

Adatbázis rendszerek 1. BSc

2.Gyak

2022.09.20.

Készítette:

Nyíri Levente BSc

Szak:

Mérnökinformatikus

Neptunkód: F023QC

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

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class NYLFileOlvas {

    Run | Debug
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("Adja meg hány adatot szeretne beolvasni: ");
        boolean ok = true;
        int olvas = 0;

        do {
            try {
                ok = true;
                olvas = sc.nextInt();
                if (olvas <= 0 || olvas > 10) {
                    System.out.println("Nem megfelelő számot adott meg!\n");
                    ok = false;
                }
            } catch (NumberFormatException e) {
                System.out.println(e);
                ok = false;
            }
        } while (!ok);

        read_text_file(olvas);

        sc.close();
    }

    public static void read_text_file(int olvas) {
        String fnev = "nyiri.txt";
        int db = 0;
        int sum = 0;
        int szam;

        try {
            File myObj = new File(fnev);
            Scanner sc = new Scanner(myObj);
            while (sc.hasNextLine() && db < olvas) {
                szam = sc.nextInt();
                sum = sum + szam;
                System.out.println("A(z) " + db + ". adat: " + szam);
                db++;
            }
            sc.close();
        } catch (FileNotFoundException e) {
            System.out.println("Hiba!\n");
            e.printStackTrace();
        }

        System.out.println("Az összeg: " + sum);
    }
}
```

2. feladat

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

Osztály neve: XYFileIr

Mentés: neptunkod_2.2.java

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

```

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class NYFileIr {

    Run|Debug
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("Add meg hogy adatot szeretn-e l beolvasni: ");
        boolean ok = true;
        int olvas = 0;
        do {
            try {
                ok = true;
                olvas = sc.nextInt();
                if (olvas <= 0 || olvas > 10) {
                    System.out.println("Nem megfellel-e szant adott meg!\n");
                    ok = false;
                }
            } catch (NumberFormatException e) {
                System.out.println(e);
                ok = false;
            }
        } while (!ok);

        write_text_file(olvas);
        read_text_file(olvas);

        sc.close();
    }

    public static void write_text_file(int olvas) {
        String fnev = "Myiri.txt";
        int db = 0;
        Scanner sc = new Scanner(System.in);
        boolean ok = true;
        String szam = "valami";

        try {
            FileWriter myWriter = new FileWriter(fnev);
            System.out.println("Adjon meg " + olvas + " db szant!");
            while (db < olvas) {

                do {
                    try {
                        ok = true;
                        szam = sc.nextLine();
                    } catch (NumberFormatException e) {
                        System.out.println(e);
                        ok = false;
                    }
                } while (!ok);

                db++;
                myWriter.write(szam + "\n");

            }
            myWriter.close();
            System.out.println("Az iras sikeres volt.");
        } catch (IOException e) {
            System.out.println("IO hiba");
            e.printStackTrace();
        }

        sc.close();
    }

    public static void read_text_file(int olvas) {
        String fnev = "Myiri.txt";
        int db = 0;
        int szam;

        try {
            File myObj = new File(fnev);
            Scanner sc = new Scanner(myObj);
            while (sc.hasNextLine() && db < olvas) {
                szam = sc.nextInt();
                System.out.println("Az " + db + ". adat: " + szam);
                db++;
            }
            sc.close();

        } catch (FileNotFoundException e) {
            System.out.println("Hiba!");
            e.printStackTrace();
        }
    }
}

```

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

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;

public class F023QC_2_3 {

    public static void main(String[] args) {

        Run | Debug
        olvasas();
    }

    public static void olvasas() {
        String sor;
        String[] szavak;
        int sorid = 0;
        try {

            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            BufferedWriter bw = null;

            while (sorid >= 0) {
                sor = br.readLine();
                if (sorid == 0) {
                    bw = new BufferedWriter(new FileWriter(sor));
                } else {
                    bw.write(sor);
                    bw.newLine();
                }
                sorid = sorid + 1;
                szavak = sor.split(regex: " ");
                for (String sz : szavak) {
                    if (sz.compareTo(anotherString: "end") == 0) {
                        br.close();
                        sorid = -1;
                    }
                }
            }
            bw.close();
            System.out.println(x: "Ok");
        } catch (Exception ee) {
            ee.printStackTrace();
        }
    }
}
```

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

public class #029QC_2_3 {

    Run | Debug
    public static void main(String[] args) {

        visszair(olvasas());

