

Programozás alapja 2.	3. ellenőrző dolgozat.	2016.04.19.	Kurz/ Terem: G4/	Elért pontszám:
Név:	Neptun:			

1. feladat

4 p

A **Könyv** osztály felhasználásával hozzon létre egy olyan **Szakkönyv** osztályt, ami a szakterület nevét (*String*) is tárolja! Az osztály **adatok()** metódusa a paraméterként kapott **std::ostream** típusú objektumra írja ki a könyv címét és a szakterület nevét! A szakterület neve legyen külön is lekérdezhető a **getSzak()** tagfüggvénnyel! A **Könyv** 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 Konyv {
    String cim;           // konyv cime
public:
    .....Konyv(const String& cim) :cim(cim) {}
    virtual void adatok(std::ostream& os) const { os << cim; }
    virtual ~Konyv() {}
};
```

Egy lehetséges megoldás:

```
class Szakkonyv :public Konyv {
    String szak; // szakterulet neve
public:
    Szakkonyv(const String& cim, const String& szak) :Konyv(cim), szak(szak) {}
    void adatok(std::ostream& os) const { Konyv::adatok(os); os << " " << szak; }
    String getSzak() const { return szak; }
};
```

2. feladat

3 p

Írjon programrészletet, amely a dinamikus memóriában létrehoz 199 db "Micimackó" című **Könyv** példányt valamint egy Ön által választott szakterületű (pl. kedvenc középiskolai tárgy) és című **Szakkönyv** példányt! A feladatot úgy oldja meg, hogy a létrehozott 200 darab objektumpéldány egyetlen 200 elemű tömb segítségével legyen elérhető! Hívja meg mind a 200 objektum **adatok()** metódusát, majd semmisítse meg a létrehozott objektumpéldányokat!

Egy lehetséges megoldás:

```
Konyv* t[200];
for (int i = 0; i < 199; i++)
    t[i] = new Konyv("Micimackó");
t[199] = new Szakkonyv("Alapok", "Matek");
for (int i = 0; i < 200; i++) {
    t[i]->adatok(std::cout);
    delete t[i];
}
```

3. feladat

3 pont

Készítsen olyan egyoperandusú predikátum osztályt, ami a konstruktorában megadott szakterületű szakkönyveket képes kiválasztani, azaz igaz értékkel tér vissza, ha a konstruktorban megadott területtel egyező szakterületű szakkönyvet kap paraméterként!

Egy lehetséges megoldás:

```
struct SzakkonyvCmp {
    String szak;
    SzakkonyvCmp(const String& szak) :szak(szak) {}
    bool operator() (const Szakkonyv& k) {
        return k.getSzak() == szak;
    }
};
```

Programozás alapja 2.	3. ellenőrző dolgozat.	2016.04.19.	Kurz/Terem: G4/	Elért pontszám:
Név:	Neptun:			

1. feladat

4 p

Az **Auto** osztály felhasználásával hozzon létre egy olyan **Teherauto** osztályt, ami a sofőr nevét (*String*) is tárolja! Az osztály **info()** metódusa a paraméterként kapott **std::ostream** típusú objektumra írja ki az autó gyártási évét és a vezető nevét! A vezető neve legyen külön is lekérdezhető a **getNev()** tagfüggvénnyel! Az **Auto** 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 Auto {
    int gyartva;          // gyártási év
public:
    .....Auto(int ev) :gyartva(ev) {}
    virtual void info(std::ostream& os) const { os << gyartva; }
    virtual ~Auto() {}
};
```

Egy lehetséges megoldás:

```
class Teherauto :public Auto {
    String nev; // sorfor neve
public:
    Teherauto(int ev, const String& nev) :Auto(ev), nev(nev) {}
    void info(std::ostream& os) const { Auto::info(os); os << " " << nev; }
    String getNev() const { return nev; }
};
```

2. feladat

3 p

Írjon programrészletet, amely a dinamikus memóriában létrehoz 699 db 1 éves **Auto** példányt valamint egy Ön által vezetett teherautót! A feladatot úgy oldja meg, hogy a létrehozott 700 darab objektumpéldány egyetlen 700 elemű tömb segítségével legyen elérhető! Hívja meg mind a 700 objektum **info()** metódusát, majd semmisítse meg a létrehozott objektumpéldányokat!

Egy lehetséges megoldás:

```
Auto* t[700];
for (int i = 0; i < 699; i++)
    t[i] = new Auto(2015);
t[699] = new Teherauto(2016, "En");
for (int i = 0; i < 700; i++) {
    t[i]->info(std::cout);
    delete t[i];
}
```

3. feladat

3 pont

Készítsen olyan egyoperandusú predikátum osztályt, ami a konstruktorában megadott nevű sofőr által vezetett teherautókat képes kiválasztani, azaz igaz értékkel tér vissza, ha a konstruktorban megadott névvel egyező nevű sofőr vezeti a paraméterként kapott teherautót!

Egy lehetséges megoldás:

```
struct TeherautoCmp {
    String nev;
    TeherautoCmp(const String& nev) :nev(nev) {}
    bool operator() (const Teherauto& a) {
        return a.getNev() == nev;
    }
};
```