

## 1 SZOFTVERTTERVEZÉS ÉS -FEJLESZTÉS II – FÉLÉVES FELADAT

---

### 1.1 ÉTELEK

Az IEheto interfész a feladatban megadottakon túl rendelkezik egy ElerhetoMennyiség tulajdonsággal, egy ElerhetoMennyiségCsokkent(szám) metódussal, ElerhetoMennyiségCsokkent() metódussal és még egy TulNagyRendelesKezelo típusú eventtel, amihez előtte létrehoztam egy delegáltat, ami egy kulcsot vár. Ezután kiépítettem egy osztályhierarchiát, 4 abstract ősosztály van(Édességek, Koleszterinben gazdag ételek, Pizzák, Semleges ételek, Tejes ételek), amelyek megvalósítják az IEheto interfészt és felülírják a ToString() metódust, úgy, hogy az étel nevét adja vissza.

Az étteremben minden ételből 100 darab van és összesen 17 féle étel van:

1. **Édességek:** abstarct osztály, cukorbetegnek nem fogyaszthatják
  - a. Dobostorta:
    - i. ár: 1000 Ft
    - ii. kedvezmény: 10 és fölötti rendelés esetén 10% kedvezmény van rá
  - b. Kókuszgolyó:
    - i. ár: 40 Ft
    - ii. kedvezmény: ha 80 fölött rendelünk belőle, akkor csak 80 darabért kell fizetnünk
  - c. Piskótatekerecs:
    - i. ár: 300 Ft
    - ii. kedvezmény: ha 40 fölött rendelünk belőle, akkor csak 40 darabért kell fizetnünk
2. **Koleszterinben gazdag ételek:** abstarct osztály, akiknek magas a koleszterinszintje ők nem fogyaszthatják
  - a. Cigánypecsenye:
    - i. ár: 1900 Ft
    - ii. Kedvezmény: --
  - b. Rántott sajt:
    - i. ár: 600 Ft
    - ii. Kedvezmény: 4 fölötti rendelés esetén csak 500 Ft
  - c. Tojásrántotta 4 tojásból:
    - i. ár: 350 Ft
    - ii. Kedvezmény: ha 50 fölött rendelünk belőle, akkor csak 50 darabért kell fizetnünk
3. **Pizzák:** abstarct osztály, gluténérzékenyek nem fogyaszthatják
  - a. Hawaii pizza:
    - i. ár: 1500 Ft
    - ii. kedvezmény: 5 fölötti rendelés esetén csak 1300 Ft
  - b. Magyaros pizza:
    - i. ár: 1600 Ft
    - ii. kedvezmény: 5 fölötti rendelés esetén csak 1400 Ft
  - c. Margaréta pizza:
    - i. ár: 1150 Ft
    - ii. kedvezmény: 5 fölötti rendelés esetén csak 1150 Ft

- d. Mexikói pizza:
  - i. ár: 1750 Ft
  - ii. kedvezmény: 5 fölötti rendelés esetén csak 1550 Ft
- e. Songoku pizza:
  - i. ár: 1450 Ft
  - ii. kedvezmény: 5 fölötti rendelés esetén csak 1250 Ft
- 4. **Semleges ételek:** abstarct osztály, mindenki fogyaszthatja
  - a. Alaszkai tőkehalfilé rostos:
    - i. ár: 1740 Ft
    - ii. kedvezmény: --
  - b. Pohár víz:
    - i. ár: 50 Ft
    - ii. kedvezmény: --
  - c. Bolognai spagetti:
    - i. ár: 1980 Ft
    - ii. kedvezmény: pároknak ingyenes, azaz, ha 2 rendelnek belőle akkor 0 Ft
- 5. **Tejes ételek:** abstarct osztály, laktózérzékenyek nem fogyaszthatják
  - a. Brokkolikrémleves:
    - i. ár: 840 Ft
    - ii. kedvezmény: --
  - b. Málnakrémleves:
    - i. ár: 720 Ft
    - ii. kedvezmény: --
  - c. Tejberizs:
    - i. ár: 500 Ft
    - ii. kedvezmény: --

## 1.2 AZNAPI CSOPORT

Rendelkezik egy név és egy létszám adattaggal, valamint egy a betegségeik tárolására szolgáló listával. A konstruktor alapból beállítja a létszámot 1 főre, valamint a csoport nevét is „Csoportra”, ha netán elfelejtenénk ezeket megadni.

