

Тест по информатика за кандидат-магистри

Дата: 08 октомври 2005

Име, презиме, фамилия: _____

1. Микропрограмите се отнасят към:
 - a. Приложението
 - b. Системното осигуряване
 - c. Апаратното осигуряване
 - d. Езиковото осигуряване
2. Машинните инструкции не могат да бъдат с
 - a. Фиксирана дължина и фиксиран брой операнди
 - b. Променлива дължина и променлив брой операнди
 - c. Фиксирана дължина и променлив брой операнди
 - d. Друго
3. Контекста на даден активен процес не включва
 - a. Контролния блок на процеса в операционната система
 - b. Процесната област от паметта – програма и данни
 - c. Съдържането на регистрите на централния процесор
 - d. Текущото системно време
4. RISC процесорите се характеризират (в сравнение с CISC) с
 - a. По-прост инструкционен конвейер
 - b. По-къси програми на машинен код (в брой инструкции)
 - c. По-малка регистрова памет
 - d. Повече формати за адресиране на паметта
5. Кой от следните TCP/IP протоколи се използва за преобразуване на MAC адреси до IP адреси?
 - a. PING
 - b. ARP
 - c. TELNET
 - d. RARP
6. Подмрежовата маска 255.255.255.240 за мрежа от клас C поражда:
 - a. 2 подмрежи
 - b. 12 подмрежи
 - c. 16 подмрежи
 - d. 30 подмрежи
7. Коя от следните думи не принадлежи на езика на регулярния израз: $(ab+ba)(a+b)^*$
 - a. aaa
 - b. ababba
 - c. babab
 - d. ab
8. Коя от следните думи принадлежи на езика на КДА, зададен със следната функция $\delta: Q \times X \rightarrow Q$: $\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_2$, $\delta(q_1, b) = q_3$, $\delta(q_2, a) = q_3$, $\delta(q_3, a) = q_3$, $\delta(q_3, b) = q_3$, като начално състояние на автомата е q_0 , а заключително – q_3 :
 - a. aab
 - b. baa
 - c. bbba
 - d. и трите принадлежат

9. Във файлова система на ОС LINUX (ext2) информация за свободната дискова памет се съхранява в структура:
- FAT
 - Битова карта (bit map)
 - Списък от блокове с номерата на свободните блокове
 - Каталог
10. Във файлова система на ОС MSDOS информация за свободната дискова памет се съхранява в структура:
- Битова карта (bit map)
 - Списък от блокове с номерата на свободните блокове
 - Каталог
 - FAT
11. Семафорите на Дейкстра са:
- Структура данни за представяне на взаимодействащи процеси
 - Механизъм за синхронизация на взаимодействащи процеси
 - Светлинно устройство за регулиране на движението
 - Механизъм за превключване на контекста на процесите в ядрото
12. При дисциплината за планиране на процеси FCFS се избира:
- Процеса с най-малко време за изпълнение
 - Най-стария процес
 - Процеса с най-висок приоритет
 - Случаен процес
13. Релационният модел борави с понятията:
- Класове, атрибути, кортежи
 - Същности, атрибути, кортежи
 - Същности, таблици, колони
 - Домени, атрибути, кортежи
14. Коя от посочените релационни операции не изисква двете релации да са с еднакви релационни схеми:
- Разлика
 - Обединение
 - Сечение
 - Декартово произведение
15. Аксиомите на Армстронг за функционалните зависимости включват следните правила за извод:
- Рефлексивност, симетричност, транзитивност
 - Рефлексивност, транзитивност, попълнение
 - Рефлексивност, симетричност, попълнение
 - Симетричност, транзитивност, попълнение
16. Релационна схема на една релация включва:
- Името на релацията и имената на атрибутите
 - Името на релацията и имената на домоените
 - Името на релацията и имената на кортежите
 - Имената на домоените и имената на кортежите