    }

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

        try {
            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
            BufferedWriter bw = null;
            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(regex: " ");
                for (String sz : szavak) {
                    if (sz.compareTo(AnotherStrings: "end") == 0) {
                        br.close();
                        sorid = -1;
                    }
                }
            }
            bw.close();
            System.out.println(Ok: "Ok");
        } catch (Exception ee) {
            ee.printStackTrace();
        }

        return nev;
    }

    public static void visszair(String fnev) {
        String sor;
        try {
            BufferedReader br = new BufferedReader(new FileReader(fnev));
            while ((sor = br.readLine()) != null) {
                System.out.println(sor.toUpperCase());
            }
            br.close();
            System.out.println(Ok: "Ok");
        } catch (Exception ee) {
            ee.printStackTrace();
        }
    }
}
```

5. feladat

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

Útmutató:

Mentés: neptunkod_2.5.java

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

public class F023QC_2_5 {

    Run | Debug
    public static void main(String[] args) {

        masol("FROM.txt", "TO.txt");

    }

    public static void masol(String fnevbe, String fnevki) {
        String sor;
        String[] k1 = { "1", "2", "3", "4", "5", "6", "7", "8", "9", "0" };
        String[] k2 = { " egy ", " kettő ", " három ", " négy ", " öt ", " hat ", " hat ", " nyolc ", "kilenc ",
            " nulla" };

        try {
            BufferedWriter bw = new BufferedWriter(new FileWriter(fnevki));
            BufferedReader br = new BufferedReader(new FileReader(fnevbe));
            while ((sor = br.readLine()) != null) {
                for (int i = 0; i < 10; i++) {
                    sor = sor.replace(k1[i], k2[i]);
                }
                bw.write(sor);
                bw.newLine();
            }

            br.close();
            bw.close();
            System.out.println("Ok");
        } catch (Exception ee) {
            ee.printStackTrace();
        }
    }
}
```

6. feladat

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

Útmutató

Mentés: neptunkod_2.6.java

```
import java.io.IOException;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.Scanner;

public class FEJQC_2_6 implements Serializable {

    @Override
    public static void main(String[] args) {
        hfd();
        try {
            visszahivas();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }

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

    public FEJQC_2_6(String r, String t, int a) {
        this.rva = r;
        this.tipus = t;
        this.ar = a;
    }

    public static void hfd() {
        FEJQC_2_6[] auto = { new FEJQC_2_6("E05123", "K1 Mercedes", 340000), new FEJQC_2_6("AG0132", "K2 Opel", 230000),
            new FEJQC_2_6("B0C143", "K3 Honda", 3050000) };
        try {
            ObjectOutputStream kifile = new ObjectOutputStream(new FileOutputStream("Autok.dat"));
            for (FEJQC_2_6 auto : auto) {
                kifile.writeObject(auto);
            }
            kifile.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Nincs a fájl megnyitása során!\n");
        }
        System.out.println("OK");
    }

    public static void visszahivas() throws FileNotFoundException {
        Scanner sc = new Scanner(System.in);
        System.out.println("Melyik rekordot szeretné visszanevezni?\n");
        boolean ok = true;
        int olvas = 0;
        do {
            try {
                ok = true;
                olvas = sc.nextInt();
                if (olvas < 0 || olvas > 30) {
                    System.out.println("Nem jó számot adott meg!\n");
                    ok = false;
                }
            } catch (NumberFormatException e) {
                System.out.println(e);
                ok = false;
            }
        } while (!ok);
        sc.close();

        int sor = 1;
        FEJQC_2_6 ma;
        try {
            File fn = new File("Autok.dat");
            if (fn.exists()) {
                ObjectInputStream kifile = new ObjectInputStream(new FileInputStream("Autok.dat"));
                try {
                    while (true) {
                        ma = (FEJQC_2_6) kifile.readObject();
                        if (sor == olvas) {
                            System.out.println("Rendszám: " + ma.rva);
                            System.out.println("Típus: " + ma.tipus);
                            System.out.println("Ár: " + ma.ar);
                        }
                        sor++;
                    }
                } catch (IOException e) {
                    ma = null;
                }
                kifile.close();
            }
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Nincs a fájl nyitása során!\n");
        }
        System.out.println("OK");
        sc.close();
    }
}
```


7. feladat

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

Útmutató:

