

Project (/javagyak/java-fiz-201672-bead1)

Activity (/javagyak/java-fiz-201672-bead1/activity)

f

You won't be able to pull or push project code via SSH until you **add an SSH key (/profile/keys)** to your profile

Don't show again (/profile?user%5Bhide_no_ssh_key%5D=true) | **Remind later**



java-fiz-201672-bead1 ▀

925c5dcf (/javagyak/java-fiz-201672-bead1/commit/925c5dcfa8bd0cec30bb5e1201c63056f7c6c594)

Update README.md (/javagyak/java-fiz-201672-

bead1/commit/925c5dcfa8bd0cec30bb5e1201c63056f7c6c594) · about 4 hours ago by javagyak (/javagyak)

Általános tudnivalók

Ebben az ismertetésben az osztályok, valamint a minimálisan szükséges metódusok leírásai fognak szerepelni. A feladatmegoldás során fontos betartani az elnevezésekre és típusokra vonatkozó megszorításokat, illetve a szövegek formázási szabályait.

Segédfüggvények létrehozhatóak, a feladatban nem megkötött adattagok és elnevezéseik is a feladat megoldójára vannak bízva. Törekedjünk arra, hogy az osztályok belső reprezentációját a lehető legjobban védjük, tehát csak akkor engedjük, és csak olyan hozzáférést, amelyre a feladat felszólít, vagy amit azt osztályt használó kódrészlet megkíván! Ami az osztályok leírásában nem szerepel, az nem lehet publikus, a helytelen láthatóság-szabályozás pontlevonással jár! Törekedjünk arra is, hogy a megírt forráskód kellően általános és újrafelhasználható legyen.

A beadott megoldásodnak működnie kell a mellékelt tesztprogramokkal, de ez nem elégséges feltétele az elfogadásnak. A megírt forráskód legyen kellően általános és újrafelhasználható!

A megoldást egyetlen, .zip kiterjesztésű tömörített állományban kell beadni, amely csak és kizárólag .java kiterjesztésű forrásokat tartalmazhat, a feladat leírása szerint megkövetelt könyvtárszerkezetben. A beadási hely Linux alatt az `smb://nas2.inf.elte.hu/zh/java` Windows alatt a `\\nas2.inf.elte.hu\zh\java` mappára. A fordítás során létrejövő class állományokat viszont nem szabad már mellékelni!

A fordításhoz a Java Standard Edition 8 használata kötelező.

🔗 Tesztelési ékezetekkel Windows alatt

Érdemes a fordítást `javac -encoding utf8` kapcsolóval futatni Windows-on, az ékezetek miatt.

Használható segédanyagok: Java API dokumentáció (<http://docs.oracle.com/javase/8/docs/api/>), legfeljebb egy üres lap és toll. Ha bármilyen kérdés, észrevétel felmerül, azt a felügyelőknek kell jelezni, nem a diáktársaknak!

```
Project=(%PATH%\C:\Program Files\Java\jdk1.8.0_101\bin\java -fiz-201672-bead1/activity) F
```

Figyelem! Az a metódus, amely fordítási hibát tartalmaz, automatikusan *nulla* pontot ér!

A tesztelést végző osztály neve `SzimulatorTeszt` . Ezt Linux alatt így lehet futtatni:

```
> java -cp .:simulatortesztelo.jar SzimulatorTeszt
```

2/5

Valósítsunk meg egy olyan `Gimnazista` osztályt amely megengedi, hogy egy gimnazistát az osztályfőnök foglalkoztassa és hazaküldhesse, ha a figyelmeztetések száma meghaladja az intő határát, amely legyen 100 figyelmeztetés.

Project (/javagyak/java-fiz-201672-bead1) **Activity (/javagyak/java-fiz-201672-bead1/activity)** **Issue (/javagyak/java-fiz-201672-bead1/issue)**
Használd a `java.util.Random` osztályt a szimulátorhoz, amely a diákok kiölésére! A véletlensorozat magja `(seed)` legyen `12345678`.

