

Soubory

Jazyk JAVA



České vysoké učení technické Fakulta elektrotechnická

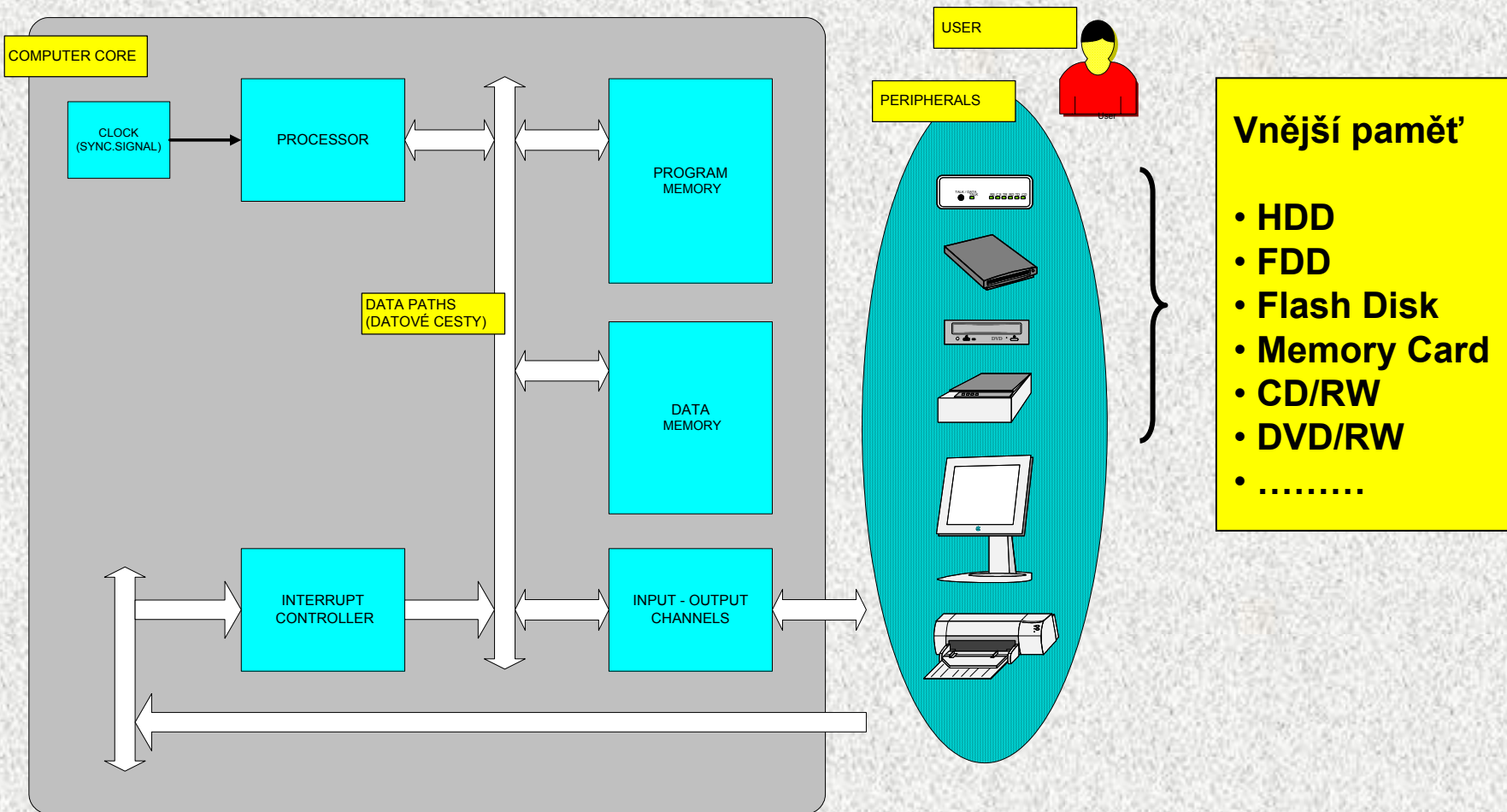
Obsah

- Soubory a vnější paměť počítače
- Soubor a adresář (File, Directory)
- Běžný adresář (Current Directory) a cesta
- Soubory, adresáře a metody
- Práce se souborem
- Postupný (sekvenční) přístup k datům v souboru
- Přímý (nahodilý) přístup k datům v souboru
- Typy souborů
- Přenos informace – dvě vrstvy
- Výjimky (Exceptions)
- Výjimky – vyhození, ošetření
- Kopie souboru – bajty
- Zápis a čtení jednoduchých typů - int
- Zápis jednoduchých typů - double
- Čtení jednoduchých typů - double

Obsah

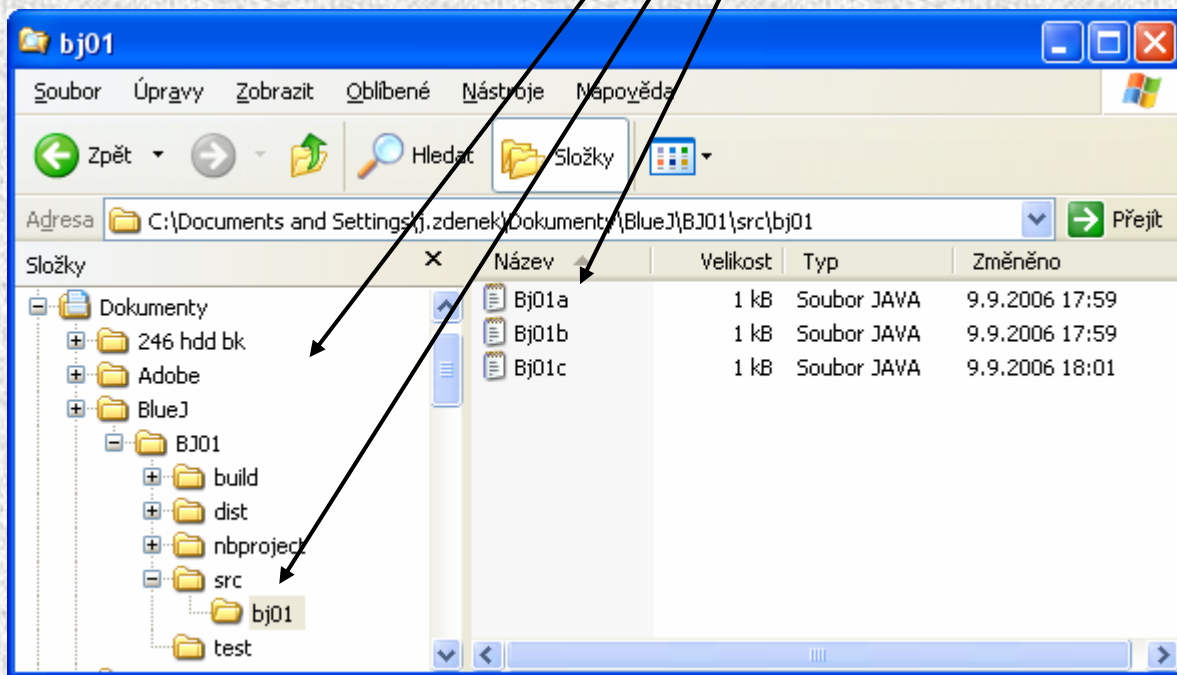
- Zápis objektů – string
- Čtení objektů – string
- Zápis objektů – pole – vyhození výjimky
- Čtení objektů – pole – vyhození výjimky
- Zápis objektů – pole – zachycení výjimky
- Čtení objektů – pole – zachycení výjimky
- Zápis objektu – vlastní třída
- Čtení objektu – vlastní třída
- Zápis textového souboru
- Čtení textového souboru
- Rozšíření textového souboru

