

A0B36PRI - PROGRAMOVÁNÍ

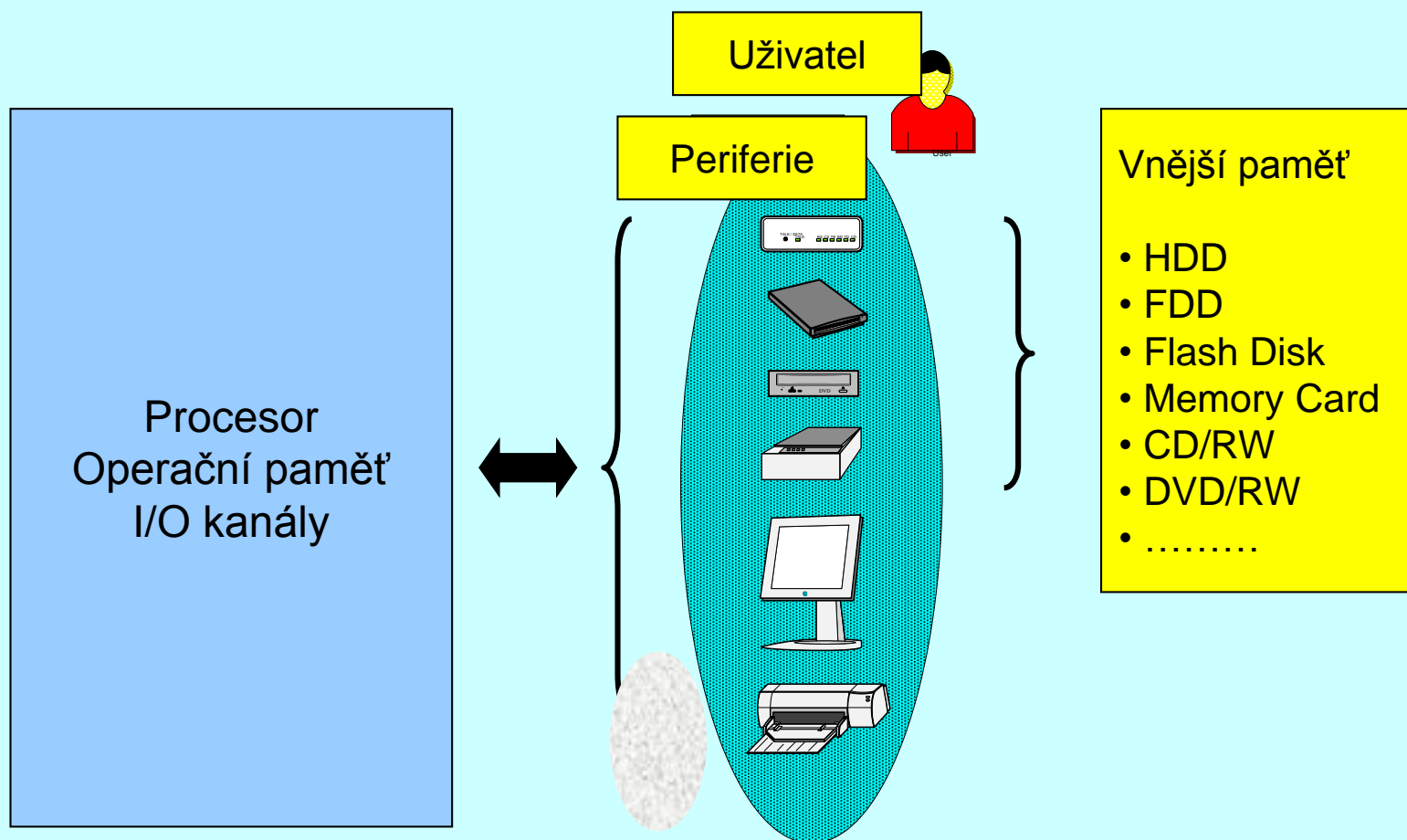
# Soubory, úvod

Jazyk JAVA



