

Имате 45мин. време за работа.
Максималният брой точки са 39т.
За оценка 3 са достатъчни 16т.
Успех!

Result: 17.5/39 - 3

1 Зад. Отговорете на въпросите: (8т.) 6т.

А) Напишете дефинициите за хардуерен, виртуален и абстрактен тип от данни.

Б) Каква е разликата между структурите и класовете.

В структурите, всички член данни се задават като public по подразбиране, докато при класовете са private

В) Кои са компонентите на голямата четворка на клас? В кои случаи за клас задължително се дефинира голяма четворка представяне?

*Конструктор, конструктор за копиране, оператор за присвояване и деструктор;
Голямата четворка е нужна, когато искаме специфично да зададем методите описани в голямата четворка. Нужна е, най-често, когато имаме работа с динамична памет, като част от член данните на обекта от този клас.*

Не е добре формулирано. По-добре да бъде:

Те са нужни, когато искаме специфично поведение, различно от това по подразбиране. Задължително се дефинират при работа с динамична памет, а при нужда - и в други случаи.

Г) Каква е ролята на указателя this?

this сочи към обекта, за когото сме извикали даден метод

Д) Дефинирайте член-функцията operator= на клас.

```
Class& operator = (const Class &other){  
    if( this != &other){  
        //присвояване на член данни  
    }  
    return *this;  
}
```

Е) Каква е разликата между конструкторите за присвояване и за копиране?

Конструкторът за присвояване изтрива данните от вече съществуващ обект и присвоява стойностите от друг обект. Конструкторът за копиране създава нов обект със същите данни като копираният.

Ж) Какви са особеностите в дефинициите на деструкторите?

3) За кои член-функции на класовете не е дефиниран указател this?

За статичните

2 Зад. Намерете резултата от изпълнението на програмата: (8т.) 3,5т.

```
#include <iostream>
#include <cassert>
using namespace std;
class A
{ public:
    A(double = 0.0, char* = "+++");
    ~A();
    A(const A&);
    A& operator=(const A&);
    void print() const;
private:
    double x;
    char* st;
};

A::A(double y, char* s)
{ cout << "A(" << y << ", " << s << ")\\n";
  x = y;
  st = new char[strlen(s)+1];
  assert(st!= NULL);
  strcpy(st, s);
}

A::~~A()
{ cout << "~A()\\n"; delete st;
}

A::A(const A& s)
{ x = s.x; cout << "A(const s)\\n";
  st = new char[strlen(s.st)+1];
  assert(st!= NULL);
  strcpy(st, s.st);
}

A& A::operator=(const A& s)
{ cout << "A::operator=()\\n";
  if (this != &s)
  { x = s.x;
    delete st;
    st = new char[strlen(s.st)+1];
```

```

        assert(st!=NULL);
        strcpy(st, s.st);
    }
    return *this;
}

void A::print() const
{ cout << x << " " << st << endl;
}

class B
{ public:
    B(double, const A&);
    B(const B&);
    B& operator=(const B&);
    void print() const;
private:
    double x;
    A a;
};

B::B(double d, const A& e) : a(e)
{ cout << "B::B(d, e)\n";
  x = d;
}

B::B(const B& p) : a(p.a)
{ cout << "B::B(const p)\n";
  x = p.x;
}

B& B::operator=(const B& p)
{ cout << "B::operator=()\n";
  if (this != &p)
  { x = p.x + 1;
    a = p.a;
  }
  return *this;
}

void B::print() const
{ cout << x << endl;
  a.print();
}

int main()
{ A a1(3.5, "---"), a2(0.5);
  B b(5.0, a1), c(1.0, a2), d(c);
  b.print();
  c.print();
}

```

```

d.print();
d = b;
d.print();
return 0;
}

```

```

A( 3,5, ---)
A( 0.5) A(0,5, +++)
A(const s)
B::B(5, a1) B::B(d, e)
A(const s)
B::B(1, a2) B::B(d, e)
A(const s)
B::B(const c) B::B(const p)
5
3.5 ---
1
0.5 +++)
1
0.5 +++)
B::operator=()
A::operator=()
5 6
3.5 ---
~A()
~A()
~A()
~A()
~A()

```

Ако трябва да сме педантични, ще изведе съобщение за грешка. Липсва библиотеката *cstring*, съответно не могат да се ползват функциите от нея.

