

# Programozás I. 3. zh

SZTE Szoftverfejlesztés Tanszék

2024. tavasz

## Általános követelmények, tudnivalók

- A feladat elkészítésére 30 perc áll a rendelkezésre. Ez szigorú határidő, a Bíró előre megadott időben zár.
- 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.
  - Aki Windowst használ, annak a gép elindítása után érdemes egyből a fejlesztőkörnyezetet elindítani, és létrehozni egy új projektet, és csak utána a böngészőt, mivel az elején egy néhány percig indexel, addig pont el lehet olvasni a feladatot.
- Bármely segédanyag használata **tilos** (a fejlesztőkörnyezetek nyújtotta segítségen kívül), aki mégis ilyen tesz, vagy próbálkozik vele, annak a dolgozata nem értékelhető és a ZH nem teljesített. Ha valakinek a padtársa segít, akkor mérlegelés nélkül mindkettő hallgató dolgozata sikertelen, a ZH nem teljesített.
- 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.
- A Java elnevezési 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://biro.inf.u-szeged.hu/Hallg/IB204L-1/1/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 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 a 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).

## A szőlők legyőzése

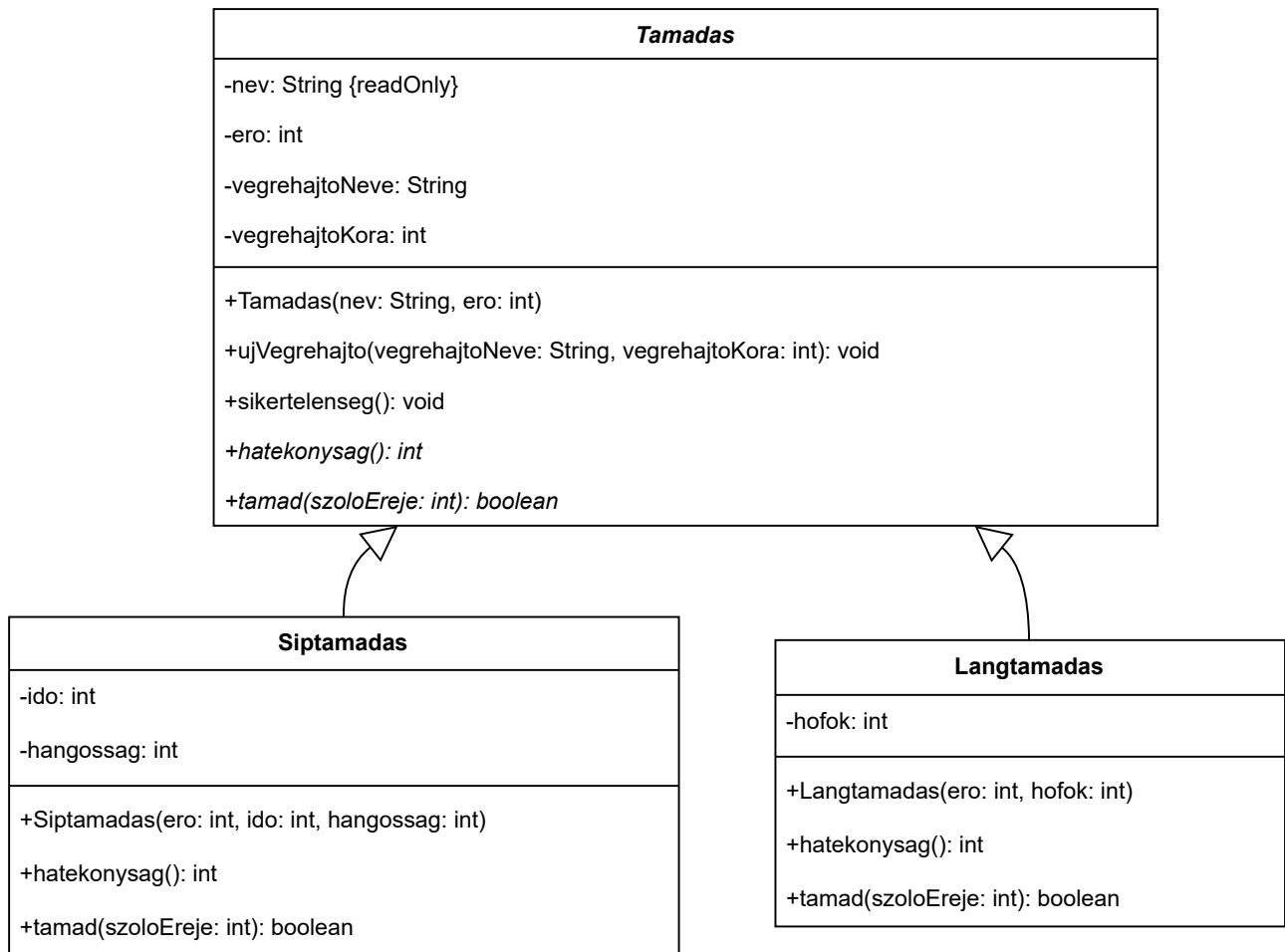
Grapeliena szőlői elkezdtek ellepni az egész világot, ezért a parasztbácsik összefogtak és mágikus ellentámadást indítottak, ahol különböző varázsfegyverekkel próbálják meg a szőlőket legyőzni: első sorban lángtámadással és mágikus sípokkal, amik már akkorák, hogy inkább egy fuvolához hasonlítanak és nem azért mert az ai nem tudott mást generálni

Ezúttal vajon sikerül a parasztbácsiknak legyőzniük Grapeliena szőlő bossait?



