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 ont.
- 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!

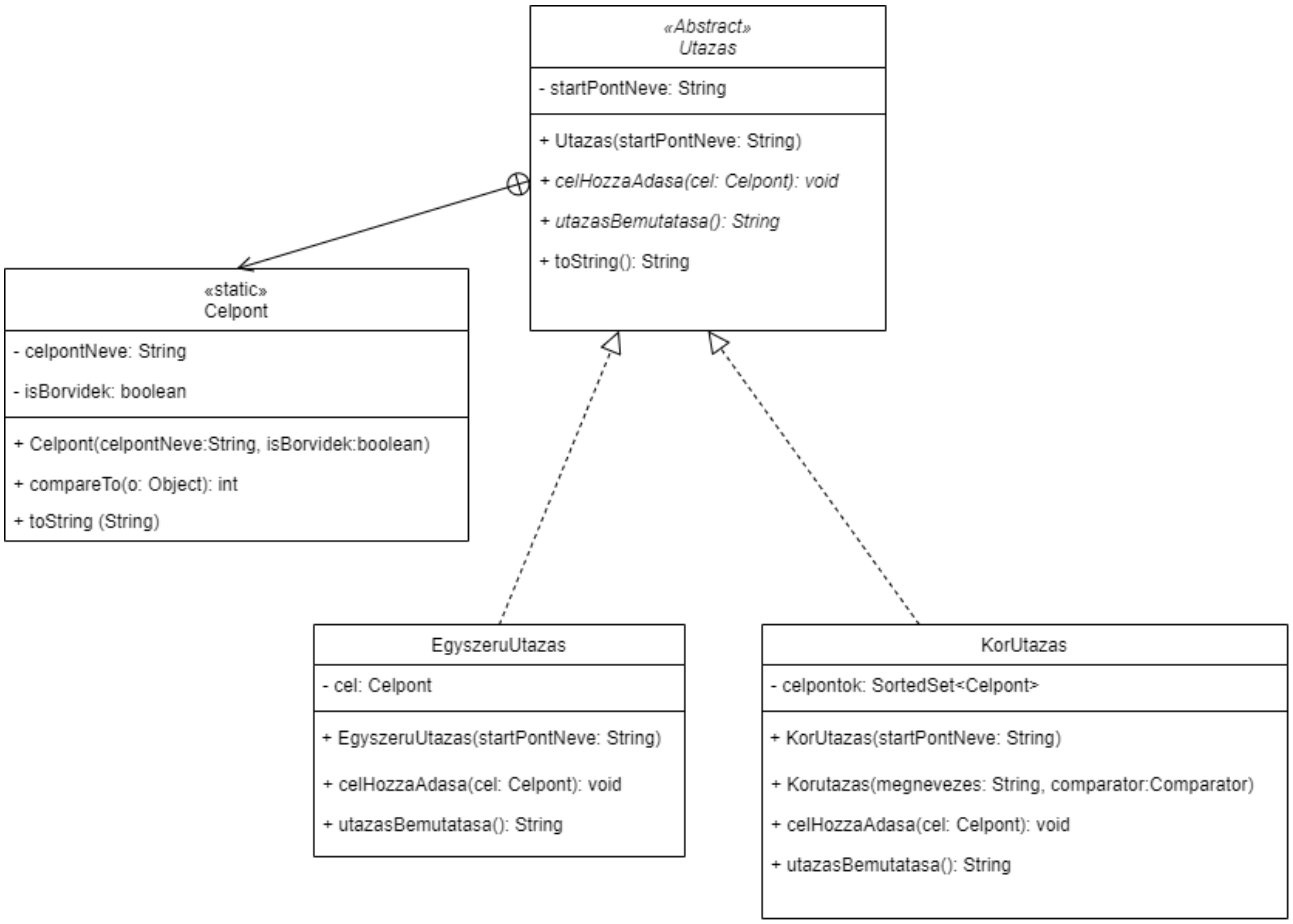
- Az elvárt kimeneteknek karakterről karakterre olyan formátumúnak kell lennie, ami a feladatban le van írva (szóközöket és sortöréseket is beleértve).

