

NÉV (olvasható):

NEPTUN kód:

max. 10p, megoldási idő: 20 perc

1. Készíts generikus függvényt, amely kiírja a kapott tömb elemeit! Írj kódrészletet, melyben bemutatom a függvény használatát! (4p)

```
template<typename T>
void print(T*t, int n) {
    for (int i = 0; i < n; i++)
        cout << t[i];
}

#include <iostream>

int main() {
    double tomb[5] = { 1.1, 2.2, 3.3, 4.4, 5.5 };
    print(tomb, 5);
    return 0;
}
```

2. Hozz létre viselet osztályt! Minden viseletnek van mennyisége és ára. Az ezeket az adatokat tároló tagváltozók legyenek kívülről elérhetetlenek, viszont legyenek lekérdezhetők és beállíthatók (tagfüggvényekkel)! Legyen konstruktor, amely inicializálja a tagváltozókat! Legyen tagfüggvény, amely kiírja a tagváltozókat!

Származtass ruha és cipő osztályt! A ruha rendelkezik szövetlágysággal (valós szám), a cipő pedig egy logikai értékkel, amely azt mondja meg, hogy fényes-e. Mindkettőhöz készíts konstruktort, amely az ősből örökölt és a saját tagváltozókat egyaránt beállítja! Készíts kiíró függvényeket is, amelyek kiírják a tagváltozók értékét, az ősből örökölteket is! (Ehhez felhasználhatod az ős osztály kiíró függvényét.) (6p)

```
#include <iostream>

class Viselet {
    int menny, ar;
public:
    Irodaszer(int menny, int ar) :menny(menny), ar(ar) {}
    void setMenny(int m) { menny = m; }
    void setAr(int a) { ar = a; }
    int getMenny()const { return menny; }
    int getAr()const { return ar; }
    void print()const { std::cout << menny << ' ' << ar; }
};

class Ruha : public Viselet {
    double lagysag;
public:
    Papir(double lagysag, int ar, int tomeg):Viselet(menny,ar),lagysag(lagysag){}
    void print()const { Viselet::print(); std::cout << ' ' << lagysag; }
};

class Cipo : public Viselet {
    bool fenyese;
public:
    Ceruza(int menny, int ar, bool feny) :Viselet(menny, ar), fenyese(feny) {}
    void print()const { Viselet::print(); std::cout << ' ' << fenyese; }
};
```