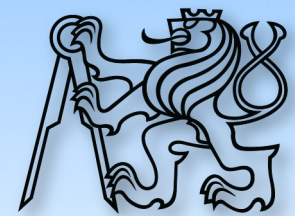


# Soubory a proudy

cvičení 24. března 2014



**Karel Čemus**  
cemuskar@fel.cvut.cz



# Aktivita

Vytvořte program, který ze souboru načte jména osob a následně je vypíše **abecedně seřazené podle příjmení**.

Jan Novak

Pepa Vomacka

Mirek Blazek

➤ **Nápověda:**

- Nemusíte řešit diakritiku
- Použijte `BufferedInputStream`
- `Scanner` pro snazší načítání dat
- Třída `Person`
- Vypisovat přes `Person#toString`
- Řadit pomocí `Collections#sort`
- Využijte třídu `Comparator<Person>`
- Termín: neděle 30. března 2014

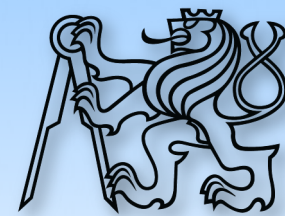


# Nápověda: Scanner

- Vytvoříme stream pro čtení ze souboru: `FileInputStream`
- Obalíme ho pokročilým streamem `BufferedInputStream`
- `Scanner` bude číst požadované datové typy ze streamu
- `Scanner#hasNext()` ověřuje, že stále ještě lze číst

```
// instance scanneru, který pracuje se streamem
Scanner s = new Scanner( new BufferedInputStream( new FileInputStream( „soubor“ ) ) );
// číst dokud je co
while ( s.hasNext() ) {
    System.out.println( s.next() );
}
```

# Nápověda: Comparator



- Java umí základní datové typy řadit správně
- Ostatní se řadí pomocí instancí rozhraní `Comparator<Typ>`
- `Comparator#compare` bere 2 argumenty a ty porovnává

```
// kolekce, kterou chceme řadit
java.util.List<String> strings = new ArrayList<String>();
// seřadit kolekci, dáváme tam instanci ANONYMNÍ třídy, která implementuje Comparator
Collections.sort( strings, new Comparator<String>() { // <String>: typ řazených objektů
    @Override
    public int compare( String o1, String o2 ) {
        return o1.compareTo( o2 ); // porovnání 2 instancí
    }
} );
```