

Име:	Фак. №	- 1 - / 16
------	--------	------------

СОФИЙСКИ УНИВЕРСИТЕТ "СВ. КЛИМЕНТ ОХРИДСКИ"
ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА

ДЪРЖАВЕН ИЗПИТ
ЗА ПОЛУЧАВАНЕ НА ОКС "БАКАЛАВЪР ПО ИНФОРМАТИКА",
03-04.07.2004 г.

ЧАСТ I (ПРАКТИЧЕСКИ ЗАДАЧИ)

Задача1.{4т.}

Като параметри на командна процедура са зададени различни имена (идентификатори) на потребители на системата. Да се състави командна процедура на езика на командния интерпретатор `bash`, която извежда на стандартния изход:

- 1) имената на онези от тях, които са стартирали сесия,
- 2) броя на сесиите, стартирани от потребителя с идентификатор *student*, ако това име е измежду зададените.

Име:	Фак. №	- 2 - / 16
------	--------	------------

Задача 2.{2 т.}

Ако съдържанието на директория /home е

```
dum.exe
err1
gref.out
mnn.exe
opit.exe
st1
us1.out
```

каква стойност ще изведе на стандартния изход командната процедура (верният отговор да се огради с кръгче)

```
br=2
cd /home
for var in *.exe
do
    echo $var >> fnam
    br=`expr $br + 1`
done
br=`grep out fnam | wc -l`
echo $br
```

a) 2 б) 5 в) 0

Задача 3 { 3 т. }

Даден е следният фрагмент от програма на C:

```
#define CMD "who"
#include <stdio.h>
main( int argc, char *argv[])
{
    int i=2, status;
    if ( fork() ){--i;
        printf("Stoinostta na i e %d\n",i);
    }
    else { i++;
        printf("Stoinostta na i e %d\n",i);
        if( execlp(CMD,CMD,0) == -1 )
            i=i+2; }
}
```

Напишете какво се извежда на стандартния изход:

Име:	Фак. №	- 3 - / 16
------	--------	------------

Задача 4 { 5т. }

Даден е крайният недетерминиран автомат А:

$$A = \langle \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7\}, \{a, b, c\}, q_0, \delta, \{q_4, q_5\} \rangle$$

$$\delta: \quad \delta(q_0, a) = \{q_1, q_3, q_5\} \quad \delta(q_5, c) = \{q_6\}$$

$$\delta(q_0, b) = \{q_5\} \quad \delta(q_6, c) = \{q_7\}$$

$$\delta(q_1, c) = \{q_2\}$$

$$\delta(q_2, b) = \{q_4\} \quad \delta(q_7, a) = \{q_7\}$$

$$\delta(q_3, b) = \{q_3, q_4\} \quad \delta(q_7, b) = \{q_5\}$$

а) Да се построи краен детерминиран автомат В, еквивалентен на автомата А.

б) Да се състави автоматна граматика, която поражда езика, разпознаван от автомата В.

Име:	Фак. №	- 4 - / 16
------	--------	------------

Задача 5 { 5т. }

Да се провери пълно ли е множеството A:

а) $A = ((L \cap M) \setminus T_1) \cup (S \cap T_1)$

б) $A = (M \setminus S) \cup (L \cap S)$

Задача 6 { 2т. }

Да се посочи броят на двоичните функции $f(x_1, x_2, x_3, \dots, x_n)$, такива че $f(x_1, x_2, x_3, \dots, x_n) = f(x_2, x_1, x_3, \dots, x_n)$.

