorů (např. seznam.cz, google.com)
D pro přiřazování TLD (top-level-domain, např. CZ, COM, ORG) jednotlivým registrátorům
E pro překlad symbolických názvů počítačů v internetu na numerické adresy, opačný směr překladu není možný

---

12 Samoopravné kódy používané při přenosu dat umožňují přijímači

A informovat vysílač o chybách vzniklých během přenosu dat
B informovat vysílač o chybách vzniklých během přenosu dat a vyžádat zopakování přenosu chybně přenesených dat
***C** detekovat a opravit téměř všechny chyby vzniklé během přenosu dat
D detekovat všechny a opravit téměř všechny chyby vzniklé během přenosu dat
E detekovat a opravit všechny chyby vzniklé během přenosu dat

---

13 Koncové aplikace jsou na úrovni transportní vrstvy identifikovány pomocí:

A síťové adresy
B jednoznačně vygenerovaného identifikátoru cesty
C hardwarové adresy
***D** čísla portu
E jednoznačně vygenerovaného identifikátoru kanálu

---

14 Sdílení přenosové kapacity souběžnými komunikacemi pomocí digitálních signálů

A lze realizovat ve frekvenčním prostoru i v časovém prostoru
***B** lze realizovat v časovém prostoru, Time-Division Multiplexing (TDM)
C nelze realizovat v časovém prostoru
D lze realizovat ve frekvenčním prostoru, Frequency-Division Multiplexing (FDM)
E nelze realizovat ani ve frekvenčním prostoru ani v časovém prostoru

---

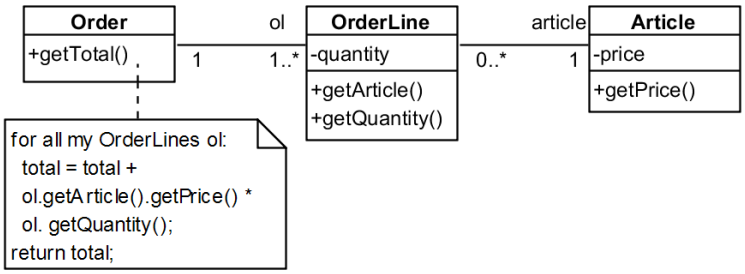
15 Model zajišťování síťových služeb pomocí přepínání paketů

A umožňuje poskytovat pouze spojované síťové služby
B nelze implementovat bez ustavení přímého fyzického spojení mezi komunikujícími uzly
***C** umožňuje poskytovat spojované i nespojované síťové služby
D umožňuje poskytovat pouze nespojované síťové služby
E je aplikovatelný pouze v prostředí rozlehlých sítí (WAN, Wide Area Network)

Softwarové inženýrství

---

16 Uvažujme návrh znázorněný UML diagramem tříd:



Jaký vliv z pohledu nefunkčních požadavků by mělo doplnění atributu `subtotal` (s hodnotou `article.getPrice()*quantity`) do třídy `OrderLine`?

A vyšší výkonnost při volání getTotal(), snazší udržovatelnost aplikace
B výkonnost při volání getTotal() ani udržovatelnost aplikace se vůbec nijak nezmění
***C** vyšší výkonnost při volání getTotal(), obtížnější udržovatelnost aplikace
D nižší výkonnost při volání getTotal(), obtížnější udržovatelnost aplikace
E nižší výkonnost při volání getTotal(), snazší udržovatelnost aplikace

---

17 Které z následujících tvrzení o verifikaci a validaci software je **nepravdivé**?

A Cílem regresního testování je ověřit, zda změna v systému nevnesla do systému nové chyby.
***B** Testování jednotek (unit testing) je použitelné pouze v objektově-orientovaném kódu.
C Výhodou statické analýzy oproti testování je možnost analyzovat nespustitelné části kódu či jiné dokumenty.
D Testování dokáže odhalit přítomnost (některých) chyb, nedokáže však potvrdit nepřítomnost chyb.
E Testování spadá pod obecnější úlohu verifikace a validace, kam patří i statická analýza.