A feladat megoldása során használd a `minta.zip` fájlban lévő kiindulási projektet, amelyben megtalálható egy absztrakt osztály.

1. ábra. Osztálydiagram



# Támadás



Az osztály egy támadást reprezentál, amelyet a parasztbácsik a szőlők ellen tudnak bevetni.

*Az osztály már el van készítve, de tetszőleges módon kiegészíthető új adattagokkal és metódusokkal. A meglévő adattagokon és metódusokon változtatni TILOS (sem a nevét, sem a láthatóságát, sem a típusát)!*

Az osztály adattagai:

- **nev:** a támadás neve (pl. nyilzapor, langtamadas, stb.)
- **ero:** a támadás ereje (mennyire erős a támadás)
- **vegrehajtoNeve:** a támadást végrehajtó ember neve (null, ha senki nem hajtja végre),
- **vegrehajtoKora:** a támadást végrehajtó ember életkora (0, ha senki nem hajtja végre).

Az osztályban lévő metódusok:

- **ujVegrehajto:** lecseréli a támadást végrehajtó személyt (nevét és korát),
- **sikertelenség:** törli a támadást végrehajtó személyt (nevet null-ra, kort 0-ra állítja),
- **hatekonysag:** a támadás hatékonyságát megadó érték (absztrakt metódus),
- **tamad:** a támadást elvégző metódus, amely visszaadja, hogy a támadás sikeres volt-e (absztrakt metódus).

## Síptámadás osztály (5 pont)



Készítsd el a `Siptamadas` osztályt a UML diagram alapján, amely egy speciális támadás.

Készítsd el az osztály konstruktorát! A konstruktor az erőt, az időt és a hangosságot várja. A támadás nevét minden esetben a "Siptamadas" szöveggel inicializáld!

Definiáld felül a **hatékonysag()** metódust, amely a támadás hatékonyságával tér vissza!

- Amennyiben a végrehajtó neve null, akkor a metódus 0-val térjen vissza.
- Ha a végrehajtó kora 12-nél kisebb, akkor nem tudja végrehajtani a támadást, így a metódus 0-val térjen vissza.
- Egyéb esetben a támadás hatékonysága megegyezik a hangosság és az idő szorzatával.

Definiáld felül a **tamad()** metódust, amely a támadás végrehajtásáért felelős!

- Ha a végrehajtó neve null, akkor a támadás sikertelen, térjünk vissza hamissal.
- Vizsgáld meg, hogy a támadás sikeres-e: a támadás akkor sikeres, ha a hatékonyság négyzetének és az erőnek a szorzata legalább akkora, mint a paraméterben kapott szülő ereje.
- Ha a támadás sikertelen, akkor a támadást végrehajtó ember meghal: hívd meg az `osztály sikertelenség()` metódusát.
- A metódus adja vissza, hogy sikeres volt-e a támadás.

## Lángtámadás (5 pont)



Készítsd el a `Langtamadas` osztályt a UML diagram alapján, amely egy speciális támadás.

Készítsd el az osztály konstruktorát! A konstruktor csak az erőt és a hőfokot várja. A támadás nevét minden esetben a `"Langtamadas"` szöveggel inicializáld!

Definiáld felül a `hatekonysag()` metódust, amely a támadás hatékonyságával tér vissza!

- Amennyiben a végrehajtó neve null, akkor a metódus 0-val térjen vissza.
- Ha a végrehajtó kora 18-nál kisebb vagy 40-nél nagyobb, akkor a támadás kevésbé hatékony: egészen pontosan a hőfok fele.
- Egyéb esetben a támadás hatékonysága megegyezik a hőfokkal.

Definiáld felül a `tamad()` metódust, amely a támadás végrehajtásáért felelős!

- Ha a végrehajtó neve null, akkor a támadás sikertelen, térjünk vissza hamissal.
- Vizsgáld meg, hogy a támadás sikeres-e: a támadás akkor sikeres, ha a hatékonyság és az erő szorzata legalább akkora, mint a paraméterben kapott szőlő ereje.
- Ha a támadás sikertelen, akkor a támadást végrehajtó ember meghal: hívd meg az `oszosztaly sikertelenseg()` metódusát.
- A metódus adja vissza, hogy sikeres volt-e a támadás.

Jó munkát!