

Programozás alapja 2.	3. ellenőrző dolgozat.	2016.04.18.	Kurz/ Terem: G3/	Elért pontszám:
Név:	Neptun:			

1. feladat

4 p

A **Kutya** osztály felhasználásával hozzon létre egy **KutyaGazdaval** osztályt, ami tárolja a kutya gazdájának címét (*String*) is! Legyen az osztálynak egy **info()** metódusa, ami a paraméterként kapott **std::ostream** típusú objektumra kiírja a kutya nevét és a gazda címét! A cím legyen külön is lekérdezhető a **getCim()** tagfüggvénnyel! A **Kutya** osztályt nem módosíthatja, de ha szükségesnek látja, a pontozott részre írhat! A laborgyakorlaton elkészített **String** osztály rendelkezésre áll, azt nem kell deklarálnia! Mielőtt megoldja ezt a feladatot, olvassa el a 2. feladatot is!

```
class Kutya {
    String nev;           // kutya neve
public:
    .....Kutya(const String& nev) :nev(nev) { }
    virtual void info(std::ostream& os) const { os << nev; }
    virtual ~Kutya() {}
};
```

Egy lehetséges megoldás:

```
class KutyaGazdaval :public Kutya {
    String cim;           // Gazdi címe
public:
    KutyaGazdaval(const String& nev, const String& cim) :Kutya(nev), cim(cim) {}
    void info(std::ostream& os) const { Kutya::info(os); os << cim; }
    String getCim() const { return cim; }
};
```

2. feladat

3 p

Írjon programrészletet, amely a dinamikus memóriában létrehoz 399 db **Kutya** példányt és egy **KutyaGazdaval** objektumot! A feladatot úgy oldja meg, hogy a létrehozott 400 darab objektum egyetlen 400 elemű tömb segítségével legyen elérhető! Hívja meg mind a 400 objektum **info()** metódusát, majd semmisítse meg a létrehozott objektumokat!

Egy lehetséges megoldás:

```
Kutya* t[400];
for (int i = 0; i < 399; i++)
    t[i] = new Kutya("Morgos");
t[399] = new KutyaGazdaval("Bikfic", "Lajos köz 2");
for (int i = 0; i < 400; i++) {
    t[i]->info(std::cout);
    delete t[i];
};
```

3. feladat

3 pont

Készítsen predikátum osztályt, ami alkalmas két **KutyaGazdaval** típusú objektum összehasonlítására a címek alapján! Adjon igaz értéket, ha a két cím nem egyezik.

Egy lehetséges megoldás:

```
struct KutyaGazdavalCmp {
    bool operator()(const KutyaGazdaval& a, const KutyaGazdaval& b) {
        return a.getCim() != b.getCim();
    }
};
```

Programozás alapja 2.	3. ellenőrző dolgozat.	2016.04.18.	Kurz/Terem: G3/	Elért pontszám:
Név:	Neptun:			

1. feladat

4 p

A **Hajo** osztály felhasználásával hozzon létre egy olyan **Hadihajo** osztályt, ami tárolja az ágyúk számát (*int*)! Legyen az osztálynak egy **indul()** metódusa, ami a standard kimenten kiírja az aktuális kikötő nevét, majd annyi ágyúlövést ad le, ahány ágyúja van ("durr!"). Az ágyúk száma legyen külön is lekérdezhető a **getAgyuk()** tagfüggvénnyel! A **Hajo** osztályt nem módosíthatja, de ha szükségesnek látja, a pontozott részre írhat! A laborgyakorlaton elkészített **String** osztály rendelkezésére áll, azt nem kell deklarálnia! Mielőtt megoldja ezt a feladatot, olvassa el a 2. feladatot is!

```
class Hajo {
    String hol;          // a hajó aktuális kikötőjének a neve
public:
    .....Hajo(String k = "") : hol(k) { }
    virtual void indul() const { std::cout << hol; }    // kiírja a kikötő nevét
    virtual ~Hajo() {}
};
```

Egy lehetséges megoldás:

```
class Hadihajo :public Hajo {
    int agyuk; // ágyúk száma
public:
    Hadihajo(String k, int a) :Hajo(k), agyuk(a) {}
    void indul() const { Hajo::indul();
        for (int i = 0; i < agyuk; i++) std::cout << "Durr! "; }
    double getAgyuk() const { return agyuk; }
};
```

2. feladat

3 p

Írjon programrészletet, amely a dinamikus memóriában létrehoz 229 db, Kenesén horgonyzó **Hajo** példányt és egy 5 ágyús **Hadihajo** példányt! A feladatot úgy oldja meg, hogy a létrehozott 230 darab objektumpéldány egyetlen 230 elemű tömb segítségével legyen elérhető! Hívja meg mind a 230 objektum **indul()** metódusát, majd semmisítse meg a létrehozott objektumpéldányokat!

Egy lehetséges megoldás:

```
Hajo* t[230];
for (int i = 0; i < 229; i++)
    t[i] = new Hajo("Kenesé");
t[229] = new Hadihajo("Kenesé", 5);
for (int i = 0; i < 230; i++) {
    t[i]->indul();
    delete t[i];
};
```

3. feladat

3 pont

Készítsen predikátum osztályt, ami alkalmas két **Hadihajo** típusú objektum összehasonlítására az ágyúk száma alapján! Adjon igaz értéket, ha az első ágyúinak száma nagyobb, mint a másodiké.

Egy lehetséges megoldás:

```
struct HadihajoCmp {
    bool operator() (const Hadihajo& a, const Hadihajo& b) {
        return a.getAgyuk() > b.getAgyuk();
    }
};
```