

# Memória

Készítsünk egy osztályt, mely egy memóriát kezel. Az elképzelt programozási nyelvünk kizárólag referenciatípusokkal dolgozik, a memória mérete 1 mb.

Maga a memória legyen egy byte típusú tömb. Készítsen a memória osztályon belül egy saját láncolt listát. Ez a lista fogja a memóriacímeket tárolni az egyes változókhoz (hol kezdődnek a tömbben). Azt, hogy meddig tartanak a típus saját jellemzője. String esetében ez nem ismert. De ilyenkor az utolsó utáni karakter egy terminálójel (pl: \0 vagy \$).

Készítsük el a .NET néhány típusának saját változatát reprezentáló osztályunkat: Egesz, Bajt, Karakter, Szoveg, Tort néven. Mindegyik osztálynak legyen neve és értéke. Az értéket a .NET megfelelő típusaival tároljuk (pl: Bajt osztályban legyen a változó neve stringként és az értéke byte-ként). A típusainkat fogja össze egy Valtozo ősosztály.

Rendelkezzen a memória a következő metódusokkal:

- DelVariable(string név)
- Valtozo GetVariable(string név)
- SetVariable (Valtozo)

A memória osztályon legyen meghívható egy Clean() metódus. Ez a metódus a törlések következtében kialakult helyek miatt átrendezi a memóriát, a memóriacímeket módosítja. Törekedjen arra, hogy a memóriamozgatások a leoptimálisabban történjenek meg! Ha hozzányúlunk egy változóhoz, annak legyen 5 a költsége, ha egy memóriarekesszel arrébb toljuk, annak legyen 1 a költsége.

Ha a memória betelik, akkor dobjunk el egy saját készítésű MemoriaKifutottATartomanybolException nevű kivételt. Készítsünk kivételt az olyan esetek kezelésére is, amikor létre akarunk hozni olyan változót, ami már létezik, illetve olyat akarunk törölni, lekérdezni, módosítani, ami már nincs.

Oldja meg tetszőleges módon a tömbök kezelését, és a program működését mutassa be a javított buborékos rendezésen keresztül. A rendezés legyen képes egy Egesz tömböt rendezni! Továbbá a memória legyen képes megkülönböztetni két külön program változóit. A memóriamozgatásnál azt is vegye figyelembe, hogy annál jobb egy átrendezés, minél kevesebb program változót érinti.