Nézők

Valaha egy kisebb településen társadalmi esemény volt, ha „a csapat” játszott. Ha nem is annyian, mint akkor, de ma is állandó látogatói vannak a futballmérkőzéseknek. A kor követelményeinek megfelelően minden bérlettulajdonost és jegyvásárlót azonosítanak, rögzítik a nézőtéren való megjelenésüket is. Az adatbázis egy idény bajnoki és kupamérkőzéseinek nézőkre vonatkozó adatait tartalmazza.

`meccs.txt` – fájlban található adatok részletezése — ***meccs (id, datum, kezdes, belepo, tipus, nezoszam)***

id	A meccs azonosítója (szám), ez a kulcs.
datum	A meccs megrendezésének dátuma (dátum). Egy nap csak egy mérkőzést játszanak.
kezdes	A meccs kezdési ideje (idő).
belepo	A belépő ára az adott meccsre (szám).
tipus	A meccs típusa (szöveg). Értéke lehet bajnoki vagy kupa.

`belepes.txt` – fájlban található adatok részletezése — ***belepes (nezoid, meccsid, idopont)***

nezoid	A néző azonosítója (szám), ez a kulcs.
meccsid	A meccs azonosítója (szám), ez a kulcs.
idopont	Az időpont, amikor a néző az adott meccsen belépett a nézőtérre (idő).

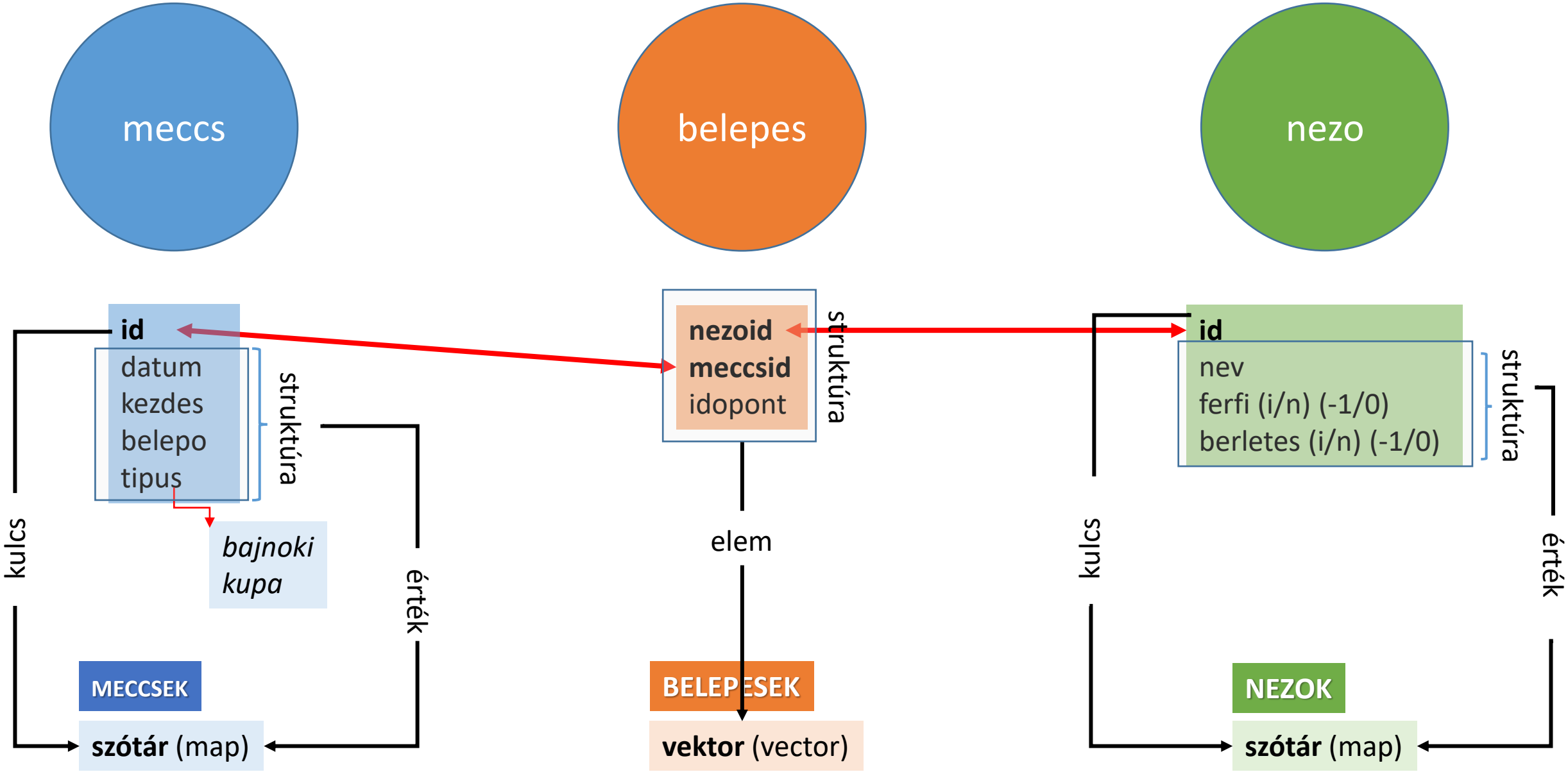
`nezo.txt` – fájlban található adatok részletezése — ***nezo (id, nev, ferfi, berletes)***

