

Órai feladat – Eseménykezelés

Bevezetés

Készítsük el egy étterem konyhájának működését eseménykezelés segítségével. A séf fogja össze a teljes folyamatot, ő hozzá futnak be a rendelések, majd pedig eseményeken keresztül értesíti a szakácsokat a teendőkről. A szakácsok megadott mennyiségű alapanyaggal dolgoznak és szintén eseményekkel jelzik, ha ez már fogytán van.

Mivel a leadott feladatokat lehetséges, hogy automatizált módon fogjuk ellenőrizni, ezért kérünk mindenkit, hogy a lenti (ékezetek nélküli) elnevezéseket tartsa meg. Szükség esetén további mezőket fel lehet venni, bár ezekre általában nincs szükség.

Ugyanígy kérünk mindenkit, hogy próbálja meg önállóan megoldani a feladatot, mivel csak így fog bármit tanulni belőle. Szükség esetén persze a laborvezetőket nyugodtan meg lehet keresni, akik segíteni fognak.

Feladatok

Alapvető osztályok

Készíts egy **Etel** nevű osztályt az alábbiakkal:

- **megnevezes** – egyszerű szöveg, az étel megnevezése, legyen egy kiolvasható tulajdonsága is
- **hozzavalok** – szövegeket tartalmazó tömb, legyen olvasható tulajdonsága is
- **konstruktor** – mind a megnevezés, mind a hozzávalók tömb a konstruktoron keresztül kapjon értéket

Készítsd el az alábbi delegáltakat:

- **RendelesTeljesitesKezelo(etelNeve):** egy szöveges paramétere van és nincs visszatérési értéke
- **HozzavaloSzuksegesKezelo(hozzavalok):** egy szöveges paramétere van és nincs visszatérési értéke
- **HozzavaloElkeszultKezelo(hozzavalok):** egy szöveges paramétere van és nincs visszatérési értéke

Készíts egy **Sef** nevű osztályt az alábbiakkal:

- **receptek** – egy Etel objektumokat tartalmazó tömb, amit már fixen feltölthetünk elemekkel, pl.:

```
Etel[] receptek = new Etel[]
{
    new Etel("poharviz", new string[] { "viz" } ),
    new Etel("leves", new string[] { "repa", "hus", "krumpli", "viz" } ),
    new Etel("rantothus", new string[] { "hus", "krumpli" } ),
    new Etel("fozelek", new string[] { "viz", "repa" } )
};
```

- **RendelesTeljesitve** – egy *RendelesTeljesitesKezelo* típusú eseménykezelő, ezt fogjuk majd hívni, ha sikerült egy rendelés teljesítése
- **RendelesNemTeljesitheto** – egy *RendelesTeljesitesKezelo* típusú eseménykezelő, ezt fogjuk majd hívni, ha nem sikerült egy rendelés teljesítése
- **HozzavaloSzukseges** – egy *HozzavaloSzuksegesKezelo* típusú eseménykezelő, ezt hívja majd meg a séf, ha valamilyen hozzávalóra van szükség az étel elkészítéséhez

- **Megrendeles(etelNeve)** – ezen a metóduson keresztül lehet leadni egy rendelést. Nézzük meg, hogy van-e a *receptek* tömbben ilyen nevű étel. Ha nincs, akkor ezt jelezzük a *RendelesNemTeljesitheto* eseményen keresztül. Ha van, akkor a megtalált *Etel* objektumot adjuk át a következő pontban leírt *Elkeszites* metódusnak
- **Elkeszites(recept)** – tároljuk el egy privát *cel* nevű mezőben a paraméterként kapott elkészítendő *Etel* objektumot. Hozzunk létre egy *suksegesHozzavaloSzam* nevű privát mezőt is, ennek kezdőértéke legyen a receptben található hozzávalók száma. Ezt követően egyesével hívjuk meg a *HozzavaloSukseges* eseményt minden egyes hozzávalóra
- **SzakacsElkeszult(hozzavalo)** – ez a metódus fog majd meghívódni akkor, ha elkészült valamelyik szakács valamelyik hozzávalóval. Ilyenkor csökkentsük a *suksegesHozzavaloSzam* mező értékét eggyel. Ha a számláló elérte a 0-t, akkor elkészült az étel, ennek megfelelően hívjuk meg a *RendelesTeljesitve* eseménykezelőt.