Soubory a vnější paměť počítače



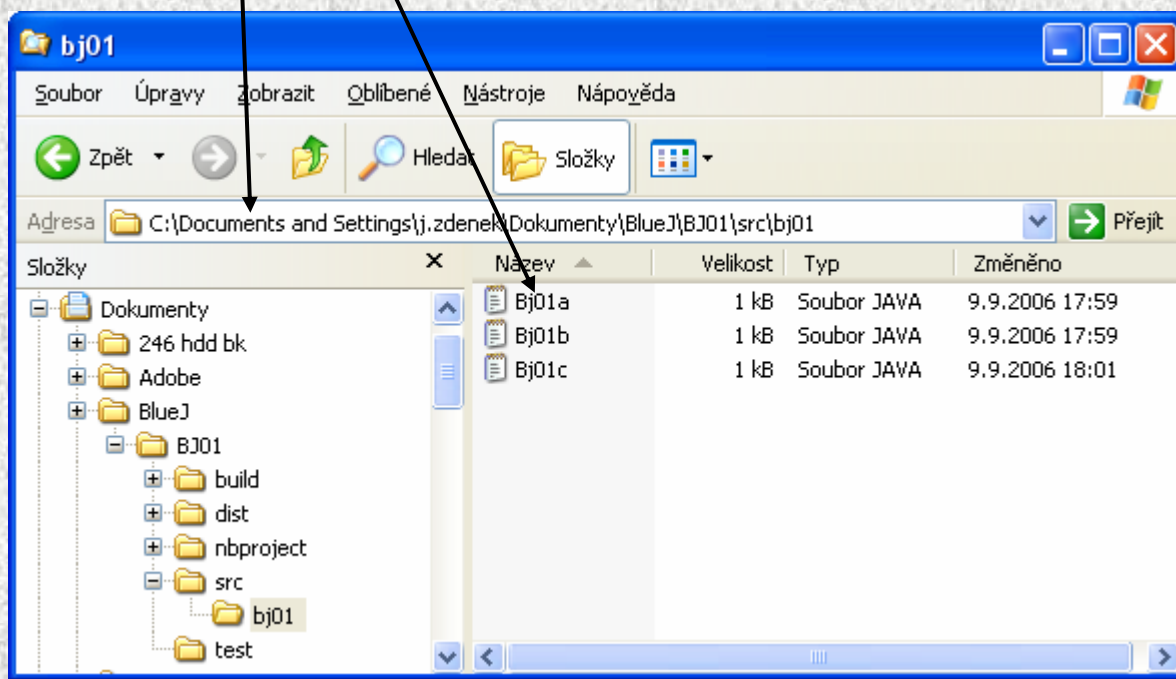
Soubor a adresář (File, Directory)

- Soubor - je tvořen daty uloženými ve vnější paměti počítače
- Vnější paměť počítače - různé druhy (HDD, FDD, CD/RW, DVD/RW, Flash, ...)
- Vnější paměť – má adresářovou strukturu
- Soubor - je uložen v určitém adresáři
- Soubor i adresáře mají svá jména



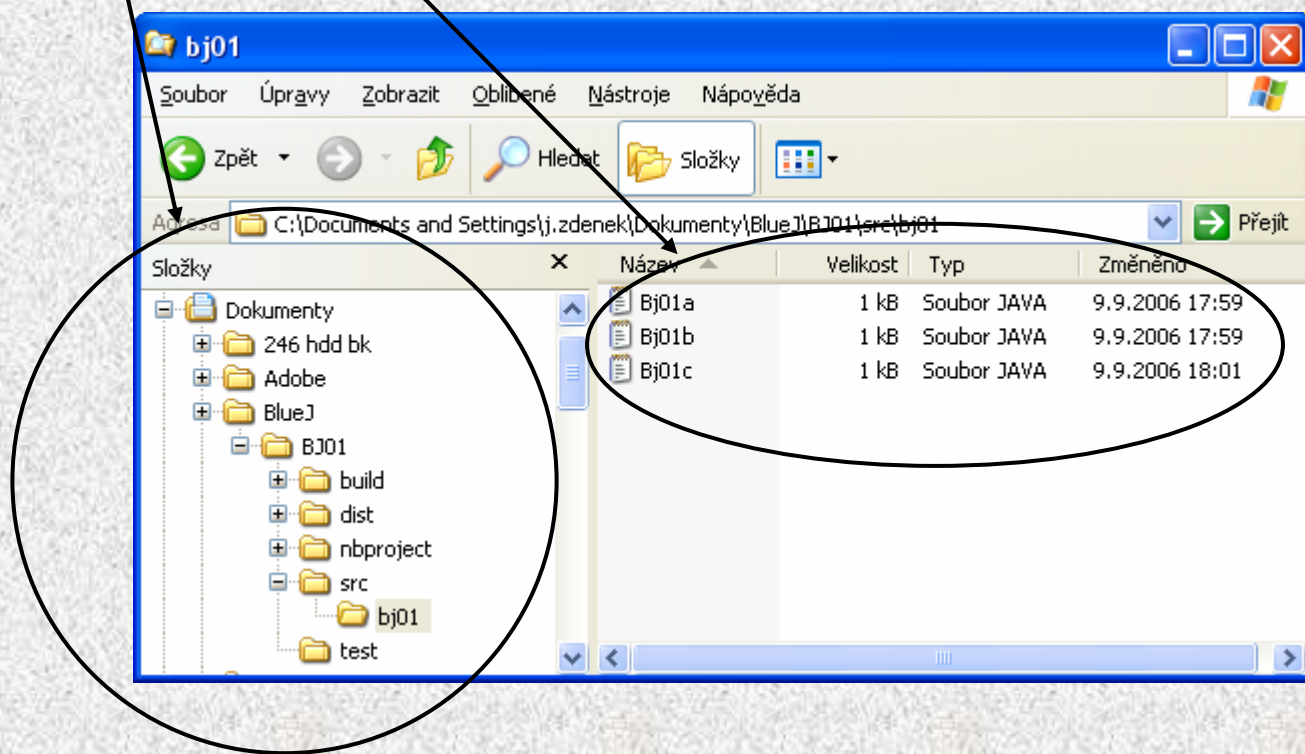
Běžný adresář (Current Directory) a cesta