## 1.3 BINÁRIS KERESŐFA

Egy a laboron megismert generikus Bináris keresőfa.

## 1.4 MENÜK

Egy étterem számára egy konzolos alkalmazást, menürendszert készítettem. Létrehoztam egy generikus abstract **Menu** őssztályt:

- Menüpont: a választható menüpontokat tartalmazó generikus tömb
- KivalasztottIndex: a Menüpont melyik elemén vagyunk „rajta”
- MenuCim: a menü címe
- konstruktor, ami beállítja a fenti adattagokat
- egy abstract Megjelenítés() metódus
- egy abstract Indítás() publikus metódus
- egy delegált

- és egy a delegált típusú esemény

Majd létrehoztam a SimaMenu és a KivalasztasosMenu osztályokat és ezeket leszármaztattam a Menu ősszostályból.

#### **SimaMenu:**

- Indítás(): egy do-while ciklus fut, amiből akkor lépünk ki ha entert nyomunk és ekkor vissza is térünk a KivalasztottIndex-el. A fel és le nyilakkal lehet lépkedni az egyes menüpontok között. A KivalasztottIndex nő, ha fel, és csökken, ha le nyilat nyomunk. Minden egyes fel és le lépés után lefut a Megjelnítés() metódus.
- Megjelenítés(): a Menupont elemeinek a ToString() metódusát írja ki a konzolra, a KivalasztottIndex indexű elemnél megcseréli a háttér és a szöveg színét, a többinél alap módon írja ki.

#### **KivalasztasosMenu:**

Az ősszostály adattagjain kívül rendelkezik még egy KivalasztottElemek listával is.

- Indítás(): Annyiban tér el a SimaMenu-s megvalósítástól, hogy ha itt entert nyomunk akkor:
  - ha a KivalasztottIndex az a Menupont tömb hossza vagy a Menupont tömb hosszánál eggyel rövidebb, akkor kilép és visszatér a KivalasztottIndexxel.
  - ha nem az előbbi esetek egyike, akkor először megnézi, hogy a KivalasztottElemek lista létre van e már hozva, ha nincs akkor létrehozza. Ezután megnézi, hogy a KivalasztottIndex benne van e már a KivalasztottElemek listában, és ha nincs benne akkor hozzáadja és Invoke-olja az IndexHozzáadás eseményt.  
Meghívja a Megjelenítés() metódust.
- Megjelenítés(): Ha a KivalasztottElemek lista nincs létrehozva akkor ugyan úgy működik mint a SimaMenu-s megvalósítás. Azonban, ha létre van hozva, akkor az azt jelenti, hogy elem is van benne és akkor úgy működik, hogy a Menupont tömb KivalasztottIndexedik elemének a ToString() metódusát fehér háttérrel és fekete betűszínnel írja ki, a KivalasztottElemek lista elemeit piros háttérrel és fehér betűszínnel jeleníti meg, a többit alap módon írja ki.

## **1.5 A MENÜRENDSZER**

### **Tartalma:**

- etelekfaja: egy bináris keresőfa, amiben az étterem összes étele van név szerint rendezve
- kiválasztotelekfaja: egy bináris keresőfa, amiben a kiválasztott ételeket fogjuk tárolni
- hányadik csoport: azt számoljuk, hogy hányadik csoportot olvastuk be fájlból
- egy Aznapi csoport, amiben az épp aktuálisan kiszolgálendő csoport adatai vannak
- egy paramétermentes konstruktor
- metódusok az a menükhöz

### **Menük:**

A legtöbb menü ugyan azon a szerkezet alapján épül fel. Van egy Menüpont tömbjük, amely tartalmazza a választható menüpontokat, valamint van egy címük. Ezután attól függően, hogy milyen

menüt akarunk létrehozni, létrehozzuk, majd futtatjuk a Menu indítás metódusát, amit kimentünk egy változóba. Ezt egy switch-case szerkezet kezeli le, és a megfelelő metódust futtatja le.

