A rendelkezésre álló idő:18 **perc**

Programozás alapja 2. 3. ellenőrző dolgozat. 2016.04.18. Kurz/**Terem**: G1/	Elért pontszám:
Név: **Neptun:**	

1. feladat 4 p

A ***Szamla*** osztály felhasználásával hozzon létre egy olyan ***BankSzamla*** osztályt, ami bankszámla azonosítóját és a számlán rendelkezésre álló összeget (*double*) tárolja! Legyen az osztálynak egy ***print()*** metódusa, ami a standard kimenetre kiírja a bankszámla azonosítóját és a számlán rendelkezésre álló összeget! A rendelkezésre álló összeg legyen külön is lekérdezhető a ***getOsszeg()*** tagfüggvénnyel! A ***Szamla*** osztályt nem módosíthatja, de ha szükségesnek látja, a pontozott részre írhat! A ***String*** osztály rendelkezésére áll, azt nem kell deklarálnia!
Mielőtt megoldja ezt a feladatot, olvassa el a 2. feladatot is!

```
class Szamla {
    String azon;           // számla azonosítója
public:
     .......Szamla(const String& azon) :azon(azon) {}
     virtual void print() const { std::cout << azon; }
     virtual ~Szamla() {}
};
```

Egy lehetséges megoldás:

```
class BankSzamla :public Szamla {
    double osszeg;  // számlán rendelkezésre álló összeg
public:
    BankSzamla(const String& azon, double osszeg) :Szamla(azon), osszeg(osszeg) {}
    void print() const { Szamla::print(); std::cout <<" " << osszeg << std::endl;}
    double getOsszeg() const { return osszeg; }
};
```

---

2. feladat 3 p

Írjon programrészletet, amely a dinamikus memóriában létrehoz 99 db ***Szamla*** példányt a saját neptunkódjával, mint számlaazonosítóval és egy db 10000 Ft-ot tartalmazó ***BankSzamla*** példányt a saját kódjával! A feladatot úgy oldja meg, hogy a létrehozott 100 darab objektumpéldány egyetlen 100 elemű tömb segítségével legyen elérhető! Hívja meg mind a 100 objektum ***print()*** metódusát, majd semmisítse meg a létrehozott objektumpéldányokat!

Egy lehetséges megoldás:

```
Szamla* t[100];
for (int i = 0; i < 99; i++)
    t[i] = new Szamla("NEPTUN");
t[99] = new BankSzamla("NEPTUN", 10000);
for (int i = 0; i < 100; i++) {
    t[i]->print();
    delete t[i];
};
```

---

3. feladat 3 pont

Készítsen egy funktort, ami két ***BankSzamla*** típusú objektumot összehasonlít a számlán levő összeg alapján! Ha a két összeg azonos, akkor adjon igaz értéket.

Egy lehetséges megoldás:

```
struct BankSzamlaCmp {
    bool operator()(const BankSzamla& a, const BankSzamla& b) {
        return a.getOsszeg() == b.getOsszeg();
    }
};
```

BBBBBBBBBBBBBBBBBBBBBBBBBBBBB rendelkezésre álló idő:18 **perc**

Programozás alapja 2. 3. ellenőrző dolgozat. 2016.04.18. Kurz/**Terem**: G1/	Elért pontszám:
Név: **Neptun:**	

1. feladat 4 p

A ***Szemely*** osztály felhasználásával hozzon létre egy olyan ***Hallgató*** osztályt, ami egy hallgató nevét és tankörének számát (*int*) tárolja! Az osztály ***write()*** metódusa a standard kimenetre írja ki a hallgató nevét és tankörének számát! A tankörszám legyen külön is lekérdezhető a getTankor() tagfüggvénnyel! A ***Szemely*** osztályt nem módosíthatja, de ha szükségesnek látja, a pontozott részre írhat! A ***String*** osztály rendelkezésére áll, azt nem kell deklarálnia!
Mielőtt megoldja ezt a feladatot, olvassa el a 2. feladatot is!

```
class Szemely {
    String nev;            // hallgató neve
public:
     .......Szemely(const String& nev) :nev(nev) {}
     virtual void write() const { std::cout << nev; }
     virtual ~Szemely() {}
};
```

Egy lehetséges megoldás:

```
class Hallgato :public Szemely {
  int tankor; // tankörszám
public:
  Hallgato(const String& nev, int tankor) :Szemely(nev), tankor(tankor) {}
  void write() const { Szemely::write(); std::cout <<" " << tankor << std::endl;}
  int getTankor() const { return tankor; }
};
```

---

2. feladat 3 p

Írjon programrészletet, amely a dinamikus memóriában létrehoz 199 db ***Hallgato*** példányt a saját nevével és tankörszámával, valamint egy ***Szemely*** példányt a saját nevével! A feladatot úgy oldja meg, hogy a létrehozott 200 darab objektumpéldány egyetlen 200 elemű tömb segítségével legyen elérhető! Hívja meg mind a 200 objektum ***write()*** metódusát, majd semmisítse meg a létrehozott objektumpéldányokat!

Egy lehetséges megoldás:

```
Szemely* t[200];
for (int i = 0; i < 199; i++)
    t[i] = new Hallgato("NEPTUN", 5);
t[199] = new Szemely ("NEPTUN");
for (int i = 0; i < 200; i++) {
    t[i]->write();
    delete t[i];
}
```

---

3. feladat 3 pont

Készítsen egy funktort, ami két ***Hallgato*** típusú objektumot összehasonlít a tankörszám alapján! Ha a két tankörszám azonos, akkor adjon hamis értéket.

Egy lehetséges megoldás:

```
struct HallgatoCmp {
    bool operator()(const Hallgato& a, const Hallgato& b) {
        return a.getTankor() != b.getTankor();
    }
};
```