

Konkurens programozás 2024.01.05. 18:00 gyakorlat

Határidő	Nincs megadva határidő	Pont	50	Kérdések	1
Elérhető	2024. jan 5, 18:30 - 2024. jan 5, 20:50 körülbelül 2 óra			Időkorlát	Nincs
Engedélyezett próbálkozások	Korlátlan				

Ez a kvíz már nem érhető el, mivel a kurzus befejeződött.

Próbálkozások naplója

	Próbálkozás	Idő	Eredmény
LEGUTOLSÓ	1. próbálkozás	134 perc	0 az összesen elérhető 50 pontból *

* Néhány kérdés még nem lett értékelve

Ezen próbálkozás eredménye: 0 az összesen elérhető 50 pontból *

Beadva ekkor: 2024. jan 5, 20:45

Ez a próbálkozás ennyi időt vett igénybe: 134 perc

1. kérdés

Még nincs értékelve / 50 pont

Konkurens programozás ZH, 2024.01.05. 18:00

Feltételek

- A feladat megoldását önállóan, más segítsége nélkül kell elkészíteni.
 - Kommunikáció csak az oktatókkal megengedett.
 - Az elkészített megoldást nemcsak a ZH végéig, hanem egészen a ZH napjának végéig nem szabad megosztani mással (pl. fórumba vagy publikus verziókezelő rendszerbe felöltés).
 - A megoldás elkészítéséhez használható a Java API dokumentációja. Ez a Canvasból letölthető, kicsomagolható.
 - Az elkészített megoldást **zip** formátumba csomagolva kell feltölteni a Canvasbe.
 - A **zip** tartalmazza a forrásfájl(oka)t, valamint a program legutolsó kimenetét egy **output.txt** fájlba beleírva.
 - A ZH végén kb. 10 percet érdemes fenntartani a kód tisztázására,

fordíthatóvá tételére, tömörítésére, beküldésére.

[A feladathoz tartozó fájlokat itt lehet letölteni.](#)

Feladatléírás

A feladatban egy olyan szimulációt kell megvalósítanunk, ahol a detektívek először kihallgatják az elkapott bűnözőket, remélve, hogy hasznos információt hallanak tőlük, majd a második részében a megkapott információt felhasználva elkapják a bűnözők feletteseit is.

A kód ad egy alapvető vázat a feladat megoldásához, minden résznél, ahol dolgozni kell rajta, `TODO` kommentekkel el van látva.

A program minden része úgy legyen megírva, hogy akkor is helyesen működjön, ha a lehető legrosszabb ütemezéssel hajtódnak végre az utasítások.

A programnak önmagától le kell állnia pár másodpercnyi futás után, és nem szabad "furcsa" jelenségeket produkálnia.

Első feladatrész (30p)

A feladat négy osztályból áll -

- `Simulation`, `Detective`, `Perpetrator` és `SharedInformation`. Ebben a részben a detektívek felváltva hallgatnak ki bűnözőket, vagy mennek kávézni, ha éppen nincs olyan bűnöző, akit ki tudnának hallgatni. A szimuláció `20` másodpercig fut, vagy addig, amíg a detektívek megszerzik a szükséges információt.

A `Simulation` osztály függvényeiben elvégzendő feladatok

Hozzuk létre

a `detectiveExecutor`, `executionExecutor` illetve `perpQueue` változókat a kommenteknek megfelelően.

`main`

- Indítsuk el a `producePerpetrator` műveletét egy külön szálon az `executionExecutor` használatával.
- Indítsuk el minden detektívnek az `interrogate` műveletét a `detectiveExecutor` használatával
- Várjunk `20` másodpercet (ezt kommenteljük ki a második részre)
- Győződjünk meg róla, hogy a programunk leáll

`producePerpetrator`

- Adjuk hozzá a `perp`-et a sorhoz, ha tudjuk, majd naplózzuk: `<perp> is ready to be interrogated`
 - Itt is és a továbbiakban is: a `<tartalom>` módon jelölt részek értelemszerűen lecserélendők
- Ha a sor teli van, naplózzuk: `<perp> will be interrogated another day`
- Várjunk `PERP_PRODUCER_WAIT_TIME_MSEC` ezredmásodpercet

A `SharedInformation` osztály függvényeiben elvégzendő feladatok

Készítsük el az `informationPercent` változót, hogy nyomon kövessük a megszerzett információ százalékát, kezdőértéke `0`. Készítsük el az `gatheringInformation` változót, hogy nyomon kövessük, hogy még információ gyűjtési fázis van-e, vagy nem - kezdőértéke `false`.