#### 1.5.1 Start()

Feltölti a fát az ételekkel és lefuttatja a FoMenu() metódust.

#### 1.5.2 FoMenu(): SimaMenu

3 menüponttal rendelkezik: csoport adatok megadása, csoport adatok beolvasása fileból, kilépés

Csoport adatok megadása: lefut a CsoportMenu() metódus. Ha ezt választjuk akkor manuálisan, kézzel tudjuk megadni az épp aktuális csoport adatait.

Csoport adatok beolvasása fileból: ez a tesztelés megkönnyítésére szolgál. Lefuttatja a Filebololvasas() metódust, ami beolvassa a „csoportAdatok.txt” file Hanyadikcsoport-odik sorát, és beállítja a csoportnak a megfelelő értékeket(név, létszám, betegségek). Majd ezután lefut a MenuEsEtelek() metódus.

Kilépés: leáll a program.

csoportAdatok.txt formai követelmény: név, létszám, felmerülő betegségek  
Pl.: Kalocsai nyugdíjasok, 7, Koleszterin, Laktózérzékenység, Gluténérzékenység  
megadható betegségek: (nagybetű-kisbetű érzékeny)

- Koleszterin
- Laktózérzékenység
- Gluténérzékenység
- Cukorbetegség

Üres sor nem lehet a végén.

#### 1.5.3 CsoportMenu(): SimaMenu

4 menüponttal rendelkezik: név, létszám, betegségek, vissza

név: megadhatjuk a csoport nevét, ha nem adjuk meg akkor automatikusan „csoport” lesz.

létszám: megadhatjuk a csoport létszámát, ha nem adjuk meg akkor automatikusan 1 lesz.

betegségek: lefut a BetegsegekMegadasa() metódus, ahol manuálisan, kézzel kitudjuk választani, hogy milyen betegségekkel rendelkezik a csoport.

vissza: lefut a FoMenu() metódus.

#### 1.5.4 BetegsegekMegadasa(): KivalasztasosMenu

6 menüponttal rendelkezik: koleszterin, cukorbetegség, laktózérzékenység, gluténérzékenység, tovább, vissza

Mivel kiválasztásos menü, ezért van egy IndexHozzaadas esemény amelyre feliratkoztattam a BetegsegBeszur() metódust, és ez a kiválasztott betegséget, hozzáadja a csoport betegségeinek listájához.

koleszterin, cukorbetegség, laktózérzékenység, gluténérzékenység: ha ezeket választjuk, hozzáadja őket a csoport betegségeinek a listájához, a fentebb leírtak alapján.

tovább: lefut a MenuEsEtelek() metódus.

vissza: lefut a CsoportMenuMetodus().

### 1.5.5 MenuEsEtelek(): SimaMenu

Itt kötünk ki, akkor is, hogy ha FoMenu-ben a filebeolvasást választjuk, és itt kötünk ki akkor is, ha kézzel adjuk meg a csoport adatait.

3 menüponttal rendelkezik: optimális menü, ételek rendelése, vissza

optimális menü: lefut az OptimalisBTS() metódus, majd a VegOsszegKiszamitas()metódus és a VegOsszeg() metódus.

ételek rendelése: lefut az EtelekKivalasztasa() metódus, ahol kézzel, kitudjuk válogatni, hogy a betegségek alapján leszűrt ételekből, melyikeket akarjuk megrendelni.

vissza: lefut a CsoportMenu() metódus.

#### 1.5.5.1 OptimalisBTS()

Feliratkoztatja az összes étel, a NemElerhető eventre, ha ez aktiválódik, akkor kitörli az elemet az etelekfajaja-ból, majd elindítjuk a VisszalepesesKereses() metódust. Kapunk egy optimális megoldást. Ezután leellenőrizzük, hogy amit kaptunk optimális megoldás, az jó-e. Végig megyünk az optimális megoldáson, és megnézzük, hogy az aktuálisan vizsgált elemünk indexe nagyobb vagy egyenlő, mint a szint. Ha igen akkor megnézzük, hogy mennyi van még belőle. Ha 0 van belőle, akkor az azt jelenti, hogy nem lehet benne az optimális megoldásban, tehát ki kell cserélnünk, de előtte el kell döntenünk, hogy utána csak ilyen ételek állnak az optimális megoldásban, vagy másfajta ételek is vannak. Miután ezt eldöntöttük, kimentjük az első indexet, ahova új elemet kell keresnünk.

