

# **Adatbázis rendszerek I.**

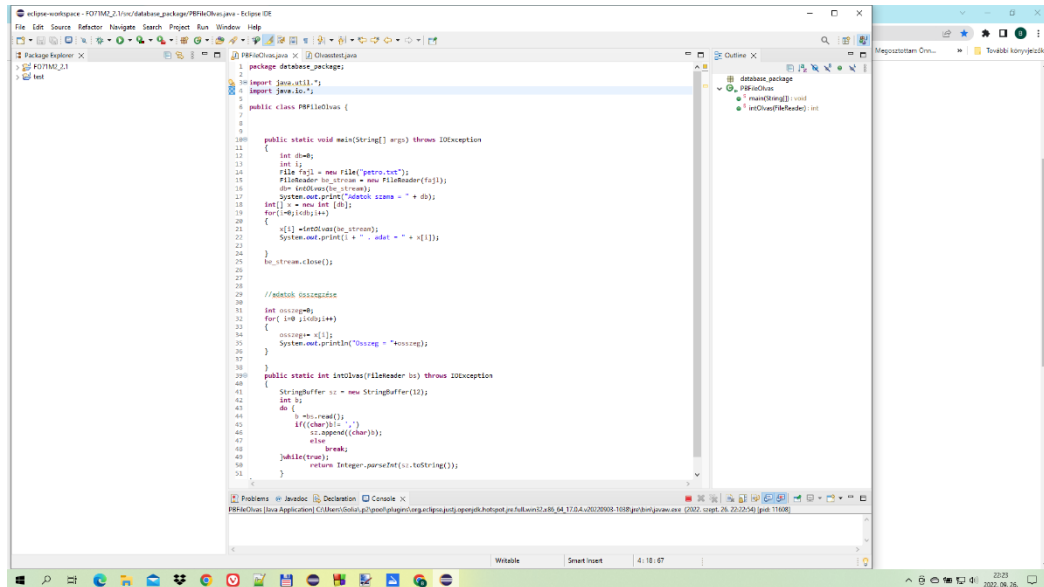
## **BSc**

**2. Gyak.**  
**2022. 09. 20**

**Készítette:**  
Petró Balázs Bsc  
Mérnökinformatikus  
FO71M2

**Miskolc, 2022**

- 1. feladat** Írjon programot, amely egész típusú adatokat beolvassa a szöveges vezeteknev.txt állományból, kiszámítja az adatok összegét és kiírja a konzolra!



```
1 package database_package;
2
3 import java.util.*;
4 import java.io.*;
5
6 public class PBP1edOlvas {
7
8
9
10
11
12 public static void main(String[] args) throws IOException
13 {
14     int db=0;
15     int sz;
16     File fszl = new File("vezeteknev.txt");
17     FileReader fr_stream = new FileReader(fszl);
18     BufferedReader br = new BufferedReader(fr_stream);
19     System.out.println("Adatok száma = " + db);
20     int[] s = new int [db];
21     for(int i=0;i<db;i++)
22     {
23         s[i] = Integer.parseInt(br.readLine());
24         System.out.println(i + " . adat = " + s[i]);
25     }
26     br.close();
27
28     //metódus összegzés
29     int osszeg=0;
30     for(int i=0;i<db;i++)
31     {
32         osszeg+=s[i];
33     }
34     System.out.println("Összeg = " + osszeg);
35
36 }
37
38 public static int irtolvas(FileReader fr) throws IOException
39 {
40     Stringbuffer sz = new StringBuffer(12);
41     int sz;
42     do
43     {
44         sz.append(fr.read());
45         if((char)sz.charAt(sz.length()-1) == '\n')
46             sz.deleteCharAt(sz.length()-1);
47     }while(true);
48     return Integer.parseInt(sz.toString());
49 }
50 }
```

- 2. feladat** Írjon egy programot, amely egész típusú adatokat ír a vezeteknev.txt állományba!  
Először meg kell adni a beírt adatok számát pl.: 3 db egész számot szeretnék beírni, majd a következő sorba külön-külön a számokat. Ezeket a számok beírja a vezeteknev.txt nevű állományba egymás alá. A következő, hogy meg is jeleníti a konzolon.