17. Какъв е резултатът от изпълнението на програмата?

```
#include <iostream.h>
class Base
{public:
virtual void spec() const
{ cout << "Base\n";
}
void print() const
{cout << "usual\n";
spec();
}
};
class Der1 : public Base
{void spec() const
{cout << "Der1\n";
}
};
class Der2 : public Base
{void spec() const
{cout << "Der2\n";
}
};
class Der3 : public Base
{ void spec() const
{cout << "Der3\n";
}
};
void main()
{Base b, *pb;
Der1 d1; Der2 d2; Der3 d3;
pb = &b; pb->print();
pb = &d1; pb->print();
pb = &d2; pb->print();
pb = &d3; pb->print();
}
```

- | | | | |
|-------|-------|-------|-----------|
| a) | b) | c) | d) грешка |
| usual | usual | Base | |
| Base | Base | usual | |
| usual | usual | usual | |
| Der1 | Base | Der2 | |
| usual | usual | Base | |
| Der2 | Base | Der3 | |
| usual | usual | Der1 | |
| Der3 | Base | Base | |

18. Какъв е резултатът от изпълнението на програмата?

```
#include <iostream.h>
class Base1
{public:
  Base1(int a = 1)
  {n = a;
  x = 1.5;
  cout << "Base1: " << n << ", " << x << endl;
  }
  Base1(const Base1& p)
  {n = p.n;
  x = p.x;
  cout << "Base1.n: " << n << endl
  << "Base1.x: " << x << endl;
  }
  Base1& operator=(const Base1& p)
  {if (this != &p)
  {n = p.n + 15;
  x = p.x + 2.5;
  cout << "Base1.n: " << n << endl
  << "Base1.x: " << x << endl;
  }
  return *this;
  }
private:
  int n;
  double x;
};
class Base2
{public:
  Base2(double b = 1)
  {n = 2;
  y = b;
  cout << "Base2: " << n << ", " << y << endl;
  }
private:
  int n;
  double y;
};
class Base3
{public:
  Base3(double b = 1)
  {n = 30;
  x = b;
  cout << "Base3: " << n << ", " << x << endl;
  }
  Base3(const Base3& p)
  {n = p.n + 25;
  x = p.x + 2.5;
  cout << "Base3.n: " << n << endl
  << "Base3.x: " << x << endl;
  }
```

```

    }
    private:
        int n;
        double x;
};
class Der : Base3, public Base2, protected Base1
{public:
    Der(int x = 10, int y = 20, int z = 30): Base1(x), Base2(y), Base3(z)
    {n = z;
    m = x+y;
    cout << "Der: " << n << ", " << m << endl;
    }
    Der& operator=(const Der& p)
    {if(this != &p)
    {Base1::operator =(p);
    n = p.n;
    m = p.m;
    }
    return *this;
    }
private:
    int n, m;
};
void main()
{Der d1, d2(1, 2), d3(3);
Der d4 = d1;
d3 = d4;
}

```

a.
 Base1: 10,1.5
 Base2: 2,20
 Base3: 30,30
 Der: 30,30
 Base1: 1,1.5
 Base2: 2,2
 Base3: 30,30
 Der: 30,3
 Base1: 3,1.5
 Base2: 2,20
 Base3: 30,305
 Der: 30,23
 Base1.n: 25
 Base1.x: 4
 Base1.n: 25
 Base1.x: 4

b.
 Base3: 30,30
 Base2: 2,20
 Base1: 10,1.5
 Der: 30,30
 Base3: 30,30
 Base2: 2,2
 Base1: 1,1.5
 Der: 30,3
 Base3: 30,30
 Base2: 2,20
 Base1: 3,1.5
 Der: 30,23
 Base3.n: 55
 Base3.x: 32.5
 Base1.n: 10
 Base1.x: 1.5
 Base1.n: 25
 Base1.x: 4

c.
 Base3: 30,30
 Base2: 2,20
 Base1: 10,1.5
 Der: 30,30
 Base3: 30,30
 Base2: 2,2
 Base1: 1,1.5
 Der: 30,3
 Base3: 30,30
 Base2: 2,20
 Base1: 3,1.5
 Der: 30,23
 Base1.n: 25
 Base1.x: 4
 Base1.n: 25
 Base1.x: 4