а) $3 \cdot 2^{n-2}$

б) $2^{2^{n-1}+2}$

в) $2^{3 \cdot 2^{n-2}}$

г) $2^{2^{n-2}}$

Име:	Фак. №	- 5 - / 16
------	--------	------------

Задача 7 {2 т.}

Нека е дадена (т.е. оценена от интерпретатора на Scheme) следната дефиниция:

```
(define (f lst el)
  (define (f-aux lst n res)
    (cond ((null? lst) (reverse res))
          ((eq? (car lst) el) (f-aux (cdr lst) (+ n 1) (cons n res)))
          (else (f-aux (cdr lst) (+ n 1) res))))
  (f-aux lst 1 '()))
```

Напишете оценката на следния израз:

(f '(x y a c d a f d u v d a) 'd) →

Задача 8 {2 т.}

Нека е дадена (т.е. оценена от интерпретатора на Scheme) следната дефиниция:

```
(define (ff l)
  (cond ((null? l) '())
        ((atom? (car l)) (cons (car l) (ff (cdr l))))
        (else (append (ff (car l)) (ff (cdr l))))))
```

Напишете оценката на следния израз:

(ff '(a (b (c d) ((e)) a) ((x (a)) (y)))) →

Задача 9 {1 т.}

Попълнете липсващите изрази в дефиницията на функцията set-diff при условие, че тя връща като резултат разликата $I1 \setminus I2$ на множествата $I1$ и $I2$, представени като списъци от техните елементи:

```
(define (set-diff I1 I2)
  (cond ((null? I1) '())
        ((member (car I1) I2) ..... )
        (else ..... )))
```

Задача 10 {2 т.}

Да се дефинира функция на езика Scheme, която намира старшия коефициент на полином, зададен във вида $((\langle \text{коеф.}_1 \rangle \langle \text{степен}_1 \rangle) (\langle \text{коеф.}_2 \rangle \langle \text{степен}_2 \rangle) \dots (\langle \text{коеф.}_n \rangle \langle \text{степен}_n \rangle))$, където всички коефициенти са различни от 0, степенните показатели са различни помежду си и между тях няма никаква наредба.

Задача 11 {2 т.}

Даден е ориентиран граф, представен чрез поредица от факти на Prolog от вида $\text{arch}(\langle \text{node1} \rangle, \langle \text{node2} \rangle)$, всеки от които означава, че в графа съществува дъга с начало $\langle \text{node1} \rangle$ и край $\langle \text{node2} \rangle$. Допишете липсващите части от дефинициите на отношенията depth_bound и extend_b при условие, че с помощта на отношението depth_bound_search се решава задачата за търсене на път Path от даден начален възел Start да целевия възел Goal, като се реализира алгоритъмът за търсене в дълбочина до определено ниво Depth:

```
depth_bound_search(Depth, Start, Goal, Path):-
  depth_bound(Depth, [[Start]], Goal, L), reverse(L, Path).
depth_bound(_, [[Goal|Path]]_, Goal, [Goal|Path]).
depth_bound(Depth, [Path|Queue], Goal, FinalPath):-
  extend_b(Depth, Path, NewPaths),
  append( ..... , NewQueue),
  depth_bound(Depth, NewQueue, Goal, FinalPath).
extend_b(Depth, [Node|Path], NewPaths):- length(Path, Len), ..... , !,
  ..... .
extend_b(_, [], []).
```

Задача 12 {3 т.}

Даден е ориентиран граф, представен на фигурата и описан чрез поредица от факти на Prolog от вида $\text{arc}(\langle \text{Node1} \rangle, \langle \text{Node2} \rangle, \langle \text{Cost} \rangle)$, всеки от графа съществува дъга с начало $\langle \text{Node1} \rangle$ и дължина (цена) $\langle \text{Cost} \rangle$.