A `Random` használatára egy példa:

```
import java.util.Random;
private static Random veletlengenerator = new Random(12345678);
veletlengenerator.nextBoolean();
veletlengenerator.nextInt(n)
```

Az osztálynak legyenek az alábbi műveletei:

- `Gimnazista()` létrehoz egy új gimnazistát és véletlenszerűen dönti el, hogy fiú-e vagy lány. (2 pont)
- `boolean fiu()` megmondja, hogy a diák fiú-e vagy lány. (1 pont)
- `foglalkoztat(int figyelmeztetesekSzama)` foglalkoztatja a gimnazistát. A foglalkozás során az osztályfőnök `figyelmeztetesekSzama`-szor figyelmezteti, amit a gimnazista megjegygez magának. (2 pont)
- `int osszesFigyelmeztetesekSzama()` megmondja, hogy összesen hányszor kapott figyelmeztetést. (1 pont)
- `boolean hazakuldhető()` megmondja, hogy elérte-e a figyelmeztetések száma az intő határát. Ha igen, akkor a diák hazaküldhető és felszabadul a helye a szobájában. (1 pont)

szallas.Szoba (11 pont)

Valósítsunk meg egy olyan `Szoba` osztályt, amelynek felhasználásával ábrázolhatjuk a diákszállót. Egy szoba tartalmazza a benne elszállásolt gimnazistákat. A szabályzat megkívánja, hogy legyenek külön lány, ill. fiúszobák. Továbbá ügyeljünk arra, hogy ne keverjük össze őket az elszállásolás során. Egy szobában a férőhelyek száma legyen 8 fő, mely 0-tól számozódik. Amennyiben egy szoba nem alkalmas vagy betelik, akkor a szimulátor körbejárja a szobákat amíg el nem tudja szállásolni a gimnazistát.

Az osztálynak legyenek az alábbi műveletei:

- `private Szoba(int, boolean)` konstruktor, amely létrehoz egy szobát a benne levő férőhelyekkel, valamint kijelöli, hogy fiúszoba-e vagy sem. (1 pont)
- `Szoba foglal(int, boolean)` ez az osztályszintű metódus az objektumgyár tervezési minta alapján ellenőrzi, hogy lehetséges-e a szobát létrehozni (azaz az első paraméterben megadott férőhelyszám nagyobb, mint 0; és a lehetséges maximális férőhelynél kisebb egyenlő). Amennyiben igen, akkor visszaadja az azonos paraméterekkel meghívott konstruktor segítségével létrehozott szobát. Ha a létrehozás feltétele nem teljesül, `null`-al térjen vissza a függvény. (2 pont)
- `boolean fiu()` megmondja, hogy a szoba fiúszoba-e vagy sem. Értéke igaz, ha csak fiúk laknak benne, hamis ha csak lányok. (1 pont)
- `int lakokSzama()` megmondja a szobában lakó gimnazisták számát. (1 pont)
- `int osszferohely()` megmondja a szoba teljes befogadóképességét. (1 pont)
- `int szabadferohely()` megmondja, hogy hány üres szabad férőhely található a szobában. (1 pont)
- `void elszallasol(Gimnazista)` megpróbál elszállásolni egy gimnazistát a szobában. (Emlékeztető: fiút csak fiúszobában, lányt csak lányszobában szabad elszállásolni. Amennyiben az elszállásolás sikertelen a gimnazista neve miatt, vagy amiatt, hogy már nincs több férőhely, a metódushívásnak nincsen semmilyen hatása a szoba állapotára)(2 pont)
- `Gimnazista lako(int)` megadja a gimnazistát, aki az adott ágyszámot kapta a szobában. (1 pont)

- `void felszabadit(int)` eltávolítja az adott ágyszámú gimnazistát a szobából és csökkenti a szobában lakók számát (a tömbelem törlésre kerül és a mérete csökken). (1 pont)

Projekt (javagyak/java-fiz-201672-bead1) · Activity (javagyak/java-fiz-201672-bead1/activity) F

Gimnazisták osztályfőnök (4 pont)

Valósítsunk meg egy olyan `Osztalyfonok` osztályt, amely a gimnazistákat és a szobákat kezeli. Egy osztályfőnök nyilvántartja magának, hogy összesen hány figyelmeztetést adott ki, valamint naponta ellenőrzi a szobákat és foglalkoztatja a gimnazistákat. A szimulátorban minden osztályfőnök minden szobát ellenőriz és minden diákot foglalkoztat, minden nap.