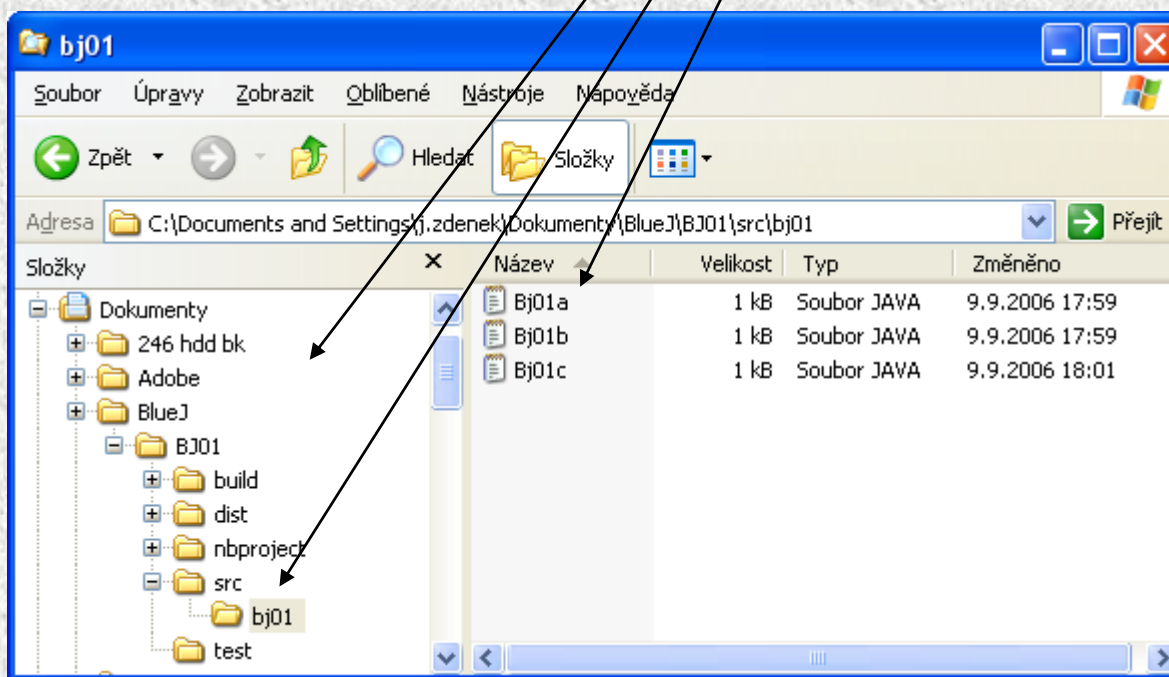
České vysoké učení technické Fakulta elektrotechnická