които означава, че в $\langle \text{Node1} \rangle$, край

Дадена е също така вида

евристичната пресмята разстоянието от

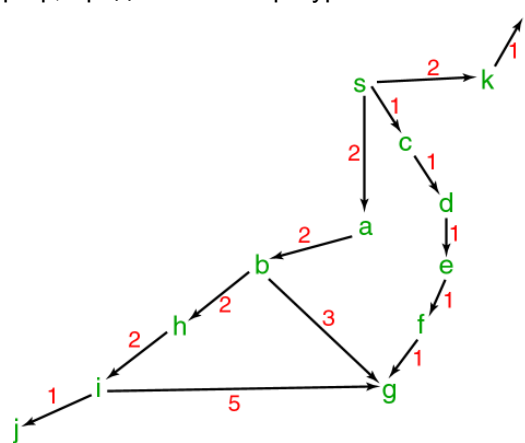
$h(a,2).$

$h(d,3).$

$h(g,0).$

$h(j,6).$

$h(s,4).$



поредица от факти на Prolog от $h(\langle \text{Node} \rangle, \langle \text{Cost} \rangle)$, дефиниращи функция, с помощта на която се приближена стойност $\langle \text{Cost} \rangle$ на върха $\langle \text{Node} \rangle$ до върха "g":

$h(b,3).$ $h(c,4).$

$h(e,2).$ $h(f,1).$

$h(h,4).$ $h(i,5).$

$h(k,5).$ $h(l,6).$

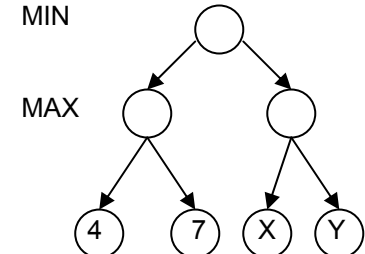
Ако се търси път в графа от върха "s" до върха "g", попълнете следната таблица:

Стратегия за търсене	Намерен път	Брой обходени възли
Hill Climbing		
Beam search (широчина на лъча BeamSize = 3)		
A* search		

Задача 13 {1 т.}

На фигурата е дадено игрово дърво.

При какви оценки на възела X може да бъде отсечен възелът Y?



Базата от правила и работната памет (базата от факти) на една система, основана на правила, са описани със средствата на езика Prolog както следва:

```
% Дефиниции на използваните оператори
:- op(900,fx,if).
:- op(890,xfx,then).
:- op(880,xfy,or).
:- op(870,xfy,and).
% Правила
if can_fly(X) then bird(X).
if can_swim(X) and has_gills(X) then fish(X).
if gives_milk(X) or has_hair(X) then mammal(X).
if mammal(X) and can_speak(X) then human(X).
if fish(X) then can_not_speak(X).
if bird(X) and can_speak(X) then parrot(X).
if can_speak(X) then intelligent(X).
% Факти (работна памет)
fact(has_gills(bobo)).
fact(has_hair(gogo)).
fact(can_fly(pepo)).
fact(can_swim(bobo)).
fact(can_speak(gogo)).
```

Име:	Фак. №	- 7 - / 16
------	--------	------------

fact(can_speak(перо)).

Предполага се, че интерпретаторът на правилата извършва прав извод (извод, управляван от данните), като на всяка стъпка от работния си цикъл записва заключението на избраното за изпълнение правило като нов факт в края на работната памет. В конфликтното множество се включват само правила, при чието изпълнение се записват нови факти в работната памет.

Задача 14 {2 т.}

Какво ще съдържа работната памет след завършване на работата на интерпретатора на правилата, ако на всяка стъпка от работния си цикъл той избира и изпълнява първото правило от конфликтното множество?

Задача 15 {2 т.}

Какво ще съдържа работната памет след завършване на работата на интерпретатора на правилата, ако на всяка стъпка от работния си цикъл той избира и изпълнява това правило от конфликтното множество, чието условие се удовлетворява от най-скоро записан в работната памет факт?

Задача 16 {3 т.}

Даден е ориентиран граф, чийто възли са именувани с уникални идентификатори. Графът е описан чрез средства на езика Scheme като асоциативен списък, в който ключове са имената на възлите, а асоциираната с даден ключ стойност е списък от имената на преките наследници на съответния възел, например:

```
(define graph '((a b c d) (b e f g) (c g h) (d f h i j)
               (h k l) (i k l m) (j m n)))
```

Функцията hill-climbing_search, която е дефинирана по-долу, връща като резултат път между два възела start и goal от графа, като за целта реализира метода Hill Climbing (метода на най-бързото изкачване).

