

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 elemekhez kell egy egyszerű osztályszerkezetet elkészíteni.
- 1) Készíts egy **Elem** osztályt, amely egy kirajzolt elem általános adatait tárolja: pozíció (x és y koordinátával, egészek), és méret (szélesség és magasság, egészek). A konstruktor várja az adatokat, és legyen hozzájuk getter (*getX*, *getY*, *getSzelesseg*, *getMagassag*). **(2 pont)**
 - a) Készíts az osztályhoz egy virtuális *kirajzol* függvényt, amely jelenítse meg mind a 4 (x, y, szélesség, magasság). **(1 pont)**
 - b) Készíts az osztályhoz egy tisztán virtuális *aktival* függvényt, amely vár egy egész paramétert és majd a gyerekosztályokban valami műveletet végez az objektummal. A függvény paraméterét el is lehessen hagyni, az értéke ekkor legyen 5. **(2 pont)**
 - 2) Származtass egy **Gomb** osztályt az **Elem**-ből, amely új adatként tárolja még a gombon szereplő szöveget (szöveg), és hogy hányszor nyomták meg (egész). A konstruktor várjon minden adatot, kivéve a megnyomások számát, az mindig 0-ról induljon. Legyenek getterek az új adatokhoz (*getSzoveg*, *getMegnyomasok*). **(3 pont)**
 - a) Fejtsd ki az őosztály *kirajzol* függvényét, ami használja az őosztály *kirajzol* függvényét, majd jelenítse meg az új adatokat is. **(2 pont)**
 - b) Fejtsd ki az őosztály *aktival* függvényét, ami növelje a megnyomások számát a paraméterben megkapott értékkel. **(1 pont)**
 - c) Készíts az osztályhoz egy **==** operátort, amely két gombot hasonlít össze, és igazgal tér vissza, ha a rajtuk lévő szöveg és a megnyomások száma is megegyezik, egyébként hamissal. Az őosztály adatait ne vegye figyelembe az operátor. **(2 pont)**
 - 3) Származtass egy **SzamBevitel** osztályt az **Elem**-ből, amely új adatként egy számot tárol (egész), ami a bevitt érték. A konstruktor várjon minden adatot, és legyen hozzá getter (*getErtek*). **(3 pont)**
 - a) Fejtsd ki az őosztály *kirajzol* függvényét, ami használja az őosztály *kirajzol* függvényét, majd jelenítse meg az új adatot is. **(2 pont)**
 - b) Fejtsd ki az őosztály *aktival* függvényét, ami módosítsa a tárolt értéket a paraméterben kapottal. Az érték csak -100 és 100 között mozoghat, ha ennél több lenne, akkor ezekhez vissza kell „vágni” (vagyis a 95-höz 10-et adva a 105 túl nagy, így 100-ra változik). **(2 pont)**
 - c) Készíts az osztályhoz egy **<<** operátort, amelynek a jobb oldalán egy egész érték van. Ha a jobb oldali érték -100 és 100 közötti, akkor állítsa be a belső egész szám értékét erre, egyébként ne csináljon semmit. **(2 pont)**

- 4) Készíts egy **Felulet** osztályt, amely tetszőleges mennyiségű elemet kezel. Az elemeket úgy kell tárolni, hogy mindig pontosan annyi hely legyen a tömbben, amennyi kell. Hozd létre az ennek megfelelő adattagokat. **(3 pont)**
- a) A **getMeret** függvénnyel legyen lekérdezhető a tárolt elemek száma. **(1 pont)**
 - b) Az osztályhoz a **hozzaad** függvénnyel lehessen egy újabb megjelenítendő elemet hozzáadni, a **main**-ben látható módon. **(4 pont)**
 - c) Az osztály destruktora szabadítsa fel a tárolt megjelenítőket. **(1 pont)**
 - d) Készíts az osztályhoz egy **kiir** függvényt, amely megjelenít minden eltárolt elemet. **(2 pont)**
 - e) Készíts az osztályhoz másoló konstruktort. **(4 pont)**
 - f) Készíts az osztályhoz egy prefix **++** operátort, amely minden tárolt elemet aktivál, az alapértelmezett értéket használva. **(3 pont)**