

# Programozás II. ZH

SZTE Szoftverfejlesztés Tanszék

2024. 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 -Wall -Werror -static -O2 -DTEST\_BIRO=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 pontozhatja a bíró!
- Egyes tesztesetekben a bíró módosított osztályt is használhat ezen kiinduló osztályok helyett, ezzel tesztelve a valóban helyes működést!

## Zh alatt használható segédanyag

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

# Kiindulás

**A kiindulási feladatban kapott osztály módosítása vagy bővítése szükséges lehet!**

## Sikidom osztály

- Az `Sikidom` osztály egy síkidomot reprezentál.
- A síkidomnak van egy típusa, amelyet egy sztring változóban tárolunk.
- Minden konkrét síkidomnak meghatározható a kerülete, ezt a `kerulet` függvénnyel lehet lekérni.
- A `keruletToString` metódus adja vissza a `Sikidom` kerületének értékét az adott formában "A <típus> kerulete <kerulet> egység".  
Példa: A negyzet kerulete 45 egység. Ez a metódus adott, ezen nem kell változtass!

## Feladatok

### Teglalap osztály

- A `Teglalap` természetesen egy speciális síkidom.
  - A `Teglalap` az oldalai hosszával legyen adott.
- Legyen egy két paraméteres konstruktora, ahol megkapja a téglalap 2 oldalhosszát.
- Valósítsd meg az osztályt úgy, hogy példányosítani lehessen!
- A `Negyzet` osztály a `Teglalap` osztályból származzon!
  - A `Negyzet` egy oldalával legyen adott.
- Legyen egy egy paraméteres konstruktora, ahol megkapja a négyzet oldalhosszát.
- A négyzet típusa "negyzet".
- Valósítsd meg az osztályt úgy, hogy példányosítani lehessen!

### Haromszog osztály

- A `Haromszog` természetesen egy speciális síkidom.
  - A `Haromszog` az oldalai hosszával legyen adott.
- Legyen egy három paraméteres konstruktora, ahol megkapja a háromszög 3 oldalának hosszát. Nem kell vizsgálni, hogy ezen adatokból lehet-e háromszög!
- A háromszög típusa "haromszog".

## Rajz osztály

- A Rajz osztály Síkidom típusú objektumokat tárol, megőrizve az eredeti sorrendjüket.
- Az osztály felel az általa eltárolt síkidomok megszüntetéséért.
- Az osztály rendelkezik default konstruktorral.
- A << operátorral adhassunk a rajzhoz síkidomokat, ahol a paraméter Síkidom\* típusú, és ennek másolatát kell eltárolni.
  - Az operator legyen láncba fűzhető!
- Az osztály legyen kiírható a stream-re úgy, hogy soronként kiírja a tárolt síkidomok kerületét.
- Az osztály ++ postfix operátora növelje a rajz objektum minden elemének méretét oly módon, hogy valamennyi oldalának hosszát növeli eggyel!
- Legyen az osztálynak copy konstruktora!
- Legyen az osztálynak értékadó operátora, de ez legyen különleges olyan értelemben, hogy egy adott rajzból csak a téglalapokat adja értékül az operátor bal oldali kifejezésének!