Предполага се, че оценката на всеки път в графа съвпада с оценката на последния възел от пътя, за по-добри се смятат по-големите оценки и оценките на възлите се съдържат в асоциативен списък, всеки елемент на който е списък от името на възел от графа и съответната му оценка, например:

```
(define h '((a 1) (b 3) (c 2) (d 4) (e 3) (f 2) (g 5) (h 5) (i 6) (j 4) (k 3) (l 7) (m 6) (n 5)))
```

Следва дефиницията на функцията hill-climbing_search:

```
(define (hill-climbing_search start goal)
  (reverse (hill-climbing (list start) goal)))
(define (hill-climbing path goal)
  (if (eq? (car path) goal)
      path
      (let ((paths (extend path)))
        (if (null? paths)
            #f
            (let ((best-succ (best paths)))
              (if (better? best-succ path)
                  (hill-climbing best-succ goal)
                  #f))))))
(define (extend path)
  (map (lambda (x) (cons x path)) (succs (car path))))
(define (succs node) (cdr (assq node graph)))
```

Име:	Фак. №	- 8 - / 16
------	--------	------------

Да се дефинират цитираните в горната дефиниция функции best и better? при условие, че (best paths) връща като резултат най-добрия път от даден списък от пътища paths, а (better? path1 path2) проверява дали пътят path1 е по-добър от пътя path2.

Име:	Фак. №	- 9 - / 16
------	--------	------------

Задача 17 {2 т.}

Какъв ще бъде резултатът от изпълнението на следния програмен фрагмент?

```
int A[100], j ;
for ( j = 1 ; j <= 10 ; j++ ) A[ j ] = j ;
for ( j = 2 ; j <= 8 ; j++ ) A[ j + 2 ] = A [ j-1 ] ;
for ( j = 3 ; j <= 7 ; j++ ) A[ 10 ] += A [ j+3 ] ;
cout << A[ 10 ] ;
```

- a) 10 b) 20 c) 30 d) 40

Задача 18 {2 т.}

При следните дефиниции на функциите f2 и f3

```
void f3(int);
void f2(int x)
{ cout << x << " "; if (x) { if (x%2) f2 (abs(x-1)); else f3 (abs(x-2)); } }
void f3(int x)
{ cout << x << " "; if (x) { if (x%3) f3 (abs(x-1)); else f2 (abs(x-3)); } }
```

резултатът от изпълнението на обръщението f2(-3) ще бъде:

Задача 19 {2 т.}

При следната дефиниция на функцията f

```
int f( int x, int *y)                      { return x++ + (*y)++;}
```

резултатът от изпълнението на програмния фрагмент

```
int x,y; x=y=1;
cout <<"\n"<< f(x, &y) +f(y, &x) + x + y;    //    ще бъде: .....
```

Задача 20 {2 т.}

Функцията bool differ(int n, int* a) трябва да проверява дали редицата a_0, a_1, \dots, a_{n-1} се състои от различни елементи, а member(int x, int n, int* a) – дали елементът x принадлежи на редицата a_0, a_1, \dots, a_{n-1} . Попълнете липсващите части, означени с многоточия, във функцията differ.

```
bool differ(int n, int* a)
{if (n == 1) return true;
return !member(....., ....., a+1) ..... differ(....., .....);
}
```

Задача 21 {9 т.}

Шаблонът на класа LList реализира свързан списък, представен чрез една връзка.

```
template <class T>
struct elem_link1
{ T inf;
  elem_link1<T> *link;
};
template <class T>
class LList
{public:
  LList();
  ~LList();
  LList(LList const&);
  LList& operator=(LList const &);
  void IterStart(elem_link1<T>* = NULL);
  elem_link1<T>* Iter();
  void DeleteElem(elem_link1<T> *p, T & x);    // изтрива сочения от р елемент и го записва в x
  .....
```