d.
 error

19. Буквите от стринга "computer", четени отляво надясно, са свързани в двоично наредено дърво. Колко сравнявания ще бъдат необходими, тръгвайки от корена, за да бъде открит връхът, в който е включена буквата "e"?
- 2
 - 3
 - 4
 - 5
20. Кое от следните твърдения не е вярно?
- Предоставянето на достъп до обекта само чрез неговия интерфейс се нарича капсулация
 - Класът е софтуерна конструкция определяща данните на обекта-екземпляр и методите за работа с него
 - Подкласът е специализация на родителския клас
 - Нито едно от изброените
21. При делегиращите езици:
- Наследяването е специален случай на типизация
 - Наследяването в обектите се реализира чрез указатели в други обекти, които отговарят на съобщенията и обикновено не променят състоянието
 - Поделят родителите си при множествено наследяване
 - Има сливане на отношенията is_a и is_a_kind_of понеже са системи с едно ниво
22. Селектор в ООП наричаме:
- Името на метода
 - Името на статичната функция
 - Името на съобщението
 - Обвързването на обекта с процедурата-функция
23. Езикът за програмиране C++ поддържа:
- Предаване на съобщения
 - Статично типизирано динамично обвързване
 - Статично обвързване
 - Никое от изброените
24. Нека е дадена (т.е. оценена от интерпретатора на Scheme) следната дефиниция:
- ```
(define (f lst)
 (cond ((null? lst) '())
 ((null? (cdr lst)) '())
 (else (cons (* (car lst) (- (length lst) 1)) (f (cdr lst))))))
```
- Каква е оценката на израза (f '(2 4 0 3)):
- (6 8 0)
  - (6 8)
  - (6 12 0)
  - (0 8 6)

25. Нека е дадена (т.е. оценена от интерпретатора на Scheme) следната поредица от дефиниции:

```
(define (f p l)
 (cond ((null? l) l)
 ((p (car l)) (cons (car l) (f p (cdr l))))
 (else (f p (cdr l))))))

(define (g l n)
 (f (lambda (x) (< (length x) n)) l))

(define l '((a b c) (a b d e) (a c d e f) (a d)))
```

Каква е оценката на израза (map (lambda (x) (cons 'g x)) (g l 4)):

- ((g a d))
  - ((g a d) (g a b c))
  - ((g a b c) (g a d))
  - ((g a b c) (g a b d e) (g a d))
26. Даден е ориентиран граф, представен чрез поредица от факти на Prolog от вида arc(<Node1>, <Node2>), всеки от които означава, че в графа съществува дъга с начало <Node1> и край <Node2>:

```
arc(s,a). arc(s,b). arc(s,c). arc(a,d). arc(a,e).
arc(b,f). arc(c,g). arc(d,g). arc(e,i). arc(f,g).
arc(i,j). arc(j,g).
```

Дадена е също така поредица от факти на Prolog от вида h(<Node>, <Cost>), дефиниращи евристичната функция, с помощта на която се пресмята приближена стойност <Cost> на разстоянието от върха <Node> до върха "g":

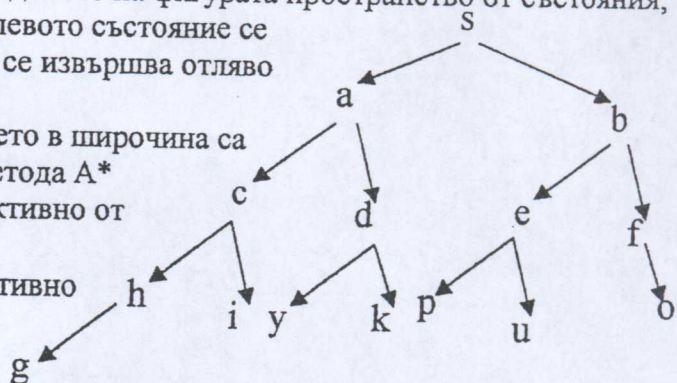
```
h(a,1). h(b,2). h(c,3). h(d,3). h(e,3).
h(f,2). h(g,0). h(i,1). h(j,1). h(s,5).
```

