# Adatbázis Rendszerek I. BSc

2. gyak

2022. szeptember 20.

#### Készítette:

Martinák Mátyás BSc Programtervező informatikus alapszak KLNSPG

Miskolc, 2022

Írjon programot, amely egész típusú adatokat beolvassa a szöveges vezeteknev.txt állományból, kiszámítja az adatok összegét és kiírja a konzolra! Osztály neve: XYFileOlvas

```
public class KLSNPG 2 1 {
private static final Pattern COMMA = Pattern.compile(",");
public static void main(String[] args) {
    int db = 0, szum = 0, i = 0;
    try {
        BufferedReader myReader =
                new BufferedReader(new FileReader("vezeteknev.txt"));
        Scanner s = new Scanner (System.in);
        String str = s.nextLine();
        db = Integer.parseInt(str);
        String line;
        System.out.println("Adatok szama = " + db);
        while ((line = myReader.readLine()) != null) {
             for (String token : COMMA.split(line)) {
                try {
                     szum += Integer.parseInt(token);
                     System.out.println((i+1) + ".adat = " + token);
                 } catch (NumberFormatException ex) {
                     System.err.println(token + " nem egy szam");
                i++;
        System.out.println("Osszeg: " + szum);
      } catch (Exception e) {
        System.out.println("Hiba.");
         e.printStackTrace();
```

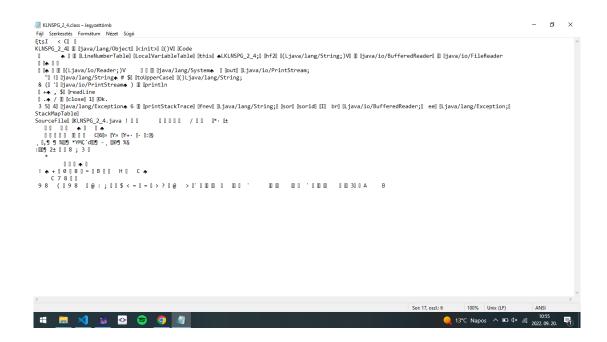
Írjon egy programot, amely egész típusú adatokat ír a vezeteknev.txt állományba!

```
public class KLNSPG_2_2 {
private static final Pattern COMMA = Pattern.compile(regex: ",");
public static void main(String[] args) {
    int db = 0, szum = 0, i = 0, szam = 0;
        BufferedWriter myReader =
               new BufferedWriter(new FileWriter(fileName: "martinak.txt"));
        Scanner s = new Scanner (System.in);
        String str = s.nextLine();
        db = Integer.parseInt(str);
        int[] szamok = new int[db];
        //String line;
        System.out.println("Adatok szama = " + db);
        while (i < db) {
                    str = s.nextLine();
                    szam = Integer.parseInt(str);
                    szum += szam;
                    System.out.println((i+1) + ".adat = " + szam);
                } catch (NumberFormatException ex) {
                    System.err.println(szam + " nem egy szam");
        myReader.close();
        s.close();
        catch (Exception e) {
            System.out.println(x: "Hiba.");
            e.printStackTrace();
        System.out.println("Osszeg: " + szum);
```

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.



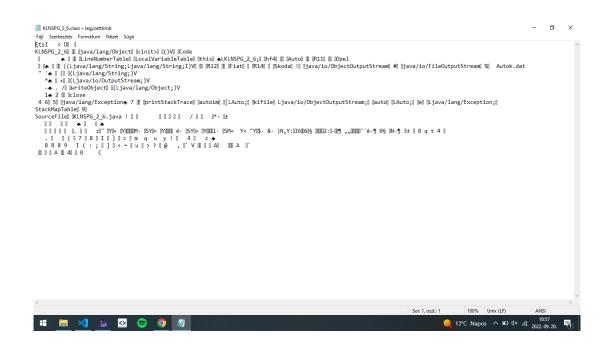
Az előbb létrehozott, lezárt állományt nyissa meg és írja vissza a lementett szöveget nagybetűs formában a képernyőre.



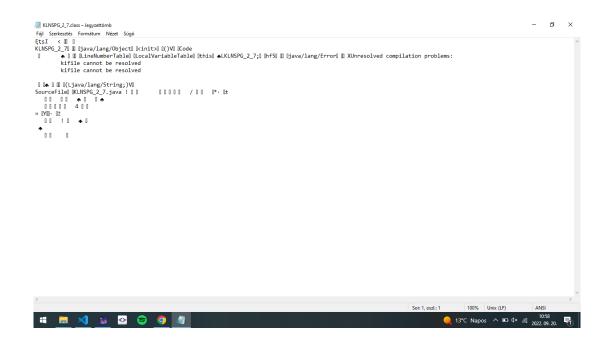
Végezze el egy fájl tartalmának másolását egy másik fájlba. Másolás közben a számjegyeket cserélje le szöveges alakra, szóközökel határolva. A másoló függvény a fileneveket az argumentumában kapja meg.



Tároljon le auto (rendszam, tipus, ar) rekordokat egymás után egy bináris állományban, majd készítsen függvényt az i. rekord visszaolvasására.



Készítsen programot, amely felőző autó nyilvántartóból kiírja a 300-nál drágább autók rendszámait.



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.



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.