id	A néző azonosítója (szám), ez a kulcs.
nev	A néző neve (szöveg). A megoldás során kihasználhatja, hogy nincsenek azonos nevű nézők.
ferfi	A néző nemét adja meg (logikai), férfi esetén igaz, nő esetén hamis.
berletes	Megadja, hogy a nézőnek van-e bérlete (logikai). Ők nem vesznek jegyet.

1. Olvasd be a fájlok tartalmát és tárold el!
2. Írd ki időrendben a bajnoki mérkőzések dátumát a bajnoki.txt fájlba!
3. Mennyibe került a jegy az idény legutolsó mérkőzésére!
4. Az egyes mérkőzéseken hány néző jelent meg! A mérkőzések dátumát és a nézőszámot jelenítse meg!
5. A jegyvásárlásokból mennyi bevétel származott 2018 szeptemberében!
6. Kiss Dorina és Varga Zsolt egy párt alkotnak. Add meg, hogy mely napokon voltak együtt meccsen!
7. Mikor volt a legtöbb bevételt hozó meccs?
8. Átlagosan hány bérletes ment el a meccsekre?

FELADAT

ADATSZERKEZETEK

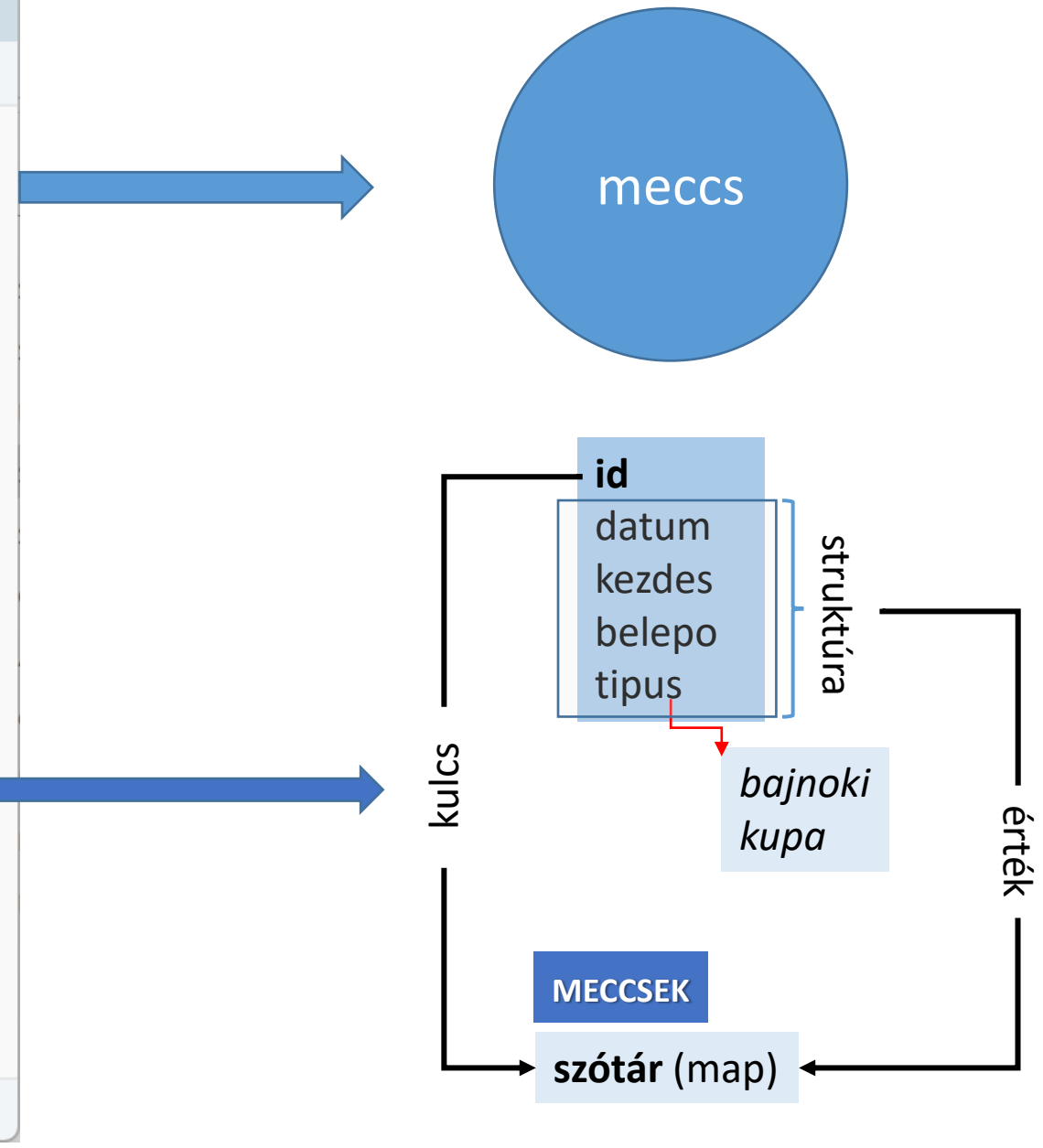


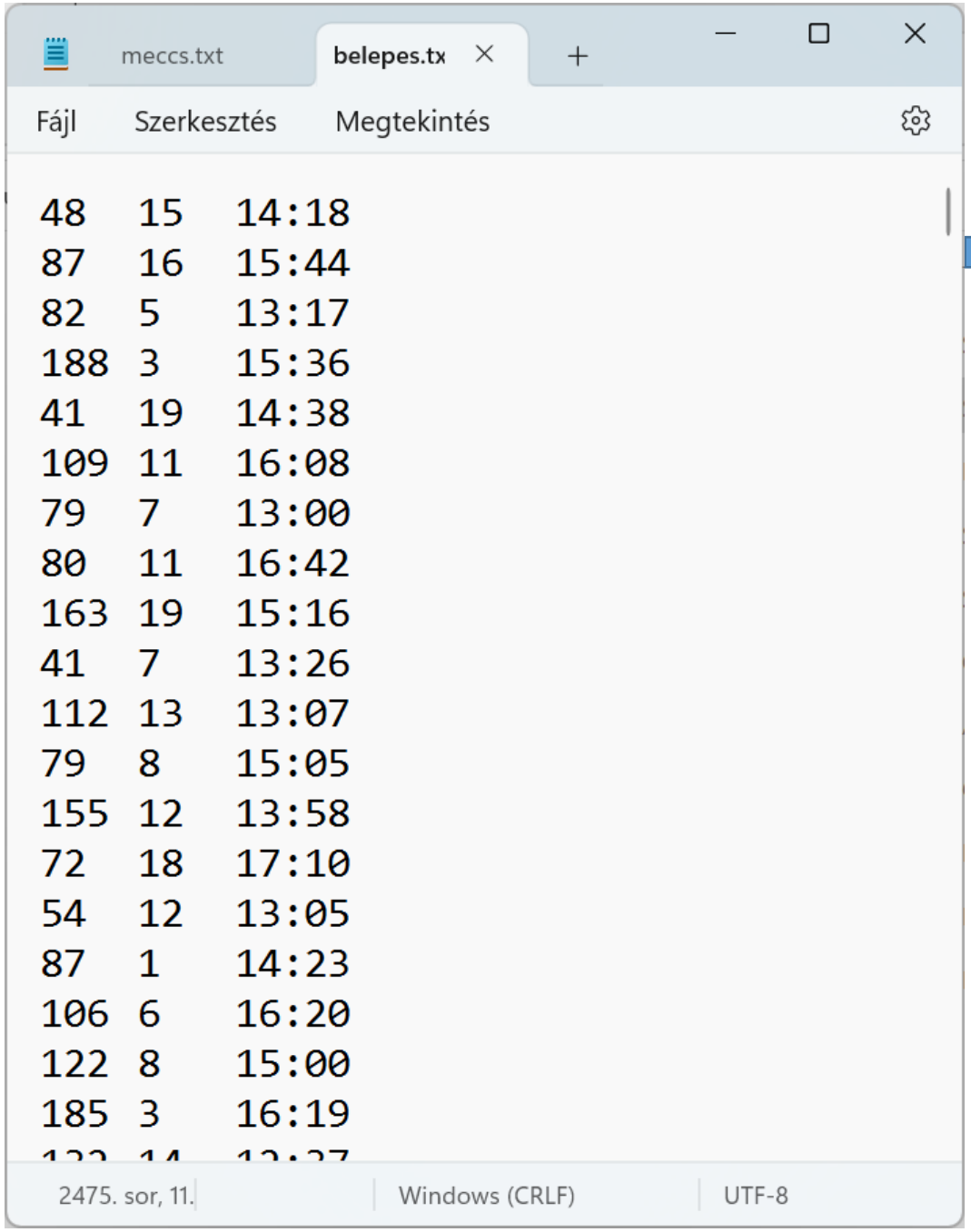
meccs.txt

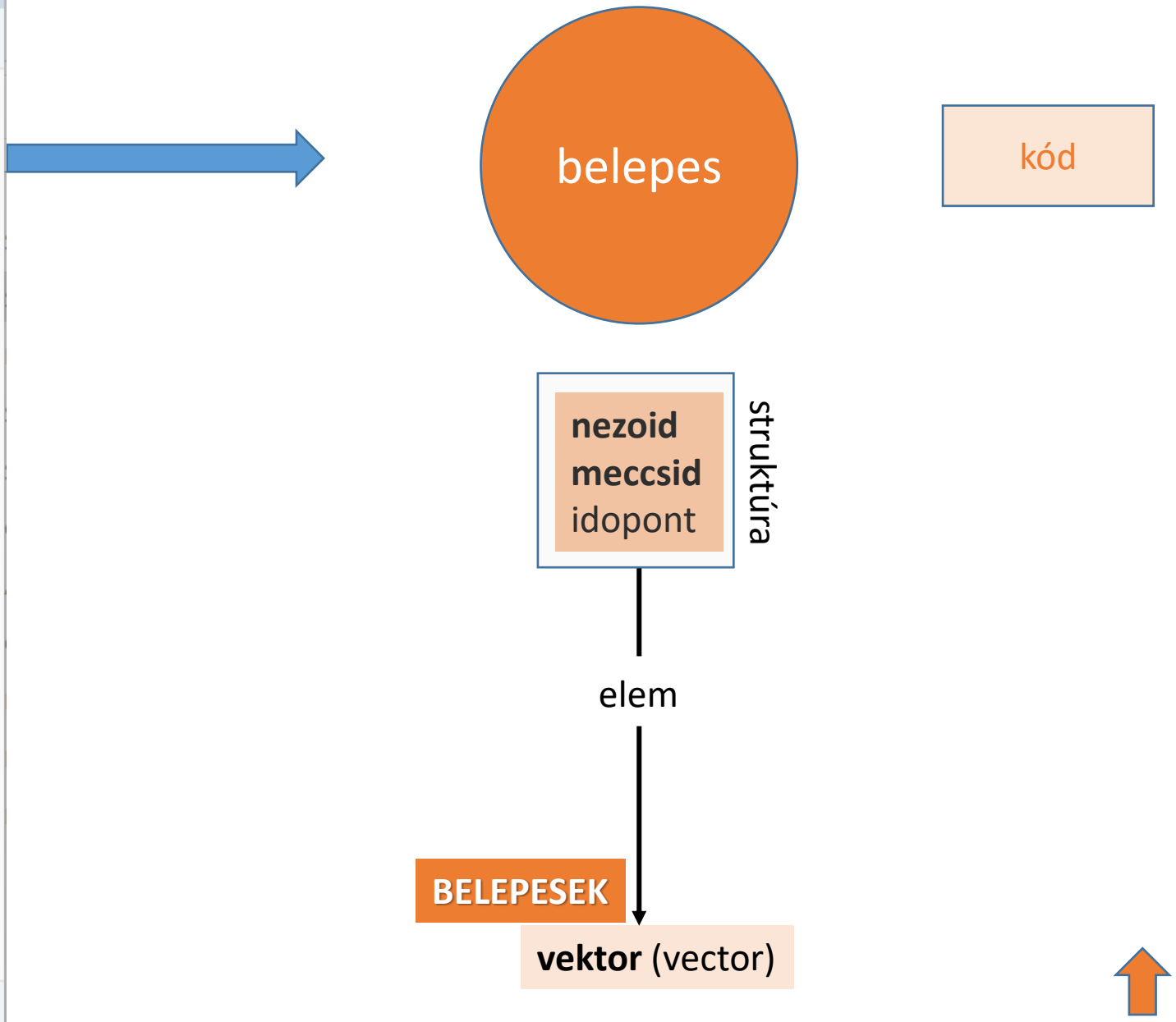
Fájl Szerkesztés Megtekintés

```
1   2019.03.23   15:00   500 bajnoki
2   2019.04.20   15:30   1000bajnoki
3   2018.09.08   16:00   500 bajnoki
4   2018.11.03   13:30   500 bajnoki
5   2018.12.01   13:30   500 bajnoki
6   2018.08.15   17:00   200 kupa
7   2018.11.17   13:30   1000bajnoki
8   2019.05.18   16:00   500 bajnoki
9   2018.08.22   17:00   200 kupa
10  2018.09.22   15:30   500 bajnoki
11  2018.08.25   17:00   500 bajnoki
12  2018.10.06   14:00   1000bajnoki
13  2018.10.20   13:30   500 bajnoki
14
15  2018.09.26   15:00   500 kupa
16  2019.05.04   16:00   500 bajnoki
17  2018.10.24   14:00   1000kupa
18  2018.08.11   17:00   500 bajnoki
19  2019.04.06   15:30   500 bajnoki
```

