

1. Feladat

A feladat egy street food stand szimulációja - szakáccsal, asztalokkal és vendégekkel. Az érkező vendégek az első szabad asztalhoz leülnek, és leadják a rendelésüket. A standban egy szakács van, akinek egy ételt elkészíteni egy másodpercig tart. Az elkészült ételeket a vendég 2 másodpercig fogyasztja (+/- fél másodperc véletlen idő), majd utána rendel még egy ételt, aminek hasonló elfogyasztása után távozik. A standnál négy asztal van, az este során harminc vendég érkezik, átlag egy másodperces (+/- fél másodperc véletlen) időközökkel.

- A szakács, és a vendégek mind külön szálon fussanak
- A négy asztal mindegyikénél csak egy vendég ülhet
- A rendelések pillanatszerűek, akármennyi rendelés felkerülhet egy rendelés-listára
- A szakács egyszerre csak egy ételt tud főzni, ha elkészült a kész ételt egy elkészült listába teszi, ahonnan a vendég elveheti a sajátját (és csakis a sajátját!)
- Semelyik résztvevő ne foglaljon feleslegesen erőforrásokat.
- Ne használjon busy-wait vagy polling módszereket a várakozásra
- Logolja az eseményeket a konzolra (érkezés, asztal elfoglalása, rendelés leadás, főzés, étkezés, távozás)

2. Feladat

Egy számológép programot kell elkészíteni, alapvető funkcionalitással a legegyszerűbb módon. A számológép, az egyszerűség és időmegtakarítás céljából - stílusosan - egy bináris számológép legyen, ahol a számjegyek a 0 és az 1 legyenek.

A számológépnek legyen tehát:

- 0 és 1 gombja
- + és - gombja (a számítás idején ne lehessen inputot bevinni)
- = az eredmény számításához (a számítás idején ne lehessen inputot bevinni)
- <- egy számjegy törléséhez
- C a kijelző törléséhez
- Szöveges mezője, ahol a bevitt számok és az eredménye jelenik meg

A számológépet a JavaFX eszközeivel kell elkészíteni, az egyes nyomógombok egérekattintásra, valamint a billentyűzeten Enter nyomására is működjenek.

Mivel ezen matematikai műveletek azonnal végrehajtódnak, a számítások lépéséhez fél-fél másodperc lassítást kérünk betenni, és ennek megfelelően reszponzív GUI-t készíteni!

A számítás módja során érdemes az `Integer.parseInt(int, int)` és a `Integer.toBinaryString(int i)` függvényeket.