Mentés: neptunkod_2.7.java

```
import java.io.IOException;
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 F023QC_2_7 implements Serializable {

    Run | Debug
    public static void main(String[] args) {

        hf4();
        hf5();

    }

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

    public F023QC_2_7(String r, String t, int a) {
        this.rsz = r;
        this.tipus = t;
        this.ar = a;
    }

    public static void hf4() {
        F023QC_2_7[] autoim = { new F023QC_2_7("BDS131", "Mercedes", 340000), new F023QC_2_7("ASR132", "Opel", 230000),
            new F023QC_2_7("RBC143", "Honda", 3894000) };
        try {
            ObjectOutputStream kifile = new ObjectOutputStream(new FileOutputStream("Autok.dat"));
            for (F023QC_2_7 auto : autoim) {
                kifile.writeObject(auto);
            }
            kifile.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Hiba a fájl megnyitása során!\n");
        }
        System.out.println("OK");
    }

    public static void hf5() {
        F023QC_2_7 ma;
        try {
            File fn = new File("Autok.dat");
            if (fn.exists()) {
                ObjectInputStream kifile = new ObjectInputStream(new FileInputStream("Autok.dat"));
                try {
                    while (true) {
                        ma = (F023QC_2_7) kifile.readObject();
                        if (ma.ar > 300) {
                            System.out.println("rendszám=" + ma.rsz);
                        }
                    }
                } catch (IOException ee) {
                    ma = null;
                }
                kifile.close();
            }
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("File nyitási hiba");
        }
        System.out.println("OK2");
    }
}
```

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**

Mentés: neptunkod_2.8.java

```

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;

public class F023QC_2_8 {

    private static F023QC_2_8 adatok[];

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int szan = 0;

        while (szan != 5) {
            System.out.println("1. Adatok felvittele\n2. Adatok torlese\n3. Adatok modositasa\n4. Adatok listazasa\n5. Kilopes\n\n");
            szan = sc.nextInt();
            switch (szan) {

                case 1:
                    int olvas = szamotker(sc);
                    adatok = new F023QC_2_8[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;

                case 4:
                    try {
                        kiolvas(sc);
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                    break;

                case 5:
                    System.out.println("Kilépés...\n");
                    System.exit(status: 0);

                default:
                    System.out.println("Érvénytelen opció!\n");
            }
        }

        sc.close();
    }
}

```

```

private static int szamotker(Scanner sc) {
    System.out.println("Menny adatot szeretnél megadni? ");
    boolean ok = true;
    int olvas = 0;
    do {
        try {
            ok = true;
            olvas = sc.nextInt();
            if (olvas <= 0) {
                System.out.println("Nem lehet ennyi adat megadni!");
                ok = false;
            }
        } catch (NumberFormatException e) {
            System.out.println(e);
            ok = false;
        }
    } while (!ok);
    return olvas;
}

private String azonosito;
private String nev;
private static int ossz;

public static int getOssz() {
    return ossz;
}

public static void setOssz(int ossz) {
    F023QC_2_8.ossz = ossz;
}

public String getAzonosito() {
    return azonosito;
}

public void setAzonosito(String azonosito) {
    this.azonosito = azonosito;
}

public String getNev() {
    return nev;
}

public void setNev(String nev) {
    this.nev = nev;
}

public F023QC_2_8(String azonosito, String nev) {
    this.azonosito = azonosito;
    this.nev = nev;
}

@Override
public String toString() {
    return "F023QC_2_8 (Azonosito: " + azonosito + ", nev: " + nev + ")";
}

```

```

public static F023QC_2_8[] beker(F023QC_2_8[] adatok, Scanner sc) throws IOException {

    FileWriter myWriter = new FileWriter(fileName: "szemelyek.dat");

    for (int i = 0; i < adatok.length; i++) {

        System.out.println("Add meg az azonosítót: ");
        String azonosito = sc.next();

        System.out.println("Add meg a nevet: ");
        String nev = sc.next();

        adatok[i] = new F023QC_2_8(azonosito, nev);
        ossz++;

        try {
            myWriter.write(adatok[i].aznosito + " " + adatok[i].nev + "\n");
            myWriter.flush();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    myWriter.close();

    return adatok;
}

```

```

public static void torles(F023QC_2_8[] adatok, Scanner sc) throws IOException {

    File file = new File(pathname: "szemelyek2.dat");
    File file2 = new File(pathname: "szemelyek.dat");
    FileWriter myWriter = new FileWriter(fileName: "szemelyek2.dat");

    System.out.println("Hanyadik rekordot szeretned tarolni?");

    int bekert = sc.nextInt();
    bekert = bekert - 1;

    for (int i = 0; i < getOssz(); i++) {
        if (bekert == 0) {
            if (i != 0) {
                myWriter.write(adatok[i].aznosito + " " + adatok[i].nev + "\n");
                myWriter.flush();
                setOssz(getOssz()-1);
            }
        } else if (bekert != 0) {
            if (i != bekert) {
                myWriter.write(adatok[i].aznosito + " " + adatok[i].nev + "\n");
                myWriter.flush();
                setOssz(getOssz()-1);
            }
        }
    }

    myWriter.close();
    file2.delete();
    file.renameTo(file2);
}

