

# Programozás I. Gyakorló feladatsor

SZTE Szoftverfejlesztés Tanszék

2023. tavasz

## Általános követelmények, tudnivalók

- A feladat elkészítési határideje: **vasárnap 23:59:59**. Ez szigorú határidő, a Bíró előre megadott időben zár, pótlásra nincs lehetőség.
- A feladatokat számítógép előtt kell megoldani, tetszőleges fejlesztői környezetben, tetszőleges operációs rendszer segítségével.
- Az elkészült programot **20** alkalommal lehet benyújtani, a megadott határidőig.
- Csak a leírásban szereplő osztályokat, metódusokat és adattagokat kell megvalósítani, egyéb dolgokért nem jár plusz pont.
- A feladat megoldása során minden megadott előírást pontosan követni kell! Tehát, ha a feladat leírása szerint egy adattag neve a "elsoFoku", akkor az alábbi elnevezések nem megfelelőek: "elsőFokú", "elsofoku", "elso\_foku", "elsőFoq". Ugyanez igaz a metódusok, osztályok elnevezésére is!
- A metódusok esetében a visszatérési típus, a név, módosítók és a paraméterek típusai (és azok sorrendje) kerülnek ellenőrzésre, azonban a paraméterek nevei tetszőlegesek lehetnek.
- Az órán tanult konvenciókat követni kell (getter/setter elnevezés, toString, indentálás, stb). Abban az esetben is, ha ezt a feladat külön nem emeli ki, az ellenőrzés során erre is építünk.
- A nem forduló kódok nem kerülnek kiértékelésre, ezt utólagosan a gyakorlatvezető sem bírálhatja felül. (Hiszen mindenki rendelkezésére áll a saját környezete, ahol fordítani, futtatni tudja a forráskódot, így feltöltés előtt ezt mindenképpen érdemes megnézni!)
- Az adattagok és konstruktorok hiányában garantáltan 0 pontos lesz a kiértékelés, mert ezek minden teszt alapját képezik.
- Ha végtelen ciklus van a programban, akkor ezt a Bíró ki fogja dobni 3 másodperc után (ha többször is meghívásra kerül ilyen metódus, akkor ez többszöri 3 másodperc, összesen akár 2 perc is lehet). Ilyenkor NE kattints még egyszer a *Feltöltés* gombra, mert akkor kifagyhat a Bíró, csak a böngésző újraindításával lehet megoldani a problémát (emellett elveszik 1 feltöltési lehetőség is).
- Kérdés/probléma esetén a gyakorlatvezetők tudnak segítséget nyújtani.
- A feladat megoldása során a default csomagba dolgozz, majd a kész forrásfájlokat tömörítve, zip formátumban töltsd fel, azonban a zip fájlt tetszőlegesen elnevezheted!

- Zip készítése: Windowson és Linuxon is lehet a GUI-ban jobb klikkes módszerrel tömörített állományt létrehozni (Windowsban pl. a 7-Zip nevű ingyenes program használatával).
  - Linux terminálon belül például a "zip feladat.zip \*.java" paranccsal is elkészíthető a megfelelő állomány.
- A feladatokban az alábbi dolgok az alapértelmezettek (**kivéve**, ha a feladat szövege mást mond)
    - az osztályok láthatósága publikus
    - az egész érték 32 bites
    - a lebegőpontos számok dupla pontosságúak
    - az olyan metódusok void visszatéréssel rendelkeznek, amelyeknél nincs specifikálva visszatérési típus.
    - a metódusok mindenki számára láthatóak
    - az adattagok csak az adott osztályban legyenek elérhetőek
  - A *riport.txt* és a fordítási log fájlok megtekinthetőek az alábbi módon:
    1. Az *Eredmények megtekintése* felületen a vizsgálandó próba új lapon való megnyitása
    2. A kapott url formátuma:  
`https://biro2.inf.u-szeged.hu/Hallg/IB204L/FELADAT/hXXXXXX/4/riport.txt`
    3. Az url-ből visszatörölve a 4-esig (*riport.txt* törlése) megkaphatók a 4-es próbálkozás adatai.
  - Szövegek összehasonlításánál az egyezés a pontos egyezést jelenti, azaz ha kis-nagy betűben térnek el, akkor már nem tekinthetők egyenlőnek (pl. a "piros" != "Piros")
  - A leírásokban bemutat 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!
  - Az elvárt kimeneteknek karakterről karakterre olyan formátumúnak kell lennie, ami a feladatban le van írva (szóközöket és sortöréseket is beleértve).

