AAAAAAAAAAAAAAAAAAAAAA

Szoftver laboratórium II. 5. ellenőrző dolgozat. 2013.05.06. Kurz/**Terem**: L1/	**15 perc**
Név: **Neptun:**	Összpont:

1. feladat **3.5 pont**

Az *Adat* osztály egy részecskefizikai kísérlet adatait tárolja. Az osztály belső szerkezetét nem ismerjük, de minden tárolt adata lekérdezhető és beállítható *getter/setter* függvényekkel, melyek interfészét pontosan ismerjük:

Függvény	**Funkció**
`void Adat::setN(std::string);`	név beállítása
`Std::string Adat::getN();`	név lekérdezése
`void Adat::setE(double);`	energia beállítása
`double Adat::getE()`	energia lekérdezése

Az osztálynak ezen kívül számos egyéb tagfüggvénye is van, melyek pontos funkcióját/leírását a fizikus kollegák ismerik, mi nem. Azt tudjuk, hogy az osztály nem perzisztens, csak default konstruktorai vannak, és nincs read/write tagfüggvénye. **Hozzon** létre az *Adat* osztályból egy perzisztens viselkedésű *PAdat* osztályt! A perzisztens változatnak minden funkciójában meg kell felelnie az *Adat* osztálynak. **Deklarálja** és **valósítsa** meg a *PAdat* osztályt úgy, hogy az alábbi programrészlet az elvárásoknak megfelelően működjön. A megvalósításban **ügyeljen** arra, hogy a névben szóköz is lehet! Tételezze fel, hogy a fájlok írása/olvasása közben nem lép fel hiba!

```
PAdat a1, a2; a1.setN("Kicsi reszecske");        // SZÓKÖZ van benne
a1.setE(12.3);  ...
std::ofstream ofs("Adat.dat"); a1.write(ofs);    // adat kiírása fájlba
...
std::ifstream ifs("Adat.dat"); a2.read(ifs);     // adat beolvasása fájlból
```

```
class PAdat : public Adat {
public:
    void read(std::istream& is) {
        size_t l;
        is  >> l; is.ignore(1);
        char *str = new char[l+1];
        is.read(str, l); is.ignore(1);
        str[l] = 0;
        setN(str);
        double e;
        is >> e;
        setE(e);
    }
    void write(std::ostream& os) {
        os << getN().size() << ' ' << getN() << ',' << getE() << std::endl;
    }

};
```

2. felAdat **2.5 pont**

Tételezze fel, hogy fizikus kollegái a *PAdat* osztályból példányosított objektumokat olyan tárolókba teszik, melyeknek van iterátora (*std::vector, std::list*)! Írjon olyan **függvénysablont** (*write*), ami képes a paraméterként kapott tárolóban levő elemek adatait a standard outputra kiírni! A kiírás formátumát Önre bízzuk. Mutassa be a sablon használatát!

```
template <class T>
void write(T t) {
    for (T::iterator it = t.begin(); it != t.end(); ++it)
        it->write(std::cout);
}
```