- Běžný adresář - aktivní (pracovní) adresář
- Jméno souboru
 - Krátké (např. Bj01a.java),
 - Úplné (Úplná cesta)
- Úplná cesta – se skládá z:
 - jména media,
 - seznamu adresářů ležících na cestě k souboru,
 - jména souboru.



Soubory, adresáře a metody

- Pro práci se soubory poskytuje Java dvě skupiny tříd (metod) pro:
 - 1) Manipulaci s adresáři (založení, změna běžného, zrušení, volba media, ...)
 - 2) Práci se soubory (čtení a zápis dat, založení a zrušení souboru, ...)
- Třídy jsou uloženy v knihovně **java.io**



Práce se souborem

- Soubor je v běžném stavu chráněn proti změnám dat v souboru (je uzavřen)
- Před vlastní manipulací s daty v souboru (čtení/zápis) musíme soubor otevřít
- Při práci s daty v souboru jsou typické tyto operace:
 - 1) otevření souboru,
 - 2) čtení nebo zápis údajů,
 - 3) uzavření souboru.
- Přístup k údajům v souboru (čtení nebo zápis) může být:
 - postupný (sekvenční)
 - přímý (nahodilý)
- Soubory se sekvenčním přístupem (sekvenční soubory) umožňují pouze postupné (sekvenční) čtení/zápis údajů
- Soubory s přímým přístupem umožňují nahodilé čtení/zápis údaje
- Způsob přístupu k údajům v souboru není zakódován v souboru, ale je dán programem a technickou podporou

Postupný (sekvenční) přístup k datům v souboru

- Otevření souboru pro čtení nastaví souborové ukazovátko na začátek souboru
- Příkazem pro čtení souboru se přečte položka na kterou ukazuje souborové ukazovátko a to se pak automaticky posune na další položku
- Sekvenční čtení může probíhat až do dosažení konce souboru

Otevři soubor

Čti položku



Čti položku



.....

.....

.....

.....

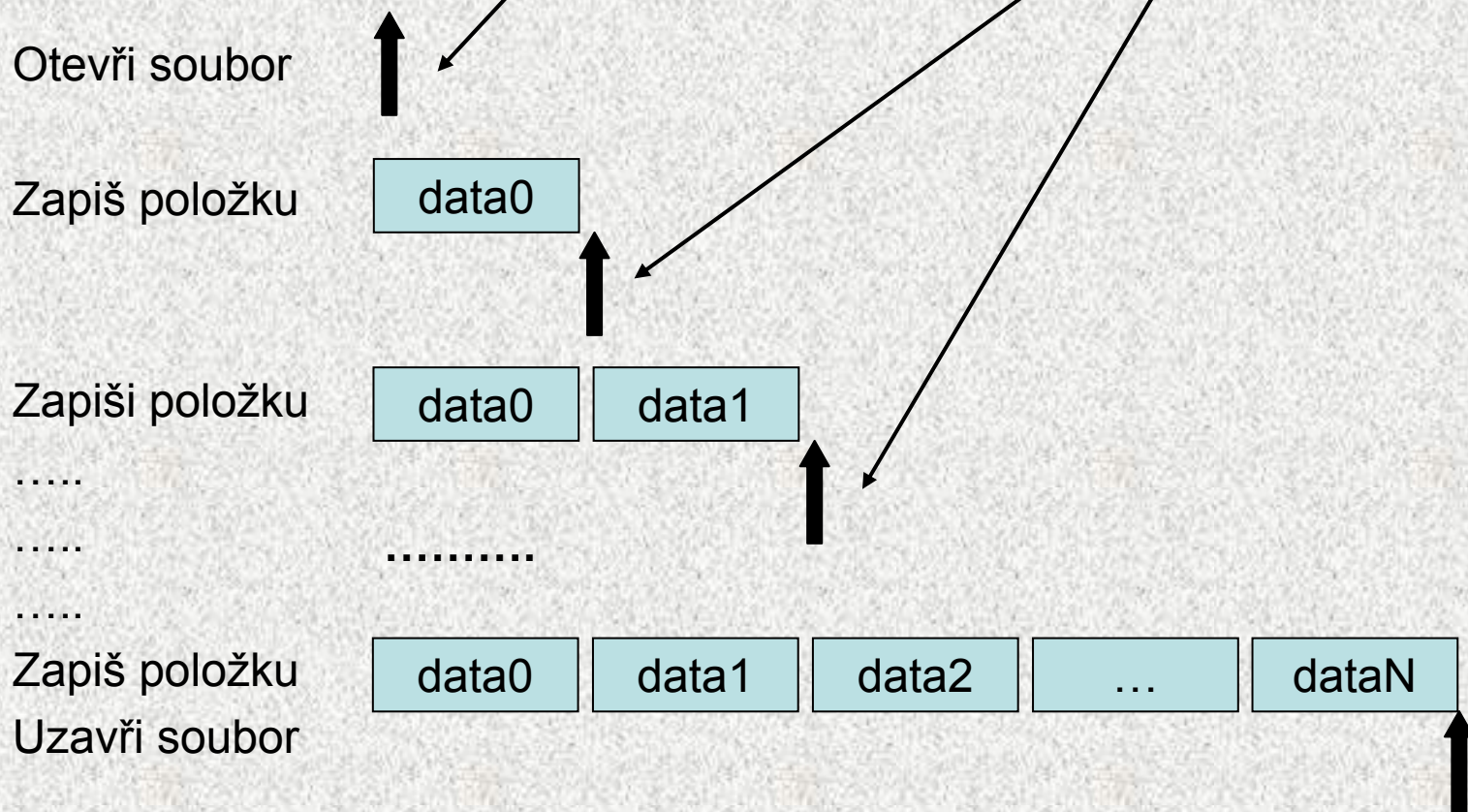
Čti položku



Uzavři soubor

Postupný (sekvenční) přístup k datům v souboru

- Otevření souboru pro zápis nastaví souborové ukazovátko na začátek souboru
- Příkazem pro zápis se запиše položka a souborové ukazovátko se pak automaticky posune na další prázdné místo
- Sekvenční zápis může probíhat až do vyčerpání kapacity media