## Feladatsor

### 1. feladat

Töltsd le a feladatkírás mellet lévő *minta.zip* fájlt! Csomagold ki egy tetszőleges helyre, majd importáld be a választott IDE-be!

A *Targy.java* és *Ollo.java* fájlok teljesen fel van kommentezve, ezek alapján értelmezd, és próbáld ki az osztályokat!

## 2. feladat: teszteljünk!

A feladat a Targy és az Ollo osztály tesztelése lesz. Ehhez készítsd el a TestTargy és TestOllo nevű publikus osztályokat. Az elkészült teszt osztályoknak JUnit 5 teszteket kell tartalmaznia. A tesztelés pontozása az alábbiak szerint alakul:

### Lefedettség (osztályonként 10 pont)

A tesztelés során a kód mekkora része van lefedve tesztekkel. A tesztelés során végrehajtási ág szerinti lefedettséget fogunk vizsgálni, és ezzel arányosan jár a pontszám. Például, ha a tesztek a végrehajtási ágak 40 százalékát használják, akkor arra 4 pont jár. Hasonlóképpen, ha a tesztek a végrehajtási ágak 65 százalékát használják, akkor arra 6,5 pont jár.

*Tipp: a lefedettséget figyelemmel követheted bármelyik IDE segítségével.*

### Használt assert metódusok (osztályonként 5 pont)

A különböző assert metódusok segíthetnek abban, hogy a tesztelés során minél specifikusabban történjen a tesztelés, hiba esetén pedig a lehető legpontosabb hibaüzeneteket kapjuk. A tesztelés során használt különböző assert metódusok alapján jár a pontszám, az alábbi assertek esetén:

#### TestTargy osztály:

- **assertEquals:** legalább 1 esetén 0,5 pont; legalább 5 esetén 1 pont.
- **assertTrue:** legalább 1 esetén 0,5 pont; legalább 3 esetén 1 pont.
- **assertFalse:** legalább 1 esetén 0,5 pont; legalább 3 esetén 1 pont.
- **assertArrayEquals:** legalább 1 esetén 0,5 pont, legalább 2 esetén 1 pont.
- **assertNotEquals:** legalább 1 esetén 0,5 pont.

#### TestOllo osztály:

- **assertEquals:** legalább 1 esetén 0,5 pont; legalább 5 esetén 1 pont.
- **assertTrue:** legalább 1 esetén 0,5 pont; legalább 3 esetén 1 pont.
- **assertFalse:** legalább 1 esetén 0,5 pont; legalább 3 esetén 1 pont.
- **assertNotEquals:** legalább 1 esetén 0,5 pont.
- **assertNull:** legalább 1 esetén 0,5 pont.
- **assertNotNull:** legalább 1 esetén 0,5 pont.

## Teszt metódusok száma (összesen 10 pont)

Hány metódusra van szétbontva a tesztelés. Figyelj, hogy mindig logikus, lehetőleg apró metódusokat készíts. A pontozás az alábbiak szerint történik (a két osztály tesztjeire összegezve):

- 1 pont, ha legalább 2 teszt metódus van.
- 2 pont, ha legalább 4 teszt metódus van.
- 3 pont, ha legalább 6 teszt metódus van.
- 4 pont, ha legalább 8 teszt metódus van.
- 5 pont, ha legalább 10 teszt metódus van.
- 6 pont, ha legalább 12 teszt metódus van.
- 7 pont, ha legalább 14 teszt metódus van.
- 8 pont, ha legalább 16 teszt metódus van.
- 9 pont, ha legalább 18 teszt metódus van.
- 10 pont, ha legalább 20 teszt metódus van.

Természetesen bármely más assert metódus is használható.

Jó munkát!