

# Szoftvertervezés és -fejlesztés I.

Rendezések  
Feladatok

# Hallgatói Tájékoztató

**A jelen bemutatóban található adatok, tudnivalók és információk a számonkérendő anyag vázlatát képezik. Ismeretük szükséges, de nem elégséges feltétele a sikeres zárthelyinek, illetve vizsgának.**

**Sikeres zárthelyihez, illetve vizsgához a jelen bemutató tartalmán felül a kötelező irodalomként megjelölt anyag, a gyakorlatokon szóban, illetve a táblán átadott tudnivalók ismerete, valamint a gyakorlatokon megoldott példák és az otthoni feldolgozás céljából kiadott feladatok önálló megoldásának képessége is szükséges.**

# Szoftvertervezés és -fejlesztés I.

Rendezések  
Feladatok

# Minimumkiválasztásos rendezés

## 3.3. Algoritmus Minimumkiválasztásos rendezés

**Bemenet:**  $x$  –  $T$  tömb,  $n$  – egész (tömb mérete); ahol  $T$  összehasonlítható

**Kimenet:**  $x$  –  $T$  rendezett tömb

```
1: eljárás MINIMUMKIVÁLASZTÁSOSRENDEZÉS(címszerint  $x$  :  $T$  tömb,  $n$  : egész)
2:   ciklus  $i \leftarrow 1$ -től  $(n - 1)$ -ig
3:      $min \leftarrow i$ 
4:     ciklus  $j \leftarrow (i + 1)$ -től  $n$ -ig
5:       ha  $x[min] > x[j]$  akkor
6:          $min \leftarrow j$ 
7:       elágazás vége
8:     ciklus vége
9:      $x[i] \leftrightarrow x[min]$ 
10:  ciklus vége
11: eljárás vége
```

### Felhasznált változók és függvények

- $x$ : A rendezni kívánt tömb. Az  $x$  tömb elemeinek összehasonlíthatónak kell lennie. Az eljárás a tömböt helyben rendez.
- $n$ : A paraméterként átadott tömb mérete.
- $min$ : A tömb egy bejárásán belül a megtalált minimális értékű elem indexe.

# Javított beillesztéses rendezés

## 3.7. Algoritmus Javított beillesztéses rendezés

**Bemenet:**  $x$  –  $T$  tömb,  $n$  – egész (tömb mérete); ahol  $T$  összehasonlítható

**Kimenet:**  $x$  –  $T$  rendezett tömb

```
1: eljárás JAVÍTOTTBEILLESZTÉSESRRENDEZÉS(címszerint  $x$  :  $T$  tömb,  $n$  : egész)
2:   ciklus  $i \leftarrow 2$ -től  $n$ -ig
3:      $j \leftarrow i - 1$ 
4:      $segéd \leftarrow x[i]$ 
5:     ciklus amíg  $(j > 0) \wedge (x[j] > segéd)$ 
6:        $x[j + 1] \leftarrow x[j]$ 
7:        $j \leftarrow j - 1$ 
8:     ciklus vége
9:      $x[j + 1] \leftarrow segéd$ 
10:  ciklus vége
11: eljárás vége
```

### Felhasznált változók és függvények

- $x$ : A rendezni kívánt tömb. Az  $x$  tömb elemeinek összehasonlíthatónak kell lennie. Az eljárás a tömböt helyben rendez.
- $n$ : A paraméterként átadott tömb mérete.
- $segéd$ : Segédváltozó, mely mindig az aktuális beillesztendő elemet tartalmazza.

# Szoftvertervezés és -fejlesztés I.

Karakter sorozatok

Feladatok

# Feladatok

## 1. Írjon programot, amely az alábbiakat valósítja meg!

1. Előállít egy tíz elemű tömböt, melynek minden eleme 1 és 20 közötti véletlen egész szám.
2. Megjeleníti a tömb elemeit.
3. Minimumkiválasztásos rendezéssel rendezi a tömb elemeit.
  - A rendezést megvalósító metódusnak paraméterként a tömböt adja át. Gondolja végig, hogy a metóduson belüli változtatások a paraméterként átadott tömbben is érvényre jutnak-e!
  - Szükséges egy Csere metódust is implementálnia, amely képes két egész számot megcserélni.  
Ügyeljen a paraméterátadás megfelelő módjára!
4. A rendezés eredményét is megjeleníti.