---

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

A Zvolím UML diagram tříd, protože je historicky nástupcem ERD, který už dnes není vhodné používat.
B Zvolím UML diagram tříd, protože nabízí silnější prostředky pro popis primárních a cizích klíčů, což je u persistentních dat podstatné.
C Zvolím ERD, protože UML diagram tříd je pro popis struktury persistentních dat nepoužitelný.
D Zvolím UML diagram tříd, protože ERD je pro popis struktury persistentních dat nepoužitelný.
***E** Zvolím ERD, protože se lépe hodí pro popis dat uložených v relační databázi, která bude v tomto případě vhodnější než objektová databáze.

---

19 Který z následujících popisů nejlépe vystihuje podstatu objektově orientované analýzy (OOA)?

A OOA pomáhá odhalovat chyby v systému ještě před jeho dokončením a je tak alternativou k testování systému.
***B** OOA identifikuje entity problémové domény (objekty), včetně jejich atributů a zodpovědností, a vztahy mezi nimi.
C OOA se zabývá analýzou nekonzistencí mezi třídami v objektově orientovaném kódu aplikace.
D OOA modeluje klíčové procesy analyzované aplikace jako síť systémových funkcí propojených datovými toky.
E OOA se zabývá analýzou nekonzistencí mezi objekty v objektově orientovaném kódu aplikace.

---

20 Jako softwarový inženýr vedete vývoj informačního systému pro nově vznikající společnost, která klade důraz na cenu, ale nemá dosud zcela vyjasněné požadavky na systém. Dáte přednost vodopádovému nebo inkrementálnímu modelu vývoje a proč?

A Zvolím vodopádový model, protože je zástupcem agilních metodik, které nabízí prostředky pro rychlý vývoj systémů s nejasnými požadavky.
B Zvolím inkrementální model, protože je historicky nástupcem vodopádového modelu, který už dnes není vhodné používat.
C Zvolím vodopádový model, protože povede k nižší ceně vývoje.
***D** Zvolím inkrementální model, protože tak dám zákazníkovi možnost začít brzy používat první verze systému, které mu pomohou ve vyjasnění požadavků.
E Zvolím vodopádový model, protože tak dám zákazníkovi možnost začít brzy používat první verze systému, které mu pomohou ve vyjasnění požadavků.

Programování

21
```
integer array[255]
void function init()
begin
  integer i = 0
  while (i < 256)
  begin
    array[i] = i * i * i * i
    i = i + 1
  end
end
```
Odhadněte k čemu lze využít globální pole array inicializované ve funkci init(). Předpokládejte, že funkce init() je zavolána ihned po začátku programu. Velikost datového typu integer je 64 bitů. Prvky pole jsou indexovány od 0.

A Jedná se o inicializaci pole hodnotou i, aby došlo k jejímu umístění do cache paměti procesoru pro rychlý následný přístup.
***B** Jedná se o pole předpočtené pro rychlé získání čtvrté mocniny ze zadaného argumentu typu celé neznaménkové 8bitové číslo.
C Jedná se o pole předpočtené pro rychlé získání čtvrté odmocniny ze zadaného argumentu typu celé neznaménkové 8bitové číslo.
D Žádná z ostatních možností není správná.
E Jedná se o pole předpočtené pro rychlé získání čtvrté mocniny ze zadaného argumentu typu celé neznaménkové 32bitové číslo.

---

22 Rozhodněte, které z uvedených tvrzení **není** v běžných OOP jazycích (C++, Java, C#) platné:

A Z metody třídy volané pozdní vazbou lze volat i metody volané včasnou vazbou téže třídy.
B Pokud je metoda třídy volána pozdní vazbou, lze její implementaci změnit v potomcích této třídy.
C Volání metod pozdní vazbou může znamenat vyšší režii spojenou s voláním.
D Implementace metody volané pozdní vazbou se použije z třídy, pro kterou byl objekt vytvořen nehledě na aktuální typ objektu.
***E** Pokud je metoda třídy volána pozdní vazbou, programátor musí v potomcích třídy vždy poskytnout novou implementaci této metody.