Име:	Фак. №	- 10 - / 16
------	--------	-------------

```
private:
    elem_link1<T> *Start,           // указател към началото
                      *End,         // указател към края
                      *Current;     // итератор
```

```
    ...
```

```
};
```

а) {1 т.} Член-функцията

```
void IterStart(elem_link1<T>* = NULL);
```

установява итератора в указания чрез параметъра адрес, ако той е ненулев, и в началото на списъка, в противен случай. Реализирай те я!

б) {1 т.} Член-функцията

```
elem_link1<T>* Iter();
```

премества итератора в следваща позиция (ако е възможно) като връща предишната му позиция. Реализирайте я!

в) {4 т.} Нека

```
typedef Llist<int> IntList;
```

Като използвате член-функциите IterStart, Iter, DeleteElem на класа IntList реализирайте външната функция

```
void deletefirst(int a, IntList& L);
```

която изтрива първото срещане на елемента *a* в списъка от цели числа *L*.

г) {3 т.} Нека

```
typedef Llist<int> IntList;
```

Като използвате член-функциите IterStart, Iter, DeleteElem на класа IntList реализирайте *рекурсивно* външната функция:

```
bool member(int x, IntList L);
```

която проверява дали елементът *x* принадлежи на списъка от цели числа *L*.

Задача 22 {6 т.}

Като използвате шаблона на класа tree, реализиращ двоично дърво с елементи от тип T:

```
template <class T>
struct node
{
    T inf;
    node *Left,
        *Right;
};
template <class T>
class tree
{
public:
```

```

tree();
~tree();
tree(tree const&);
tree& operator=(tree const&);
bool operator==(tree<T> const& t2);
bool empty() const;           // проверява дали дървото е празно
T RootTree() const;          // осъществява достъп до корена
tree LeftTree() const;       // осъществява достъп до лявото поддърво
tree RightTree() const;      // осъществява достъп до дясното поддърво
void print_tree() const
{pr(root);
 cout << endl;
}

.....
private:
 node<T> *root;
 void pr(const node<T> *) const;
.....
};

```

a) {3 т.} Предефинирайте оператора == чрез шаблон на функция-приятел на шаблона на класа tree.

б) {3 т.} Дефинирайте член-функция на шаблона на класа tree, която извежда на екрана листата на двоично дърво.

Задача 23 {6 т.} Какъв е резултатът от изпълнението на програмата?

```

#include <iostream.h>
class base
{public:
 base(int x = 1)
 {b = x;}
 base& operator=(const base &x)
 {if (this!=&x) b = x.b + 1;
  return *this;
 }
 void Print() const
 {cout << "base: " << b << endl;
 }
 private: int b;
};
class der1 : public base
{public:
 der1(int x = 1) {d = x;}
 der1& operator=(const der1& x)
 {if (this!=&x)

```

```
{d = x.d + 2;
  base::operator=(x);
} return *this;
}
void Print() const
{cout << "der1: " << d << endl;
  base::Print();
}
private: int d;
};
class der2 : public base
{public:
  der2(int x = 2) : base(x+3)
  {d = x;
  }
  der2(const der2& p)
  {d = p.d + 2;
  }
  der2& operator=(const der2& x)
  {if(this !=&x)
  {d = x.d + 3;
    base::operator=(x);
  } return *this;
  }
  void Print() const
  {cout << "der2: " << d << endl;
    base::Print();
  }
  private: int d;
};
class der3 : public der2
{public:
  der3(int x = 3)
  {d = x;
  }
  der3(const der3& p) : der2(p)
  {d = p.d+3;
  }
  void Print()
  {cout << "der3: " << d << endl;
    der2::Print();
  }
  private: int d;
};
void main()
{der1 d11(2), d12=d11;
  der2 d21(1), d22;
  der3 d31(3), d32=d31;
  cout << "d11: "; d11.Print();
  cout << "d12: "; d12.Print();
  d12 = d11; cout << "d12: "; d12.Print();
  cout << "d21: "; d21.Print();
  cout << "d22: "; d22.Print();
  d22 = d21; cout << "d22: "; d22.Print();
  cout << "d31: "; d31.Print();
  cout << "d32: "; d32.Print();
  d32 = d31; cout << "d32: "; d32.Print();
}
```

