

Gyakorló feladat 1

Az időjósítás igazán nehéz feladat, melyben detemenisztikus modelleket nem lehet elérni. Hogy minél pontosabbak legyenek az előrejelzések, érdemes minél több helyen megmérni egyes tényezőket.

- Hozzunk létre egy **meroberendezes** nevű class-t az alábbi adatagokkal: *id*, *homerseklet* (1 tizedes pontos lebegőpontos szám), *paratartalom* (egesz), *szam* (statikus adattag)
- Az osztálynak a default konstruktora az *id*-t kiszámítja, a *homerseklet* és *paratartalom* értékeket 10-re állítja.
- Az *id* számításához egy privát tagfüggvényt hívjon meg a program, az *id* 5 karakterből álló szám, az egyediséget nem kell ellenőrizni.
- Készítsük el a három adattag getter és setter függvényeit! A statikus adattagnak statikus getter és setter tagfüggvényt írjunk meg.
- Hozzunk létre egy **meres** nevű tagfüggvényt, melyet random számgenerálással szimuláljunk. A hőmérséklet 20-25, a páratartalom 30-50 között legyen mérhető.
- Készítsünk egy **kozpont** nevű osztályt, mely rendelkezzen egy *kapacitas* adattaggal és egy *meroberendezes* típusú tömbbel.
- A konstruktor kérje be a kapacitást és ennek megfelelően hozza létre a tömböt, a destruktork pedig törölje azt.
- Egy hozzáad tagfüggvény segítségével lehessen hozzáadni mérőberendezéseket a központhoz, amíg be nem telik a kozpont. Ha betelt, akkor erről adjon tájékoztatást a felhasználónak a konzolra. A jelenlegi számot mindig a statikus getter függvényből kérje le.
- Rendelkezzen a kozpont egy listázás tagfüggvénnyel, mely elvégzi a méréseket és azok eredményét kiírja egy fájlba (az eszköz *id*-jével együtt).
- A main függvényben egy állapotgépet hozzon létre, ami a konzolon keresztül lép be az egyes állapotokba.
- Az **'h'** gomb lenyomására adjon hozzá egy új eszközt.
- Az **'m'** gomb lenyomására mér és kilistázza az eredményeket.
- A **'q'** lenyomására leáll a program.

Gyakorló feladat 2

A feladat során egy gyorsétteremlánc adminisztrációs rendszerét kell elkészíteni.

- Ehhez elsőként készítsünk egy **etel** osztályt. Az osztály tartalmazza az alábbi adattagokat: *nev, ar, szam (statikus)*
- Készítse el az adattagok getter és setter függvényeit, a statikus taghoz statikus függvényekként.
- Készítsen egy default konstruktort, mely a névnek „hamburger”, árnak „590” értéket adjon meg.
- Terhelje túl a konstruktort, mely argumentumként bekéri a két szükséges adattagot és ezek alapján definiálja azok értékét.
- Definiáljon egy **etterem** osztályt. Adattagjai az alábbiak legyenek:
- `etel_max` és ennek nagyságának megfelelően egy `etel` típusú tömb, egyenleg legyen 1000 kezdetben
- Az egyenlegre készítsünk egy getter függvényt.
- Készítsen egy **menu_hozzaad** nevű függvényt, melynek megadhatjuk a rendelt étel nevét és árát. Amennyiben a tömb betelt, ne vegyük fel a menüre, hiszen nem tudunk több félélt egyszerre készíteni.
- Készítsünk egy **rendel** függvényt, amelynek argumentuma egy string. Amennyiben van az ételek között, az étterem elkészíti és kiírja, az egyenleghez ennyit hozzáad.
- Készítsünk egy **reklamas** függvényt, mely esetén az elégedetlen vevők vissza akarják vinni a terméket. Amennyiben az étel szerepel az étlapon, elnézést kérése mellett szolgáltatassuk vissza a termék árát, ezt vonjuk le az egyenlegből.

Gyakorló feladat 3

Készítsünk egy osztályt, amely string műveleteket tud végrehajtani.

- Készítsünk két string privát adattagot.
- A konstruktor kérje be a két tárolandó adattagot.
- Határozzuk meg egy tagfüggvény segítségével a leghosszabb azonos szövegrészt a két adattagban. (alma és almafa esetén az „alma”, malom és halom esetén az „alom”)
- Készítsünk egy **tomorites** függvényt, mely a stringeket tömöríti az alábbi metódus szerint: Amennyiben van egymás után több azonos betű, vonja őket össze és írja ki, hogy hány darab betűt lehetett összevonni. pl. aaab = a3b.