```

```

    }

    public static void kiolvas(Scanner sc) throws IOException {

        String fnev = "szemelyek.dat";
        int db = 0;
        String valami;

        try {
            File myObj = new File(fnev);
            BufferedReader buff1 = new BufferedReader(new FileReader(myObj.getAbsolutePath()));
            sc = new Scanner(buff1);
            while (sc.hasNextLine()) {
                valami = sc.nextLine();
                System.out.println(db + ". adat: " + valami);
                db++;
            }
            buff1.close();
        } catch (FileNotFoundException e) {
            System.out.println("Hiba!\n");
            e.printStackTrace();
        }
    }

    public static void modosit(F023QC_2_8[] adatok, Scanner sc) throws IOException {

        System.out.println("Hanyadik rekordot szeretned modositani? " + getOssz() + "db adat van az allomanyban.");
        int szam = sc.nextInt();
        System.out.println("Mit szeretnel modositani?\n1.Azonosito\n2.Nev\n");
        int mit = sc.nextInt();

        FileWriter myWriter = new FileWriter(fileName: "szemelyek.dat");

        for (int i = 0; i < getOssz(); i++) {
            if(i==(szam-1)) {
                switch(mit){
                    case 1:
                        System.out.println("Add meg az azonositot! ");
                        String azonosito = sc.next();
                        adatok[i].azonosito= azonosito;
                        break;

                    case 2:
                        System.out.println("Add meg az új nevet! ");
                        String nev = sc.next();
                        adatok[i].nev=nev;
                        break;

                    default: System.out.println("Nincs ilyen opcio, kilopes.."); System.exit(-1);
                }
            }

            myWriter.write(adatok[i].azonosito + " " + adatok[i].nev + "\n");
            myWriter.flush();
        }

        myWriter.close();
    }
}

```

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.**

Mentés: neptunkod_2.9.java

```

F023QC_2.9.java > F023QC_2.9 > @main@main@
import java.io.Serializable;

public class F023QC_2.9 implements Serializable {

    private static int db = 0;

    @SuppressWarnings("unchecked")
    public static void main(String[] args) {
        try {
            atlag();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        try {
            lagdragabb();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }

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

    public F023QC_2.9(String r, String t, int a) {
        this.rax = r;
        this.tipus = t;
        this.ar = a;
    }

    public static void main() {
        F023QC_2.9[] auto = { new F023QC_2.9("F02111", "E1 Mercedes", 140000), new F023QC_2.9("F02112", "E1 Opel", 23000),
            new F023QC_2.9("F02113", "E1 Honda", 300000);
        db = auto.length;

        try {
            ObjectOutputStream kifile = new ObjectOutputStream(new FileOutputStream("Autok.dat"));
            for (F023QC_2.9 auto : auto) {
                kifile.writeObject(auto);
            }
            kifile.close();
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Hiba a faji meggyitasa során!\n");
        }
        System.out.println("OK");
    }

    public static void atlag() throws FileNotFoundException {
        int szam = 0;
        F023QC_2.9 ma;
        try {
            File fn = new File("Autok.dat");
            if (fn.exists()) {
                ObjectInputStream kifile = new ObjectInputStream(new FileInputStream("Autok.dat"));
                try {
                    while (true) {
                        ma = (F023QC_2.9) kifile.readObject();
                        szam = szam + ma.ar;
                    }
                } catch (EOFException ee) {
                    ma = null;
                }
                kifile.close();
            }
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Hiba a faji meggyitasa során!\n");
        }
        System.out.println("Az autok atlagara: " + szam/db);
    }

    public static void lagdragabb() throws FileNotFoundException {
        F023QC_2.9 ma;
        try {
            File fn = new File("Autok.dat");
            if (fn.exists()) {
                F023QC_2.9 max;
                ObjectInputStream kifile = new ObjectInputStream(new FileInputStream("Autok.dat"));
                ma = (F023QC_2.9) kifile.readObject();
                try {
                    while (true) {
                        ma = (F023QC_2.9) kifile.readObject();
                        if (ma.ar > max.ar) {
                            max = ma;
                        }
                    }
                } catch (EOFException ee) {
                    ma = null;
                }
                kifile.close();
                System.out.println("A lagdragabb auto rendszara: " + max.rax + " @ " + max.ar);
            }
        } catch (Exception e) {
            e.printStackTrace();
            System.out.println("Hiba a faji meggyitasa során!\n");
        }
    }
}

```