3 Зад За шаблона на класа *base*: (4т.) 4т.

```

template <typename T, typename V>
class base
{
    T f1;
    V f2;
public:
    base(T f, V s)
    { f1 = f;
      f2 = s;
    }
};

```

```

        cout << f1 << " " << f2 << endl;
    }
    ~base()
    { cout << "~ base()\n";
    }
};

```

определете коя от дефинициите на обекта *X* е правилна?

A) base X(1,2);

Б) base<double, int> X;

X.f1 = 0.5;

X.f2 = 5;

X В) base<int, char> X(5, 'a');

Г) base <double> X(1.5, 2.3);

Д) всички посочени по-горе

Е) никоя от посочените по-горе.

4.Зад. За йерархията (4 т.) 0т.

```

class base                      class der1 : public base
{ private: int a1;              { private: int a4;
  protected: int a2;            protected: int a5;
  public: int a3();              public: int a6();
};                               };

class der2 : der1               class der3 : protected der2
{ private: int a7;              { private: int a10;
  protected: int a8;            protected: int a11;
  public: int a9();              public: int a12();
};                               };

class der4 : protected der1     class der5 : public der2
{ private: int a13;              { private: int a16;
  protected: int a14;            protected: int a17;
  public: int a15();              public: int a18();
};                               };

```

base b; der4 d4;

der1 d1; der5 d5;

der2 d2;

der3 d3;

определете възможностите за достъп на обектите: b, d1, d2, d3, d4 и d5 до компонентите на класовете.

b: a3()

d1: a6(), a3()

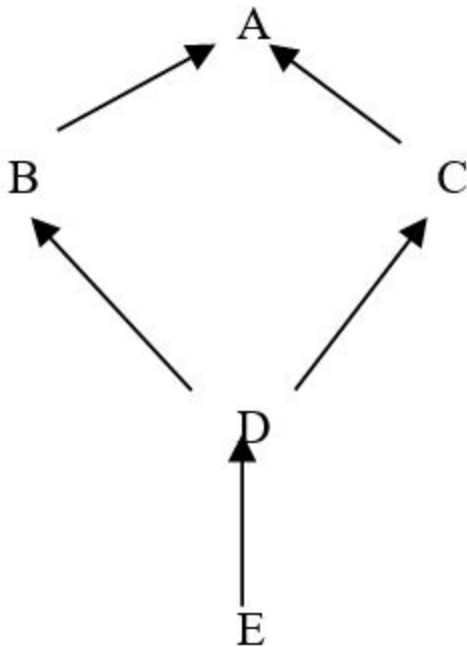
d2: a9()

d3: a12()

d4: a15()

d5: a18(), a9()

5. зад. (7 т.) Нека A е виртуален основен клас и всеки клас на йерархията има компонента с име x. От.



1) Какво е разпределението на паметта за обект от клас E?

E съдържа член-данните на D

D съдържа член-данните на B и на C

B съдържа член-данните на A

C съдържа член-данните на A (същите като на B)

2) Дефинирайте конструктор за присвояване на класа E.

6. Отговорете на въпросите (всеки въпрос носи по 2т.) 4т..

6.1.

```
Имаме следната йерархия
class A{...};
class B : public virtual A{...};
class C{...};
class D : public C, public B{...};
```

Напишете в какъв ред ще се извикат конструкторите при създаване на обект от клас D

(Пример D, C, D, B, A)

C, A, B, D *Не съм особено сигурна тук обаче*

6.2. Чисто виртуална функция може да бъде извикана от класа, в който е декларирана

а) Лъжа **X**

б) Истина

6.3. Кои са методите от каноничната четворка на клас A (голяма четворка)

а) A& operator=() const

б) A& operator=(A&) const

в) A(const A&) **X**

г) A(const A)

д) ~A() **X**

е) A& operator=(A&)

ж) A() **X**

з) A(A&) const

и) ~A(const A&)

й) A(bool)

к) A operator=()

л) Друг (неописан тук)

м) bool operator ==(const A&) const

н) A& operator=(const A&) const

о) A& operator=(const A&) **X**

//6.4. Всеки текстов файл е двоичен файл ??????????????

а) Лъжа

б) Истина *Евентуално, ако се има предвид, че може да се преведе до бинарен код....*

6.5. При чисто множествено наследяване, наследника получава пълни копия на базовите си класове

а) Лъжа *По-скоро това*

б) Истина