Készíts egy **Szakacs** nevű osztályt az alábbiak szerint:

- **nev** – a szakács neve, tartozzon hozzá egy csak olvasható tulajdonság
- **specialitas** – annak a hozzávalónak a neve, amit a szakács el tud készíteni
- **konstruktor** – ezen keresztül kap értéket a fenti két mező
- **HozzavaloElkeszult** – egy *HozzavaloElkeszultKezelo* típusú esemény, amivel értesíti a szakács a séfet, ha elkészült
- **SefKerValamit(hozzavalo)** – ellenőrzi, hogy a paraméterként átadott *hozzavalo* megegyezik-e a szakács *specialitás*ával. Ha igen, akkor meghívja az alábbi *Foz()* metódust
- **Foz()** – egyelőre feltételezzük, hogy a szakács mindig tudja teljesíteni a feladatát, tehát csak annyit csinál, hogy meghívja a *HozzavaloElkeszult* eseménykezelőt

Egészítsük ki a *Sef* osztályt egy metódussal:

- *Felvesz(szakacs)* – itt tud jelentkezni egy szakács munkára. Ehhez a szakácsot nem is kell eltárolni, csak az eseménykezelőket kötjük össze. A séf *HozzavaloSukseges* eseménykezelőjére iratkozzon fel a jelentkező szakács *SefKerValamit* metódusa. A szakács *HozzavaloElkeszult* eseményére pedig iratkozzon fel a séf *SzakacsElkeszult* metódusa.

Teszt1: hozzunk létre egy *Sef* objektumot, aminek a *RendelesTeljesitve* és *RendelesNemTeljesitheto* eseményeire iratkozzon fel két metódus, amelyek kiírják a képernyőre, hogy melyik ételt sikerült vagy nem sikerült elkészíteni. Hozzunk létre egy szakácsot, akinek „víz” a specialitása és vegyük őt fel a séfhez. Próbáljunk meg rendelni „poharviz” és „kecskesajt” ételleket. Ha menet közben kiírjuk a képernyőre, hogy mikor melyik metódus hívódik meg, akkor valami hasonlóra számíthatunk:

```
Séf: Rendelés beérkezett 'poharviz'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
* Sikeres rendelés 'poharviz'
Séf: Rendelés beérkezett 'kecskesajt'
* Sikertelen rendelés 'kecskesajt'
```

Korlátok kezelése

Készíts egy **KorlatosSzakacs** osztályt, ami a *Szakacs* leszármazottja az alábbiak szerint:

- **mennyiseg** – egész szám mező, ami mutatja, hogy a saját specialitásából hányat tud még készíteni
- **konstruktor(nev, specialitas, mennyiseg)** – az első kettőt továbbítsa az ős konstruktornak, a mennyiséget adja értékül a fenti mezőnek

- **HozzavaloNemKeszithetoEl** – egy *HozzavaloElkeszultKezelo* típusú esemény, ami akkor fog tüzelni, ha a szakács nem tud elkészíteni egy hozzávalót
- **Foz()** – módosítsuk úgy az örökölt főz metódust, hogy minden hívásakor csökkenjen eggyel a *mennyiség* értéke. Ha még nagyobb mint 0, akkor hívja meg az *ős Foz* metódusát. Ha elfogyott a hozzávalo, azaz a szakács nem tudja elkészíteni a specialitását, akkor pedig hívja meg a *HozzavaloNemKeszithetoEl* eseményt

Egészítsük ki a **Sef** osztályt az alábbiakkal:

- **SzakacsHibatJelez(hozzavalo)** – ez a metódus hívódik majd meg, ha egy szakácsnál elfogyott a hozzávaló. Maga a metódus csak annyit csinál, hogy meghívja a *RendelesNemTeljesitheto* eseményt
- Egészítsük ki úgy a *Felvesz* metódust, hogy ha a paraméterként átadott szakács *KorlatosSzakacs*, akkor annak *HozzavaloNemKeszithetoEl* eseményére iratkozzon fel a *SzakacsHibatJelez* metódus

Teszt: az előző példát egészítsük ki azzal, hogy hozzunk létre és vegyünk fel a séfhez egy *KorlatosSzakacs* objektumot, akinek „repa” a specialitása, és 2 a kezdő mennyisége. Ezt követően rendeljünk meg 5 darab „fozelek”-et.

```
Séf: Rendelés beérkezett 'poharviz'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
* Sikeres rendelés 'poharviz'
Séf: Rendelés beérkezett 'kecskesajt'
* Sikertelen rendelés 'kecskesajt'
Séf: Rendelés beérkezett 'fozelek'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
Robert főz 'repa'-et
Séf: Elkészül a 'repa'
* Sikeres rendelés 'fozelek'
Séf: Rendelés beérkezett 'fozelek'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
Robert főz 'repa'-et
Séf: Elkészül a 'repa'
* Sikeres rendelés 'fozelek'
Séf: Rendelés beérkezett 'fozelek'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
Séf: Szakács hibát jelzett 'repa'
* Sikertelen rendelés 'fozelek'
Séf: Rendelés beérkezett 'fozelek'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
Séf: Szakács hibát jelzett 'repa'
* Sikertelen rendelés 'fozelek'
Séf: Rendelés beérkezett 'fozelek'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
Séf: Szakács hibát jelzett 'repa'
* Sikertelen rendelés 'fozelek'
```

Opcionális feladatok

- Készítsünk még egy delegáltat tetszőleges elnevezéssel, amin keresztül egy *KorlatosSzakacs* jelezni tudja majd azt, ha a hozzávalója fogytán van már (pl. 2-nél kevesebb van).
- Módosítsuk úgy a *KorlatosSzakacs*-ot, hogy ha fennáll a fenti állapot, akkor ezen az eseménykezelőn keresztül jelezni tudja ezt.
- Készítsünk egy *Raktar* osztályt, ami egy megadott hozzávalót tartalmaz valamilyen mennyiségben (konstruktorban megadott értékek).

- A raktár iratkozzon fel a *KorlatosSzakacs* fenti eseményére egy metódusával, és ha egy szakács azt jelzi, hogy már fogytán van valamilyen hozzávalója, akkor a Raktár ezt automatikusan pótolja 1-gyel.

Teszt: például a fenti 5 főzelék elkészítése úgy, hogy a szakácsnak is 2 répája van, és hozzákapcsolunk egy raktárat további 2 répával:

```
Séf: Rendelés beérkezett 'fozelek'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
Robert főz 'repa'-et
Séf: Elkészül a 'repa'
* Sikeres rendelés 'fozelek'
Séf: Rendelés beérkezett 'fozelek'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
Raktár 'repa': 'Robert segítséget kér - adtunk 1-et
- de nincs már
Robert főz 'repa'-et
Séf: Elkészül a 'repa'
* Sikeres rendelés 'fozelek'
Séf: Rendelés beérkezett 'fozelek'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
Raktár 'repa': 'Robert segítséget kér - adtunk 1-et
Robert főz 'repa'-et
Séf: Elkészül a 'repa'
* Sikeres rendelés 'fozelek'
Séf: Rendelés beérkezett 'fozelek'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
Raktár 'repa': 'Robert segítséget kér - de nincs már
Robert főz 'repa'-et
Séf: Elkészül a 'repa'
* Sikeres rendelés 'fozelek'
Séf: Rendelés beérkezett 'fozelek'
Bela főz 'viz'-et
Séf: Elkészül a 'viz'
Raktár 'repa': 'Robert segítséget kér - de nincs már
Séf: Szakács hibát jelzett 'repa'
* Sikertelen rendelés 'fozelek'
```