# Feladatok

## 2. Írjon programot, amely az alábbiakat valósítja meg!

1. A konzolról beolvas néhány szöveget.
2. A javított beillesztéses rendezés segítségével növekvő sorrendbe rendezi a bekért szövegeket.
  - Két string típusú változóról el kell tudni dönteni, hogy melyik „kisebb”.
  - Ehhez kell írnia egy összehasonlítást megvalósító metódust.
3. Megjeleníti rendezett sorrendben a szövegeket.



# Feladatok

## 3. Írjon programot egy kávézó napi forgalmi adatainak kezelésére!

1. A programban megadható, hogy hány nap adatait akarjuk eltárolni.
2. Minden napnál megadható legfeljebb 10 ital, amit aznap felszolgáltak.
3. A program megadja, hogy a vizsgált napokon milyen italokat szolgáltak fel.
4. Minden ital esetén meg lehet határozni, hogy hány esetben szolgálták fel.

# Feladatok

## 3. Írjon programot egy kávézó napi forgalmi adatainak kezelésére!

```
Hány nap eladásait akarod rögzíteni? 3
1. napi italok (max 10 darab, ha kevesebb üss üres entert)
kávé
kávé
tea
2. napi italok (max 10 darab, ha kevesebb üss üres entert)
tea
capuccino
kávé
3. napi italok (max 10 darab, ha kevesebb üss üres entert)
tea
narancslé
kávé
kávé

capuccino: 1 darab
kávé: 5 darab
narancslé: 1 darab
tea: 3 darab
```

# Feladatok

## 3. Írjon programot egy kávézó napi forgalmi adatainak kezelésére!

- `int BekerNapokSzama ()`  
Bekéri a vizsgált napok számát.
- `string[,] EladottItalokBekerese ()`  
Bekéri az egyes napokon eladott italok listáját.
- `int AdottNapiItalokSzama (string[,] italokNaponta, int index)`  
Megadja, hogy adott napon hány darab fogyasztás volt.
- `string[] AdottNapotKivalogat (string[,] italokNaponta, int index)`  
Az adott nap italait kiválogatja. A kimenet hossza megegyezik a tényleges fogyasztási mennyiséggel.

# Feladatok

## 3. Írjon programot egy kávézó napi forgalmi adatainak kezelésére!

- `bool StringKisebbE (string első, string második)`  
Összehasonlítja a két bemeneti stringet.
- `void NapiItalListatRendez (string[] italok)`  
A bemeneti tömböt növekvő módon rendezi.
- `string[] IsmetlodoItalokKiszurese (string[] italok)`  
A rendezett bemeneti tömbből kiszűri az ismétlődő elemeket.  
A kimeneti tömb mérete megegyezik a benne ténylegesen eltárolt italok számával.

# Feladatok

## 3. Írjon programot egy kávézó napi forgalmi adatainak kezelésére!

- `string[] KetListatOsszefuttat(string[] elsoLista, string[] masodikLista)`  
Az összefuttatás tételt alkalmazva összefuttatja a két rendezett bemeneti tömböt.  
A kimeneti tömb mérete megegyezik a benne eltárolt italok számával.
- `string[] Italok(string[,] italokNaponta)`  
Visszaadja a bemeneti tömbben található italok rendezett és ismétlődésmentes listáját.
- `int[] EladottMennyisegek(string[,] italokNaponta, string[] italok)`  
Megadja, hogy az egyes italokból ténylegesen hány eladás történt.

# Irodalom, feladatok

- **Kotsis-Légrádi-Nagy-Szénási: Többnyelvű programozástechnika, PANEM, Budapest, 2007**
- **Faraz Rasheed: C# School, Synchron Data, 2006**  
**<http://www.programmersheaven.com/2/CSharpBook>**
- **Reiter István: C# jegyzet, DevPortal, 2010,**  
**<http://devportal.hu/content/CSharpjegyzet.aspx>**