Задача 24 {8 т.} Задраскайте грешните линии на програмата. Какъв е резултатът от изпълнението на програмата след отстраняване на грешките в нея? (Напишете резултата от страни на програмата).

```
#include <iostream.h>
class Base
{public:
    virtual void func1()
    { cout << "func1() \n";
    }
    void help()
    { cout << "help()\n";
      func2();
      func1();
      func3();
    }
private:
    virtual void func2()
    { cout << "func2()\n";
    }
protected:
    virtual void func3()
    { cout << "func3()\n";
    }
};
class Der : public Base
{virtual void func1()
 { cout << "Der-class\n";
 }
protected:
    virtual void func2()
    { cout << "Der-func2()\n";
    }
public:
    virtual void func3()
    { cout << "Der-func3()\n";
    }
};
void main()
{Base b; Der d;
  Base *p = &b;
  b.func1();
  p->func1();
  p->func2();
  p->func3();
  Base *q = &d;
  q->func1();
  q->func2();
  q->func3();
  p->Base::func1();
  Der *r = new Der;
  r->func2();
  r->func1();
  r->func3();
  p->help();
  q->help();
  r->help();
}
```

Задача 25 {3 т.}

Да означим с F_1 множеството на едноместните частични функции в естествените числа. Нека Γ_i , $i=1, 2, 3$, са следните оператори в F_1 :

$$\Gamma_1(f)(x) \equiv \text{if } x \equiv 0 \pmod{2} \text{ then } f(x+1) \text{ else } f(x-1)$$

$$\Gamma_2(f)(x) \equiv f(0) + \dots + f(x)$$

$$\Gamma_3(f)(x) \equiv \text{if } f(x) \equiv 0 \text{ then } 0 \text{ else } 1.$$

Кое/кои от следващите твърдения са верни:

- а) само Γ_1 е компатктен;
- б) само Γ_1 и Γ_2 са компатктни;
- в) всички оператори са компактни;
- г) всяко едно от горните три твърдения не е вярно.

Задача 26 {2т.}

Дадени са операторите:

$$\Gamma_1(f)(x) \equiv \text{if } x \leq 1 \text{ then } x \text{ else } 2.f(x-1) + 3.f(x-2)$$

$$\Gamma_2(f)(x) \equiv \text{if } x \equiv 0 \pmod{2} \text{ then } (f(x/2))^2 \text{ else } (f((x-1)/2))^2$$

$$\Gamma_3(f)(x) \equiv \text{if } x \equiv 0 \pmod{2} \text{ then } x/2 \text{ else } f(3x+1).$$

Нека f_{Γ_i} е най-малката неподвижна точка на оператора Γ_i , $i=1, 2, 3$. Кое/кои от следващите твърдения са верни:

- а) само f_{Γ_1} е тотална;
- б) само f_{Γ_1} и f_{Γ_2} са тотални;
- в) и трите функции f_{Γ_1} , f_{Γ_2} и f_{Γ_3} са тотални;
- г) нито едно от горните три твърдения не е вярно.

