

Iterator

- viz minulé cvičení
- ve skutečnosti je to generický typ

```
interface Iterable<T> {  
    Iterator<T> iterator();  
}
```

```
interface Iterator<T> {  
    boolean hasNext();  
    T next();  
    void remove();  
}
```

Čtení z textového souboru

- balíček java.io
 - skoro všechno vyhazuje výjimku IOException
- sekvenční čtení
- otevření pro čtení po znacích

```
FileReader fr = new FileReader(jmeno_souboru);
int ch;
while ((ch = fr.read()) != -1) {
    // dělej něco s načteným znakem
}
fr.close();
```

- čtení po řádcích

```
BufferedReader br = new BufferedReader(new
                                         FileReader(jmeno_souboru));

String line;
while ((line = br.readLine()) != null) {
    .....
}
br.close();
```

Standardní vstup

- System.in
- kopírování ze std. vstupu na std výstup

```
try (BufferedReader input = new BufferedReader(  
    new InputStreamReader(System.in))) {  
    int c;  
    while ((c = input.read()) != -1) {  
        System.out.print((char) c);  
    }  
} catch (IOException ex) {  
    System.out.println("Nastala IOException")  
}
```

Úkol 1

- napište třídu `MyString`, která bude použitelná jako modifikovatelný řetězec
 - bude mít alespoň následující metody a konstruktory
 - `MyString()`
 - `MyString(String str)`
 - `void append(String str)`
 - `void append(char ch)`
 - `void insert(int pos, String str)`
 - `void insert(int pos, char ch)`
 - `void delete(int pos, int length)`
 - korektně předefinovaná metoda `String toString()`
 - bude mít iterátor po jednotlivých znacích

Úkol 2

- napište program, který vypíše textový soubor po slovech, tj. každé slovo na nový řádek
 - oddělovače slov jsou bílé znaky
 - jméno souboru bude zadáno jako parametr programu při spuštění
- napište program, který vypíše textový soubor zalámaný na danou délku řádku
 - jméno souboru a délka řádku budou zadány z příkazového řádku jako parametry programu při spuštění

Testy...

Test 1



ZKOUŠKOVÝ
PŘÍKLAD

- Co program udělá

```
public class Null {  
  
    public static void main(String[] argv) {  
        ((Null) null).hello();  
    }  
  
    public static void hello() {  
        System.out.println("Hello world!");  
    }  
}
```

- A nelze přeložit
- B vyhodí za běhu NullPointerException
- C vypíše Hello World!

Test 2

- Co program vypíše

```
public class Test01 {  
  
    public static void main(String[] argv) {  
        run();  
        System.out.println("Konec");  
    }  
  
    public static void run() {  
        try {  
            run();  
        } finally {  
            run();  
        }  
    }  
}
```

- A nejde preložit
- B Konec
- C nic
- D vyhodí výjimku

Test 3

- Co program vypíše

```
class A {  
    public String className = "A";  
}
```

```
class B extends A {  
    private String className = "B";  
}
```

```
public class Test02 {  
    public static void main(String[] argv) {  
        System.out.println(new B().className);  
    }  
}
```

- A nejde preložit - chyba ve třídě B
- B nejde preložit - chyba ve třídě Test02
- C A
- D B
- E vyhodí výjimku
- F něco jiného



Verze prezentace PJ04.cz.2018.01

Tato prezentace podléhá licenci [Creative Commons Uveďte autora-Neužívejte komerčně 4.0 Mezinárodní License](#).