

## Feladat

- A feladat elkezdéséhez a mellékelt projekt tartalmaz **main** függvényt. A teszteléshez a **main** függvényben a nem használt kódokat ki lehet kommentezni, de módosítani nem szabad a kódokat. A feladathoz csak azokat a függvényeket készítsd el, amiket a feladat kér.
  - Ügyelj arra, hogy minden lefoglalt memória kerüljön megfelelően felszabadításra.
  - A feladatban egy egyszerű társasjáték megvalósítása a cél, ahol a karakterek akciókat hajtanak végre, és ezért pontokat kapnak.
  - A feladatban többféle karakter is létezik, akik más módon kezelik az egyes akciókat. Ezt létrehozásukkor adjuk majd meg.
- 1) Készíts egy **Akcio** osztályt, amely tárolja egy végrehajtható akció adatait: akció típusa (szöveg), pontértéke (egész) és nehézségi szintje (egész). A konstruktor várja az adatokat, és legyen hozzájuk getter (*getTipus*, *getErtek*, *getNehezseg*). (2 pont)
    - a) Készíts az osztályhoz egy `*` operátort, amely visszaadja lebegőpontos értéként, hogy az akciónak a pontértéke milyen arányban van a nehézségével (a kettő hányadosa). (2 pont)
  - 2) Készíts egy **Karakter** osztályt, amely tárolja egy karakter nevét (szöveg) és pontszámát (egész, lehet negatív). A konstruktor várja az adatokat, és legyen hozzájuk getter (*getNev*, *getPont*). (2 pont)
    - a) Az osztály absztrakt ősosztály lesz. Legyen benne egy tisztán virtuális *vegrehajt* függvény, amelynek paramétere egy **Akcio**. A függvény logikai értékkel tér majd vissza (végrehajtás sikeressége), és módosít is majd a belső adatokon, amikor a gyerekosztályok kifejtik. (1 pont)
  - 3) Származtass a **Karakter** osztályból egy **Mohokarakter** osztályt. Ez tároljon még egy nehézségi szintet (egész érték). Az új adatot is a konstruktor várja, és legyen hozzá getter (*getNehezseg*). (2 pont)
    - a) Fejtsd ki a *vegrehajt* függvényt úgy, hogy a karakter akkor hajta végre az akciót ha annak az értéke legalább 5, de a nehézsége nem haladja meg a karakter által tárolt nehézséget. Ekkor a függvény igazzal tér vissza, és a karakter megkapja az akcióval járó pontszámot (az értéke). Egyéb esetben térjen vissza hamissal, és nem változik a pontszám. (2 pont)
    - b) Legyen az osztályban egy statikus adattag, amivel az objektumszámlálást meg lehet valósítani. Legyen egy *getDarab* függvény, amellyel a program bármely pontján lekérdezhető, hogy az adott ponton mennyi **Mohokarakter** objektum létezik. (3 pont)
  - 4) Származtass a **Karakter** osztályból egy **Valogatos** osztályt. Ez tároljon még egy szöveget, ami a karakter által választott akció típus. Az új adatot is a konstruktor várja, és legyen hozzá getter (*getTipus*). (2 pont)
    - a) Fejtsd ki a *vegrehajt* függvényt úgy, hogy a karakter csak azokat az akciókat hajtsa végre, amelyek típusa megegyezik az általa tárolttal, azokat viszont mindet. Ekkor a függvény igazzal tér vissza, és a karakter megkapja az akcióval járó pontszámot (az értéke). Egyéb esetben térjen vissza hamissal, és nem változik a pontszám. (2 pont)
  - 5) Készíts egy **Jatekosok** osztályt, amely karaktereket fog tárolni tetszőleges mennyiségben. A konstruktor várja, hogy legfeljebb mennyi karaktert tárolhat, és foglalja le a megfelelő dinamikus tömböt. Kezdetben egy karaktert sem tárol, majd később adjuk hozzá. A *getDarab* függvénnyel lehessen lekérni az eltárolt karaktereket számát. (3 pont)

- a) Készíts az osztályhoz << operátort, amelynek segítségével új karaktert lehet hozzáadni a tárolóhoz (paraméterként mutatót vár). Ha a karakterek száma már elérte a maximumot, akkor ne adja hozzá az újat (memóriaszemét ekkor se legyen!). (2 pont)
- b) Készíts [] operátort az osztályhoz, amely paraméterként egy egész indexet vár, és visszaadja a belső tömb ennyiedik helyén lévő mutatót. Ha az index érvénytelen (kicímez a tömbből, vagy csak nem tárolunk még ennyit), akkor null mutatóval térjen vissza. (2 pont)
- c) Készíts += operátort az osztályhoz, melynek bal oldalán egy Jatekosok objektum van, a jobb oldalán egy Akcio. Az operátor minden eltárolt karakternek adja át az akciót végrehajtásra. Az operátor számolja meg, hogy hány karakter hajtotta végre az akciót, és ezzel az egész értékkel térjen vissza. (3 pont)
- d) Készíts az osztályhoz << operátort, amivel tetszőleges kimeneti folyamra meg lehet jeleníteni az adatait. A kimeneten szerepeljen a tárolt karakterek maximális és tényleges száma, valamint a karakterek adatai. A gyerekosztályokban tárolt adatokat nem kell megjeleníteni, viszont minden karakternél írd ki hogy melyik típusba tartozik (Mohokarakter, vagy Valogatos). (4 pont)

Pontozás:

0-15: 1

16-19: 2

20-23: 3

24-27: 4

28-32: 5