

# **Adatbázis rendszerek 1. BSc**

**2.Gyak**

**2022.09.20.**

**Készítette:**

Bodnár László BSc

Szak:

Mérnökinformatikus

Neptunkód D1H8VP **2022.09.20.**

## **1. feladat**

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

**Mentés: neptunkod\_2.1.java A futtatás eredménye:**

**Adatok száma = 2**

**0.adat = 10**

**1.adat = 20**

**Összeg: 30**

```
1 package Csomag;
2
3 import java.io.BufferedReader;
4 import java.io.BufferedWriter;
5 import java.io.FileWriter;
6 import java.io.InputStreamReader;
7
8 public class D1H8VP {
9
10     public static void main(String[] args) {
11         olvasas();
12     }
13
14     public static void olvasas() {
15         String sor;
16         String[] szavak;
17         int sorid = 0;
18         try {
19             System.out.println("Az első szöveg a txt fájl neve\n");
20             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
21             BufferedWriter bw = null;
22             System.out.println("Írj szavakat, a program az 'end' szöveg olvas!\n");
23
24             while (sorid >= 0) {
25                 sor = br.readLine();
26                 if (sorid == 0) {
27                     bw = new BufferedWriter(new FileWriter(sor));
28                 } else {
29                     bw.write(sor);
30                     bw.newLine();
31                 }
32                 sorid = sorid + 1;
33                 szavak = sor.split(" ");
34                 for (String sz : szavak) {
35                     if (sz.compareTo("end") == 0) {
36                         br.close();
37                         sorid = -1;
38                     }
39                 }
40             }
41             bw.close();
42         } catch (Exception e) {
43             e.printStackTrace();
44         }
45     }
46 }
```

**2. feladat**

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

**Osztály neve: XYFileIr**

## Mentés: neptunkod\_2.2.java

```
package diH8vp;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.InputStreamReader;

public class D1H8VP {

    public static void main(String[] args) {

        visszair(olvasas());

    }

    public static String olvasas() {
        String sor;
        String[] szavak;
        int sorid = 0;
        String nev="valami";

        try {
            System.out.println("Az első szó a txt fájl neve");
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            BufferedWriter bw = null;
            System.out.println("Írj szavakat, a program 'end' szöveg olvas!\n");
            while (sorid >= 0) {
                sor = br.readLine();
                if (sorid == 0) {
                    bw = new BufferedWriter(new FileWriter(sor));
                    nev = sor;
                } else {
                    bw.write(sor);
                    bw.newLine();
                }

                sorid = sorid + 1;
                szavak = sor.split(" ");
                for (String sz : szavak) {
                    if (sz.compareTo("end") == 0) {
                        br.close();
                        sorid = -1;
                    }
                }
            }
        }
    }
}
```

**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.**

**A futás eredménye:**

**Adatok száma = 3**

**0.adat = 10**

**1.adat = 20**

**3.adat = 30**

**Adatok kiírása: vezeteknev.txt állományba**

**10**

**20**

**30**

```

1 package Csomag;
2
3 import java.io.BufferedReader;
4 import java.io.BufferedWriter;
5 import java.io.File;
6 import java.io.InputStreamReader;
7
8 public class D1H8VP {
9
10     public static void main(String[] args) {
11
12         olvasas();
13     }
14
15     public static void olvasas() {
16         String sor;
17         String[] szavak;
18         int sorid = 0;
19         try {
20             System.out.println("Az első szó a txt fájl neve\n");
21             BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
22             BufferedWriter bw = null;
23             System.out.println("Írj szavakat, a program az 'end' szóig olvas!\n");
24
25             while (sorid >= 0) {
26                 sor = br.readLine();
27                 if (sorid == 0) {
28                     bw = new BufferedWriter(new FileWriter(sor));
29                 } else {
30                     bw.write(sor);
31                     bw.newLine();
32                 }
33                 sorid = sorid + 1;
34                 szavak = sor.split(" ");
35                 for (String sz : szavak) {
36                     if (sz.compareTo("end") == 0) {
37                         br.close();
38                         sorid = -1;
39                     }
40                 }
41             }
42             bw.close();
43         } catch (Exception e) {
44             e.printStackTrace();
45         }
46     }
47 }

```

### 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.

Útmutató:

