

Programozás alapjai 2.	2. ellenőrző dolgozat.	2014.03.27.	Kurz/Terem: G1/
Név:	Neptun:	Összpont:	

## 1. feladat

6 pont

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

```
class Vekker {
    int hour;
    int min;
public:
    Vekker(int h = 0, int m = 0) :hour(h), min(m) { }
    double getH() const { return hour; }
    double getM() const { return min; }
    void setH(int h){ hour = h; }
    void setM(int m){ min = m; }
    virtual void clock();
};

void Vekker::clock() {
    min++;
    if (min >= 60) {
        min = 0; hour++;
    }
    if (hour >= 24) hour = 0;
}
```

A **Vekker** osztály felhasználásával, annak módosítása nélkül hozzon létre egy **CsorgoOra** osztályt, ami a konstruktorában megadott órákor „csörög”, azaz adott óra nulla perckor a clock tagfüggvénye kiírja, hogy „BRRR”! (A példában 12:00-kor csörög.) Valósítson meg minden olyan függvényt, ami az alábbi kódrészlet működéséhez kell! A kódrészlet a végrehajtáskor a megjegyzésben szereplő szöveget írja ki standard kimenetre!

```
...
    Vekker v1;
    v1.clock();
    std::cout << v1 << std::endl; // 0:1
    Vekker *vp = new CsorgoOra(11,50,12);
    std::cout << *vp << std::endl; // 11:50
    vp->setM(59);
    vp->clock(); // BRRR
    vp->clock();
    std::cout << *vp << std::endl; // 12:1
    vp->setM(0);

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

class CsorgoOra : public Vekker {
    int alarm;
public:
    CsorgoOra(int h, int m, int alarm) :Vekker(h, m), alarm(alarm) {}
    void clock();
};

void CsorgoOra::clock() {
    Vekker::clock();
    if (getM() == 0 && alarm == getH())
        std::cout << "BRRR" << std::endl;
}
```