

# Programozás II. ZH

SZTE Szoftverfejlesztés Tanszék

2025. ősz

## Technikai ismertető

- A programot C++ nyelven kell megírni.
- A megoldást a *Bíró* fogja kiértékelni.
  - A Feladat beadása felületen a Feltöltés gomb megnyomása után ki kell várni, amíg lefut a kiértékelés. **Kiértékelés közben nem szabad az oldalt frissíteni vagy a Feltöltés gombot újból megnyomni** különben feltöltési lehetőség veszik el!
- Feltöltés után a *Bíró* a programot g++ fordítóval és a  
-std=c++20 -static -O2 -DTEST\_BIR0=1  
paraméterezéssel fordítja és különböző tesztesetekre futtatja.
- A program működése akkor helyes, ha a tesztesetek futása nem tart tovább 5 másodpercnél és hiba nélkül (0 hibakóddal) fejeződik be, valamint a program működése a feladatkiírásnak megfelelő.
- A *Bíró* által a `riport.txt`-ben visszaadott lehetséges hibakódok:
  - Futási hiba 6: Memória- vagy időkorlát túllépés.
  - Futási hiba 8: Lebegőpontos hiba, például nullával való osztás.
  - Futási hiba 11: Memória-hozzáférési probléma, pl. tömb-túindexelés, null pointer használat.
- A `riport.txt` és a fordítási log fájlok megtekinthetők az alábbi módon:
- A programot 20 alkalommal lehet benyújtani, a megadott határidőig.
- A programban szerepelhet `main` függvény, amely a pontszámításkor nem lesz figyelembe véve. Azonban ha fordítási hibát okozó kód van benne az egész feladatsor 0 pontos lesz.

## Általános követelmények, tudnivalók

- Csak a leírásban szereplő osztályokat, metódusokat és adattagokat kell megvalósítani, egyéb dolgokért nem jár plusz pont.
- Minden metódus, amelyik nem változtatja meg az objektumot, legyen konstans! Ha a paramétert nem változtatja a metódus, akkor a paraméter legyen konstans!
- string összehasonlításoknál az egyezés a pontos egyezést jelenti, azaz ha kis-nagy betűben térnek el, akkor már nem tekinthetők egyenlőnek (pl. a "piros" != "Piros")
- A leírásokban bemutatott példákban a stringek köré rakott idézőjelek nem részei az elvárt kimenetnek, azok csak a string határait jelölik. Például ha az szerepel, hogy a példa bemenetre az elvárt kimenet az, hogy "3 alma", akkor az elvárt kimenet idézőjelek nélkül az 3 alma, de a szóköz szükséges!
  - A tesztesetekben nem lesz ékezetes szöveg kiírása.
- Az elvárt kimeneteknek karakterről karakterre olyan formátumúnak kell lennie, ami a feladatban le van írva (szóközöket és sortöréseket is beleértve).
- Ha az objektum másolása nem triviális (azaz a fordító által generált másolás nem elegendő), akkor a megfelelő másolást is meg kell valósítani.

## Kiindulási projekt, megoldás feltöltése

- **A megoldáshoz az előre kiadott osztályok módosítása szükséges lehet.**
  - Nem minden ZH esetében van kiindulási projekt.
- Feltöltéskor ezeket az osztályokat is fel kell tölteni és a módosításokat is pontosan a bírónak!
- Egyes tesztesetekben a bírónak módosított osztályt is használhat ezen kiinduló osztályok helyett, ezzel tesztelve a valóban helyes működést!

## Zh alatt használható segédanyag

- A ZH során használható segédanyag elérhető bírónak.
  - <https://biro.inf.u-szeged.hu/kozos/prog2/>

## Alkatrész osztály

Készíts egy `Alkatrész` osztályt! Ez az osztály egy alkatrészt fog reprezentálni, pl. Alaplap, GPU, CPU, memória stb.

- Az alkatrészt lehessen példányosítani az alkatrész nevének és árának megadásával, ebben a sorrendben! (2 pont)
- Valósítsd meg az `std::string`-re való konverziót! Ez az alkatrész nevét adja vissza! (2 pont)
- Valósítsd meg az `unsigned`-re való konverziót! Ez az alkatrész árát adja vissza!<sup>1</sup> (2 pont)
- Tedd lehetővé, hogy az alkatrész setben használható legyen! Ez a viselkedés legyen azzal azonos, mintha csak az alkatrész neve kerülne a setbe! (3 pont)

## PC osztály

- Készíts egy `PC` osztályt! Egy `PC`-ben alkatrészek vannak, minden alkatrészből csak 1 szükséges.
- Valósítsd meg, hogy a `+=` operátorral, hogy `Alkatrészt` lehessen a `PC`-hez adni. Az operátor egymás után meghívható legyen! (2 pont)
- Készíts egy `beepitve` metódust ami megmondja egy `Alkatrészről`, hogy benne van-e a `PC`-ben vagy sem. (3 pont)
- Készíts egy `stats` nevű metódust ami kiírja lexikografikus rendezésben a `PC`-ben található `Alkatrészeket`! Egy sorba egy alkatrészt ír ki a következő formában: `<alkatrész neve> (szóköz) <alkatrész ára>` Az utolsó alkatrész után is van sortörés! (2 pont)
- Készíts egy `koltseg` metódust, ami visszaadja, hogy a `PC`-ben található alkatrészeket legalább hány ezressel tudjuk kifizetni. (4 pont)
- Valósítsd meg 2 `PC` összeadását! Az eredmény egy új `PC`, ami tartalmazza mindkét `PC` alkatrészeit! Ha egy alkatrészből mindkét operandusban lenne, a drágábbik kerüljön bele! (5 pont)

---

<sup>1</sup>Megjegyzés: az eddigi 3 feladatra csak akkor tud pont járni, ha legalább a konstruktorok és konverziós operátorok deklarációja megvan mind.