Přímý (nahodilý) přístup k datům v souboru

- Po otevření souboru se před čtením každé položky dat příkazem nastaví souborové ukazovátko na hledanou položku
- Příkazem pro čtení souboru se položka přečte
- Položky se uvedeným způsobem mohou číst v libovolném pořadí

Otevři soubor

Nastav ukazovátko

Čti položku



Nastav ukazovátko

Čti položku



.....

.....

Nastav ukazovátko

Čti položku



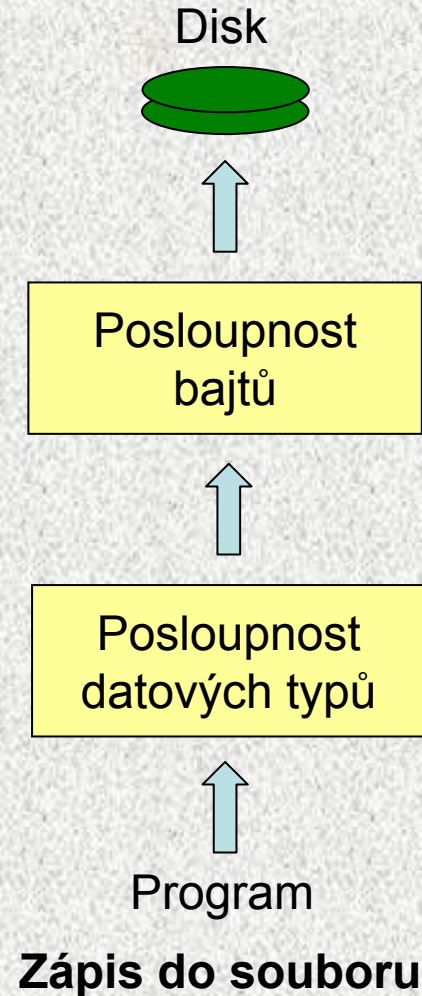
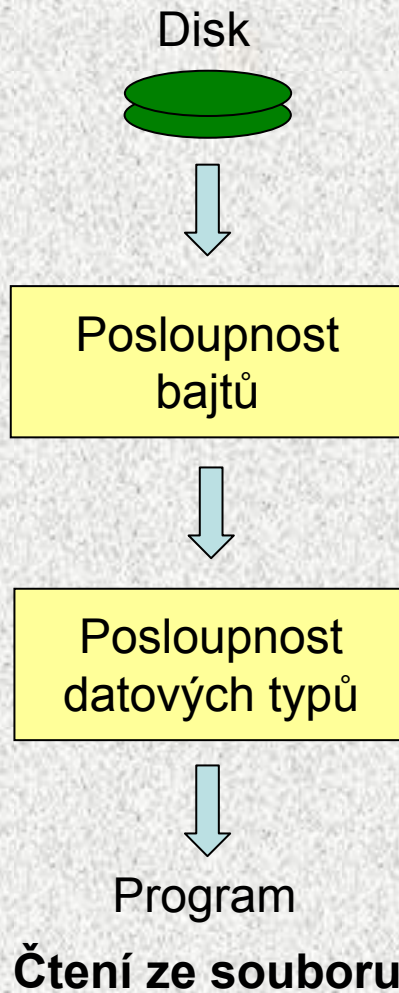
Uzavři soubor

Typy souborů

- Podle způsobu kódování informace v souboru známe:
 - Soubory textové
 - Soubory binární
- Textový soubor je posloupnost znaků členěná na řádky
 - každý znak je reprezentován jedním bytem
 - členění na řádky je závislé na platformě a obvykle je dáno jedním nebo dvěma řídicími znaky (*CR*, *CR LF*, 0x0d 0x0c, “\r\n”)
 - Textový soubor je čitelný textovým editorem
- Binární soubor je posloupnost byteů a informace je kódována vnitřním kódem počítače. Do binárního souboru mohou být zapsány:
 - byte, jednoduché proměnné, pole, data celých objektů, ...
- Informace o typu souboru (textový, binární) ani o způsobu kódování informace není v souboru obsažena. Správnou interpretaci přečteného souboru musí zajistit uživatelský program

Přenos informace – dvě vrstvy

- Přenos informace z/do souboru je rozdělen do vrstev



Výjimky (Exceptions)

