

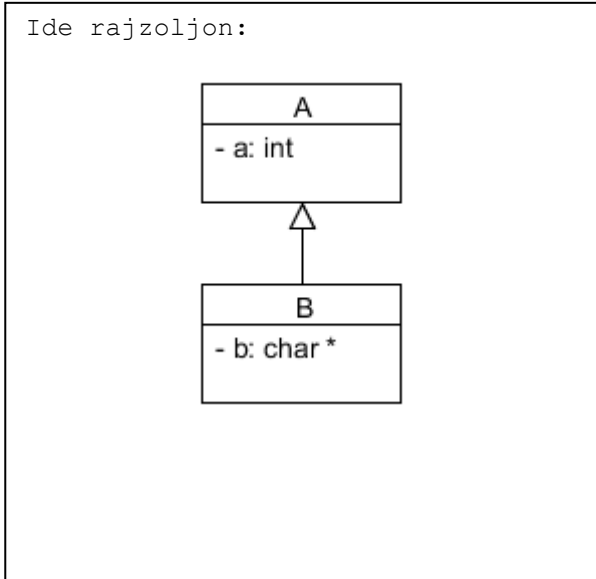
Szoftver laboratórium II. 4. ellenőrző dolgozat. 2012.04.16. Kurz/Terem: L1/	15 perc
Név: _____ Neptun: _____	Összpont: _____

1.feladat

1 pont

Rajzolja le az alábbi programban deklarált **A** és **B** osztályok kapcsolatát! Jelölje tagváltozók típusát és láthatóságát, de a tagfüggvényeket most **ne** jelölje, hogy a megadott helyen elférjen a rajz!

```
#include <iostream>
using namespace std;
class A {
    int a;
public:
    A(int i) :a(i)
        { cout << "A ctor " << a; }
    A(const A& ar) :a(ar.a)
        { cout << "A copy "; }
    void operator=(const A& ar)
        { a = ar.a; cout << "A assign"; }
    virtual void print(ostream& os) const
        { os << a; }
    virtual ~A() { cout << "A dtor "; }
};
class B : public A {
    char *b;
public:
    B(int i = 0, char* d = "B0") :A(i), b(d) { cout << " B ctor " << d; }
    B(const B& br) :A(br), b(br.b) { cout << " B copy "; }
    void print(ostream& os) const { os << b; }
    ~B() { cout << "B dtor "; }
};
int main() {
    A a1(1);
    B b1;
    B b2(2, "B2");
    B b3 = b2;
    b1 = b3;
    A *pa = &b1;
    pa->A::print(cout);
    pa->print(cout);
}
```



```

// _____
A a1(1);          cout << endl; // A°ctor°1 _____
B b1;           cout << endl; // A°ctor°0°B°ctor°B0 _____
B b2(2, "B2");  cout << endl; // A°ctor°2°B°ctor°B2 _____
B b3 = b2;      cout << endl; // A°copy°B°copy° _____
b1 = b3;        cout << endl; // A°assign _____
A *pa = &b1;     cout << endl; // _____
pa->A::print(cout); cout << endl; // 2 _____
pa->print(cout);  cout << endl; // B2 _____
}                // B°dtor°A°dtor°B°dtor°A°dtor°B°dtor°A°dtor°A°dtor° _____
```

2.feladat

4 pont

Írja a vonalakra, hogy mit ír ki a fenti C++ program a standard outputra az egyes utasítások végrehajtásakor! Jelölje a szóközőket is! (lehet, hogy nem kell minden vonalra írni.)

3.feladat

1 pont

Egészítse ki a fenti programot úgy, hogy az alábbi kódrészlet a megjegyzésben leírtaknak megfelelően működjön! A tagváltozó láthatóságát ne változtassa meg! (A standard outputon bármi keletkezhet!)

```
cerr << A(12); // 12-t ír a standard hibakimenetre
cerr << A(234); // 234-et ír a standard hibakimenetre
cerr << A(0) << endl; // 0-t, majd egy soremelést ír a standard
// hibakimenetre
```

```
ostream& operator<<(ostream &os, const A& a) {
    a.print(os);
    return os;
}
```