

Szoftver laboratórium II. 5. ellenőrző dolgozat. 2012.04.23. Kurz/Terem: L1/	15 perc
Név:	Összpont:

1. feladat**4 pont**

Az *Atlag* osztály módosítása nélkül, a *Serializable* osztály felhasználásával szeretnénk létrehozni annak perzisztens változatát (*PAtlag*). A perzisztens változatnak mindenben meg kell felelnie az *Atlag* osztálynak. **Egészítse ki** a *PAtlag* osztály deklarációját úgy, hogy az alábbi program írja ki és olvassa vissza az objektumok állapotát egy fájlból! Csak a pontozott részekre írjon, de nem kell minden helyre írnia! Tétélezze fel, hogy a fájlok írása/olvasása közben nem lép fel hiba.

```
#include <iostream>
#include <fstream>
#include <vector>
using namespace std;
class Serializable {
public:
    virtual void write(ostream&) const = 0; // kiíró tagfüggvény
    virtual void read(istream&) = 0;      // beolvasó tagfüggvény
    virtual ~Serializable() {}
};
class Atlag {
protected:
    long azonosito;      // azonosító
    double atlag;       // átlag
public:
    Atlag(long az = 0, double atl = 0) :azonosito(az), atlag(atl) { }
};
class PAtlag :public Atlag, public Serializable {
public:
    PAtlag(long az = 0, double atl = 0) :Atlag(az, atl) {
}

    void write(ostream& os) const {
        // Ügyeljen arra, hogy majd egymás után több adatot is vissza tudjon olvasni!
        os << azonosito << ' ' << atlag << endl;
    }

    void read(istream& is) {
        is >> azonosito >> atlag;
    }
};
int main() {
    vector<Serializable*> vec(10), vec2(10);
    for(int i = 0; i < 10; i++) vec.at(i) = new PAtlag(i, 2*i);
    ofstream ofs("adat.dat");
    for(int i = 0; i < 10; i++) vec.at(i)->write(ofs); // állapot kiírása
    ofs.close();
    ifstream ifs("adat.dat");
    for(int i = 0; i < 10; i++) {
        vec2.at(i) = new PAtlag;
        vec2.at(i)->read(ifs); // állapot visszaolvasása
    }
}
```

2. feladat**2 pont**

A feladatlap hátoldalán rajzolja le a feladatban megjelenő osztályok osztálydiagramját! Szintén a hátoldalra írjon egy kódrészletet, ami felszabadítja a fenti programban lefoglalt dinamikus memóriaterületet!

```
for (int i = 0; i < vec.size(); i++)
    delete vec[i];
for (int i = 0; i < vec2.size(); i++)
    delete vec2[i];
```