Ha nincs másféle elem az optimális megoldásban az éppen elfogyott elem indexe után, akkor újra le kell futtatni az OptimalisBTS()-t, de csak attól a szinttől, ahol először fogyott el az étel.

Ha másféle elemek is vannak az optimális megoldásban az éppen elfogyott elem indexe után, akkor ezt azt jelenti, hogy az OptimalisBTS()-t meg kell ismételnünk a 0-ás szinttől, ezért vissza kell állítani az ételek eredeti darabszámát, mert lehet benne se lesznek a következő optimális megoldásban.

mivel

Pl.: ha az optimális megoldásunk a következő lenne:

bolognai spagetti, bolognai spagetti, pohár víz, pohár víz, pohár víz, pohár víz, pohár víz

pohár víz: 100 db

ár: 50 Ft, kedvezmény: --

bolognai spagetti: 1 db

ár: 1980 Ft, kedvezmény: 2 esetén ingyenes

Ugye csak 1 spagettink maradt, tehát 2 nem lehet az optimális megoldásban, és ha ugyan úgy járnánk el, mintha csak ugyan olyan elem következne utána, akkora visszalépéses keresésnek átadnánk azt, hogy 1. Lefutna a keresés, és megkapnánk, hogy: bolognai spagetti, pohár víz, pohár víz, pohár víz, pohár víz, pohár víz, pohár víz.

És ez nem lenne optimális, hiszen a Bolognai spagetti, csak úgy a nagyon kedvező, hogyha 2 rendelünk belőle, itt viszont 1 van belőle, tisztán látszik, hogy jobban járnánk, ha csak pohár vizet rendelnénk. Ezért kell máshogy eljárni, mint akkor, ha csak ugyan olyan elemek következnek.

#### 1.5.5.2 VisszalepesesKereses()

Az előadáson megismert "optimális megoldás keresése"-t valósítottam meg.

Az Fk függvény abban az esetben ad vissza hamis értéket, hogyha az elérhető mennyiség az nulla.

A Josag() metódusom pedig visszaadja egy lehetséges variáció összértékét. Ezzel a metódussal hasonlítom össze a most talált megoldást és az eddigi optimális megoldást. Ha „nagyobb jószágút”

találok, azaz a kisebb árat, mint az eddigi optimális akkor, akkor kicserélem a most talált megoldásra az optimális megoldást.

Úgy hozom olyan formára, hogy az megfeleljen a visszalépéses keresésnek, hogy a szinteken a szűrés alapján megmaradt ételek vannak, és annyi szintünk van ahány fős a csoport, azaz hány darab ételt keresünk.

#### 1.5.6 EtelekKivalasztasa() : KivalasztasosMenu

Változó számú menüponttal rendelkezik: ezt a Szures() metódus határozza meg, de biztosan van egy megrendel és egy vissza menüpontja

Mivel kiválasztásos menü, ezért van egy IndexHozzaadas esemény amelyre feliratkoztattam a KivalasztotEtelekHozzaadasa() metódust, ami a kiválasztott ételt hozzáadja a kiválasztottetelekfajához.

Miután kiválasztottuk az ételleket, lefut az MennyiségMegadása() metódus, itt meg tudjuk adni, hogy miből mennyit szeretnénk rendelni, ha annyit rendelünk hogy elfogyna az étel akkor töröljük az etelekfaja-ból, majd miután megadtuk minden ételnél, kiszámolja az összértéket.

megrendel: lefut a VegOsszeg() metódus.

vissza: lefut a CsoportMenu() metódus.

#### 1.5.7 VegOsszeg() : SimaMenu

Kiírja a csoport nevét, és a végösszeget.