Задача 27 {4 т.}

Нека R е следната рекурсивна програма в типа данни Nat :

$R: F(X, Y) \text{ where}$

$$F(X, Y) = \text{if } X = 0 \text{ then } Y \text{ else } F(X-1, G(Y))$$

$$G(X) = \text{if } X \leq 1 \text{ then } X \text{ else } G(X-2).$$

Да означим с $D_V(R)$ семантиката на R с предаване на параметрите по стойност. Кое/кои от следващите твърдения за $D_V(R)$ са верни:

- а) $\forall x \forall y (D_V(R)(x, y) \Rightarrow D_V(R)(x, y) \leq 1)$;
- б) $\forall x \forall y (D_V(R)(x, y) \Rightarrow D_V(R)(x, y) \equiv y \pmod{2})$;
- в) $\forall x \forall y (x > 0 \Rightarrow \neg D_V(R)(x, y))$;
- г) $\forall x \forall y (\neg D_V(R)(x, y) \Rightarrow x > 0)$.

Задача 28 {3т.}

Коя/кои от изброените формули са тавтологически верни (т.е. са тавтологии):

- а) $((\forall X p(X)) \Rightarrow (\forall X q(X))) \Rightarrow \exists X(p(X) \Rightarrow q(X))$;
- б) $(\forall X \exists Y p(X, Y)) \Rightarrow (\exists Y \forall X p(X, Y))$;
- в) $(\forall X (p(X) \vee q(X))) \Leftrightarrow ((\forall X p(X)) \vee (\forall X q(X)))$;
- г) $(p \Rightarrow (q \Rightarrow r)) \Leftrightarrow ((p \wedge q) \Rightarrow r)$.

Задача 29 {2т.}

Коя/кои от изброените субституции е най-общ унификатор за $f(X, f(Y, V))$ и $f(g(Z), Z)$:

- а) $[X | g(f(Y, Y)), Z | f(Y, Y)]$;
- б) $[X | g(f(Y, Y)), Z | f(Y, Y), V | Y]$;
- в) $[X | g(f(Y, V)), Z | f(Y, V)]$;
- г) $[X | g(f(W, V)), Y | W, Z | f(W, V)]$.

Задача 30 {6 т.}

В базата данни със схема:

Classes (class, type, country, numGuns, bore, displacement)

Ships (name, class, launched)

Battles (name, date)

Outcomes (ship, battle, result)

се съхранява информация за кораби и тяхното участие в битки по време на Втората Световна Война.

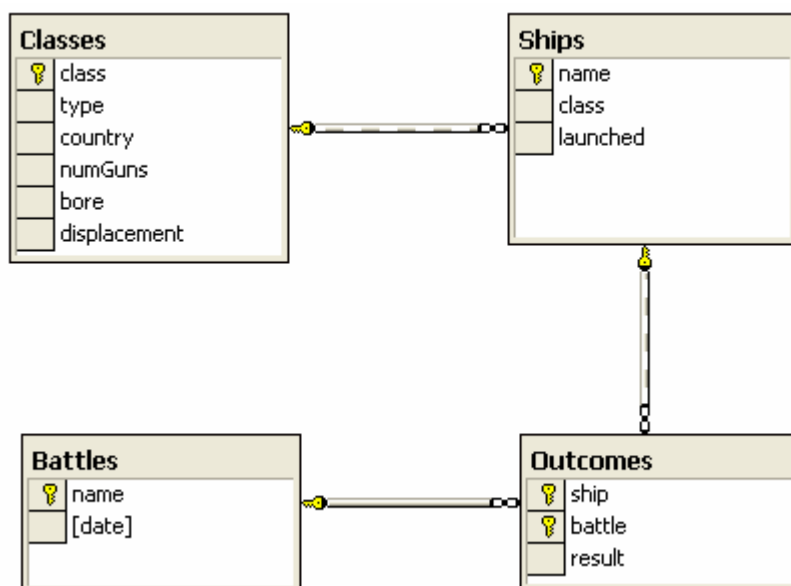
Всеки кораб е построен по определен стереотип, определящ класа на кораба. Обикновено класът носи името на първия построен кораб от този клас.

