Név: Papp Márk

Neptun: DCQEB4

Feladat leírás: DCQEB4.pdf

Védésnek a videó linkje: <https://youtu.be/eWQdPOdoyAs>

**Teszt fájlok és felépítésük:**

2 fájl található (felevesfeladat\_DCQEB4\felevesfeladat\_DCQEB4\bin\Debug) *feladat.csv* és *dolgozok.csv* .

Azért használok .csv fájl formátumot és nem .txt-t mert azt könnyen be lehet olvastatni az excel-el, ha az elválasztás a „ ; ”-el történik.

A*feladat.csv* fájlban csak maguk a feladatok neve szerepel és ez alapján képezi le a program a különböző feladatokat. Fontos tudni, hogy pontosan ugyan azok a nevek kellenek, amik a feladatok neve, mert különben hibára fut a program. A feladatok szintjei automatikusan képződnek, nem kell és nem is lehet külön megadni őket. A feladatok szintjei 1-4 között vannak (orvosi fel.=4, etetés=3, beszerzés=2, jegy eladás=2, ketrec tisztítás=1).

A *dolgzok.csv* fájlban találhatjuk az első oszlopban a nevüket és a másodikban a képzettségüket pontos vesszővel (;) elválasztva. Fontos, hogy a program automatikusan adja meg kinek milyen pozíció jár a képzettségük alapján. A pozíciók a képzettségek alapján a következőek: 4 vagy annál több=orvos, 3=állat gondozó, 2=pénztáros, 1=gondnok. Program kezeli, ha valakinek a képzettsége több, mint 4 ő orvos lesz és olyan feladatokat is kap. Program kezeli, ha valakinek nem éri el a képzettsége az 1-et, ilyenkor *Kepzettsegiszint* exception-t dob a program, melyet a gondnok konstruktor le is kezeli és elküldi továbbképzésre, ahol a képzettsége 1 lesz.

**Class-ok és tulajdonságaik:**

**IAllatkertiDolgozo**.cs feladatban leírt tulajdonságok. Plusz hozzáraktam az IComparable-t.

**IFeladat**.cs feladatban leírt tulajdonságok. Plusz hozzáraktam az IComparable-t.

**Dolgozók**(Gondnok.cs,Orvos.cs,stb…) megvalósítják a IAllatkertiDolgozo interfészt és felülírják a ToString metódust. Konstruktoruk Kepzettsegiszint kivételt dob ha a képzettség 1-nél kissebb.

**Feladatok**(Jegyeladas.cs,Etetes.cs,stb…) megvalósítják a IFeladat interfészt és felülírják a ToString metódust. Konstruktoruk a szintet magától osztja.

**Kepzetsegiszint**.cs kivétel mely tartalmaz egy üzenetet és egy dolgozót. Fentebb írt esetben dobódik.

**IO**.cs statikus class itt történik a file beolvasás külön-külön metódusban létrehozódik kettő lista egy LancoltListaDolgozo a dolgozókkal és egy LancoltListaFeladat a feladatokkal. Itt kapódik el a Kepzettsegiszint kivétel és itt lesz a dolgozó elküldve továbbképzésre.

**Allatkert**.cs tartalmaz egy feladat listát és egy dolgozó listát illetve lehet a konstruktorából meghívni a két filet vagy külön-külön metódusokban beolvasni őket. Tartalmaz még egy Tovabbkepzes metódust ahol egy dolgozónak a képzettsége 1-re lesz állítva.

**LancoltListaFeladat**.cs egy lista melynek az alábbi tulajdonságai vannak és csak Ifeladat-ot képes felvenni magába.

* Beszúrás: szint szerinti csökkenő sorrendbe rendezve szúr be.
* Törlés: Törli az azonos Ifeladat-ot.
* Bejárás: bejárja a listát és event-ként hivatkozva visszaadja őket sorrendben.
* Tömbbe: Ifeladat tömbbe rendezi a listát. Ezt majd a LancoltListaDolgozo listában használom.

**LancoltListaDolgozo**.cs egy lista melynek az alábbi tulajdonságai vannak és csak IAllatkertiDolgozo-t képes felvenni magába.

* Beszúrás: képzettség szerinti csökkenő sorrendbe rendezve szúr be.
* Törlés: Törli az azonos IAllatkertiDolgozo-t.
* Bejárás: bejárja a listát és event-ként hivatkozva visszaadja őket sorrendben.
* Feladatkiosztás: Itt történik a feladatkiosztás a dolgozóknak és event-ként hivatkozva adja vissza, hogy kinek melyik feladat lett kiosztva, illetve ha valakinek nincs munkája vagy valamelyik feladat nem lett kiosztva és annak az indokját. A program.cs-ben van az event handling-je.

**Program**.cs IAllatkertiDolgozo event handling-je illetve a Tovabbkepzes metódus event handling-je itt történik. Illetve itt van tesztelve maga a program működése az Allatker class példányosításával és IAllatkertiDolgozo metódusának meghívásával.