Az osztálynak legyenek az alábbi műveletei:

- `int osszesKiosztottFigyelmeztetesekSzama()` megmondja, hogy hány figyelmeztetést adott ki összesen az osztályfőnök. (1 pont)
- `void ellenoriz(Szoba)` ellenőrzi a megadott szobát és foglalkoztatja az ott lakó gimnazistákat. Egy gimnazistára legfeljebb négyszer szólhat rá egy nap egy osztályfőnök, és ha a gimnazista egy intőnyi figyelmeztetést gyűjtött össze, akkor hazaküldi, felszabadítva a helyét a szobában. A szimulátor kisorsolja, hogy a gimnazista mennyi figyelmeztetést szedett össze a metódus legutolsó hívása óta. Ezt úgy számoljuk, hogy minden hívásnál a figyelmeztetések számát véletlenszerűen egy 0 és 4 közötti értékkel növeljük meg. (3 pont)

szimulátorApp.Osztalykirandulas (7 pont)

Valósítsunk meg egy olyan `Osztalykirandulas` osztályt, amely a szimulációhoz megadott paraméterek alapján létrehozza a megfelelő számú osztályfőnököt, gimnazistát és szobát. Legyen minden páros szoba fiúszoba, minden páratlan szoba lányszoba.

Ezután meghívja a `kirandulas(int, int, int)` metódust, amely addig fut, amíg tart az osztálykirándulás. Egy szimulációs ciklus egy nap és addig tart a kirándulás, amíg az egyik osztályfőnök ki nem kényszerült osztani az ezredik figyelmeztetést vagy el nem fogynak a diákok. Minden osztályfőnök minden nap ellenőrzi az összes szobát.

A szimuláció végen az alkalmazás kiírja, hogy hány napig tartott az osztálykirándulás és megadja az osztályfőnökök statisztikáját indexük szerint, hogy ki hány figyelmeztetést volt kénytelen kiosztani.

Az osztálynak legyenek az alábbi műveletei:

- `void main(String[])` a szimulátor belépési pontja, induláskor három egész számot vár paraméterként. Az első az osztályfőnökök száma, a második a gimnazisták száma és a harmadik a szobák száma. Ezután meghívja a `kirandulas(int, int, int)` metódust. (1 pont)
- `private static void kirandulas(int osztalyfonokszam, int gimnazistaszam, int szobakszama)` maga a szimulátor, paraméterként vár három egész számot. Ellenőrzi, hogy legalább két szoba van, különben hibaüzenettel kilép. Ezután létrehozza a szobákat, valamint az osztályfőnököket. Létrehozza a gimnazistákat és egyből elszállásolja őket a szabályzatnak megfelelő szobákban az `int kovetkezoSzoba(int szobaszam, int szobakszama)` segédmetódus segítségével. (4 pont)
- `private static int kovetkezoSzoba(int szobaszam, int szobakszama)` segédmetódus, amely a szálláson található szobákon lép végig. Körbejár, azaz ha a végére ért, akkor előről kezdi. (2 pont)

Példák a szimuláció futtattatásra:

```
java szimulatorApp/Osztalykirandulas 3 44 12
Az osztálykirándulás 11 napig tartott.
Az osztályfőnökök ennyi figyelmeztetőt osztottak ki:
Project (/javagyak/java-fiz-201672-bead1) Activity (/javagyak/java-fiz-201672-bead1/activity) f
1.: 902
2.: 1004
```

Hibakezelés:

```
java szimulatorApp/Osztalykirandulas
Indítási hiba: három paraméter szükséges.
```

Bónusz feladat: többször futtatva a szimulációt, mi derül ki: átlag meddig tart egy kirándulás?

Pontozás

- 1: 0 — 6
- 2: 7 — 10
- 3: 11 — 14
- 4: 15 — 18
- 5: 19 — 29