3. **feladat** A szabvány billentyűzetről olvasson be sorokat, egészen a “end” szóig. A beolvasott sorokat írja ki egy szövegfile-ba. A szövegfile nevét a bevitel első sorában adja meg. Az így létrehozott, lezárt állományt utána nyissa meg és írja vissza a lementett szöveget nagybetűs formában.
4. **feladat** Az előbb létrehozott, lezárt állományt nyissa meg és írja vissza a lementett szöveget nagybetűs formában a képernyőre.

A két feladat megoldása 1 javán belül:

```
1 package main;
2
3
4 import java.nio.file.Path;
5 import java.util.List;
6 import java.nio.charset.StandardCharsets;
7 import java.io.IOException;
8 import java.nio.file.Files;
9 import java.nio.file.OpenOption;
10 import java.nio.file.Paths;
11 import java.util.ArrayList;
12 import java.util.Scanner;
13
14 public class FO71M2_NOV20_B
15 {
16     static Scanner sc;
17
18     static {
19         FO71M2.sc = new Scanner(System.in);
20     }
21
22     public static void main(final String[] args) {
23         String line = "";
24         final List<String> lines = new ArrayList<String>();
25         while (!line.contains("end")) {
26             line = FO71M2.sc.next().toString();
27             lines.add(line);
28         }
29         final Path path = Paths.get(String.valueOf(lines.get(0)) + "petro.txt", new String[0]);
30         lines.remove(0);
31         try {
32             Files.write(path, lines, new OpenOption[0]);
33         }
34         catch (IOException e) {
35             System.out.println("network");
36         }
37         List<String> biglines = null;
38         try {
39             biglines = Files.readAllLines(path, StandardCharsets.UTF_8);
40         }
41         catch (IOException e2) {
42             System.out.println("network 2");
43         }
44         biglines.stream().forEach(k -> System.out.println(k.toUpperCase()));
45     }
46 }
```

**5. feladat** Végezze el egy fájl tartalmának másolását egy másik fájlba. Másolás közben a számjegyeket cserélje le szöveges alakra, szóközzel határolva. A másoló függvény a file neveket az argumentumában kapja meg.



```
C:\Users\Golia\Desktop\2,5,6,7,8,9,DBSEMTASK\2\Java_progis_hf_F_file.java - Notepad++
Fájl Szerkesztés Keresés Nézet Kódolás Nyelv Beállítások Eszközök Makró Futtatás Bővítmények Ablakok ?

package main;

public class Auto
{
    String rsz;
    String tipus;
    int ar;

    public Auto(String rsz, String tipus, int ar)
    {
        this.rsz = rsz;
        this.tipus = tipus;
        this.ar = ar;
    }

    @Override
    public String toString()
    {
        return rsz + " " + tipus + " " + ar;
    }
}
```

