

Programozás alapjai 2.	2. ellenőrző dolgozat.	2014.03.27.	Kurz/Terem: G3/
Név:	Neptun:	Összpont:	

1. feladat

6 pont

Adott az alábbi osztály, amely egy egyszerű órát modellez. Az óra csak percet és másodpercet „mutat”. A clock tagfüggvénye másodpercenként meghívódik, s így méri az időt.

```
class Ora {
    int min;
    int sec;
public:
    Ora(int m = 0, int s = 0) :min(m), sec(s) { }
    double getM() const { return min; }
    double getS() const { return sec; }
    void setM(int m){ min = m; }
    void setS(int s){ sec = s; }
    virtual void clock();
};
void Ora::clock() {
    sec++;
    if (sec >= 60) {
        sec = 0; min++;
    }
    if (min >= 60) min = 0;
}
```

Az **Ora** osztály felhasználásával, annak módosítása nélkül hozzon létre egy **Stopper** osztályt, ami megállítható és elindítható. Példányosításakor mindig álló helyzetben jöjjön létre, és legyen megadható az óra induló értéke! Valósítson meg minden olyan függvényt, ami az alábbi kódrészlet működéséhez kell! A kódrészlet a végrehajtáskor a megjegyzésben szereplő szöveget írja ki standard kimenetre!

```
...
Ora o1;
o1.clock();
std::cout << o1 << std::endl; // 0:1
Stopper *op = new Stopper(1,50);
std::cout << *op << std::endl; // 1:50
op->clock();
std::cout << *op << std::endl; // 1:50
op->start();
op->clock();
std::cout << *op << std::endl; // 1:51
op->stop();

std::ostream& operator<<(std::ostream& os, const Ora& v) {
    os << v.getM() << ':' << v.getS();
    return os;
}

class Stopper : public Ora {
    bool run;
public:
    Stopper(int m, int s) :Ora(m, s), run(false) {}
    void start() { run = true; }
    void stop() { run = false; }
    void clock();
};

void Stopper::clock() {
    if (run) Ora::clock();
}
```