

Programozás alapjai II. 1. ellenőrző dolgozat. 2012.03.08. Kurz/Terem: G3/		
Név:	Neptun:	Összpont:

1. feladat

3 pont

Valósítson meg C++ nyelven egy egyszerű osztályt (**Kabin**), amit egy lift modellezéséhez kívánunk felhasználni! A kabin csak az utasok számát (**egész**) tárolja! Egy kabinban maximum 12 fő tartózkodhat. Legyen az osztálynak:

- alapértelmezett konstruktora, ami nullára állítja az utasok számát;
- olyan tagfüggvénye (**addPassenger**), amivel 1-gyel növeli az utasok számát, feltéve, ha van férőhely. Ha nincs, ne növeljen;
- olyan tagfüggvénye (**freeSpace**), ami visszaadja, hogy mennyi szabad hely van a kabinban.

Az osztálynak legyen legalább egy nem inline tagfüggvénye! Az osztály adatait közvetlenül ne lehessen módosítani!

```
class Kabin {
    int utasok;
public:
    Kabin() :utasok(0) {}
    void addPassenger();
    int freeSpace() const { return 12 - utasok; }
};
```

```
void Kabin::addPassenger() { if (utasok < 12) utasok++; }
```

2. feladat

3 pont

Írja a vonalakra, hogy mit ír ki az alábbi C++ program a standard outputra! Jelölje a szóközőket is (nem biztos, hogy minden vonalra kell írni)!

```
#include <iostream>
using namespace std;
class Zh1 {
    char a;
public:
    Zh1() :a('A') { cout << 'K'; }
    char& operator[] (int) { cout << "I"; return a; }
    const char& operator[] (int) const { cout << "J"; return a; }
    ~Zh1() { cout << "D"; }
};
```

```
int main() {
    Zh1 v1;
    Zh1 *pt = new Zh1[2];
    pt[0][0] = '8';
    delete[] pt;
    const Zh1 v2 = v1;
    cout << v2[3];
    return 0;
}
```

```
// _____
cout << endl; // K _____
cout << endl; // KK _____
cout << endl; // I _____
cout << endl; // DD _____
cout << endl; // _____
cout << endl; // JA _____
// DD _____
```