

DEBRECENI EGYETEM

Magasszintű programozási nyelvek 2

2022/2023 I félév
Zárthelyi Dolgozat

Általános információk

Hozzon létre egy *IntelliJ* projektet (de akár használhat más fejlesztői környezetet is), melynek neve az Ön **neptunkódja**! A ZH feladatokat ebben implementálja Java nyelven, majd munkája végeztével ugyanilyen néven tömörítse az elkészült forrás állományokat, és küldje el a megadott email címre!

A kicsomagolás után nem beazonosítható dolgozatok automatikusan elégtelenek! Aki elégtelent kap, azoknak lesz egy pót ZH lehetőség a vizsgaidőszak elején!

A dolgozat megírása során semmilyen segédanyagot, órai feladatot nem használhat!

A dolgozat megírása során semmilyen kommunikációs csatorna (levelezőrendszer, chat, stb. . .) nem lehet nyitva és nem léphet kapcsolatba senkivel.

Bármilyen a fentiekre utaló magatartás esetén azonnal elégtelen a ZH eredménye!

Törekedjen rész megoldásokra!

A dolgozat fókuszában az osztálystruktúra és a program összetettsége áll. Éppen ezért ahol nem fejtjük ki, hogy egy metódus milyen módon működik, ott törekedjen a legegyszerűbb megvalósításra!

Pontozás és idő

Minden gondolatjellel (-) kezdődő sor 3 pontot ér. Ez elérhető maximális pontszám $40 * 3$ pont = 120 pont. Részpontok adhatók. Ponthatárok:

- Elégtelen(1): 0% ... 50%,
- Elégséges(2): 51% ... 60%,
- Közepes(3), 61% ... 70%,
- Jó(4): 71% ... 80%,
- Kiváló(5): 81% ... 100% .

A dolgozat 90 percig írható.

Feladatkiírás

A zárthelyi dolgozatban egy 3×2 km-es területen elhelyezett szenzorok (hőmérők és kamerák) tárolását és menedzselését végző alkalmazást kell implementálnia!

A feladat megoldásához mellékelve kap néhány fájlt forráskódokkal, melyeket beilleszthet a forráskódjába! Éppen ezért figyeljen arra, hogy kövesse a dolgozat által előírt elnevezési konvenciókat!

Csatolt forrásfájlok

Annak érdekében, hogy ne kelljen mindent teljesen előlről megírnia, mellékelten talál néhány forrásfájlt. Kezdje azzal, hogy a **Pozicio.java** fájlt behúzza a projektbe! A **szenzorok.csv** fájlt tegye a

C:\Temp

alkönyvtárba. A **Kepformatum.java**-t is húzza be! A **Main.java**-t vagy húzza be, vagy a meglévő **Main.java** tartalmát írja felül a megadottal! A **toString.java** is behúzható, de abban csak kimásolandó kódrészletek vannak.

Interfészek

IKAMERA

- Hozzon létre egy interfészt **IKamera** néven, és hozza létre benne az alábbi metódusok szignatúráit!
- **ketpetKeszit** nevű paraméter nélküli szöveges típussal visszatérő függvény.
- **getFormatum** nevű Kepformatum típusú függvény.

IHOMERO

- Hozzon létre egy interfészt **IHomero** néven, és hozza létre benne az alábbi metódusok szignatúráit!
- **homersekletetMer** nevű paraméter nélküli valós értékkel visszatérő függvény.
- **hatarokatBeallit** nevű kettő darab egész értékű paraméterrel (egész: alsoHatar, egész: felsoHatar) rendelkező eljárás.

Osztályhierarchia

A program lényegi részét a különböző szenzorok képezik. Ezek alapja a *Szenzor* absztrakt osztály, amelynek két gyermekosztálya van, a *Homero*, és a *Kamera*.

SZENZOR

- Hozzon létre egy absztrakt osztályt **Szenzor** néven, és implementálja benne az alábbiakat!

Mezők

- **pozicio** nevű Pozicio típusú mező, melynek van kívülről hívható **gettere** és csak gyermek osztályok által hívható **settere**. Maga a mező legyen a gyermek osztályok számára sem elérhető.

Konstruktor

- Készítsen egy konstruktort, mely bekéri a pozíciót és eltárolja az abban található értéket!

Metódusok

- **isAktiv** nevű igaz/hamis típusú public absztrakt metódus.
- **adatkuldes** nevű paraméter nélküli **absztrakt** eljárás.
- A **Szenzor** osztály felüldefiniálja az Object osztályban lévő **clone** metódust! Mivel az osztály absztrakt (nem lehet belőle példányosítani), így a kapott **clone** metódust tegye **absztrakttá**!
- Írja felül a **toString** metódust! A mellékelt fájlban megtalálja az ide másolható forráskódot!

KAMERA

- Hozzon létre egy osztályt **Kamera** néven, jelölje meg ősként a **Szenzor** osztályt, és implementálja az **IKamera** interfészt!

Mezők

- **formatum** nevű Kepformatum típusú private mező, melynek van kívülről hívható **gettere**, ami megvalósítja az **IKamera** interfészben megadott **getFormatum** metódust.

Konstruktor

- Készítsen konstruktort, mely bekér két egész számot, mint a pozíció x és y koordinátái, valamint egy képformátumot! A koordinátákból készítsen Pozíció példányt és adja át az ős konstruktorának! A képformátumot a konstruktor törzsében mentse el a **formatum** mezőbe!