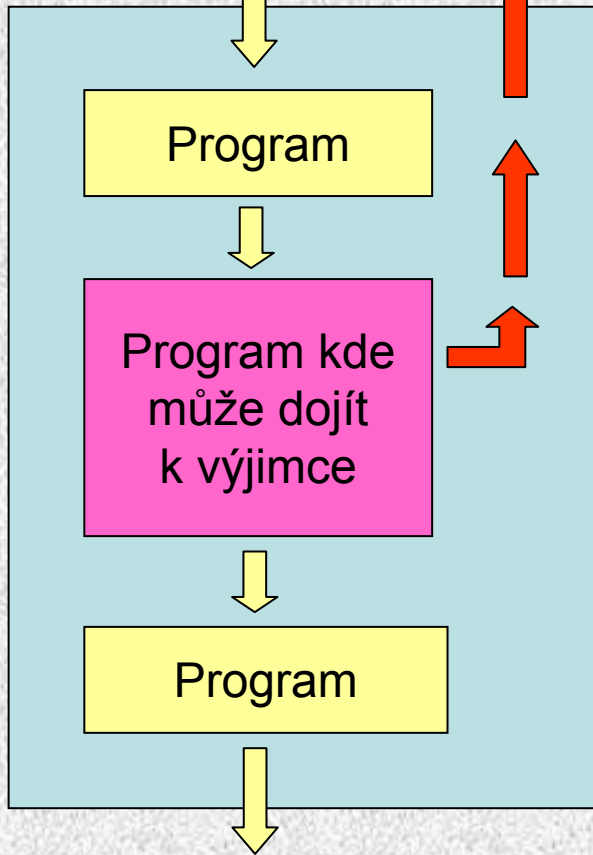
- Při operacích se soubory (i při jiných akcích) mohou nastat chyby, které program sám neumí zvládnout pokud nemá předem připravené řešení (alternativu) programátorem.
- Těmto chybám se říká „Běhové chyby“ (Run-time errors) a při překladu programu je nelze odhalit (nejsou to syntaktické chyby) ani předem stanovit zda vzniknou.
- Příkladem takové chyby je pokus o čtení z neexistujícího souboru nebo pokus o zápis na zaplněný disk.
- Chyba nemusí znamenat ukončení programu – chybu lze ošetřit a program může pokračovat dál.
- K ošetření chyb v Javě je určen mechanismus výjimek (exceptions)
- Java ošetření výjimek vynucuje (je to bezpečnostní vlastnost), bez zápisu v programu, jak se bude výjimka řešit, nelze program v Javě přeložit.

Výjimky (Exceptions)

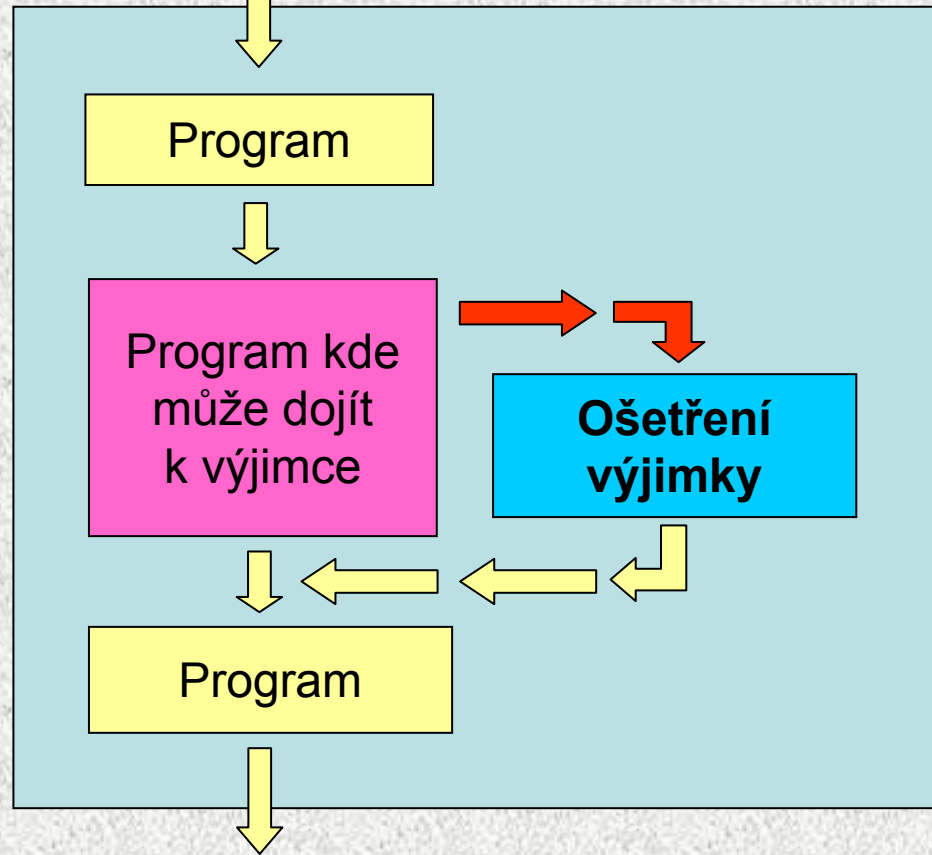
- Princip řešení výjimek v jazyku Java:
 - metoda či konstruktor může skončit standardně (tj. bez chyby) nebo nestandardně tzv. vyhozením výjimky určitého typu v případě, že nastala chyba. Výjimka je objekt.
 - pokud příkaz skončil vyhozením výjimky, další příkazy se neprovedou a řízení se předá konstrukci ošetřující výjimku daného typu (s touto konstrukcí se seznámíme později)
 - pokud taková konstrukce v těle metody či konstruktoru není, skončí nestandardně a výjimka se šíří na nadřazenou úroveň
 - není-li výjimka ošetřena ani ve funkci *main()*, vypíše se a program skončí
 - pro rozlišení různých typů výjimek je v jazyku Java zavedena řada knihovných tříd, výjimky jsou instancemi (objekty) těchto tříd

Výjimky – vyhození, ošetření

metoda **Vyhození výjimky**



metoda

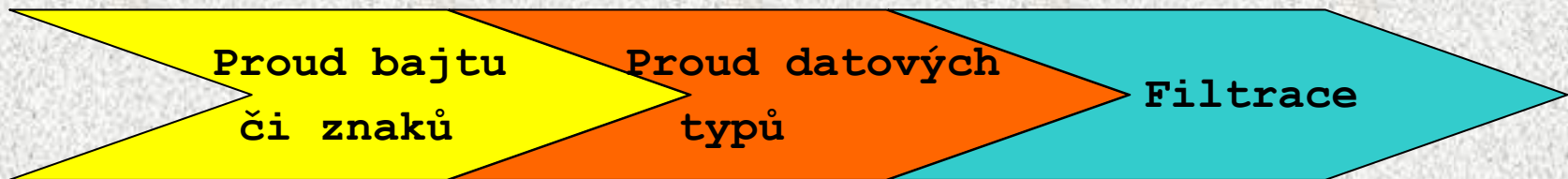


Třídy pro práci se soubory v Javě

- Třída (viz knihovna `java.io.*`)
 - Použití
 - Adresáře a vlastnosti souborů
 - Binární soubory, postupný přístup
 - `FileOutputStream` Čtení bajtů
 - `FileInputStream` Zápis bajtů
 - `DataOutputStream` Čtení jednoduchých typů
 - `DataInputStream` Zápis jednoduchých typů
 - `ObjectOutputStream` Zápis dat objektů
 - `ObjectInputStream` Čtení dat objektů
 - Textové soubory, sekvenční přístup
 - `FileWriter` Zápis do textového souboru
 - `FileReader` Čtení textového souboru
 - `BufferedReader` Čtení textového souboru
 - Binární soubory, přímý (nahodilý) přístup
 - `RandomAccessFile`
- Při práci se souborem se nepoužívá jeho fyzické jméno (tj. jméno které zobrazuje souborový manažer – např. “Průzkumník”. Fyzickému jménu se přiřadí jméno (objekt) logické a s tím se dále pracuje viz třída `File`.

