

Izuzeci

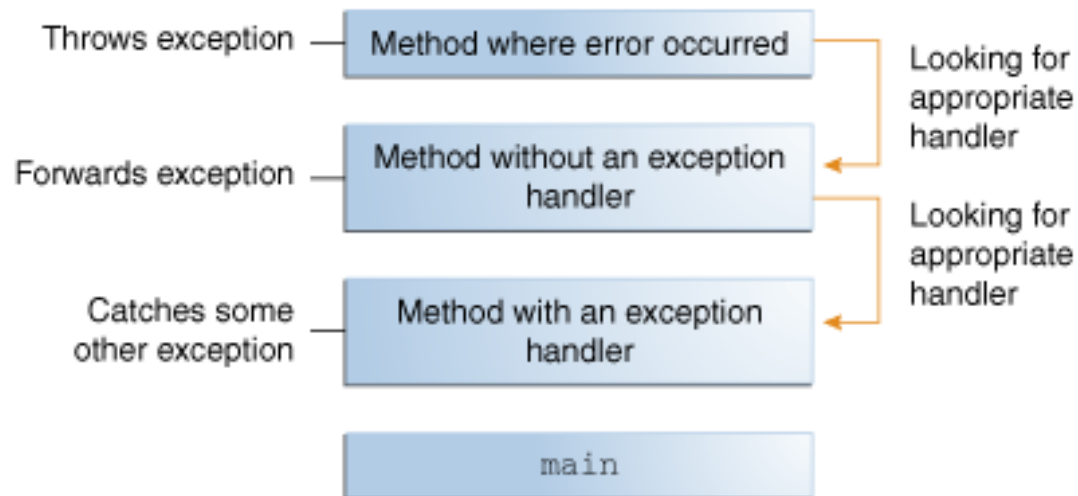
Šta je izuzetak?

- Izuzetak je zapravo greška koja se javlja u toku izvršavanja programa, dakle tzv. „runtime“ greška.
- Zašto se ne bavimo „compile time“ greškama?
- Šta su sve izuzeci:
 1. Očekivani (checked) izuzetak – npr. situacija kada otvaramo datoteku, a ona ne postoji.
 2. Neočekivani (unchecked) izuzetak:
 - A. Greške u hardveru i slično.
 - B. Logički izuzeci, npr. vrednost objekta je null.

Šta se dešava prilikom pojave izuzetka?

1. Kada se izuzetak desi, pravi se „izuzetak objekat“ sa informacijama o tome šta ga je izazvalo.
2. Taj objekat se:
 - A. ili prosleđuje (propagira) roditeljskoj metodi,
 - B. ili se greška razrešava odmah.
3. Na kraju se može desiti da izuzetak objekat stigne sve do main-a.

Redosled poziva



Hvatanje i razrešavanje izuzetaka

```
public class ListOfNumbers {
    private List<Integer> list;
    private static final int SIZE = 10;
    public ListOfNumbers () {
        list = new ArrayList<Integer>(SIZE);
        for (int i = 0; i < SIZE; i++) {
            list.add(new Integer(i));
        }
    }
    public void writeList() {
        PrintWriter out = new PrintWriter(new FileWriter("OutFile.txt"));
        for (int i = 0; i < SIZE; i++)
            out.println("Value at: " + i + " = " + list.get(i));
        out.close();
    }
}
```

try, catch

```
public class ListOfNumbers {  
    ...  
    public void writeList() {  
        try{  
            PrintWriter out = new PrintWriter(new FileWriter("OutFile.txt"));  
            for (int i = 0; i < SIZE; i++)  
                out.println("Value at: " + i + " = " + list.get(i));  
            out.close();  
        }catch(...){  
        }  
    }  
}
```

try, catch

```
public class ListOfNumbers {  
    ...  
    public void writeList() {  
        try{  
            PrintWriter out = new PrintWriter(new FileWriter("OutFile.txt"));  
            for (int i = 0; i < SIZE; i++)  
                out.println("Value at: " + i + " = " + list.get(i));  
            out.close();  
        }catch (IndexOutOfBoundsException e) { //neproverljiv izuzetak  
            System.out.println("IndexOutOfBoundsException: " + e.getMessage());  
        } catch (IOException e) { //proverljiv izuzetak  
            System.out.println("Caught IOException: " + e.getMessage());  
        }  
    }  
}
```

Neproverljive je bolje sprečiti ako je moguće!

```
public class ListOfNumbers {  
    ...  
    public void writeList() {  
        try{  
            PrintWriter out = new PrintWriter(new FileWriter("OutFile.txt"));  
            for (int i = 0; i < list.size(); i++)  
                out.println("Value at: " + i + " = " + list.get(i));  
            out.close();  
        } catch (IOException e) { //proverljiv izuzetak  
            System.out.println("Caught IOException: " + e.getMessage());  
        }  
    }  
}
```

JER ONI ZAPRAVO PREDSTAVLJAJU BAGOVE U PROGRAMU!

try, catch, finally

```
public class ListOfNumbers {  
    ...  
    public void writeList() {  
        try{  
            PrintWriter out = new PrintWriter(new FileWriter("OutFile.txt"));  
            for (int i = 0; i < list.size(); i++)  
                out.println("Value at: " + i + " = " + list.get(i));  
            out.close();  
        } catch (IOException e) {  
            System.out.println("Caught IOException: " + e.getMessage());  
        } finally { //ovaj blok se uvek izvršava, i kad ima i kad nema izuzetka  
            if (out != null) {  
                System.out.println("Closing PrintWriter");  
                out.close();  
            } else  
                System.out.println("PrintWriter not open");  
            }  
        }  
    }  
}
```

Propagiranje izuzetka - throws

```
public class ListOfNumbers {  
    ...  
    public void writeList() throws IOException {  
        PrintWriter out = new PrintWriter(new FileWriter("OutFile.txt"));  
        for (int i = 0; i < list.size(); i++)  
            out.println("Value at: " + i + " = " + list.get(i));  
        out.close();  
    }  
  
    public static void main(String args[]){  
        try{  
            ListOfNumbers list=new ListOfNumbers();  
            list.writeList();  
        }catch (...)  
    }  
}
```

Zadatak

- Importovati paket **izuzeci** u projekat pod nazivom Javalzuzeci.
- Paket se nalazi zipovan na adresi:
<http://www.math.rs/~kartelj/nastava/MGOOP2014/11/izuzeci.zip>
- Razrešiti sve izuzetke na način opisan u kodu i popraviti sve namerne greške.