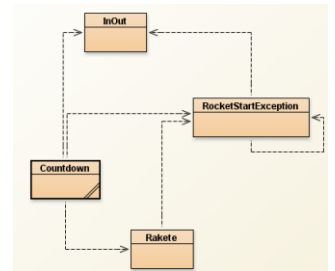


## Java Exceptions beim Raketenstart

(arbeite analog zum Projekt Eigene\_Exceptions\_werfen!)

Das Java-Projekt, welches dieses Prüfszenario für einen Raketenstart objektorientiert abbildet, besteht aus den Klassen **Rakete** (das Objekt Rakete festlegen), **Countdown** (die Dateneingabe, Das Erstellen des Raketenobjekts und der Versuch die Rakete zu starten oder einen Fehler ausgeben) sowie der benutzerdefinierten Exception **RocketStartException** (hier werden die Startbedingungen überprüft und bei Fehlern die Exceptions ausgelöst)



**Wenn nicht alle Bedingungen zusammenpassen, wird der Start der Rakete abgebrochen.**

**wetter\_perfekt** hat den Wert **0**, **treibstoff** hat den Wert **"voll"** und **technik\_OK** hat den Wert **true**.

**wetter\_perfekt** speichert einen Ganzzahlwert, der zufällig mit 50% Wahrscheinlichkeit **1** oder **0** ist. Damit soll die Unbeständigkeit des Wetters simuliert werden

--> `(int) (Math.random() * 2);` //erzeugt Ausgabe 0 oder 1

Die benutzerdefinierte Exception-Klasse ist recht unspektakulär. Es werden drei unterschiedliche Exception-Typen erstellt, die je nach Problemfall für sich ausgelöst werden könnten.

**checkTechnik, checkTreibstoff, checkWetter**

```
public class RocketStartException extends Exception {
    static String meldung="\nHouston es gibt ein Problem - wir können nicht starten!\n";
    public RocketStartException (String message) {
        super(message);
    }
    public static void checkWetter(int wetter_perfekt) throws RocketStartException
    {
        ...
    };
    public static void checkTechnik(boolean technik_OK) throws RocketStartException
    {
        ...
    };
    public static void checkTreibstoff(String treibstoff) throws RocketStartException
    {
        ...
    };
}
```

Nun zur wichtigsten Klasse – die Rakete:

```
int wetter_perfekt;
String treibstoff;
boolean technik_OK;
public Rakete(boolean technik_OK, String treibstoff, int wetter_perfekt) throws RocketStartException
{
    RocketStartException.checkTechnik(technik_OK);
    this.technik_OK=technik_OK;
    RocketStartException.checkTreibstoff(treibstoff);
    this.treibstoff=treibstoff;
    RocketStartException.checkWetter(wetter_perfekt);
    this.wetter_perfekt=wetter_perfekt;
}
```

Eine **RocketStartException** wird dann ausgelöst, wenn eine der drei Bedingungen false ist. Es tritt eine Ausnahmesituation ein und es wird die Exception **RocketStartException** geworfen. Sind hingegen alle drei Bedingungen **true** wird die Methode vollständig ausgeführt und die Rakete kann starten.

Zur Objekterzeugung einer Rakete und des dazugehörigen Exception-Handlings verwenden wir die Klasse **Countdown**. Mit dieser Klasse startest du deine Rakete:

```
public class Countdown
{
    // hier die benötigten globalen Variablen festlegen -->
    int wetter_perfekt;
    String treibstoff;
    boolean technik_OK;

    public Countdown()    // Konstruktor -->
    {
        // hier die Werte für treibstoff/technik_OK mithilfe der Klasse InOut einlesen und an die
        // globale Variable übergeben. Den Wert für wetter_ok mit Zufallsfunktion erzeugen -->
        ...
        eingabe_und_start ( );
    }

    public void eingabe_und_start ( ) // Methoden -->
    {
        try
        {
            Rakete r = new Rakete (technik_OK,treibstoff,wetter_perfekt );
            System.out.println("_____ \n");
            System.out.println("Countdown läuft!"); }

        catch ( ... )
        {
            ...
        }

        finally
        {
            // wird unabhängig des Ergebnisses von try ... catch immer ausgeführt -->
            System.out.println("_____ \n");
            ...
        }
    }
}
```

Erstelle die Ausgabeoptionen für die 4 Fälle (es gibt natürlich weitere Kombinationen)

#### 1. Das Wetter passt nicht

```
Technik ist ok? --> true oder false?.. true
Treibstoff? --> voll oder leer?.. voll

Houston es gibt ein Problem - wir können nicht starten!

_____

Das Wetter ist für den Start ungeeignet!
Technik: alles ok!
Tank: voll
```

#### 2. Die Technik streikt

```
Technik ist ok? --> true oder false?.. false
Treibstoff? --> voll oder leer?.. voll

Houston es gibt ein Problem - wir können nicht starten!

_____

Wetter ist prima!
Technik: Ein Fehler ist aufgetreten!
Tank: voll
```

#### 3. Der Tank ist leer

```
Technik ist ok? --> true oder false?.. true
Treibstoff? --> voll oder leer?.. leer

Houston es gibt ein Problem - wir können nicht starten!

_____

Wetter ist prima!
Technik: alles ok!
Tank: leer
```

#### 4. Alles ok

```
Technik ist ok? --> true oder false?.. true
Treibstoff? --> voll oder leer?.. voll

_____

Countdown läuft!

_____

Wetter ist prima!
Technik: alles ok!
Tank: voll
```