Soubory a proudy

- Java rozlišuje soubory (file) a proudy (stream)
 - **soubory** je množina údajů uložená ve vnější paměti počítače
 - **proudy** jsou nástroje k přenosu informací např. z/do souboru, ale také do/ze sítě, paměti, jiného programu, atd.
- informace může mít tvar znaků, bajtů, skupin bajtů (obrázky...), objektů,...
- přenos informace se děje dvoj/třívrstevně v proudech (**streams**):
 1. **otevření** přenosového proudu pro **bajty** či **znaky**
 2. **otevření** přenosového proudu pro **datové typy Javy**
 3. **filtrace** dat podle požadavků – bufferování, řádkování, ...



1. Soubor jako **posloupnost bytů**

- V jazyku Java slouží pro práci se soubory třídy definované v balíku *java.io*
- Soubor jako **posloupnost bytů** reprezentují třídy:
 - ***FileInputStream*** vstupní soubor (bude se z něj číst)
 - ***FileOutputStream*** výstupní soubor (bude se do něj zapisovat)
- Příklad: kopie souboru

```
import java.io.*;

public class Kopie1 {

    public static void main(String[] args) throws IOException {
        FileInputStream in = new FileInputStream("vstup.txt");
        FileOutputStream out = new FileOutputStream("vystup.txt");
        int b = in.read();
        while (b != -1) {
            out.write(b);
            b = in.read();
        }
        out.close();
        in.close();
    }
}
```

**Vstupní
soubor**

**Výstupní
soubor**

**Proud bajtu
či znaků**

Soubor jako posloupnost bytů

- Komentář k příkazům
 - ► `FileInputStream in = new FileInputStream("vstup.txt");`
 - vytvořený objekt `in` reprezentuje vstupní soubor uložený na disku v aktuálním adresáři pod názvem `vstup.txt`; pokud takový soubor neexistuje, nastane chyba
 - `FileOutputStream out = new FileOutputStream("vystup.txt");`
 - vytvořený objekt `out` reprezentuje výstupní soubor, který bude uložen do aktuálního adresáře pod názvem `vystup.txt`; pokud by soubor nebylo možné vytvořit, nastane chyba

Metody:

- ► `b = in.read();`
 - ze souboru `in` se přečte jeden byte (číslo v rozsahu 0..255) a uloží do `b`; není-li v souboru žádný nepřečtený byte, výsledkem metody je `-1`
- ► `out.write(b);`
 - do souboru `out` se zapíše jeden byte s hodnotou `b`
- ► `out.close();`
- ► `in.close();`
 - uzavření souborů `in` a `out`

2. Soubor jako **posloupnost primitivních typů**

- Příklad: program, který vytvoří soubor obsahující 100 náhodných čísel typu *double* a pak soubor přečte a vypíše součet čísel

```
public class Cisla {  
    public static void main(String[] args) throws Exception {  
        DataOutputStream out = new DataOutputStream(  
            new FileOutputStream("temp.bin"));  
  
        for (int i=0; i<100; i++)  
            out.writeDouble(Math.random());  
  
        out.close();  
  
        DataInputStream in = new DataInputStream(  
            new FileInputStream("temp.bin"));  
  
        double soucet = 0;  
        while (in.available()>0)  
            soucet = soucet + in.readDouble();  
        System.out.print(soucet);  
    }  
}
```

Zavření
souboru

Výstupní
soubor

Vstupní
soubor

Proud bajtu
či znaků

Proud datových
typů

Užitečná informace

3. Soubor **primitivních typů a objektů**

- Soubor obsahující jak primitivní typy tak objekty reprezentují třídy `ObjectOutputStream` a `ObjectInputStream`
- Příklad: program, který vytvoří soubor obsahující dvě hodnoty typu `int` a dva řetězce, a pak soubor přečte a vypíše

```
public class CislaRetezce {  
    public static void main(String[] args) throws Exception {  
        ObjectOutputStream out = new ObjectOutputStream(  
            new FileOutputStream("temp.bin"));  
  
        out.writeInt(1);  
        out.writeInt(2);  
        out.writeObject("prvni retezce");  
        out.writeObject("druhy retezce");  
        out.close();  
  
        ObjectInputStream in = new ObjectInputStream(  
            new FileInputStream("temp.bin"));  
  
        System.out.print(in.readInt()+" "+in.readInt());  
        String s1 = (String)in.readObject();  
        String s2 = (String)in.readObject();  
  
        System.out.print (s1+" "+s2);  
    }  
}
```

Zavření souboru

Výstupní soubor

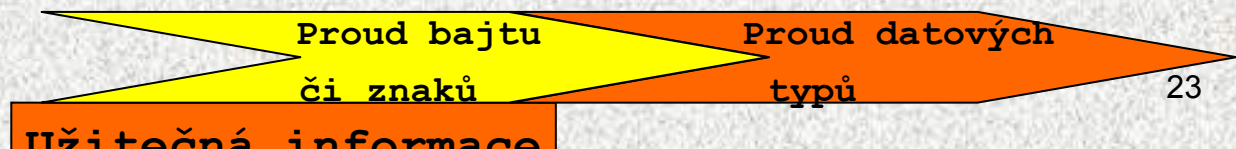
Vstupní soubor

**Proud bajtu
či znaků**

**Proud datových
typů**

Operace se souborem primitivních typů a objektů

Uvedenými metodami lze zapisovat a číst pouze tzv. **serializovatelné objekty** (patří mezi ně implicitně např. řetězce a pole primitivních typů), jinak je třeba „serializovat“ (**implementovat rozhraní `Serializable` !**)