---

23 Rozhodněte, které z uvedených tvrzení je v běžných OOP jazycích (C++, Java, C#) platné:

A Standardní knihovna neobsahuje žádné předdefinované třídy výjimek, programátor si vždy deklaruje vlastní.
B Blok, ve kterém se provádí zachytávání výjimky, je obalen klauzulí throw {}.
C Po obsloužení výjimky se pokračuje v kódu na řádku následujícím po řádku, ve kterém došlo k vyvolání výjimky.
D Výjimky nelze zachytávat, lze je jen vyvolávat.
***E** Pokud není výjimka obsloužena v aktuální funkci, tak se propaguje v zásobníku volání o úroveň výš do volající funkce.

24
```
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. Jaký bude výstup programu po vytištění hodnot a, b, c?

***A** a = 1, b = 7, c = 17.
B a = 2, b = 7, c = 16.
C a = 1, b = 2, c = 3.
D a = 1, b = 1, c = 1.
E a = 7, b = 3, c = 17.

25
```
program main()
begin
  B obj1 = new A
  C obj2 = new C
  C obj3 = new A
end
```

Předpokládejte, že máte objektově orientovanou hierarchii tříd A, B a C. (`B obj1 = new A` vytvoří objekt typu A a uloží ho do proměnné obj1 typu B). Pro které z uvedených možností jde kód přeložit?

A nelze přeložit za žádných okolností (nelze přiřazovat různé datové typy)
***B** A je potomek B, A je potomek C
C B je potomek A, C je potomek A, B je potomek C
D C není potomek B, A není potomek C, A není potomek B
E B je potomek A, C je potomek A

Počítačové systémy

---

26 Vyberte **nepravdivé** tvrzení o reprezentaci záporných čísel na počítačích:

A V přímém kódu odpovídá binární číslo 10000011 dekadickému -3.
B V přímém kódu odpovídá binární číslo 10111111 dekadickému -63.
***C** Ve dvojkovém doplňkovém kódu odpovídá binární číslo 11110001 dekadickému -17.
D Nevýhodou inverzního kódu je tzv. problém dvojí nuly.
E V inverzním kódu odpovídá binární číslo 11110100 dekadickému -11.

---

27 Nechť ve frontě připravených procesů máme procesy P1, P2, P3 a P4 v tomto pořadí. Proces P1 potřebuje pro svůj běh ještě 22 časových jednotek CPU, proces P2 ještě 11, proces P3 ještě 3 a proces P4 ještě 8 jednotek. Pokud je plánování CPU prováděno algoritmem Round Robin (RR) s časovým kvantem 10 jednotek, jaký proces bude mít k dispozici procesor v časovém okamžiku 38 jednotek od současného okamžiku, kdy se procesor uvolnil a algoritmus RR bude právě provádět své plánovací rozhodnutí?

A P3
B žádný z těchto procesů
***C** P1
D P2
E P4

---

28 Které číslo ve dvojkové soustavě je ekvivalentem čísla vyjádřeného v šestnáctkové (hexadecimální) soustavě jako 4B6A?

A 0101 0110 1101 0010
***B** 0100 1011 0110 1010
C 0100 1010 0100 1011
D 19306
E 1010 0110 1011 0100

---

29 Pro řešení problému vzájemného vyloučení se **nevyužívá**:

A instrukce typu TestAndSet
B event
C mutex
***D** jutex
E semafor

---

30 Nutnou podmínkou uváznutí **není**:

A zákaz předbíhání (odebrání)
B vzájemné vyloučení při přístupu ke zdroji
C ponechání si již alokovaného zdroje a čekání na další zdroj (tzv. hold and wait)
***D** existence právě jedné instance každého zdroje
E kruhové čekání
