PNy Java ZH gyakorlati feladat 2022.06.02. 15:30

Határidő Nincs megadva határidő **Pont** 20 **Kérdések** 1
Elérhető jún 2, 15:45 - jún 2, 17:30 körülbelül 2 óra **Időkorlát** Nincs
Engedélyezett próbálkozások Korlátlan

Ezt a kvízt ekkor zárolták: jún 2, 17:30 .

Próbálkozások naplója

	Próbálkozás	Idő	Eredmény
LEGUTOLSÓ	1. próbálkozás	96 perc	18 az összesen elérhető 20 pontból

Ezen próbálkozás eredménye: **18** az összesen elérhető 20 pontból
Beadva ekkor: jún 2, 17:28
Ez a próbálkozás ennyi időt vett igénybe: 96 perc

1. kérdés **18 / 20 pont**

Programozási Nyelvek Java ZH, 2022.06.02. 15:30 turnus

Feltételek

A kódról.

- Feltételezhető, hogy meghíváskor érvényes adatokat kapnak a konstruktorok és metódusok: pl. a megkapott fájlnevek értelmesek, a hivatkozások nem `null` értékűek stb.
  - Kivétel: a szöveg explicit kérheti bizonyos adatok ellenőrzését.
- A kód legyen jó minőségű, a tanultaknak minél inkább megfelelő.
  - A mezők láthatósága mindig a lehető legszűkebb legyen.
  - A metódusok legyenek nyilvános láthatóságúak.
  - Legyen jól működő egységbe zárás: ne legyen adatszivárgás.
- A megoldás elkészítéséhez a **Java API dokumentáció (https://kitlei.web.elte.hu/exammaterials/java-api/)** használható, más segédeszköz nem.
  - Ha valami nem egyértelmű, érdemes az oktatóktól kérdezni.
  - A részfeladatokat a leírt sorrendben kell megoldani.

A beadásról.

- A félreértések elkerülése végett: a ZH napján tilos megosztani az elkészített megoldást bármilyen módon.
- Az elkészített megoldás forrásfájljait **zip** formátumba csomagolva kell feltölteni a Canvasbe.
  - A `zip` csak a `java` forrásfájlokat tartalmazza.
  - A **ZH végén kb. 10 percet érdemes fenntartani** a kód tisztázására, fordíthatóvá tételére, tömörítésére, beküldésére.

Feladat: Lukid bolygó lakóinak részleges virtuális megvalósítása

Készíts egy implementációt a Lukid bolygó lakosainak a következőkben meghatározott módon.

Alapvető tudnivaló, hogy ezen a bolygón olyan lakosok élnek, amelyek a nevük alapján egyértelműen beazonosíthatóak. Tehát nem létezhet két azonos nevű lukid a bolygón.

`NamedEntity` osztály

Hozz létre `lukidPlanet.NamedEntity` néven osztályt, amely lekezeli, hogy egy adott névvel csak egyetlen példány legyen létrehozható. Ehhez legyen egy `Set` típusú osztályszintű mezője, amiben valamilyen tetszőleges megvalósítást választva azt tárold el, hogy milyen nevek lettek már lefoglalva. Az osztályból származó osztályok láthassák ezt a mezőt, esetleges módosítások miatt, viszont mások ne.

Legyen `name` nevű adattagja, amelyet getteren keresztül érhetnek kívülről el.

A konstruktor oldja meg, hogy ne lehessen többször azonos névvel létrehozni objektumot, amennyiben kétszer próbálnának meg létrehozni azonos névvel ilyen típusú példányt, váltson ki `lukidPlanet.errors.OccupiedNameException` kivételt a már foglalt névvel.

A `NamedEntity` osztályt ne lehessen közvetlenül példányosítani, de a leszármazott számára legyen látható a konstruktor.

`OccupiedNameException` osztály

Hozz létre `lukidPlanet.errors.OccupiedNameException` néven ellenőrzött kivétel osztályt, amelynek paraméteres konstruktora az ősosztály paraméteres konstruktorát hívja meg azzal a kiegészítéssel, hogy `The following name is occupied: <lukid>`, ahol `<lukid>` a szöveg típusú paraméter.

Legyen továbbá egy paraméter nélküli konstruktor is definiálva, ahol az ősosztály paraméter nélküli konstruktora hívódik meg.

`Lukid` osztály

Készíts `lukidPlanet.Lukid` néven osztályt a Lukid bolygó lakosainak.

Az osztály származzon a már megírt `NamedEntity` osztályból. A konstruktora ne legyen látható kívülről, viszont a lefutása végén írja ki a standard kimenetre, hogy `<lukid> has been born.`, ahol `<lukid>` a konstruktor paramétere.

Hozz létre egy gyártóműveletet `createLukid` néven, amely a paraméterül kapott névvel megpróbál létrehozni egy egyedet, amelynek sikertelensége (kivétel) esetén a következőt írja ki: `This name is occupied: <lukid>`, ahol `<lukid>` a függvény paraméterében szereplő név.

Hozz létre egy `die` nevű függvényt, ami az adott példányt kitörli a bolygóról, tehát ezentúl lehessen létrehozni ilyen nevű lukidot ismét. Amikor ez a függvény lefut, legyen a standard kimeneten a következő üzenet: `<lukid> has died.`, ahol `<lukid>` jelöli az adott nevet, amelyet lekér a `NamedEntity`-ből a megfelelő módon.

Hozz létre két osztályszintű adattagot, amelyek a két őslukidot valósítsák meg, melyeknek nevei legyenek `"Luko"` és `"Luka"`, ezen példányok halhatatlanok, ezt a `die` függvényben kezelni kell a következő hibaüzenettel: `"Luko and Luka are immortal."`.

Hozz létre `equals` metódust, amely megfelel a "szerződésben elvártaknak", és úgy különböztesse meg a két objektumot, hogy nem csak név alapján, hanem létrehozás sorszáma alapján is. (Tehát vezess be valamilyen id-szerű viselkedést.) (Ez a jelenség akkor figyelhető meg, ha egy Lukid-ra meghívjuk a `die` függvényt, majd utána ugyanilyen névvel létrehozunk egy újabbat.) Írj `hashCode` metódust, amely megvalósítja azokat a feltételeket, amelyek elvárunk egy ilyen metódustól.

Ebben az osztályban definiáld felül a `NamedEntity` getterét úgy, hogy a név elé kerüljön egy `"Lukid "` szövegrész.

`LukidMain` osztály

Hozd létre a névtelen csomagban a futtatható `LukidMain` osztályt.

Legyen a futtatás során négy Lukid létrehozva: `l1`, `l2`, `l3`, `l4`.
Először `l1` "Luki" névvel, majd `l2` "Luko" névvel, majd `l3` "Luka" névvel. Ezután írasd ki l1 `hashCode`-ját, majd töröld `l1`-et a `die` függvényhívással. Hozz létre egy újabb Lukid-ot "Luki" névvel, és hasonlítsd össze `l1` és `l4`-et az `equals` metódussal és írasd ki az eredményt. Írd ki `l4` `hashCode`-ját is a végén és hasonlítsd össze a korábbi `l1` `hashCode`-jával.

JAVA_TóthZsoltDániel.zip
(https://canvas.elte.hu/files/1781989/download)

Kvízeredmény: **18** az összesen elérhető 20 pontból