Изпит по СДП 04.07.2012 г.

Име ..... **Група**..... **Фак. №** 20101

Задача 1. Да се открие и коригира грешката в следния фрагмент от дефинирането на клас Array и на предефинираната функция за равенство:

```
class Array {
public:
. . .
int operator==(const Array &) const;
. . .
private:
int *ptr; // указател към първия елемент на масива
int size; // брой на елементите на масива
};
int Array::operator==(const Array &right) const
{ for(int i=0; i<size; i++)
if (ptr[i] != right.ptr[i])
return 0;
return 1; •
}
```

Задача 2. Намерете и задраскайте грешните оператори на програмата. Обяснете в какво се състои грешката. Какъв е резултатът от изпълнението на програмата след отстраняването (задраскването) на грешките в нея? Напишете резултата отстрани 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 3. Какъв ще бъде резултатът от изпълнението на следната програма. Напишете резултата отстрани на програмата.

```
#include <iostream.h>
class KLAS
{private:
int a, b;
public:
KLAS(int=0, int=1);
KLAS(const KLAS&);
KLAS& operator=(const KLAS&);
~KLAS( );
void print( ) const;
};
KLAS::KLAS(int x, int y)
{a = x;
b = y;
cout << "Конструктор \n";
}
KLAS::KLAS(const KLAS& r) : a(r.a+3),
b(r.b+5)
{cout << "Конструктор за присвояване \n";
}
KLAS& KLAS::operator=(const KLAS& p)
{cout << "Оператор =\n";
```

```
~KLAS( )
{cout << "Деструктор( )\n";
}
void KLAS::print( ) const
{cout << a << " " << b << endl;
}
void main( )
{KLAS a, b(2), c(3,4), d(c);
a.print( ); b.print( ); c.print( );
KLAS f = d; f.print( );
KLAS g(c); g.print( );
d = g; d.print( );
}
```

Задача 4. Напишете резултата от изпълнението на програмата. Обяснете как сте получили отговора.

```
#include <iostream.h>
class CPolygon {
 protected:
  int width, height;
 public:
  void set_values (int a, int b)
   { width=a; height=b; }
  virtual int area ()
   { return (0); }
};
class CRectangle: public CPolygon {
 public:
  int area ()
   { return (width * height); }
};
class CTriangle: public CPolygon {
 public:
  int area ()
   { return (width * height / 2); }
};
int main () {
 CRectangle rect;
 CTriangle trgl;
 CPolygon poly;
 CPolygon * ppoly1 = &rect;
 CPolygon * ppoly2 = &trgl;
 CPolygon * ppoly3 = &poly;
 ppoly1->set_values (4,5);
 ppoly2->set_values (4,5);
 ppoly3->set_values (4,5);
 cout << ppoly1->area() << endl;
 cout << ppoly2->area() << endl;
 cout << ppoly3->area() << endl;
 return 0;}
```

Задача 5. Функцията push добавя елемент в свързан стек, а главната функция я изпълнява. Многоточието да се замени с липсващия оператор в следния фрагмент. Обяснете как сте получили отговора.

```
struct stack_el
{ int info;
stack_el *link;};
void push(stack_el **t, int x)
{ stack_el *p;
if ((p=new stack_el) == NULL)
{ cout << "\nНяма свободна памет\n";
exit(1);
}
p->info = x;
. . .
*t = p;
}
void main()
{ . . .
push(&stack, 1);
push(&stack, 2);
}
```

Задача 6. Шаблонът на класа BinOrdTree реализира двоично наредено дърво от тип T.

```
template <class T>
struct node_bin
{T inf;
node_bin<T> *Left;
node_bin<T> *Right;};
template <class T>
class BinOrdTree
{public:
BinOrdTree();
~BinOrdTree();
BinOrdTree(BinOrdTree<T> const&);
BinOrdTree<T>& operator=(BinOrdTree<T>
const&);
bool empty()const; //проверява дали дървото е
празно
T RootTree() const; // връща корена на двоично
наредено дърво
BinOrdTree LeftTree() const; // връща лявото
поддърво
BinOrdTree RightTree() const; // връща дясното
поддърво
```

```
void AddNode(T const & x) // включва указания
element
{ Add(root, x);
void DeleteNode(T const&); // изтрива указания
element
void Create3(T const&,BinOrdTree<T> const&,
BinOrdTree<T> const&);
// ...
private:
node_bin<T> *root;
void DeleteTree(node_bin<T>* &) const; //
изтрива двоично наредено дърво
void Copy(node_bin<T> * &, node_bin<T>*
const&) const;
void CopyTree(BinOrdTree<T> const&); //
копира указаното в неявното двоично наредено
дърво
void Add(node_bin<T> *&, T const &) const;
// .....
};
```

Функцията-елемент void AddNode(T const & x) на шаблона на класа включва указания като параметър елемент в неявното двоично наредено дърво. Тя използва помощната функция-елемент

oid BinOrdTree<T>::Add(node_bin<T>* &p, T onst & x) const, която включва елемента x в воично наредената дърво, зададено чрез казателя p. Попълнете липсващите части, начени с многоточия, в дефиницията й по-лу.

nplate <class T>
d BinOrdTree<T>::Add(node_bin<T>* &p, T st & x) const
!p)
new node_bin<T>;
nf = ...
.....

```
.....
}
else
if (x < p->inf) .....
else .....
}
```

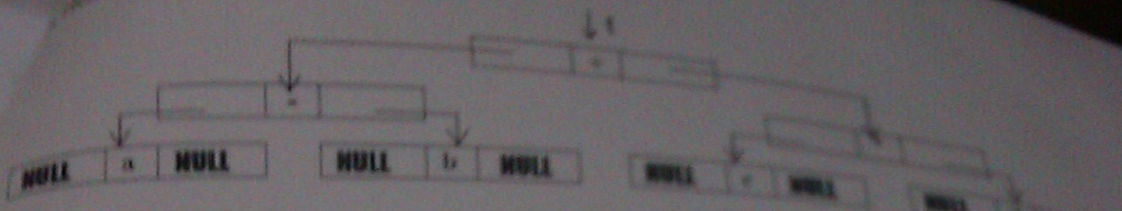
Задача 7. Функцията dequeue изключва елемент от свързана опашка с първи фиктивен елемент, който се сочи от f.

```
struct queue_el
{ float info;
queue_el *link;};
float dequeue(queue_el *f, queue_el **r)
{ float x;
queue_el *p = f->link;
if (p == NULL)
{ cout << "\nОпашката е празна\n";
exit(1); }
. . .
x = p->info;
delete p;
if (f->link == NULL)
. . .
return x;}
void main()
{ float y;
queue_el Q, *F = &Q, *R;
y = dequeue(F, &R);}
```

a) Многоточията във функцията dequeue да се заменят с липсващите оператори. Да се обясни какво правят новите оператори.

b) Да се открият, обяснят и коригират грешките във функцията main .

...ава е дадено двоичното дърво:




```
struct btree_el //тип на елементите във върховете на дървото
{ btree_el *left; //указател към ляво поддърво
char key;
btree_el *right; //указател към дясно поддърво
};
```

Какво ще изведе функцията?

```
void preorder(btree_el *t)
{ if (t)
{ putchar(t->key);
preorder(t->left);
preorder(t->right);
}
}
```

Обяснете как сте получили отговора.