`addNewInformation`

- Biztosítsuk be, hogy az `informationPercent` nem mehet 100 fölé, adjuk hozzá a `percent` értéket
- Mikor először eléri a 100-at az `informationPercent`, írjuk ki: `Got all the information needed`, majd állítsuk át a `gatheringInformation` értéket `false`-ra
- Ha az `informationPercent` a `percent` hozzáadása után nem érte még el a `100`-at, írjuk ki az `informationPercent` új értékét: `Information gathering is at <informationValue új értéke>%`

`isGatheringInformation`

- Adjuk vissza a `gatheringInformation` értékét

A `Perpetrator` osztály függvényeiben elvégzendő feladatok

Konstruktor

- Legyen egyedi azonosítójuk (`id`), `0`-tól indulva, egyesével növekedve

`interrogationResult`

- Adjunk vissza `MINIMUM_INFORMATION_PERCENT` és `MAXIMUM_INFORMATION_PERCENT` között egy random értéket

A `Detective` osztály függvényeiben elvégzendő feladatok

`interrogate`

- Próbáljunk kiszedni a sorból egy bűnözőt, de adjuk fel `PERP_WAITING_TIME_MSEC` msec után
- Amennyiben nem sikerült találnunk bűnözőt, hívjuk meg a `drinkCoffee()` metódust, majd próbáljuk újra
- Amennyiben találtunk egy bűnözőt, hívjuk meg az `interrogatePerp(perp)` metódust
- A ciklusból kilépve írjuk ki `<detektív neve> finished interrogation`

`drinkCoffee`

- Várjunk `COFFEE_DRINKING_TIME_MSEC` ezredmásodpercet

`interrogatePerp`

- Várjunk `PERP_INTERROGATION_TIME_MSEC` ezredmásodpercet

Második feladatrész (20p)

Ebben a feladatrészben csak a `Simulation` és a `Detective` osztályokban kell dolgozni. Kiegészítve az első feladatrészt. Most amint megszerzik a detektívek a szükséges információt, el is fogják kapni a nagyobb és gonoszabb bűnözőket, mint akiket eddig kihallgattak. A 20 másodperces időlimitet itt ki is vesszük, a szimuláció akkor ér véget, ha a detektívek sikerrel jártak küldetésükben

A `Simulation` osztály függvényeiben elvégzendő feladatok

Csináljunk egy változót, amivel nyomon követjük, hogy vége van-e a szimulációnak, vagy nem (`simulationOver`), kezdőértéke legyen false

`main`

- Indítsuk el a `catchCrimeBosses` műveletét egy külön szálon az `executionExecutor` használatával
- Várjunk, amíg vége a szimulációnak (az 1. rész időzített várakozása helyett, azt kommenteljük ki)
 - Tipp: Erre célszerű a `simulationOver` változót, illetve wait-notify-t használni, bátran vegyünk fel új változókat

`catchCrimeBosses`

- Várjunk, amíg az összes információt megszerezték a detektívek, és mindegyik készen áll
 - Tipp: Bátran vegyünk fel új paramétert a `Detective` osztály `interrogate` metódusához.
 - Tartsuk figyelembe, hogy csak akkor lépünk túl ezen a ponton, ha az összes detektív jelzett már!

- Állítsuk `true`-ra a `simulationOver` értéket
- Jelezzük, hogy vége van a szimulációnak

A `Detective` osztály függvényeiben elvégzendő feladatok

`interrogate`

- Írjuk ki: `<detektív neve> gets ready to catch crime bosses`
- Várjunk `DETECTIVE_PREPARATION_TIME_MSEC` ezredmásodpercet
- Jelezzük, hogy a detektív kész arra, hogy elkapja a többi bűnözőt (erre a jelzésre várunk a `Simulation` osztály `catchCrimeBosses` metódus elején)

[↓ src.zip \(https://canvas.elte.hu/files/2660799/download\)](https://canvas.elte.hu/files/2660799/download)

Kvízeredmény: **0** az összesen elérhető 50 pontból