## Mentés: neptunkod\_2.3.java

```
2
3 import java.io.BufferedReader;
4 import java.io.BufferedWriter;
5 import java.io.FileReader;
6 import java.io.FileWriter;
7
8 public class D1H8VP {
9
10 public static void main(String[] args) {
11
12     masol("ebbol.txt", "ebbe.txt");
13
14 }
15
16 public static void masol(String fnevbe, String fnevki) {
17     String sor;
18     String[] k1 = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "0" };
19     String[] k2 = { " egy ", " kettő ", " három ", " négy ", " öt ", " hat ", " hét ", " nyolc ", " kilenc ",
20         " nulla" };
21
22     try {
23         BufferedWriter bw = new BufferedWriter(new FileWriter(fnevki));
24         BufferedReader br = new BufferedReader(new FileReader(fnevbe));
25         while ((sor = br.readLine()) != null) {
26             for (int i = 0; i < 10; i++) {
27                 sor = sor.replace(k1[i], k2[i]);
28             }
29             bw.write(sor);
30             bw.newLine();
31         }
32
33         br.close();
34         bw.close();
35         System.out.println("Ok");
36     } catch (Exception ee) {
37         ee.printStackTrace();
38     }
39 }
```

### 4. feladat

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

Útmutató

## Mentés: neptunkod\_2.4.java

```
import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;

public class D1H8VP implements Serializable {

    public static void main(String[] args) {

        hf4();
        hf5();

    }

    private static final long serialVersionUID = 1L;
    String rsz;
    String tipus;
    int ar;

    public D1H8VP(String r, String t, int a) {
        this.rsz = r;
        this.tipus = t;
        this.ar = a;
    }

    public static void hf4() {
        D1H8VP[] autoim = { new D1H8VP("R11", "Opel", 333), new D1H8VP("R12", "Fiat", 233),
            new D1H8VP("R14", "Skoda", 364) };
        try {
            ObjectOutputStream kifile = new ObjectOutputStream(new FileOutputStream("Autok.dat"));
            for (D1H8VP auto : autoim) {
                kifile.writeObject(auto);
            }
            kifile.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("File nyitási hiba");
        }
        System.out.println("OK");
    }

    public static void hf5() {
```

## 5. feladat

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

Útmutató:

## Mentés: neptunkod\_2.5.java

```
import java.io.EOFException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;

public class D1H8VP implements Serializable {

    public static void main(String[] args) {

        hf4();
        hf5();
    }

    private static final long serialVersionUID = 1L;
    String rsz;
    String tipus;
    int ar;

    public D1H8VP(String r, String t, int a) {
        this.rsz = r;
        this.tipus = t;
        this.ar = a;
    }

    public static void hf4() {
        D1H8VP[] autoim = { new D1H8VP("R11", "Opel", 333), new D1H8VP("R12", "Fiat", 233),
            new D1H8VP("R14", "Skoda", 364) };
        try {
            ObjectOutputStream kifile = new ObjectOutputStream(new FileOutputStream("Autok.dat"));
            for (D1H8VP auto : autoim) {
                kifile.writeObject(auto);
            }
            kifile.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("File nyitási hiba");
        }
        System.out.println("OK");
    }

    public static void hf5() {
```

## 6. feladat

**Tároljon le auto (rendszer, típus, ár) rekordokat egymás után egy bináris állományban, majd készítsen függvényt az i. rekord visszaolvasására.**

## Útmutató



# Mentés: neptunkod\_2.6.java

```
public class D1H8VP {  
  
    private static D1H8VP adatok[];  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
  
        int szam = 0;  
  
        while (szam != 5) {  
            System.out.println(  
                "Mit szeretnél csinálni? \n1. Adatok felvitele\n2. Adatok torlese\n3. Adatok modositasa\n4. Adatok listazasa\n5. Kilepes\n\n");  
            szam = sc.nextInt();  
            switch (szam) {  
  
                case 1:  
                    int olvas = szamotker(sc);  
                    adatok = new D1H8VP[olvas];  
                    try {  
                        beker(adatok, sc);  
                    } catch (IOException e) {  
                        e.printStackTrace();  
                    }  
                    break;  
                case 2:  
                    try {  
                        torles(adatok, sc);  
                    } catch (IOException e) {  
                        e.printStackTrace();  
                    }  
                    break;  
                case 3:  
                    try {  
                        modosit(adatok, sc);  
                    } catch (IOException e1) {  
                        e1.printStackTrace();  
                    }  
                    break;  
            }  
        }  
    }  
}
```