# Soubory a vnější paměť počítače



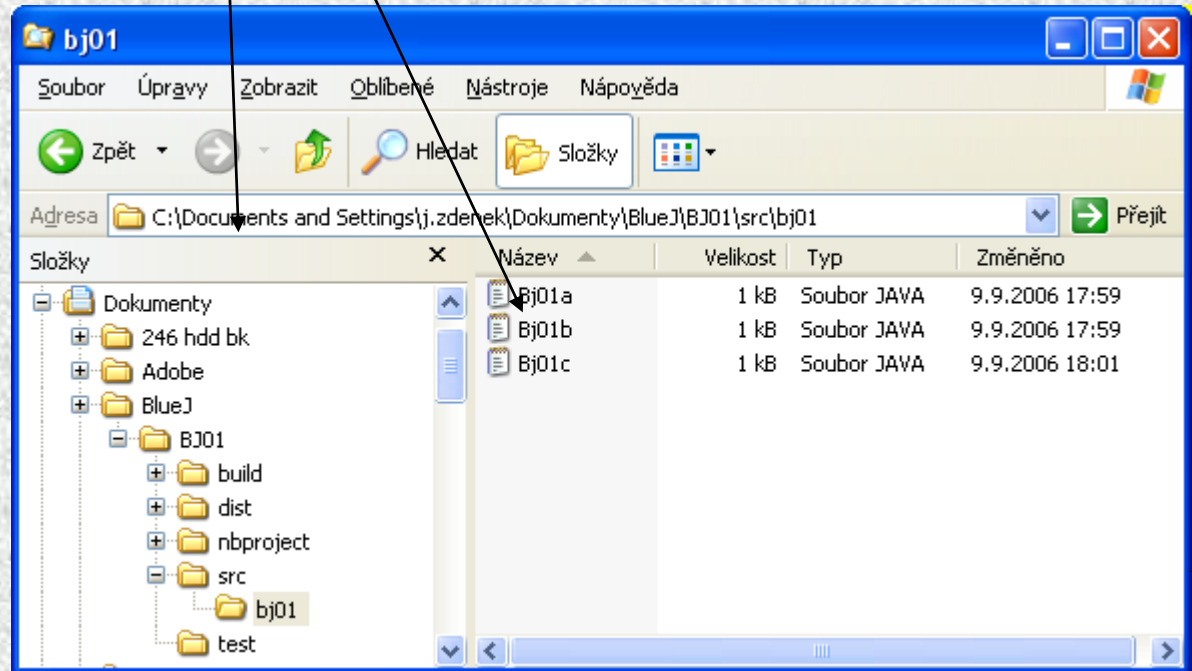
# Soubor a adresář (File, Directory)

- 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



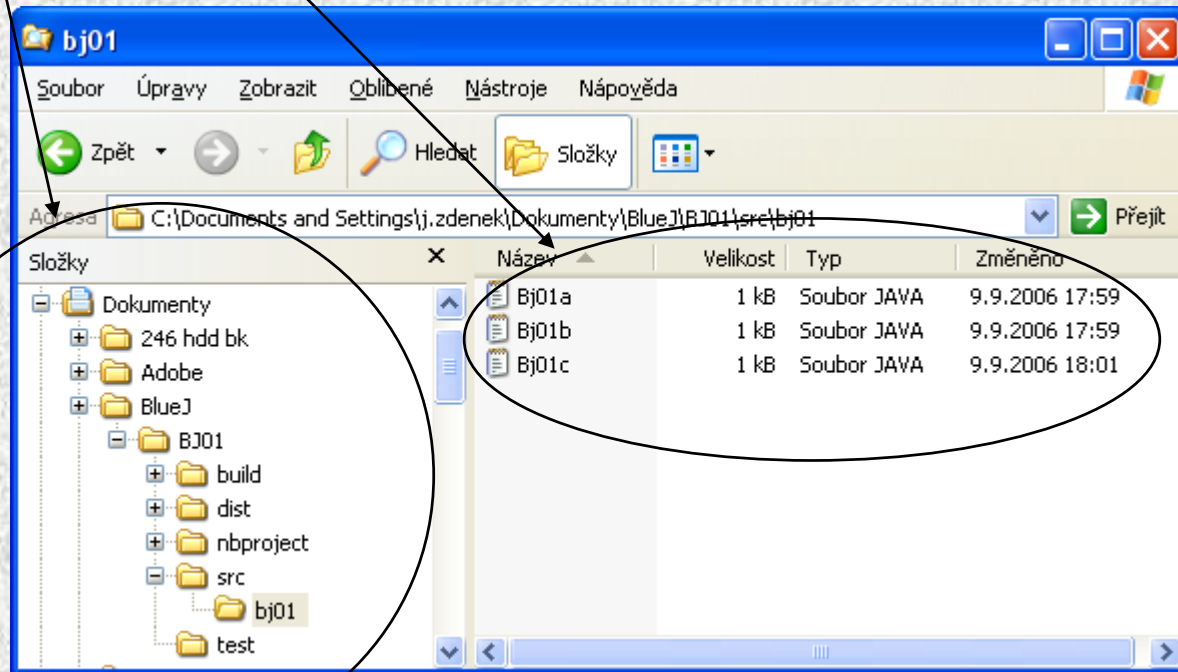
# Běžný adresář (Current Directory) a cesta