1. sor, 1. oszl Windows (CRLF) UTF-8




meccs.txt
belepes.tx
Fájl
Szerkesztés
Megtekintés
48 15 14:18
87 16 15:44
82 5 13:17
188 3 15:36
41 19 14:38
109 11 16:08
79 7 13:00
80 11 16:42
163 19 15:16
41 7 13:26
112 13 13:07
79 8 15:05
155 12 13:58
72 18 17:10
54 12 13:05
87 1 14:23
106 6 16:20
122 8 15:00
185 3 16:19
2475. sor, 11.
Windows (CRLF)
UTF-8


belepes
kód
nezoid
meccsid
idopont
struktúra
elem
BELEPESEK
vektor (vector)

nezo.txt

Fájl Szerkesztés Megtekintés

```
1   Halk János   -1   -1
2   Kertész Bence    -1   0
3   Molnár Alex  -1   -1
4   Bóka Nátán   -1   -1
5   Réti Dániel  -1   -1
6   Varga Zsolt  -1   0
7   Elek Miklós  -1   0
8   Balázs Domonkos  -1   0
9   Tóth Ákos    -1   -1
10  Sári Péter   -1   0
11  Kiss Dorina  0    -1
12  Csuka Beáta  0    0
13  Kun Tamás    -1   -1
14  Zsolczai Balázs  -1   -1
15  Bertók Dániel    -1   -1
16  Fehér Orsolya    0    -1
17  Márton Hunor -1   0
18  Vass András  -1   0
19  Szepes Edit  0    0
20  Kónya Barbara    0    0
```

190. sor, 21. Windows (CRLF) ANSI



kód

id
nev
ferfi (i/n) (-1/0)
berletes (i/n) (-1/0)

struktúra

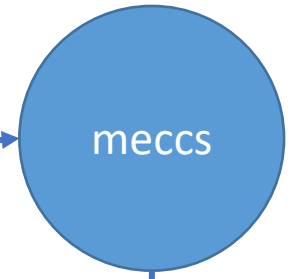
kulcs

érték

NEZOK

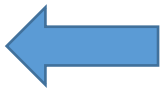
szótár (map)

```
void F1meccs(map<int,meccs> &m)
{
   ifstream fbe;
   fbe.open("meccs.txt");
   meccs t;
   t.nezoszam=0;
   int id;
   while(fbe>>id&&fbe>>t.datum&&fbe>>t.kezdes&&fbe>>t.belepo&&fbe>>t.tipus)
   {
      m[id]=t;
   }
   /*
   for(auto ms:m)
   {
      t=ms.second;
      cout << ms.first << " " << t.datum << " " << t.kezdes << " " << t.belepo << " " << t.tipus << endl;
   }
   */
   fbe.close();
}
```

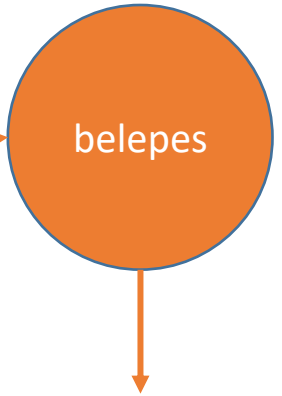



```
struct meccs
{
   string datum;
   string kezdes;
   int belepo;
   string tipus;
   int nezoszam;
};
```

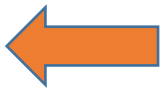
számított adat



```
void F1belepes(vector<belepes> &b)
{
   ifstream fbe;
   fbe.open("belepes.txt");
   belepes t;
   while(fbe>>t.nezoid&&fbe>>t.meccsid&&fbe>>t.idopont)
   {
      b.push_back(t);
   }
   /*
   for(auto bel:b)
   {
      cout << bel.nezoid << " " << bel.meccsid << " " << bel.idopont << endl;
   }
   fbe.close();
   */
}
```

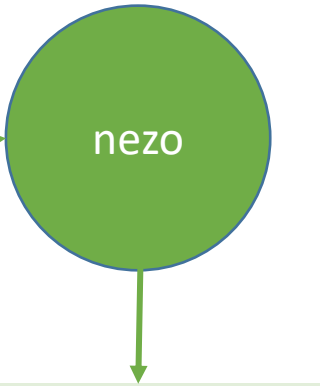



```
struct belepes
{
   int nezoid;
   int meccsid;
   string idopont;
};
```