Java source file length: 288 lines: 21 Ln: 3 Col: 19 Pos: 36 Windows (CR LF) UTF-8 INS 16:00 2022. 11. 20.

```
C:\Users\Golia\Desktop\2,5,6,7,8,9,DBSEMTASK\2\Java_progis_hf_E_file.java - Notepad++
Fájl Szerkesztés Keresés Nézet Kódolás Nyelv Beállítások Eszközök Makró Futtatás Bővítmények Ablakok ?

package main;

import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;
import java.util.ArrayList;
import java.util.List;

public class FO7IM2_NOV20_E
{
    public static void main(String[] args)
    {
        Auto[] cars = {new Auto("R11", "Opel", 333),
                       new Auto("R12", "Fiat", 233),
                       new Auto("R13", "Skoda", 364)};

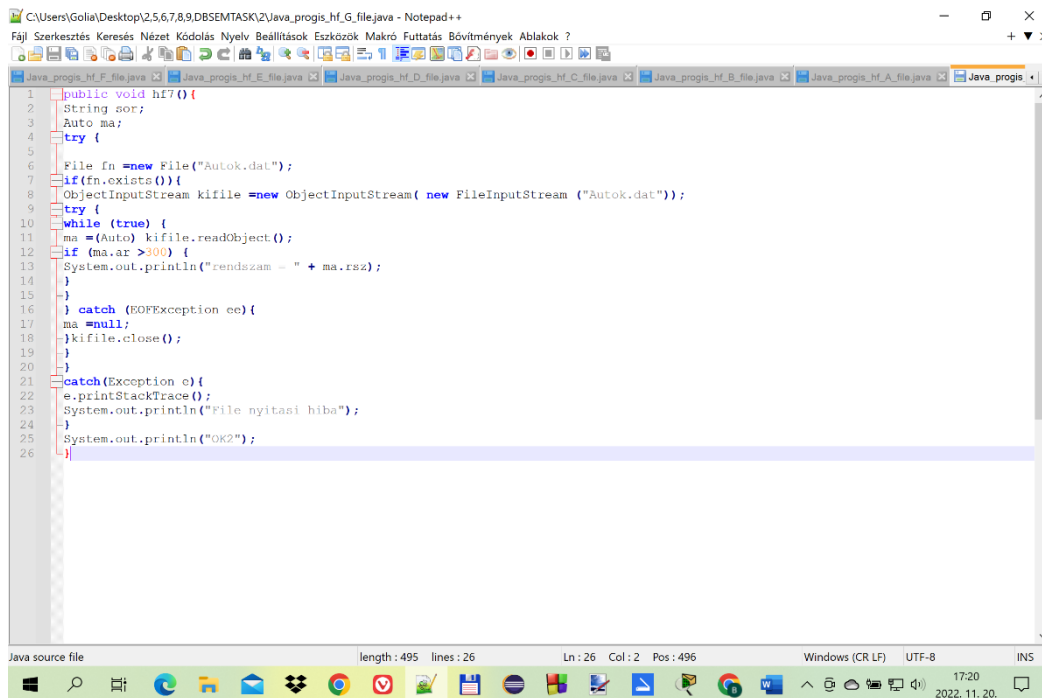
        List<Auto> carlist = new ArrayList<Auto>();
        for (Auto car : cars) carlist.add(car);

        try
        {
            FileOutputStream writeData = new FileOutputStream("outputFO7IM2.ser");
            ObjectOutputStream writeStream = new ObjectOutputStream(writeData);

            writeStream.writeObject(carlist);
            writeStream.flush();
            writeStream.close();
        } catch (IOException e)
        {
            System.out.println("mincs kimenet");
        }
    }
}
```

Java source file length: 825 lines: 37 Ln: 10 Col: 28 Pos: 201 Windows (CR LF) UTF-8 INS 16:12 2022. 11. 20.

**7. feladat** Készítsen programot, amely az előző autó nyilvántartóból kiírja a 300-nál drágább autók rendszámait.



```
1 public void hf7() {
2     String sor;
3     Auto ma;
4     try {
5         File fn = new File("Autok.dat");
6         if (fn.exists()) {
7             ObjectInputStream kifile = new ObjectInputStream(new FileInputStream("Autok.dat"));
8             try {
9                 while (true) {
10                     ma = (Auto) kifile.readObject();
11                     if (ma.ar > 300) {
12                         System.out.println("rendszám = " + ma.rs2);
13                     }
14                 }
15             } catch (EOFException ee) {
16                 ma = null;
17             } kifile.close();
18         }
19     } catch (Exception e) {
20         e.printStackTrace();
21         System.out.println("File nyitási hiba");
22     }
23     System.out.println("OK2");
24 }
25 }
```

**8. feladat** Készítsen programot, mely fel tud vinni személyeket (azonosító és név) bináris fájlba.

Készítsen függvényt a) új rekordot létrehozatalára, b) létező rekord törlésére c) létező rekord módosítására.

Útmutató:

- használjon bináris file-t
- használjon saját osztályt szerkezetet
- törlésnél másolja át a maradó részt

**9. feladat** Az autókat tároló adatfile-ban végezze el az alábbi lekérdezési műveleteket:

Számítsa ki a fájlban eltárolt autók átlagárát.

Kérdezze le az eltárolt piros autók darabszámát.

Keresse meg a legdrágább autót a fájlban.

Útmutató:

olvassa át az állományt rekordonként

a szükséges adatokat emelje ki a rekordból

végezze ez a szükséges számításokat memória változók segítségével.