- 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

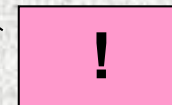
- Pro práci se soubory poskytuje Java **dvě skupiny tříd** (metod) pro:
  - 1) **Manipulaci s adresáři** (založení, změna běžného, zrušení, volba media, ...)
  - 2) **Práci se soubory** (čtení a zápis dat, založení a zrušení souboru, ...)
- Třídy jsou uloženy v knihovně **java.io**





# 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
- **Důležité: 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

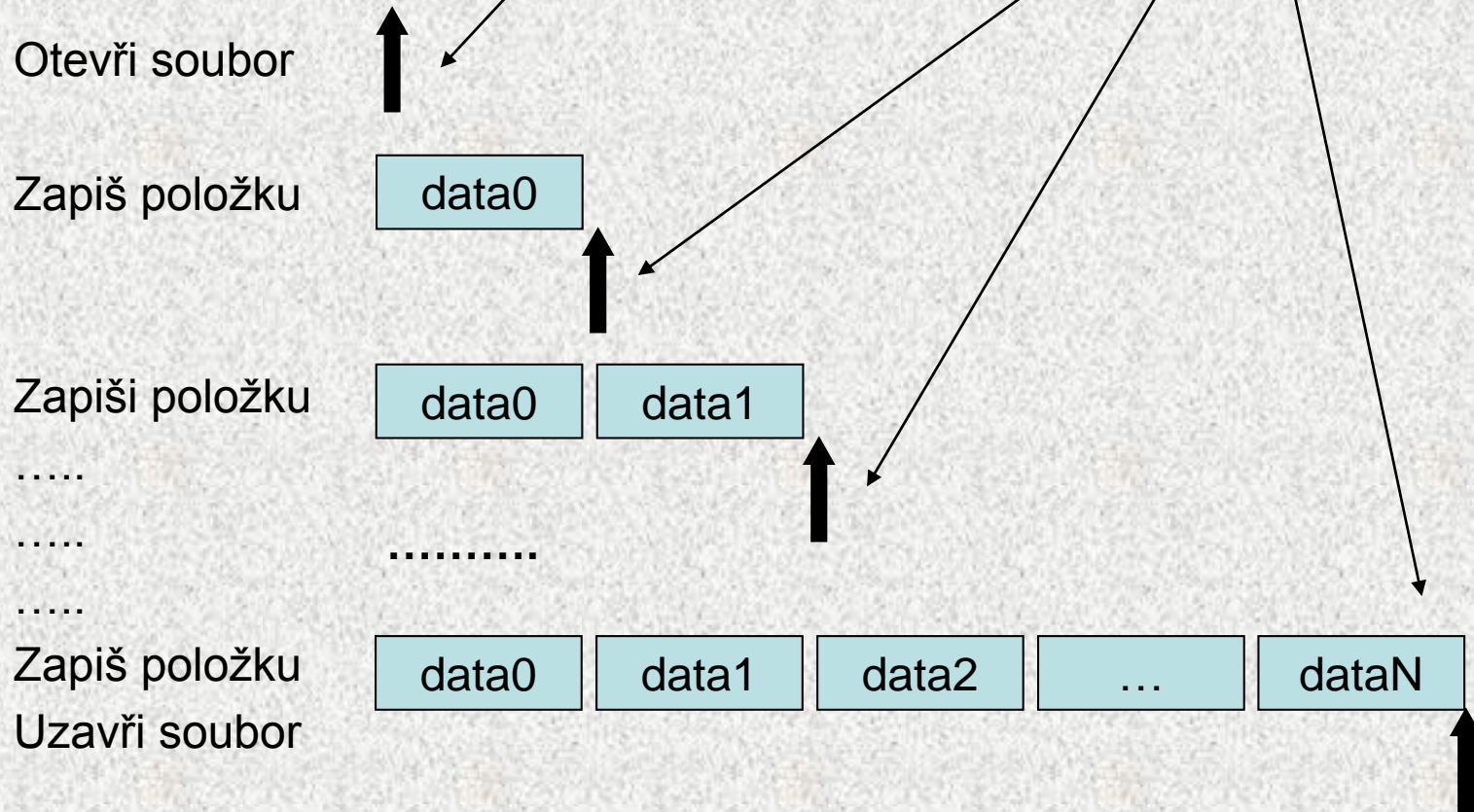


Uzavři soubor



# 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

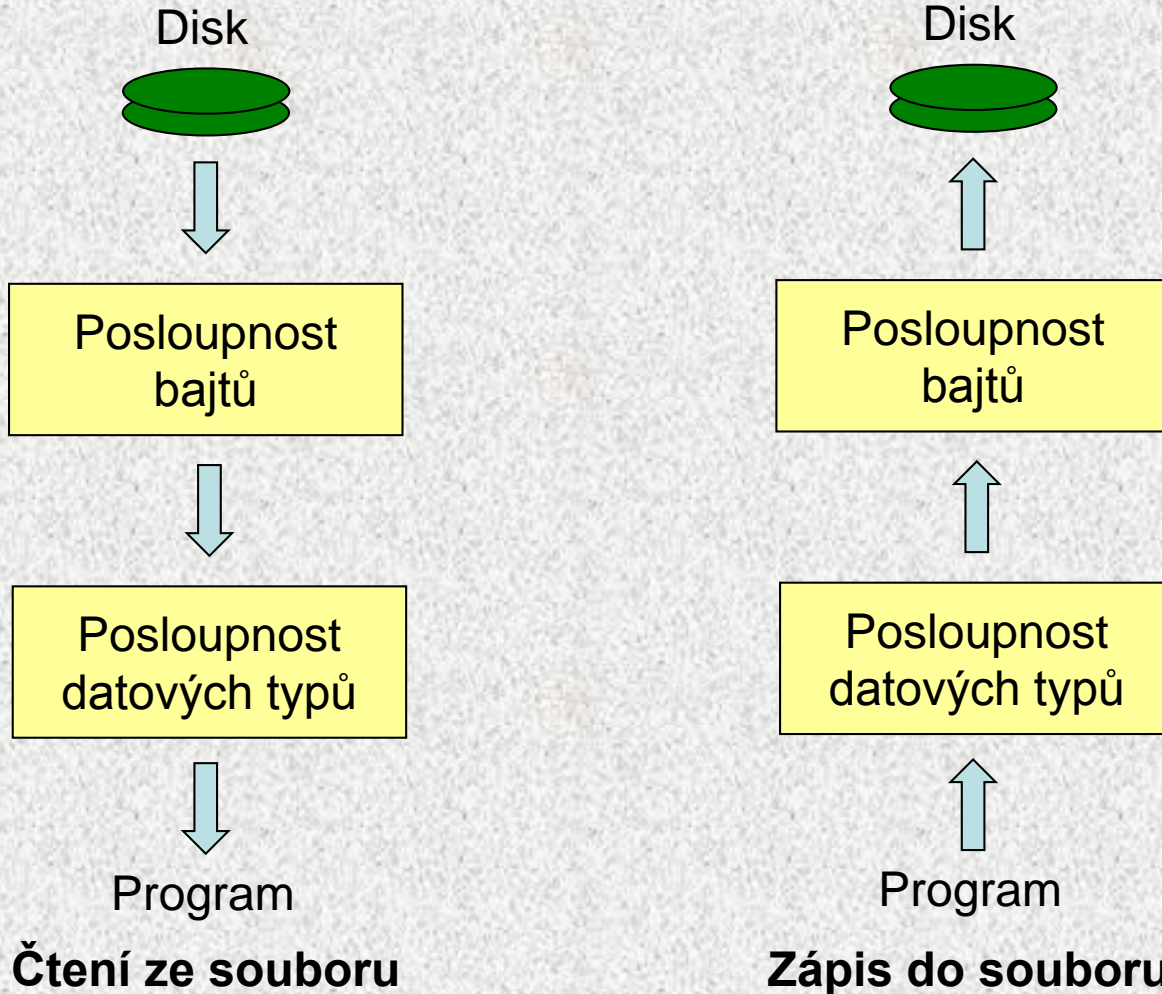


Uzavři soubor



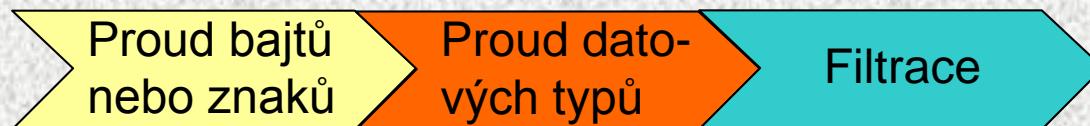
# Přenos informace v Javě– dvě vrstvy

- Přenos informace z/do souboru je rozdělen do vrstev



# 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, ...



# Soubor jako posloupnost jednoduchých 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);  
    }  