4. Ukládání objektů do souboru

```
class ProObjekt implements Serializable {
```

```
    int i;
```

```
    String jmeno;
```

```
    int telefon;
```

```
    boolean pohlavi;
```

```
    double vaha;
```

```
    public ProObjekt(int j) {
```

```
        i = j;
```

```
        jmeno = "JMENO-" + j;
```

```
        telefon = 111 + j;
```

```
        pohlavi = i%2==0;
```

```
        vaha = 25 - i;
```

```
    }
```

```
    public String toString() {
```

```
        return (i+"\t"+ jmeno+"\t"+ telefon+"\t"+
```

```
            vaha); }
```

Datové
složky

Konstruktor

Jak zobrazit
objekt

Ukládání objektů do souboru

```
public static void main(String[] args) throws IOException,  
                                         ClassNotFoundException {  
    FileOutputStream fwJm = new FileOutputStream("objekty.bin");  
    ObjectOutputStream fw = new ObjectOutputStream(fwJm);  
    ProObjekt[] poleObjektu = new ProObjekt[3];  
    for (int i = 0; i < poleObjektu.length; i++) {  
        poleObjektu[i] = new ProObjekt(i);  
    }  
    System.out.println("Ulozeni");  
    for (int i = 0; i < poleObjektu.length; i++) {  
        System.out.println(poleObjektu[i]);  
    }  
    for (int i = 0; i < poleObjektu.length; i++) {  
        fw.writeObject(poleObjektu[i]);  
    }  
    fwJm.close();  
    for (int i = 0; i < poleObjektu.length; i++) {  
        poleObjektu[i] = null;}  
}
```

Výstupní
soubor

Zavření
souboru

Ukládání objektů do souboru

```
FileInputStream frJm = new FileInputStream("objekty.bin");
ObjectInputStream fr = new ObjectInputStream(frJm);
for (int i = 0; i < poleObjektu.length; i++) {
    poleObjektu[i] = (ProObjekt) fr.readObject();
}
fwJm.close();
System.out.println("Cteni");
for (int i = 0; i < poleObjektu.length; i++) {
    System.out.println(poleObjektu[i]);
}
}
```

Vstupní
soubor

Ulozeni

0	JMENO-0	111	true	25.0
1	JMENO-1	112	false	24.0
2	JMENO-2	113	true	23.0

Cteni

0	JMENO-0	111	true	25.0
1	JMENO-1	112	false	24.0
2	JMENO-2	113	true	23.0

Příklady práce se soubory

- Zápis a čtení jednoduchých typů – int
- Zápis jednoduchých typů – double
- Čtení jednoduchých typů – double
- Zápis objektů – string
- Čtení objektů – string
- Zápis objektů – pole – vyhození výjimky
- Čtení objektu – vlastní třída
- Zápis textového souboru
- Čtení textového souboru
- Rozšíření textového souboru

Zápis a čtení jednoduchých typů - int

```
import java.io.*;
public class TestInt1 {
    public static void main(String[] args){
        String jmenoSouboru = "test1.txt";
        File f = new File(jmenoSouboru);
        FileOutputStream fos = new FileOutputStream(f);
        DataOutputStream dos = new DataOutputStream(fos);
        for(int i = 0; i < 10; i++){
            dos.writeInt(i);          // Zapis do souboru
        }
        dos.close();

        FileInputStream fis = new FileInputStream(f);
        DataInputStream dis = new DataInputStream(new FileInputStream(f));
        int x;
        while(dis.available() > 0){
            x = dis.readInt();        // Cteni ze souboru
            System.out.println("x = "+x);
        }
        dos.close();
    } // main() END
} // class TestInt1 END
```


Zápis jednoduchých typů - double

```
import java.io.*;
public class TestDouble1 {

    public static void main(String[] args){
        File g = new File("test2.txt");
        try{                // Osetreni vyjimky
            FileOutputStream fos1 = new FileOutputStream(g);
            DataOutputStream dos1 = new DataOutputStream(fos1);
            for(int i = 0; i < 10; i++){
                dos1.writeDouble(i+0.2);
            }
            dos1.close();
        }// try{} END
        catch(IOException e){
            System.out.println("Chyba pri zapisu do souboru");
        }// main() END

    }// class TestDouble1 END
```

Čtení jednoduchých typů - double

```
import java.io.*;
public class TestDouble2 {

    public static void main(String[] args){
        File f = new File("test2.txt");
        try{
            FileInputStream fis = new FileInputStream(f)
            DataInputStream dis = new DataInputStream(fis);
            double d;
            while(dis.available() > 0){
                d = dis.readDouble();
                System.out.println("d="+d);
            }
            dis.close();
        }
        catch(IOException e){
            System.out.println("Chyba pri cteni ze souboru");
        }
    } // main() END

} // class TestDouble1 END
```

Zápis objektů - string

```
import java.io.*;
public class TestString3 {

    public static void main(String[] args)
        throws IOException, ClassNotFoundException {
        String[] s = {"Jana","Karel","Petr","Helena","Agata"};
        String jmenoSouboru = "test3.txt";
        zapisPoleString(jmenoSouboru,s);
    }// main() END

    static void zapisPoleString(String jmenoSouboru, String[] s)
        throws IOException{
        FileOutputStream fos = new FileOutputStream(new File(jmenoSouboru));
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        for(int i = 0; i < s.length; i++){
            oos.writeObject(s[i]);
        }
        oos.close();
    }// zapisPoleString() END

}// class TestString3 END
```

Čtení objektů - string

```
import java.io.*;
public class TestString4 {
    public static void main(String[] args)
        throws IOException, ClassNotFoundException {
        String[] r = cteniPoleStringu("test3.txt");
        for(int i = 0; i < r.length; i++){
            System.out.println(r[i]);
        }
    } // main() END

    static String[] cteniPoleStringu(String jmenoSouboru)
        throws IOException, ClassNotFoundException{
        FileInputStream fis = new FileInputStream(new File(jmenoSouboru));
        ObjectInputStream ois = new ObjectInputStream(fis);
        String[] s = new String[10];
        int i;
        for(i = 0; fis.available() > 0 && i < delka; i++){
            s[i] = (String)ois.readObject();
        }
        ois.close();
        String[] r = new String[i];
        System.arraycopy(s,0,r,0,i);
        return(r);
    } // cteniPoleStringu() END
} // class TestString4 END
```


Zápis objektů – pole – vyhození výjimky

```
import java.io.*;
public class TestPole5 {

    public static void main(String[] args)
        throws IOException, ClassNotFoundException {
        String[] s = {"Jana", "Karel", "Petr", "Helena", "Agata"};
        String jmenoSouboru = "test5.xxx";
        zapisPoleStringu(jmenoSouboru, s);
    } // main() END

    static void zapisPoleStringu(String jmenoSouboru, String[] s)
        throws IOException{
        FileOutputStream fos = new FileOutputStream(new File(jmenoSouboru));
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(s);
        oos.close();
    } // zapisPoleString() END

} // class TestPole5 END
```

