

ФИО Холопкин

1	2	3	4	5
+	+	±	±	+

Задание 1. Решить задачу.

Есть ли ошибки в приведенном фрагменте программы на C++? Если есть, то объясните, в чем они заключаются. Ошибочные конструкции вычеркнуть из текста программы. Что будет выведено на экран?

```

class A {
public:
    virtual void f() {
        cout << "A_f ";
    }
    void h(int) {
        cout << "A_h ";
    }
    void g() {
        cout << "A_g ";
    }
};
class B : public A {
public:
    virtual void g() = 0;
    void h() {
        cout << "B_h ";
    }
};
class C: public B {
public:
    void h(int) {
        cout << "C_h(int) ";
    }
    void f() {
        cout << "C_f ";
    }
    void h() {
        cout << "C_h ";
    }
    void g() {
        cout << "C_g ";
    }
};

```

Выход:

A_f A_g A_h
A_h A_g A_h
C_f A_g A_h



```

main() {
    A a;
    B b;
    C c;
    A *pa;
    pa = &a; pa->f(); pa->g(); pa->h(0); pa->h();
    pa = &b; pa->f(); pa->g(); pa->h(0); pa->h();
    pa = &c; pa->f(); pa->g(); pa->h(0); pa->h();
}

```

// надо int main, нельзя не указывать тип возвращаемого значения

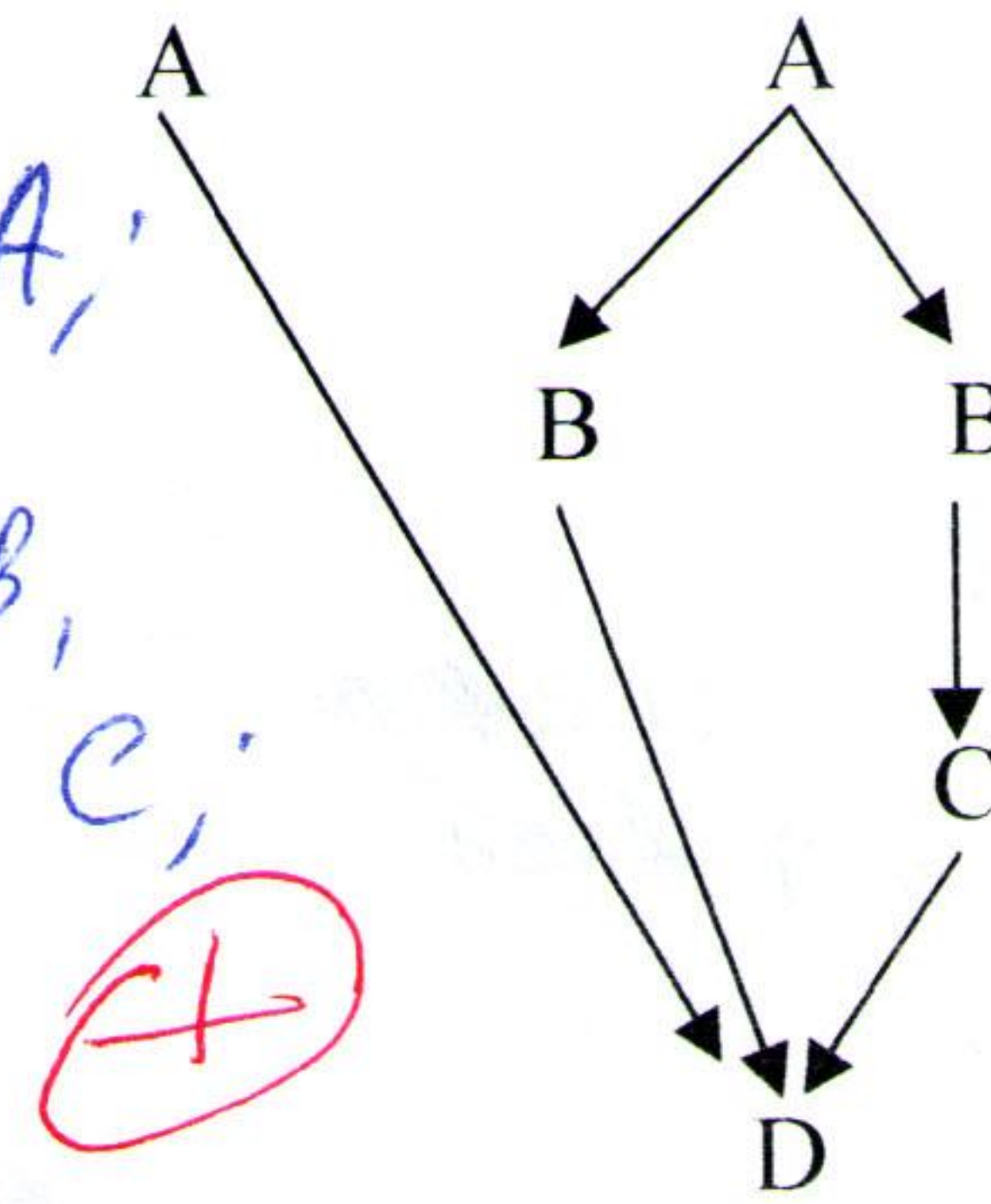
B - абстрактный

err: в A нет функции h без параметров

взятие адреса несуществующего объекта класса

Задание 2. Можно ли построить следующее дерево наследования классов? Если возможно, то написать заголовки классов. Если нет, то объяснить ВСЕ причины невозможности.

```
class A;
class B: virtual public A;
class C: public B;
class D: public A, public B, public C;
```



Задание 3. Написать класс и решить задачу.

Напишите класс Isklyuchish так, чтобы следующая программа выполнялась, и на экран выводилось "Bad-bad-bad":

```
int main() {
    try{
        throw Isklyuchish();
    }
    catch (std::exception& ex) {
        std::cout << "Bad-bad-bad";
    }
    return 0;
}
```

ответ на обороте

Подробно опишите действия в программе.

Задача 4. Реализовать классы.

ответ на обороте

