

PB162 Programování v jazyce Java

8. cvičenie

od pochybenia k zlyhaniu

- pochybenie (*fault*, „bug“)
 - abnormálna podmienka v okolí alebo nedokonalosť v systéme
 - (v prostredí programových systémov) vzniká (najmä) nedokonalosťou ľudských mentálnych pochodov
 - napr. programátor očakáva, že $x \% 4$ vracia čísla 0..3
 - aritmetické výrazy, podmienky, konštrukcie, ktorých sémantika je iná, ako programátor očakáva

od pochybenia k zlyhaniu

- chyba (*error*)
 - manifestácia (prejav) pochybenia vnútri hraníc systému
 - napr. v premennej y je záporná hodnota
- zlyhanie (*faulire*)
 - systém sa nespráva podľa očakávaní
 - preruší poskytovanie služby
 - službu poskytuje v rozpore so špecifikáciou
 - napr. kadmiové tyče sa úplne vysúvajú z aktívnej zóny...

hranice systému

- vždy záleží na hraniciach systému
- ak berieme ako systém celú elektrárň:
- chyba
 - „kadmiové tyče...“.
- zlyhanie
 - prerušenie dodávky elektrickej energie
 - dodávka ionizujúceho žiarenia a spadu

pochybenia

- boli, sú, budú...
- pochybenia programátora (*bugs*, boty), problémy v okolí (hardware, sieť)...
- musíme zabrániť, aby prerástli v zlyhanie
- vo vnútri komponenty – hľadáme **testovaním**
- v okolí – ošetrujeme **výnimkami**

výnimky

- záležitosť komunikačného rozhrania
- komponent reaguje na **neprípusté okolnosti**, z ktorých sa na úrovni komponentu nemožno rozumne zotaviť
 - nevhodné argumenty metódy
 - nevhodný stav komponenty
 - problém na nižších vrstvách
 - ...

výnimky – delenie

1. behové (*runtime*)

– potomok `RuntimeException`

2. kontrolované

- v prípade, že sa klient môže z daného stavu zotaviť, je vhodné použiť kontrolovanú výnimku

výnimky – životný cyklus

- vyhodenie výnimky

```
throw new SomeException(parameters...);
```

- kontrolované výnimky musia byť uvedené klauzulou **throws**

```
public void myDangerousMethod() throws  
    RecoverableException {  
    ...  
}
```


výnimky – životný cyklus

- spracovanie výnimky
 - poslať vyššie (do volajúcej metódy)
 - zachytiť

```
try {  
    myDangerousMethod();  
} catch (RecoverableException e) {  
    doSomething(e);  
} finally {  
    cleanImportatnResource();  
}
```

„oblíbené“ výnimky

- **behové**
 - `NullPointerException`
 - `IllegalArgumentException`
 - `InvalidStateException`
 - `ArrayIndexOutOfBoundsException`
 - `ClassCastException`

Stack

- pop, peek na prázdnom zásobníku
 - vrátiť `null`?
 - vyhodit' výnimku?
 - akú?
- vidno oba prístupy, i v jedej triede
 - Deque
 - metódy vracajúce `null`
 - metódy vyhadjúce výnimku