

Programozás I. Gyakorló feladatsor

SZTE Szoftverfejlesztés Tanszék

2024. 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.
- 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!

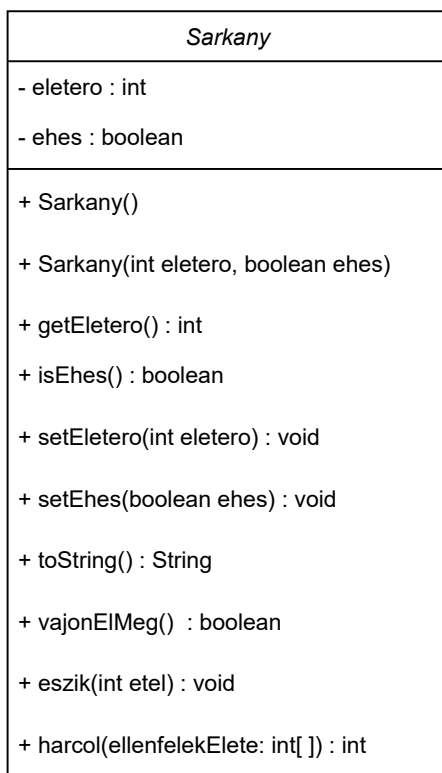
- Zip készítése: Windowson és Linuxon is lehet a GUI-ban jobb klikkes módszerrel tömörített állományt létrehozni (Windowsban pl. a 7-Zip nevű ingyenes program használatával).
- Linux terminálon belül például a "zip feladat.zip *.java" paranccsal is elkészíthető a megfelelő állomány.
- A feladatokban 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://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 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).

Feladatsor

Írj egy *Sarkany* nevű osztályt, ami a programbeli sárkányainkat reprezentálja. Az adattagokat, valamint a szükséges metódusokat a ?? ábrán láthatjuk. Ügyelj a megfelelő láthatóságok használatára! **(1 pont)**

A *setEletero* paraméterben várja az új életerőt. Ha az negatív, 0 értéket állítson be, és írja ki az alapértelmezett hibacsatornára, hogy "megolted a sarkanyt". A kiíratásban légy karakterpontos, és a sort zárja sortörés. **(1 pont)**

1. ábra. Sárkány osztálydiagram



A `setEhese` paraméterben várja az új értéket. Ha a sárkány életerege kevesebb, mint 50, akkor mindenképpen igazra állítja, ellenkező esetben a kapott paramétert állítja be. **(1 pont)**

Alapértelmezetten a sárkány életerege 100, és éhes, így készítsd el az alapértelmezett konstruktort. **(1 pont)**

A paraméteres konstruktor kövesse a settereknél leírt szabályokat, azonban hiba esetén ne írjon semmit a hibacsatornára. **(1 pont)**

Definiáld felül az osztályban a `toString` metódust úgy, hogy az az alábbi szöveggel térjen vissza: "A sarkany életerege `eletero`, es jelenleg `ehese`-e `ehese`." Az `eletero` helyére az életerőt írja, az `ehese`-e helyere pedig a következő kifejezést:

- ha éhes, akkor azt, hogy "rettenetesen"
- ha nem éhes, akkor azt, hogy "veletlenül nem" **(1 pont)**

A `vajonElMeg` metódus igaz értékkel tér vissza, ha az életerő pozitív, különben hamissal. **(1 pont)**

Az `eszik` metódus a paraméterül kapott értéket hozzáadja az életerőhöz. Ha nem pozitív számot kapott, akkor ne adjon hozzá semmit, hanem írja ki a hibacsatornára: "en a helyedben nem eheztenek egy sarkanyt". **(1 pont)**

A `harcol` metódus a sárkány ellenfelei életerejének tömbjét kapja paraméterként. Egy sárkány csak akkor hajlandó küzdelemre, ha éhes, ekkor addig harcol, amíg el nem fogynak az

ellenfelei, vagy meg nem hal, egyébként azonnal elmenekül. A sárkány sorban haladva küzd meg az ellene felvonult ellenfelekkel, minden ellenfél az életerejének felét (lefelé kerekítve) képes belesebezni a sárkányba. A módszer a legyőzött ellenfelek számával térjen vissza, ha a sárkány elesik az ütközet során, azt az ellenséget nem kell számolni, amelyik sebzése a veszét okozta. Értelemszerűen, ha elmenekül, akkor 0 ellenfelet győzött le. **(2 pont)**

Jó munkát!