Придумать класс S, отображающий студента. Класс должен содержать имя и возраст студента. Переопределить функции класса так, чтобы здесь:

```
S x("Alexey", 21), y("Olga", 19);
x ^ y;
```

не было ошибок, а имена в выбранных Вами переменных менялись местами, но возраст оставался прежним.

Задача 5. Решить задачу.

```
class A {
protected:
    A() { std::cout << "A::A\n"; }
public:
    virtual ~A() = 0 { std::cout << "A::~A\n"; }
};
class B: public A {
    A* x;
public:
    B() { std::cout << "B::B\n"; }
protected:
    ~B() { std::cout << "B::~B\n"; }
};
int main()
{
    A* pA = new B();
    delete pA;
}
```

A::A
B::B
B::~B
A::~A



Есть ли ошибки в приведенном фрагменте программы на C++? Если есть, то объясните, в чем они заключаются, и исправьте их, не меняя функцию main() и не удаляя функций из классов. Что будет выведено на печать?

Задача 3.

```

class Isklyuchish : public exception
{
public:
    Isklyuchish() throw() {}
    ~Isklyuchish() throw() {}
};

```



При выполнении оператора throw (вызывается деструктор) создается временный объект типа Isklyuchish (по ссылке) и передается исключение этого типа. Если собственный catch-блок может принимать и все типы, исключения типа exception и всех типов, представляющих собой классы, наследующие от exception, т.е. класс Isklyuchish является таким классом, то catch-блок перехватит временное исключение, и на экран выведется "Bad-bad-bad". После этого вызывается деструктор временного объекта *Всегда?!* на выходе из catch.

Задача 4.

```

#include <iostream>
#include <string>
using namespace std;
class S
{
public:
    S(const char * src_name, int src_age);
    void operator ^ (S &src_second);
private:
    string name;
    int age;
};
S::S(const char * src_name, int src_age)
{
    name = src_name;
    age = src_age;
}
void S::operator ^ (S &src_second)
{
    string temp;
    temp = name;
    name = src_second.name;
    src_second.name = temp;
}

```



```

int main()
{
    S x ("Alexey", 21), y ("Olga", 19);
    x ^ y;
    return 0;
}

```