Metódusok

- **isAktiv** nevű igaz/hamis típusú kívülről hívható metódus. Ez az ősben lévő **isAktiv** metódus kifejtése. Megvizsgálja az aktuális időt, majd 7 és 21 óra között igaz értékkel tér vissza, egyébként hamissal (7 : 00-re és 21 : 00-re is igazat ad, de már 21 : 01-re hamisat).
- **kepetKeszit**: Ha a szenzor nem aktív, dobjon kivételt! Ehhez készítsen saját kivételt **SzenzorInaktivException** néven! Ez a kivétel származzon a **RuntimeException** osztályból! Ellenkező esetben térjen vissza az "elkészített" kép generált nevével az alábbi minta szerint, melyben az aktuális dátumot és időt kell használni, illetve a kamera pozíciójának x és y koordinátáit! Pl.:

2022-05-13 _16-53 _x250 _y1780.PNG

- **adatkuldes**: Meghívja a **kepetKeszit** függvényt, és a generált fájlnevet felhasználva a minta szerint kiírja a standar ouput-ra az alábbi szöveget. Pl.:

Kép mentve: 2022-05-13 _16-53 _x250 _y1780.PNG

- **clone**: készítsen egy klónt az adott példánnyal és térjen vissza vele!
- Írja felül a **toString** metódust! A mellékelt fájlban megtalálja az ide másolható forráskódot!

HŐMÉRŐ

- Hozzon létre egy osztályt **Homero** néven, jelölje meg ősként a **Szenzor** osztályt, és implementálja az **IHomero** interfészt!

Mezők

- **alsoHatar** nevű egész típusú private mező, melynek van **gettere** és kívülről nem látható **settere**. A setter ellenőrizze, hogy az alsó határ nem lehet -60°C -nál kisebb! Ellenkező esetben dobjon saját kivételt **AlacsonyAlsoHatarException** néven! Ez a kivétel származzon a **RuntimeException** osztályból!
- **felsőHatar** nevű egész típusú mező, melynek van **gettere** és kívülről nem látható **settere**.
- **aktív**: egy igaz/hamis private mező. Írjon egy **setAktív** setter metódust a szokott módon.

Konstruktor

- Készítsen konstruktort, mely bekér két egész számot, mint a pozíció x és y koordinátái, valamint két további egész számot (alsó határ és felső határ)! A koordinátákból készítsen Pozíció példányt és adja át az ős konstruktorának! Mentse el a határokat a **hatarokatBeallit** metódus segítségével, lásd lejjebb, és állítsa aktívra a szenzort!

Metódusok

- **isAktív**: Az **isAktív** metódus megvalósítja az ős **isAktív** metódusát, egyszerűen visszaadja az **aktív** mező értékét.
- **hatarokatBeallit**: Elmenti a paraméterben kapott értékeket a megfelelő setterek segítségével. Megvalósítja az **IHomero** azonos nevű metódusát.
- **homersekletetMer**: Ha a szenzor nem aktív, dobjon kivételt! Ehhez (ha még nincs) készítsen saját kivételt **SzenzorInaktivException** néven! Ez származzon a **RuntimeException** osztályból! Ellenkező esetben térjen vissza egy véletlenül generált 2 tizedesjegy pontosságú valós értékkel az $[alsoHatar, felsőHatar)$ intervallumból!
- **adatkuldes**: Meghívja a **homersekletetMer** függvényt, és a generált értéket felhasználva a minta szerint megjeleníti az alábbi szöveget. Pl.:

Hőmérséklet a(z) (50;730) pozíción 2022.05.13 16:53 időpontban: 32,05°C

- **clone**: készítsen egy klónt az adott példánnyal és térjen vissza vele!
- Írja felül a **toString** metódust! A mellékelt fájlban megtalálja az ide másolható forráskódot!

Konténerosztály

A konténerosztály feladata kezelni a szenzorok listáját, és bizonyos lekérdezések eredményeit szolgáltatni.

SZENZORHÁLÓZAT

- Hozzon létre egy osztályt **SzenzorHalozat** néven és implementálja az **Iterable<Szenzor>** interfészt!

Mezők

- **szenzorok**: kívülről nem elérhető **Szenzor** típusú adatokat tartalmazó lista. Ne írjon hozzá se getter-t, se setter-t!

Metódusok

- **telepit**: Szenzor típusú paraméterrel rendelkező eljárás. Feladata elmenteni a paraméterben kapott szenzort a listába. Nem kell ellenőriznie semmit!
- **aktivSzenzorok**: Gyűjtse listába az aktív szenzorok klónjait és térjen vissza azzal!
- **iterator()**: Adja vissza az aktív szenzorok klónjait fordított sorrendben!

Főprogram

Húzza be a projektbe a mellékelt Main.java fájl tartalmát, vagy másolja be a meglévő Main.java-ba a tartalmát! Ez tartalmazza az adatok beolvasásának vázát!

- Készítse fel a beolvasást arra, hogy esetleg rossz fájlnevet adtak meg: Írja meg az ehhez tartozó elkapó részt!
- Készítse fel a beolvasást az Ön által definiált kivételekre! Úgy oldja meg, hogy csak a hibás sor ne legyen beolvasva, az utána levők ettől függetlenül működjenek!

További feladatok:

- Jelenítse meg az aktív szenzorokat az iterátor segítségével egy foreach ciklusban!