# 2. gyakorlat

## Berezvai Dániel jegyzete <http://elte.3ice.hu/>

Hárman jöttünk konzultálni, én csak azért, mert lyukas órám van.

Gregorics Tibor tanár úr órájára beültem, kaptunk egy feladatot, amit, aki 15 perc alatt meg tud írni, az már el is végezte a tárgyat, csak el kell, hozza a beadandókat.

Elkezdtem írni a második beadandót.

## BigNum

Valósítsa meg a nagyon nagy természetes számok típusát! Ábrázoljuk a számokat a számjegyeik láncolt listáival és implementálja az összeadást és a szorzást operátor túldefiniálással! Ne feledkezzen meg a beolvasó (operátor>>) és kiíró (operátor<<) metódusokról sem! Az összeadás műveletigénye $Ơ\left(m+n\right)$, a szorzásé $Ơ\left(m\*n\right)$, ahol $m$és $n$a két szám számjegyeinek száma.

Az fenti feladat típusát egy **osztály segítségével** valósítsa meg. Az összes megvalósítandó típus azonos típusú elemek összetett adatszerkezetű gyűjteményét írja le, amelyet **láncolt listában** kell elhelyezni, ezért az osztályban meg kell valósítani a **másoló konstruktort** és az **értékadás operátort** is. Ahol a feladat szövege nem definiálja, az elemi típus az egész számok típusa. (Ne alkalmazzon sablonokat!)

Egy osztály szolgáltatásainak (metódusainak) bemutatásához olyan főprogramot kell készíteni, amelyik egy **menü segítségével** teszi lehetővé a metódusok tetszőleges sorrendben történő kipróbálását. A főprogram példányosítson egy objektumot, amelyre a menüpontok közvetítésével lehet meghívni az egyes metódusokat. Természetesen szükség lesz minden tevékenység után az objektum állapotának kiírására vagy egy az objektum állapotát kiíró külön menüpontra. Azoknál az osztályoknál, ahol vannak olyan metódusok (esetleg barát függvények), amelyek több objektum közötti műveleteket valósítanak meg, a főprogram több objektum létrehozására és azok állapotának kiírására is adjon lehetőséget.

Alkalmazzon **operátor felüldefiniálást** és **kivételkezelést**! Bontsa modulokra a programját!