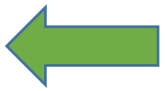


```
void F1nezo(map<int,nezo> &n)
{

   ifstream fbe;
   fbe.open("nezo.txt");
   nezo t;
   int id;
   while(fbe>>id&&fbe>>t.vnev&&fbe>>t.knev&&fbe>>t.ferfi&&fbe>>t.berletes)
   {
      n[id]=t;
   }
   /*
   for(auto ms:n)
   {
      t=ms.second;
      cout << ms.first << " " << t.vnev << " " << t.knev << " " << t.ferfi << " " << t.berletes << endl;
   }
   */
   fbe.close();
}
```




```
struct nezo
{
   string vnev;
   string knev;
   int ferfi;
   int berletes;
};
```



```cpp
// Érettségi Kézikönyv 2.0 - (C) Ódor Péter 2023

#include <iostream>
#include <fstream>
#include <vector>
#include <map>
#include <list>
#include <algorithm>

using namespace std;

struct nezo
{
    string vnev;
    string knev;
    int ferfi;
    int berletes;
};

struct meccs
{
    string datum;
    string kezdes;
    int belepo;
    string tipus;
    int nezoszam;
};

struct belepes
{
    int nezoid;
    int meccsid;
    string idopont;
};

void F1nezo(map<int,nezo> &n)
{

    ifstream fbe;
    fbe.open("nezo.txt");
    nezo t;
    int id;
    while(fbe>>id&&fbe>>t.vnev&&fbe>>t.knev&&fbe>>t.ferfi&&fbe>>t.berletes)
    {
        n[id]=t;
    }
    /*
    for(auto ms:n)
    {
        t=ms.second;
        cout << ms.first << " " << t.vnev << " " << t.knev << " " << t.ferfi << " "
<< t.berletes << endl;
    }
    */
    fbe.close();
}

void F1meccs(map<int,meccs> &m)
{
    ifstream fbe;
    fbe.open("meccs.txt");
    meccs t;
    t.nezoszam=0;
    int id;
    while(fbe>>id&&fbe>>t.datum&&fbe>>t.kezdes&&fbe>>t.belepo&&fbe>>t.tipus)
    {
```

```
 66          m[id]=t;
 67      }
 68      /*
 69      for(auto ms:m)
 70      {
 71          t=ms.second;
 72          cout << ms.first << " " << t.datum << " " << t.kezdes << " " << t.belepo <<
" " << t.tipus << endl;
 73      }
 74      */
 75      fbe.close();
 76  }
 77
 78  void F1belepes(vector<belepes> &b)
 79  {
 80      ifstream fbe;
 81      fbe.open("belepes.txt");
 82      belepes t;
 83      while(fbe>>t.nezoid&&fbe>>t.meccsid&&fbe>>t.idopont)
 84      {
 85          b.push_back(t);
 86      }
 87      /*
 88      for(auto bel:b)
 89      {
 90          cout << bel.nezoid << " " << bel.meccsid << " " << bel.idopont << endl;
 91      }
 92      fbe.close();
 93      */
 94  }
 95
 96  void sorting_a_vector(vector<string> &r,bool novekvo)
 97  {
 98      if(novekvo) sort(r.begin(),r.end());
 99      else sort(r.begin(),r.end(),greater<>());
100  }
101
102  void F2(map<int,meccs> m)
103  {
104      cout << "---------------------------------- 2.feladat
----------------------------------" << endl;
105      cout << "Mérkõzések idõrendben:" << endl;
106      vector<string> datumok;
107      for(auto ms:m)
108      {
109          if(ms.second.tipus=="bajnoki") datumok.push_back(ms.second.datum);
110      }
111      sorting_a_vector(datumok,true);
112      //sort(datumok.begin(),datumok.end());
113      ofstream fki;
114      fki.open("bajnoki.txt");
115      for(auto d:datumok)
116      {
117          cout << d << endl;
118          fki << d <<endl;
119      }
120      fki.close();
121  }
122
123  void F3(map<int,meccs> m)
124  {
125      cout << "---------------------------------- 3.feladat
----------------------------------" << endl;
126      cout << "Utolsó meccsre a jegy ára:" << endl;
127      vector<string> datumok;
128      for(auto ms:m)
```

```
129      {
