5. heti feladatok

1 Készítsen egy olyan alkalmazást, amivel szimulálni tudják élelmiszerek ásott veremben történő tárolását.

- Hozzon létre egy **FoodIngredient** osztályt.
  - Az osztályban tárolja el egy élelmiszer fajta nevét, tárolt mennyiségét és a mennyiség egységét.
  - Mennyiségi egységként liter, kilogramm, darab és csomag legyen megadható.
  - Legyen az osztálynak egy felülírt `string ToString()` metódusa.
  - Tesztelje az osztály `string ToString()` metódusát.
- Kivételek kezelése érdekében hozzon létre három kivétel osztályt.
  - A **StackException** osztályban tárolja el, hogy melyik **IngredientStack** objektum dobta el a kivételt.
  - A **StackEmptyException** osztály legyen a **StackException** osztály leszármazottja.
  - A **StackFullException** osztály legyen a **StackException** osztály leszármazottja. Tárolja el, hogy melyik az az élelmiszer, amit nem sikerült betenni a verembe.
- Készítsen egy **IngredientStack** osztályt a verem reprezentálására.
  - A veremben lévő élelmiszereket tárolja el egy tömbben, aminek a méretét a konstruktoban állítsa be. Új elemeket mindig a tömb elejétől nézve első üres helyre szúrjon be, amikor pedig kivesz valamit a veremből, a kivett elem a tömbben eltárolt utolsó elem legyen.
  - Legyen az osztálynak egy publikus `bool Empty()` metódusa, ami információt szolgáltat arról, hogy a verem üres-e.
  - Legyen az osztálynak egy publikus `void Push(FoodIngredient newItem)` metódusa, ami betesz egy új elemet a verem tetejére. Ha a verem már tele volt, akkor sikertelen a verembe helyezés, ezért dobjon el egy **StackFullException** kivételt.
  - Legyen az osztálynak egy publikus `FoodIngredient Pop()` metódusa, ami kiveszi a legfelül lévő elemet a veremből. Ha a verem üres volt, akkor dobjon el egy **StackEmptyException** kivételt.
  - Legyen az osztálynak egy publikus `FoodIngredient Top()` metódusa, ami megadja, hogy mi van a verem tetején, de azt nem veszi ki onnan. Ha üres a verem, akkor `null` értékkel térjen vissza.
  - Tesztelje teljes körűen a publikus metódusokat.
- Készítsen egy **IngredientStackHandler** osztályt, ami egy példányosításkor megadott verem kezelését valósítja meg.
  - Legyen az osztálynak egy publikus `FoodIngredient[] AddItems(FoodIngredient[] foodIngredients)` metódusa. A metódus beteszi a verembe a paraméterként megkapott élelmiszerek közül azokat, amik befférnek a verembe. Visszatérési értéke tartalmazza azokat az élelmiszereket, amik már nem fértek be a verembe.
  - Tesztelje a metódus működését.
  - Próbálja meg úgy is tesztelni a metódus működését, hogy közben kimockolja a háttérben használt **IngredientStack** osztályt. Ennek érdekében, ha szükséges, hozzon létre megfelelő interfészt is.

2 Sportóra által rögzített edzések adatait csv formátumú fájlban menti el a sportórához adott alkalmazás. Egy csv fájl tartalma, például az alábbi.

```
type,distance,time,elevation,heart_rate
Cycling,"35,8","1:28:03","314","142"
Running,"11,95","1:12:04","15","138"
Hiking,"24,3","4:47:02","687","116"
Swimming,"2,4","0:58:35","0","143"
```

Ahogy a példából is látszik négy féle sportág edzéseinek rögzítésére képes a sportóra. A számértékkel rendelkező adatok a csv állományban mindig idézőjelek között vannak. A távolság lebegőpontos, az emelkedés és a pulzus pedig egész számként van eltárolva. Az idő minden esetben `h:mm:ss` formátumú, ahol az óra akár egynél több számjegyet is tartalmazhat.

Írjon csv fájl feldolgozó alkalmazást, ami be tudja olvasni az egyes sorokban tárolt adatokat. Az esetleges hibás formátumú sorok esetén az alkalmazás írja ki a konzolra, hogy hiba történt, majd az adott sor figyelmen kívül hagyása mellett folytatódjon a fájl feldolgozása.

Egy lehetséges megvalósítási módhoz ötletek:

- Hozzon létre egy **Time** osztályt, amelyben óra, perc és másodperc adatok legyenek eltárolva. Írjon ehhez az osztályhoz egy statikus `Parse` metódust. Ha a parszolás közben azt tapasztalja, hogy nem megfelelő a bemenet formátuma, akkor dobjon el egy **TimeException** kivételt. Ha az óra érték negatív, akkor dobjon el egy **HourException** kivételt, ami a **TimeException** leszármazottja. Ha a perc vagy másodperc érték negatív vagy 59-nél, akkor is dobjon el megfelelő kivételeket.
- Hozzon létre egy **Workout** osztályt, amelyben el tudja tárolni egy edzés adatait. Legyen ennek az osztálynak is egy saját parszere, ami egy sorban lévő sztringet fel tud dolgozni. Ha közben valami problémát tapasztal dobjon el **WorkoutException** kivételt.
- Legyen egy **CSVProcessor** osztálya, aminek példányosításkor megadja a csv fájl elérési útját. Az osztály `Workout[] AllItems()` metódusa végzi a csv fájl feldolgozását, lekezeli az esetleg előálló kivételeket és visszaadja az értelmezhető sorokban lévő edzések tömbjét.
- Tesztelje az egyes publikus metódusokat.