

PB162 Programování v jazyce Java

12. a závěrečné cvičenie

rozhranie Iterator

- priechod kolekciou
- interne využívaný foreach cykom
- nutný, ak treba z kolekcie mazať (filter)

```
for (Iterator<Person> it = electorat.iterator();  
    it.hasNext();  
    ) {  
    Person person = it.next();  
    if (person.getSalary() > 25000) {  
        it.remove();  
    }  
}
```

rozhranie Comparator

- Comparable definuje **prirodzené** usporiadanie
- alternatívne usporiadania: `Comparator<T>`
- metóda `compare (T x, T y)`
 - $-1 \Leftrightarrow x < y$
 - $0 \Leftrightarrow x.\mathbf{equals} (y)$
 - $1 \Leftrightarrow x > y$
- `Collections.sort (personList, new PersonComparator ())`
- `SortedSet personSet = new TreeSet<Person> (new PersonComparator ());`

vstup a výstup

- `java.io`
- binárne dáta – prúdy (*streams*)
- znakové dáta – čitatelia/zapisovatelia (*readers/writers*)
- rozdiely?
- rôzne zdroje, formáty, sémantika dát
- riešenie pomocou modulárnej „matrioškovej“ architektúry

architektúra „matriošky“



znakové dáta dáta

- poskladať vhodný reader/writer
 - syntaktické skratky
 - kódovanie
- v cykle prečítať a parsovať/zapísať dáta
- zachytiť i/o a vyhodiť doménové výnimky
- **finally**
 - zatvorenie zdrojov (suboru...)

```
BufferedWriter bw = null;  
try {  
    bw = new BufferedWriter(  
        new OutputStreamWriter(  
            new FileOutputStream(  
                new File(„file.txt“)), "UTF-8"));  
    for (object o: objects) {  
        bw.append(o.get...);  
        bw.newLine();  
        ...  
    }  
} catch (UnsupportedEncodingException ex) {  
    throw new DomainException(  
        "unsupported encoding", ex);  
} catch (IOException ex) {  
    ...  
} finally {  
    try {  
        if (bw != null) {bw.close();}  
    } catch (IOException ex) {  
        throw new DomainException(  
            "output error while closing", ex);
```

```

BufferedReader br = null;
try {
    br = new BufferedReader(
        new InputStreamReader(..., "UTF-8"));
    String line = br.readLine();
    while (line != null) {
        String data[] = line.split("\\s+", 3);
        if (data.length < 6) {
            throw new DomainException(
                "not enoug data items on line " + line);
        }
        Racer racer = new RacerImpl(data[2]);
        ....
        line = br.readLine();
    }
} catch ...
} finally {
    try {
        if (br != null) br.close();
    } catch (IOException ex) {
        throw new DomainException(
            "i/o error while closing" + ex);
    }
}

```


konverzie dátových typov

- `String` \rightarrow `Integer` (**int**)
 - `Integer.valueOf(s)`
 - vyhadzuje `NumberFormatException`
- `int` \rightarrow `String`
 - `String.valueOf(i)`
- `Integer` \rightarrow **int** a naopak
 - vo väčšine prípadov automaticky:
autoboxing/unboxing

split a regulárne vyrazy

- `String.split (String regex[, int limit])`
 - slúži na „parovanie“ reťazca – vytvára `String[]`
 - `regex` – regulárny výraz
 - `limit` – max dĺžka poľa (delí sa od začiatku)
- regulárny výraz
 - `"a"` – jednoduchý znak (napr `":"`)
 - `"a|b"` – buď–alebo (napr `":|;"`)
 - `"a+"`, `"a*"` – aspoň jedno, ľub. počet opak.
 - `"\\s"`, `"\\S"` – biely, nebiely znak

zoznam (List)

- môže byť na skúške!
- precvičiť ešte pred opravkou/predtermínom
- nadviazať na príklad z cd playerom:
 1. `CompactDisk` premenovať na `ArrayCompactDisk`
 2. Vytvoriť rozhranie `CompactDisk`
 3. Vytvoriť implementáciu `ListCompactDisk`, v ktorej je pole nahradené zoznamom

kontajnery kontajnerov

- `Tagging`: implementácia tagov pre cdčka
- zadanie pošlem v priebehu sviatkov
- **oboje** (`ArrayCompactDisk` i `Tagging`)
 - otázky na diskusnom fóre skupiny 17
 - môžete vkladať kusy kódu
 - za správne odpovede bonusové body :-)

6. úloha

- domácu časť odovzdať do soboty!