3 menüponttal rendelkezik: ételek megnézése, menü elejére, kilépés

ételek megnézése: lefut az EtelekMegnezese() metódus.

menü elejére: lefut a FoMenu() metódus.

kilépés: leáll a program.

#### 1.5.8 EtelekMegnezese() : SimaMenu

Kiírja a megrendelt ételleket.

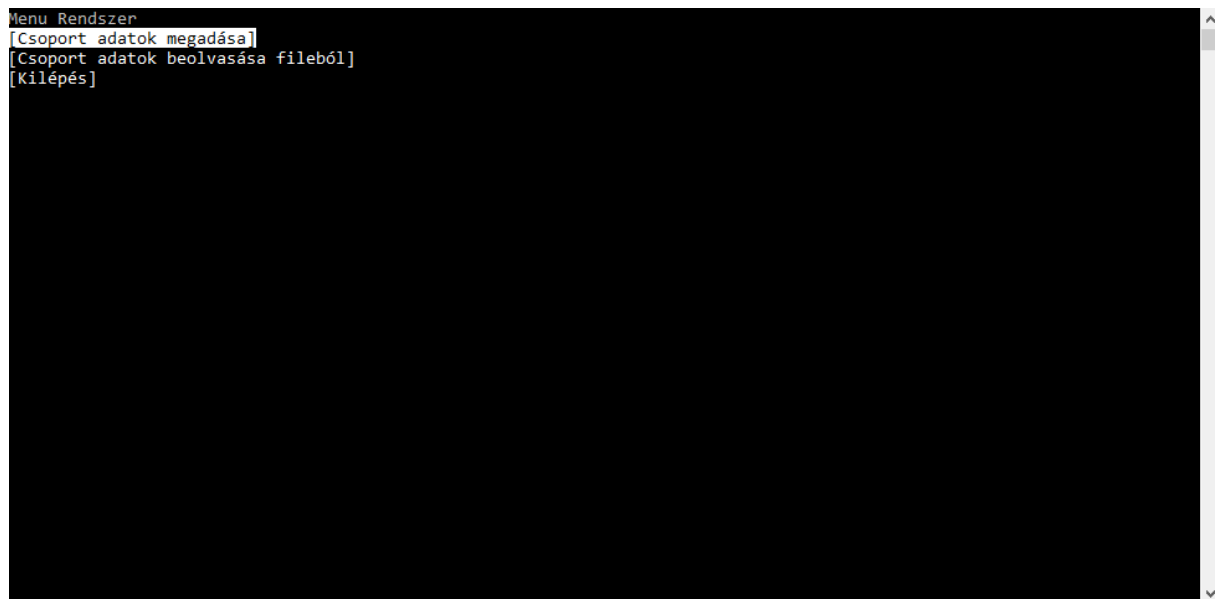
2 menüponttal rendelkezik: menü elejére, kilépés

menü elejére: lefut a FoMenu() metódus.

kilépés: leáll a program.

### 1.6 A PROGRAM MŰKÖDÉSE

Az indítás után, a következő kép fogad:



A fel és le nyilakkal lehet a megfelelő menüpontra lépni és az enterrel ki lehet választani, ugyan így a többinél.

## 2 TARTALOM

1	Szoftverttervezés és -fejlesztés II – Féléves feladat .....	1
1.1	Ételek .....	1
1.2	Aznapi csoport .....	2
1.3	Bináris keresőfa .....	2
1.4	Menük.....	2
1.5	A Menürendszer .....	3
1.5.1	Start().....	4
1.5.2	FoMenu(): SimaMenu.....	4
1.5.3	CsoportMenu(): SimaMenu.....	4
1.5.4	BetegsegekMegadasa(): KivalasztasosMenu.....	4
1.5.5	MenuEsEtelek(): SimaMenu .....	5
1.5.6	EtelekKivalasztasa() : KivalasztasosMenu.....	6
1.5.7	VegOsszeg() : SimaMenu .....	6
1.5.8	EtelekMegnezese() : SimaMenu .....	6
1.6	A program működése .....	6