Ако задачата е да се намери път от върха "s" до върха "g" по метода Best-first search, какво ще бъде полученото решение:

- s-a-d-g
- s-a-e-i-j-g
- s-b-f-g
- s-c-g

27. Кое от следните твърдения е вярно за даденото на фигурата пространство от състояния, ако търсенето започва от възела "s", целевото състояние се представя от възела "g", а обхождането се извършва отляво надясно?

- Търсенето в дълбочина и търсенето в широчина са по-ефективни от търсенето по метода A\*
- Търсенето в дълбочина е по-ефективно от търсенето в широчина
- Търсенето в широчина е по-ефективно от търсенето в дълбочина
- Търсенето по метода A\* е по-ефективно от търсенето в дълбочина и търсенето в широчина



28. Дадени са следните факти на Prolog:

parent(pam,bob).  
parent(tom,bob).  
parent(tom,liz).  
parent(bob,ann).  
parent(bob,pat).  
parent(pat,sam).

Предполага се, че  $\text{parent}(P,Ch)$  описва релацията "P е родител на Ch". За коя от следните дефиниции на предиката  $\text{predecessor}(Pr,Succ)$ , описващ релацията "Pr е предшественик на Succ", ще се получи даденият по-долу резултат?

| ?- predecessor(X,sam).

X = pam ? ;

X = tom ? ;

X = bob ? ;

X = pat ? ;

no

- $\text{predecessor}(X,Z):-\text{predecessor}(X,Y),\text{parent}(Y,Z).$   
 $\text{predecessor}(X,Z):-\text{parent}(X,Z).$
- $\text{predecessor}(X,Z):-\text{parent}(X,Z).$   
 $\text{predecessor}(X,Z):-\text{predecessor}(X,Y),\text{parent}(Y,Z).$
- $\text{predecessor}(X,Z):-\text{parent}(X,Z).$   
 $\text{predecessor}(X,Z):-\text{parent}(X,Y),\text{predecessor}(Y,Z).$
- $\text{predecessor}(X,Z):-\text{parent}(X,Y),\text{predecessor}(Y,Z).$   
 $\text{predecessor}(X,Z):-\text{parent}(X,Z).$

29. Каква задача решава предикатът  $\text{guess}/3$  със следната дефиниция:

$\text{mystery}(X,[X|_]).$

$\text{mystery}(X,[_|T]):-\text{mystery}(X,T).$

$\text{guess}([],X,[]).$

$\text{guess}([X|T],Y,Z):-\text{mystery}(X,Y),!,\text{guess}(T,Y,Z).$

$\text{guess}([X|T],Y,[X|Z]):-\text{guess}(T,Y,Z).$

- Намира обединението на две множества, представени чрез списъци
- Намира разликата на две множества, представени чрез списъци
- Намира сечението на две множества, представени чрез списъци
- Намира декартовото произведение на две множества, представени чрез списъци

30. Кое от следните твърдения не е вярно?

- Експертните системи трябва да могат да работят с несигурни знания
- Механизмът за извод, използван в експертните системи, е независим от областта на приложението
- Някои задачи, типични за експертните системи, изискват извършване на абдуктивен извод
- Експертните системи трябва да могат да извършват извод по аналогия

31. Кое от следните твърдения не е вярно?

- Семантичните мрежи са графи, формулиращи таксономична информация за множество обекти и техните свойства
- Възможно е да се извършва само монотонен извод върху семантични мрежи
- При семантичните мрежи е възможно да се използват дъги с едно и също име, означаващи релации от различни типове
- Формализмът на т. нар. разделени семантични мрежи (partitioned semantic nets) е въведен с цел подобряване на изразителната сила на семантичните мрежи



32. Кое от следните твърдения е вярно?

