* **Slecht vb van werken met Exceptions:**

public void setDatum(int dag, int maand, int jaar) {

**try {**

if (jaar < MIN\_JAAR || jaar > MAX\_JAAR) {

**throw new DatumException("Het jaartal kan enkel van 1584 tem 4099 lopen");**

}

else if (maand < 1 || maand > 12) {

**throw new DatumException("Een maand kan 1 tot 12 zijn");**

} else if (dag < 1 || dag > aantalDagenInDeMaand(jaar, maand)) {

**throw new DatumException("Deze dag is niet mogelijk voor deze maand");**

} else{

this.dag =dag;

this.maand =maand;

this.jaar=jaar;

}

**} catch (DatumException e) {**

System.out.println(e.getMessage());

}

}

In een try-catch zelf nieuwe exceptions aanmaken en smijten om die onmiddellijk op te vangen is niet de bedoeling!

**Wel:**

public void setDatum(int dag, int maand, int jaar) **throws DatumException** {

if (jaar < MIN\_JAAR || jaar > MAX\_JAAR) {

**throw new DatumException("Het jaartal kan enkel van 1584 tem 4099 lopen");**

}

else if (maand < 1 || maand > 12) {

**throw new DatumException("Een maand kan 1 tot 12 zijn");**

} else if (dag < 1 || dag > aantalDagenInDeMaand(jaar, maand)) {

**throw new DatumException("Deze dag is niet mogelijk voor deze maand");**

} else{

this.dag =dag;

this.maand =maand;

this.jaar=jaar;

}

}

Je methode smijt de exception, zodat degene die je methode gebruikt met parameters die niet toegelaten zijn, de exceptie kan opvangen (hij roept de methode op in een try-catch, zie onderstaand) en vervolgens deze exceptie kan afhandelen (loggen, bericht aan user, ...).

**try {**

setDatum(5, *20* , 2015);

**} catch (DatumException e) {**

System.out.println(e.getMessage()); // print dan "Een maand kan 1 tot 12 zijn"

**}**

* Wat denk je: **kan of beter mag een no arg. constructor** (= een constructor zonder parameters) **een Exception smijten?**