

Feladat

- A feladat elkezdéséhez a mellékelt projekt tartalmaz **main** függvényt. A teszteléshez a **main** függvényben a nem használt kódokat ki lehet kommentezni, de módosítani azokat nem szabad. A feladathoz csak azokat a függvényeket készítsd el, amiket a feladat kér, illetve ami ezen felül feltétlenül szükséges a helyes működéshez.
 - Ügyelj arra, hogy minden lefoglalt memória kerüljön megfelelően felszabadításra.
 - **A nem forduló kód 0 pont.**
 - A feladatban felhasználói felületen található beviteli mezőkhöz kell egy egyszerű osztályszerkezetet elkészíteni.
- 1) Készíts egy **BeviteliMezo** osztályt, amely egy beviteli mező általános adatait tárolja: érték megnevezése (szöveg), és hogy szerkeszthető-e (logikai érték). A konstruktor várja az adatokat, és legyen hozzájuk getter (*getMegnevezes*, *getSzerkesztheto*). A logikai értékhez legyen setter is (*setSzerkesztheto*). **(2 pont)**
 - a) Készíts az osztályhoz egy tisztán virtuális *megfelelo* függvényt, amely majd a gyerekosztályokban visszadja, hogy a bevitt érték megfelel-e a feltételeknek. **(1 pont)**
 - 2) Származtass egy **SzovegMezo** osztályt a **BeviteliMezo**-ból, amely új adatként tárolja a beírt szöveget (szöveg), és a szöveg maximális hosszát (egész). A konstruktora csak a maximális hosszt várja (és az őszosztály adatait), a szöveget üres szövegre állítsa be. A szöveghez legyen setter és getter (*setSzoveg*, *getSzoveg*), a maximális hosszhoz pedig csak getter (*getMaxhossz*). **(3 pont)**
 - a) A *setSzoveg* függvényt úgy valósítsd meg, hogy csak akkor módosítsa az adattagot, ha az őszosztályban a szerkeszthetőség igaz-ra van állítva. **(1 pont)**
 - b) Fejtsd ki az őszosztály *megfelelo* függvényét, ami igazat adjon vissza, ha a tárolt szöveg hossza nem több, mint a megadott max hossz, egyébként hamisat. **(2 pont)**
 - c) Készítsd el az osztályhoz az unáris *~* operátort, amely visszatér a tárolt szöveggel. **(2 pont)**
 - 3) Származtass egy **SzamMezo** osztályt a **BeviteliMezo**-ból, amely új adatként tárolja a bevitt szám értéket (előjeles egész). A konstruktora csak az őszosztály adatait várja, az új értéket állítsa 0-ra, de legyen hozzá setter és getter is (*setErtek*, *getErtek*). **(3 pont)**
 - a) A *setErtek* függvényt úgy valósítsd meg, hogy csak akkor módosítsa az adattagot, ha az őszosztályban a szerkeszthetőség igaz-ra van állítva. **(1 pont)**
 - b) Fejtsd ki az őszosztály *megfelelo* függvényét, ami igazat adjon vissza, ha a tárolt szám értéke 0 és 100 között van, egyébként hamisat. **(2 pont)**
 - c) Készíts az osztályhoz egy *=* operátort, amelynek bal oldalán az osztály, a jobb oldalán egy egész szám áll. Az operátor módosítsa a tárolt értéket a jobb oldali számra (akkor is, ha egyébként nem lenne szerkeszthető). **(2 pont)**
 - d) Az osztályban legyen egy statikus adattag, ami a beviteli mezőkbe írható maximális értéket tárolja (egész). Kezdetben az értéke legyen 100. Készítsd el hozzá a setter és getter függvényeket (*setMaxertek*, *getMaxertek*). **(3 pont)**
 - e) A *megfelelo* függvény működését módosítsd úgy, hogy akkor jó, ha 0 és a statikus maximális érték között van. **(1 pont)**

- 4) Készíts egy **AdatBekero** osztályt, amely tetszőleges mennyiségű beviteli mezőt kezel. A beviteli mezőket úgy kell tárolni, hogy mindig pontosan annyi hely legyen a tömbben, amennyi kell. Hozd létre az ennek megfelelő adattagokat. **(3 pont)**
- a) A **getDarab** függvénnyel legyen lekérdezhető a tárolt mezők száma. **(1 pont)**
- b) Az osztályhoz a **+=** operátorral lehessen egy újabb beviteli mezőt hozzáadni, a **main**-ben látható módon. **(4 pont)**
- c) Az osztály destruktora szabadítsa fel a tárolt mezőket. **(1 pont)**
- d) Készíts egy **<<** operátort az osztályhoz, amellyel az osztály tartalmát tetszőleges kimeneti folyamra meg lehet jeleníteni. Minden beviteli mezőhöz jelenjen meg a megnevezése, valamint hogy az általa érték megfelelő-e. **(3 pont)**
- i) + Oldd meg, hogy minden mezőhöz jelenjen meg a benne tárolt érték is. **(3 pont)**
- e) Készíts az osztályhoz egy **[]** operátort, ami megkap egy tömb indexet, és visszatér azzal, hogy az adott indexen tárolt mezőben lévő érték megfelelő-e. Érvénytelen index esetén hamissal térjen vissza. **(2 pont)**