Таблицата **Classes** съдържа информация за име на класа, тип ('bb' за бойни кораби и 'bc' за бойни крайцери), страната, която строи такива кораби, броя на основните оръдия, калибъра им (диаметъра на отвора на оръдието в инчове) и водоизместимостта (тегло в тонове).

Таблицата **Ships** съдържа информация за име на кораб, име на неговия клас и годината, в която корабът е пуснат на вода.

Таблицата **Battles** съхранява имена и дати на провеждане на битки.

Таблицата **Outcomes** съдържа информация за резултатата от участието на даден кораб в дадена битка (потънал - 'sunk', повреден - 'damaged', победил - 'ok').



1. {2 т.} Посочете заявката, изтриваща от таблицата Outcomes всички записи, в които участва кораб, пуснат на вода по-късно от дата на провеждане на битката.

- DELETE FROM Outcomes A
INNER JOIN Ships B ON A.ship=B.name
INNER JOIN Battles C ON C.name=A.battle
WHERE B.launched > C.date
- DELETE FROM Outcomes
WHERE ship IN (SELECT s.name
FROM Ships s, Outcomes o, Battles b
WHERE s.launched > b.date AND
b.name=o.battle AND
o.ship=s.name)
- DELETE FROM Outcomes
WHERE (SELECT date
FROM Battles
WHERE name=Outcomes.battle) < (SELECT launched
FROM Ships
WHERE name=Outcomes.ship)

Име:	Фак. №	- 16 - / 16
------	--------	-------------

d) DELETE FROM Outcomes
WHERE ship IN (SELECT ship
FROM Outcomes
JOIN Ships ON Ships.name = Outcomes.ship
JOIN Battles ON Battles.name = Outcomes.battle
WHERE Ships.launched > Battles.date)
AND
battle IN (SELECT battle
FROM Outcomes
JOIN ships ON Ships.name = Outcomes.ship
JOIN battles ON Battles.name = Outcomes.battle
WHERE Ships.launched > Battles.date)

2. {2 т.} Посочете заявката, която извежда имената на всички кораби с по-голям брой оръдия от оръдията на кораба *Yamato*.

- a) SELECT name
FROM Ships s, Classes c
WHERE s.class=c.class AND
numGuns <(SELECT numGuns
FROM Ships s, Classes c
WHERE s.class=c.class AND name='Yamato')
- b) SELECT s.name
FROM Ships s,Classes c
WHERE s.class=c.class AND
c.numGuns > (SELECT numGuns
FROM classes
WHERE class='Yamato')
- c) SELECT s.name
FROM Ships s,Classes c,Classes Yamato
WHERE s.class=c.class AND
Yamato.class='Yamato' AND
c.numGuns > Yamato.numGuns
- d) SELECT s.name
FROM Ships s,Classes c, Ships ys, Classes yc
WHERE s.class=c.class AND
ys.name='Yamato' AND
ys.class=yc.class AND
c.numGuns > yc.numGuns

3. {2 т.} Посочете заявката, която намира в колко битки е участвала всяка страна.

- a) SELECT c.country,COUNT(DISTINCT o.battle)
FROM Classes c
LEFT OUTER JOIN Ships s ON c.class=s.class
LEFT OUTER JOIN Outcomes o ON o.ship=s.name
GROUP BY c.country
- b) SELECT c.country, COUNT(battle)
FROM Classes c
LEFT OUTER JOIN (Ships s
LEFT OUTER JOIN Outcomes o ON o.ship=s.name) ON c.class=s.class
GROUP BY c.country
- c) SELECT c.country, COUNT(*)
FROM Classes c, Outcomes o, Ships s
WHERE c.class = s.class AND s.name = o.ship
GROUP BY c.country
- d) SELECT c.country, SUM(1)
FROM Classes c, Outcomes o, Ships s
WHERE c.class = s.class AND s.name = o.ship
GROUP BY c.country