}
```

Ošetření  
chyb

Otevření  
souboru

Výstupní  
soubor

Zavření  
souboru

Vstupní  
soubor

Konec  
souboru?

Proud bajtů  
nebo znaků

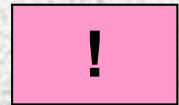
Proud dato-  
vých typů





# 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.





## Další možnosti souboru jednoduchých typů

- `out.writeDouble();`
- `out.writeInt();`
- `out.writeBoolean();`
- `out.writeChar();`
- `out.writeFloat();`
- .....

Zápis do  
souboru

- `in.readDouble();`
- `in.readInt();`
- `in.readBoolean();`
- `in.readChar();`
- `in.readFloat();`
- .....

Čtení ze  
souboru



# Zápis textového souboru

```
import java.io.*;
public class FileWriter1 {
    public static void main(String[] args) throws Exception{
        File f = new File("test1.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
}
```



# Čtení textového souboru

```
import java.io.*;
public class Konverse {
    public static void main(String[] args) throws Exception{
        FileReader fr = new FileReader("test1.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
}
```

Konec  
souboru?

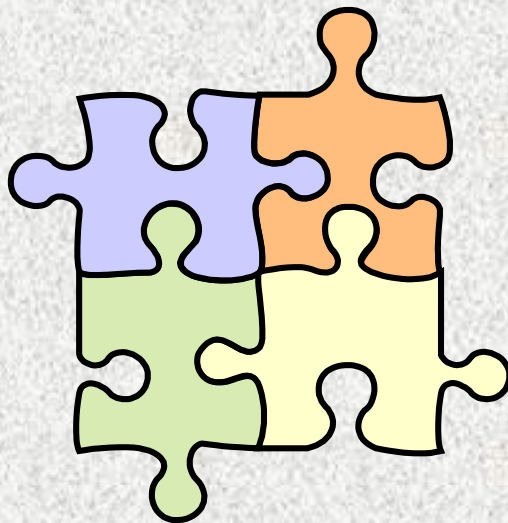


# Rozšíření textového souboru

```
import java.io.*;
public class BufferedWriter1 {
    public static void main(String[] args) throws Exception{
        File f = new File("test3.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 (za puvodni konec souboru)
        bw1.write("Karel");
        bw1.newLine();
        bw1.write("Eva");
        bw1.close();
    } // main() END
}
```



- Soubory – speciální přednáška č. 12
- Další informace k této přednášce hledejte např. v:  
Herout, P.: Učebnice jazyka Java, Kopp, Č.Budějovice, 2010,  
str.114 – 128, 305 – 309.





A0B36PRI - PROGRAMOVÁNÍ

# Předávání parametrů, soubory

Jazyk JAVA

## Konec



České vysoké učení technické Fakulta elektrotechnická