130          datumok.push_back(ms.second.datum);
131      }
132      sorting_a_vector(datumok,false);
133      for(auto ms:m)
134      {
135          if(ms.second.datum==datumok[0])
136          {
137              cout << ms.second.belepo << " Ft (" << datumok[0] <<")" << endl;
138              break;
139          }
140      }
141  }
142
143  void F4(map<int,meccs> &m,vector<belepes> b)
144  {
145      cout << "---------------------------------- 4.feladat
----------------------------------" << endl;
146      cout << "Nézõszámok az egyes meccseken:" << endl;
147      vector<pair<string,int>> datum_nezoszam;
148      map<string,int> temp;
149      for(auto be:b)
150      {
151          m[be.meccsid].nezoszam++;
152      }
153      for(auto be:b)
154      {
155          temp[m[be.meccsid].datum]=m[be.meccsid].nezoszam;
156      }
157      for(auto items:temp)
158      {
159          datum_nezoszam.push_back(items);
160      }
161      sort(datum_nezoszam.begin(),datum_nezoszam.end());
162      for(auto elem:datum_nezoszam)
163      {
164          cout << elem.first << " " << elem.second << endl;
165      }
166  }
167
168  void F5(map<int,meccs> m,map<int,nezo> n,vector<belepes> b)
169  {
170      cout << "---------------------------------- 5.feladat
----------------------------------" << endl;
171      cout << "Bevétel 2018 szeptemberében jegyeladásból:" << endl;
172      int bevetel=0;
173      for(int i=0;i<b.size();i++)
174      {
175          if(n[b[i].nezoid].berletes==0)
176          {
177              if(m[b[i].meccsid].datum.substr(0,7)=="2018.09") bevetel+=m[b[i].meccsid
].belepo;
178          }
179      }
180      cout << bevetel << " Ft" << endl;
181  }
182
183  void F6(map<int,meccs> m,map<int,nezo> n,vector<belepes> b)
184  {
185      cout << "---------------------------------- 6.feladat
----------------------------------" << endl;
186      cout << "Kiss Dorina és Varga Zsolt ezeken a meccseken voltak együtt:" << endl;
187      vector<string> datum_kiss;
188      vector<string> datum_kozos;
189      for(int i=0;i<b.size();i++)
190      {
```

```
191         if(n[b[i].nezoid].vnev+" "+n[b[i].nezoid].knev=="Kiss Dorina")
192         {
193             datum_kiss.push_back(m[b[i].meccsid].datum);
194         }
195     }
196     for(auto datum:datum_kiss)
197     {
198         for(int i=0;i<b.size();i++)
199         {
200             if(n[b[i].nezoid].vnev+" "+n[b[i].nezoid].knev=="Varga Zsolt"&&m[b[i].
meccsid].datum==datum)
201             {
202                 datum_kozos.push_back(m[b[i].meccsid].datum);
203             }
204         }
205     }
206     sort(datum_kozos.begin(),datum_kozos.end());
207     for(auto datum:datum_kozos)
208     {
209         cout << datum << endl;
210     }
211 }
212
213 void F7(map<int,meccs> m,map<int,nezo> n,vector<belepes> b)
214 {
215     cout << "---------------------------------- 5.feladat
----------------------------------" << endl;
216     cout << "Bevétel a teljes idényben jegyeladásból:" << endl;
217 }
218
219 int main()
220 {
221     setlocale(LC_ALL,"");
222     map<int,nezo> nezok;
223     map<int,meccs> meccsek;
224     vector<belepes> belepesek;
225     F1nezo(nezok);
226     F1meccs(meccsek);
227     F1belepes(belepesek);
228     F2(meccsek);
229     F3(meccsek);
230     F4(meccsek,belepesek);
231     F5(meccsek,nezok,belepesek);
232     F6(meccsek,nezok,belepesek);
233     return 0;
234 }
```