Čtení objektů – pole – vyhození výjimky

```
import java.io.*;
public class TestPole6 {

    public static void main(String[] args)
        throws IOException, ClassNotFoundException {
        String jmenoSouboru = "test5.xxx";
        String[] r = cteniPoleStringu(jmenoSouboru);
        for(int i = 0; i < r.length; i++){
            System.out.println(r[i]);
        }
    } // main() END

    static String[] cteniPoleStringu(String jmenoSouboru)
        throws IOException , ClassNotFoundException{
        FileInputStream fis = new FileInputStream(new File(jmenoSouboru));
        ObjectInputStream ois = new ObjectInputStream(fis);
        String[] s =(String[])ois.readObject();// Pretypovani
        ois.close();
        return(s);
    } // cteniPoleStringu() END

} // class TestPole6 END
```

Zápis objektů – pole – zachycení výjimky

```
import java.io.*;
public class TestPole7 {

    public static void main(String[] args){
        String[] s = {"Jana", "Karel", "Petr", "Helena", "Agata", "Imre"};
        String jmenoSouboru = "test7.xxx";
        if(zapisPoleStringu(jmenoSouboru, s) == true) {
            System.out.println("Chyba pri zapisu do souboru");
        }
    } // main() END

    static boolean zapisPoleStringu(String jmenoSouboru, String[] s) {
        boolean error = false;
        try{
            FileOutputStream fos = new FileOutputStream(new File(jmenoSouboru));
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            oos.writeObject(s);
            oos.close();
        }
        catch(IOException e){
            error = true;
        }
        return error;
    } // zapisPoleString() END
} // class TestPole7 END
```

Čtení objektů – pole – zachycení výjimky

```
import java.io.*;
public class TestPole8 {

    public static void main(String[] args){
        String[] r;
        if ((r = cteniPoleStringu("test7.xxx"))== null) {
            System.out.println("Chyba pri ctení souboru");
        } else {
            for (int i = 0; i < r.length; i++) {
                System.out.println(r[i]);
            }
        }
    }
} // main() END

static String[] cteniPoleStringu(String jmenoSouboru){
    String[] s = null;
    try{
        FileInputStream fis = new FileInputStream(new File(jmenoSouboru));
        ObjectInputStream ois = new ObjectInputStream(fis);
        s = (String[]) ois.readObject();// Pretypovani
        ois.close();
    }
    catch(Exception e){;}// null nastaven při deklaraci
    return (s);
} // cteniPoleStringu() END
} // class TestPole8 END
```


Zápis objektu – vlastní třída

```
class Osoba implements Serializable{

    static int pocetObjektu=0;
    String jmeno;
    String telefon;

    public Osoba(String jmeno, String telefon){
        this.pocetObjektu++;
        this.jmeno=jmeno;
        this.telefon=telefon;
    }

    String getJmeno(){
        return(this.jmeno);
    }

    String getTelefon(){
        return(this.telefon);
    }

} // Osoba class
```

Zápis objektu – vlastní třída

```
import java.io.*;
public class TestOsoba9 {

    public static void main(String[] args)
        throws IOException, ClassNotFoundException {

        Osoba o1 = new Osoba("Novak", "1234");
        Osoba o2 = new Osoba("Dvorak", "4567");
        Osoba o3 = new Osoba("Horak", "9999");
        File f = new File("Osoby.xxx");
        FileOutputStream fos = new FileOutputStream(f);
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        oos.writeObject(o1);
        oos.writeObject(o2);
        oos.writeObject(o3);
        oos.close();
    } // main() END

} // class TestOsoba9 END
```

Čtení objektu – vlastní třída

```
import java.io.*;
public class TestOsoba10 {

    public static void main(String[] args){
        Osoba.pocetObjektu = 0;
        Osoba[] osoby = new Osoba[5];
        File g = File("Osoby.xxx");
        FileInputStream fis = new FileInputStream(g);
        ObjectInputStream ois = new ObjectInputStream(fis);
        try {
            for (int i = 0; i < osoby.length; i++) {
                osoby[i] = (Osoba)ois.readObject();
                // POZOR promenne tridy se ze souboru neobnovuji
                Osoba.pocetObjektu = i + 1;
            }
        }
        catch (EOFException e) {
            // EOF - konec souboru
        }
        finally {
            ois.close();
        } // try/catch/finally END
    } // main END
} // class TestOsoba10 END
```

Zápis textového souboru

```
import java.io.*;

public class TestText11 {

    public static void main(String[] args){
        File f = new File("test11.txt");
        FileWriter fw = new FileWriter(f);
        BufferedWriter bw = new BufferedWriter(fw);
        for(int i=0;i<10;i++){
            bw.write(i+"\r\n");
            //fw.newLine(); // nebo misto "\r\n"
        }
        bw.close();
    } // main() END

} // class TestText10 END
```


Čtení textového souboru

```
import java.io.*;

public class TestText12 {

    public static void main(String[] args){
        FileReader fr = new FileReader("test11.txt");
        BufferedReader br = new BufferedReader(fr);
        int x;
        String s;
        // Cteni String a konverse string na int
        while((s=br.readLine()) != null){
            x = Integer.parseInt(s);
            System.out.println("Konverse String->int x=" + x);
        }
        br.close();
    } // main() END

} // class TestText12 END
```

Rozšíření textového souboru

```
import java.io.*;
public class TestText13 {

    public static void main(String[] args){
        File f = new File("test13.txt");
        FileWriter fw = new FileWriter(f);
        BufferedWriter bw = new BufferedWriter(fw);
        bw.write("Hugo");
        bw.newLine(); //Vlozeni znaku "Novy radek"
        bw.close();
        final boolean APPEND = true;
        FileWriter fw1 = new FileWriter(f,APPEND);
        BufferedWriter bw1 = new BufferedWriter(fw1);
        // Zapis (PRIDANI) dat do souboru (na puvodni konec souboru)
        bw1.write("Karel");
        bw1.newLine();
        bw1.write("Eva");
        bw1.close();
    } // main() END

} // TestText13 END
```

Soubory

Jazyk JAVA

Konec

