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).

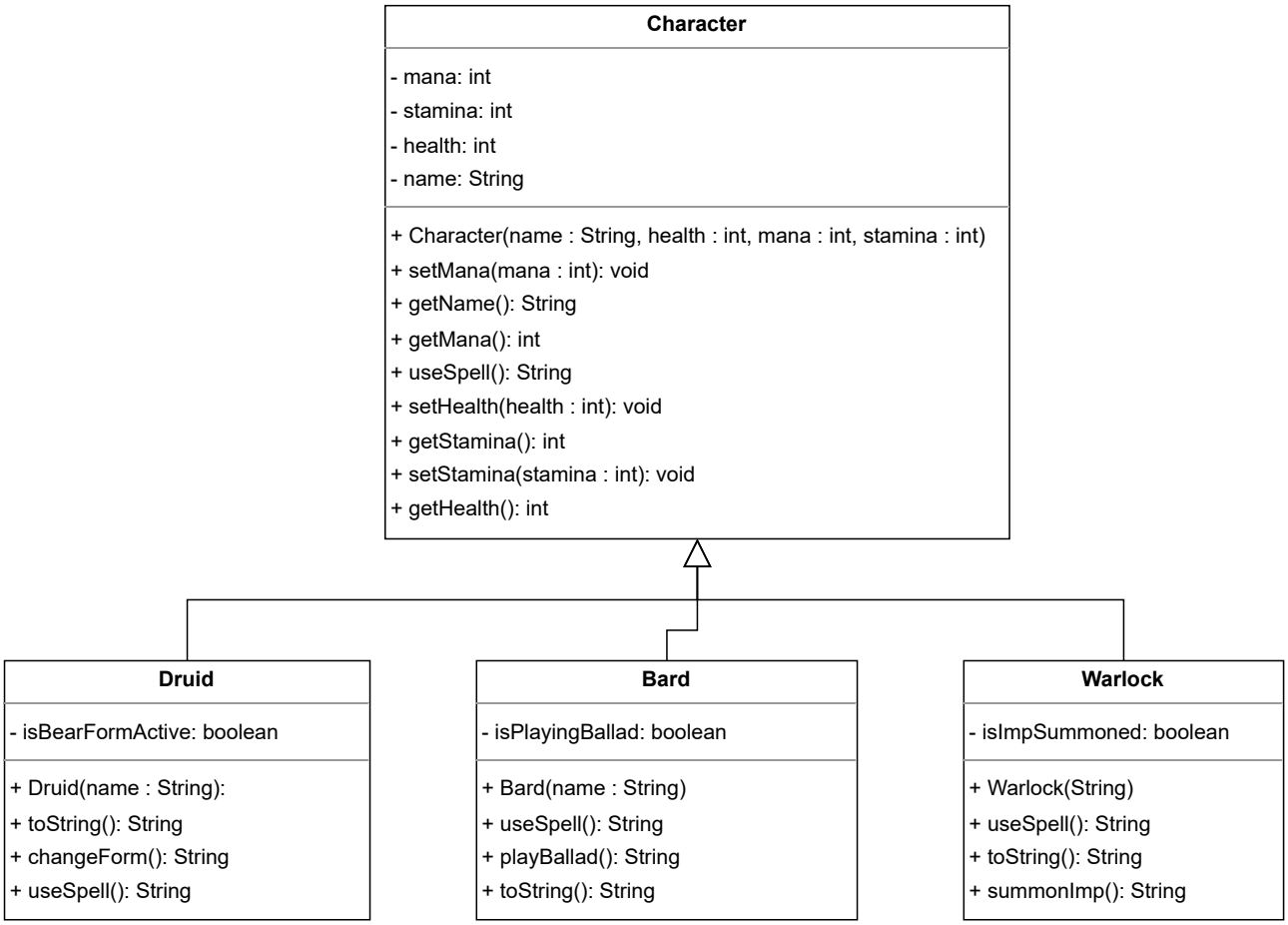
Feladatsor

Character és a leszármazott osztályok (15 pont)

Írj egy `Character` nevű osztályt, ami egy alap karaktert reprezentál. Az adattagokat, valamint a szükséges metódusokat a **??**. ábrán láthatjuk. Ügyelj a megfelelő láthatóságok és getter/setterek használatára.

Készítsünk a **Character**-ből öröklődő osztályokat, melyek rendre legyenek a **Druid**, **Bard** és **Warlock**.

1. ábra. Character osztálydiagram



Metódusok

A **Character** osztályban található `useSpell()` metódus egy üres Stringgel térjen vissza, és ezen kívül ne csináljon semmit.

A **Character** ősosztályban lévő `useSpell()` metódust írd felül minden gyerek osztályban a következő módon.

- **Druid**: írja ki, hogy "`Moonbeam`" és használjon el 3 mana pontot.
- **Bard**: írja ki, hogy "`Charm Person`" és használjon el 2 mana pontot.
- **Warlock**: írja ki, hogy "`Eldritch Blast`", és használjon el 4 mana pontba kerül.

Abban az esetben, ha nincs a karakternek elég mana pontja, hogy az adott varázslatot használni tudja, írja ki a "`Not enough mana.`" stringet.

Konstruktorok

Az osztályok alapértelmezett értékeit a következőképpen állítsd be a konstruktoraikban:

- **Druid**: `Health`: 10, `Mana`: 6, `Stamina`: 4.

- **Bard**: `Health`: 6, `Mana`: 5, `Stamina`: 10.
- **Warlock**: `Health`: 5, `Mana`: 10, `Stamina`: 2.

Ezen felül lehessen egy neve is a karakternek.

Osztályspecifikus metódusok/adattagok

A **Druid** át tud változni medve formába a `changeForm()` nevű metódussal, ami egy boolean értéket állít át, hogy maci formában van-e. Ennek a költsége 2 stamina. A visszaváltozás nem használ el staminát.

- Amikor átvált maci formába, térjen vissza a `"Bear form is on."` szöveggel.
- Amikor kikapcsolja a medve formát, térjen vissza a `"Bear form is off."` szöveggel.

A **Bard** tud zenélni a `playBallad()` nevű metódussal, ami szintén egy boolean értéket állít, hogy éppen zenél-e vagy nem. Ez a mutatvány 3 staminát használ el, amikor bekapcsolja. A zenét abbahagyni itt sem használ el staminát.

- Amikor elkezd zenélni, térjen vissza a `"Ballad is on."` stringgel.
- Amikor abbahagyja a zenélést, térjen vissza a `"Ballad is off."` stringgel.

A **Warlock** le tud idézni egy imp segítőt maga mellé a `summonImp()` nevű metódussal. Az, hogy az imp éppen a Warlock mellett van vagy nem szintén egy boolean érték. Ez a varázslat 1 stamina és 2 health pontba kerül. Az impet vissza küldeni nem használ sem staminát, sem életet.

- Amikor leidézi a túlvilági segítőjét, térjen vissza az `"Imp summoned."` stringgel.
- Amikor elküldi a segítőjét, akkor térjen vissza az `"Imp sent away."` stringgel.

`Fontos`: Az adott boolean változót minden osztály metódusának újra meghívásával lehessen visszaállítani.
Amikor nem tudják már újra használni a képességet, akkor egységesen a `"No can do."` szöveggel térjenek vissza a metódusok.

Definiáld felül az osztályokban a `toString` metódust úgy, hogy kiírja az osztály adattagjait az alábbi formátumban: `"<Class name>: Health:<Actual Health>, Mana:<Actual Mana>, Stamina:<Actual Stamina>"`.
Például: `Druid: Health:10, Mana:2, Stamina:4`

Jó munkát!