- Правилото за извод Modus Ponens е теоретична основа на работата на интерпретатора на знанията при системите за представяне и използване на знания с помощта на продукционни правила
- Наименованието "N-търсене" е синоним на правия извод при системите, основани на правила
- Наименованието "Z-търсене" е синоним на обратния извод при системите, основани на правила
- Стратегиите за разрешаване на конфликтите (conflict resolution strategies) са методи за избор на стойност в случай на конфликт при множествено наследяване

33. Дадени са група обекти от т. нар. свят на кубчетата: маса и три кубчета. Ще наричаме едно кубче свободно, ако върху него няма поставено друго кубче. Всяко свободно кубче може да бъде премествано или върху друго свободно кубче, или върху масата. На масата има достатъчно място, където могат да бъдат поставени всички кубчета, т.е. можем да считаме, че масата винаги е свободна за поставяне на кубчета.

Нека в алгоритъма за планиране STRIPS са зададени следните правила за света на кубчетата:

```
rule(puton(X,Y),[clear(X),on(X,Z),clear(Y)],[on(X,Y),clear(Z)],
[clear(Y),on(X,Z)],(nonvar(X),nonvar(Y),nonvar(Z),X\==Y,Y\==Z,X\==Z)).
rule(puton(X,table),[clear(X),on(X,Y)],[on(X,table),clear(Y)],
[on(X,Y)],(nonvar(X),nonvar(Y),X\==Y)).
```

където:

- on(X,Y) описва релацията "кубчето X е върху кубчето Y"
- on(Z,table) описва релацията "кубчето Z е върху масата (table)"
- clear(K) описва релацията "върху обекта K няма друг обект"

Нека е дадено следното начално състояние: [on(a,b),on(b,c),on(c,table),clear(a)] и е дадено целево състояние: [on(a,c)].

Какво ще е крайното състояние, което ще бъде достигнато при изпълнение на алгоритъма STRIPS при така зададените правила и начално и целево състояние?

- [on(c,b),clear(a), on(b,table),on(a,c)]
- [on(c,table), on(b,a),on(a,c),clear(b)]
- [on(c,table),clear(a), on(b,table),on(a,c),clear(b)]
- [on(c,table),on(a,table), on(b,table),on(a,c),clear(b)]

34. При алгоритъма Situation Calculus се извършва търсене в пространството на състоянията по метода:

- Търсене в ширина
- Търсене в дълбочина
- Търсене в ограничена дълбочина
- Итеративно търсене по нива

35. Кой от графичните файлови формати е текстов (ASCII):

- JPEG (\*.jpg)
- 24-битов Bitmap (\*.bmp)
- Drawing Exchange Format (\*.dxf)
- AVI (\*.avi)

36. Какво представлява графичния 3D –ускорител:

- Аритметичен процесор за изчисления с плаваща запетая
- Софтуер за реалистично изобразяване на 3D обекти
- Алгоритъм за отстраняване на невидими елементи от триизмерни сцени
- Нито едно от трите неща споменати по-горе

37. Алгоритмите на Брезенхам служат за:
- Изчертаване на отсечки и окръжности
  - Създаване на светлосенки
  - Определяне на пресечните точки на два многоъгълника
  - За нито една от трите задачи, споменати по-горе
38. Отсичащия екран при алгоритъма на Сайръс – Бек задължително трябва да е с форма на:
- Правоъгълник
  - Квадрат
  - Изпъкнал многоъгълник
  - Елипса
39. Алгоритмите за запълване на многоъгълник чрез вертикални или хоризонтални линии използват:
- Пресмятане на лицето на многоъгълника
  - Сортиране на ръбовете на многоъгълника
  - Случаен избор на вътрешни точки от многоъгълника
  - Подреждане на средите на ръбовете в опашка
40. Алгоритъмът на ван Акен за средната точка не се използва за:
- Изчертаване на елипси
  - Изчертаване на окръжности
  - Изтриване на невидими елементи от дадена сцена
  - Някоя от задачите задачи, споменати по-горе