Utazás

1. Utazás (5 pont)

Készítsük el az `Utazas` absztrakt osztályt a diagramon látható adattagokkal és metódusokkal.

1. ábra. A feladathoz tartozó osztálydiagram



Definiáljuk felül az osztályban a `toString` metódust, ami az osztály specializációinak megfelelően meghívja és visszatér a `utazasBemutatasa` metódus visszatérési értékével.

2. Utazás célpont (4 pont)

Hozzunk létre egy `Celpont` osztályt, melyben definiáljuk felül a `toString` metódust, ami a `celpontNeve` adattag értékével térjen vissza.

3. Egyszerű utazás (5 pont)

Készítsünk egy `EgyszeruUtazas` osztályt, melynek adattgjai és metódusai a diagramról olvashatók le.

A `utazasBemutatasa` metódus térjen vissza a következő sztringgel: "*{startPontNeve} - {cel.celpontNeve} ({(borvidek)|(nem borvidek).})*" szöveggel, ahol a mondat végén levő borvidék/nem borvidék kiíratás attól függ, hogy az adott célpont borvidék-e. Amennyiben a `cel` adattag még nincs beállítva, akkor a visszaadott érték az "Ez az utazas nincs meg megtervezve." üzenet legyen.

4. Körutazás (6 pont)

Készítsünk egy `KorUtazas` osztályt.

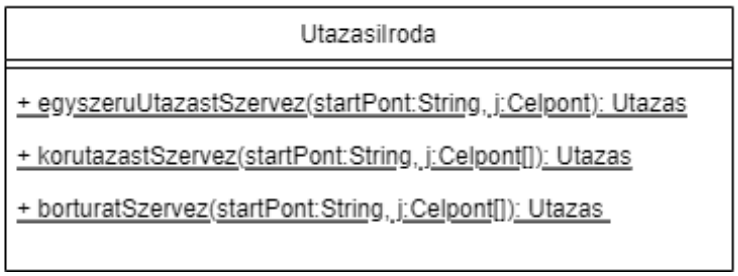
Az osztály konstruktora, mely a `String` attribútumot várja paraméterül, egyszerűen inicializálja a `celpontok` halmazt.

A `utazasBemutatasa` metódus térjen vissza a következő sztringgel: "*{startPontNeve} - {cel.celpontNeve} ({(borvidek)|(nem borvidek)}) - ... - {cel.celpontNeve} ({(borvidek)|(nem borvidek).})*" szöveggel, ahol a célpontokat a halmaz egyszerű bejárásával soroljuk fel, és ahol a borvidék/nem borvidék kiíratás nyilvánvalóan attól függjön, hogy az adott célpont borvidék-e. Amennyiben az utazásnak még nincsenek célpontjai, akkor a visszaadott érték az "Ez az utazas nincs meg megtervezve." üzenet legyen.

5. Utazási iroda (5 pont)

Hozzunk létre egy `UtazasiIroda` osztályt, a rendszerfunkcionalitások kipróbálása érdekében. Az osztály nem kell, hogy példányosítható legyen, valamennyi metódust az osztályon keresztül tudjuk elérni.

2. ábra. UtazasiIroda diagram



A feladat 3 metódus megvalósítása, melyek az `egyszeruUtazastSzervez`, a `korUtazastSzervez`, valamint a `borturatSzervez`.

A `borturatSzervez` metódusban létrehozott `KorUtazas` objektumot, úgy inicializáljuk, hogy a `celpontok` rendezése kövesse azt a szabályt, hogy azon célpontok, amik borvidékek, előzzék meg a nem borvidék célpontokat, ugyanakkor egy kategórián belül legyenek a célpontok ábécé szerint rendezve.

Jó munkát!