

## Was sind Laufzeitfehler

Laufzeitfehler sind Fehler, die erst zur Laufzeit des Programms erkannt werden können und nicht etwa schon beim Kompilieren. Es handelt sich daher nicht um Syntaxfehler im Programm, sondern um unerlaubte Operationen während des Programmlaufs, z.B. Division durch 0, Zugriff auf ein nichtexistentes Vektorelement usw. Manche dieser Fehler führen zu einem Programmabbruch durch das C++-Laufzeitsystem (z.B. Division durch 0), in anderen Situationen rechnet das Programm falsch weiter oder reagiert nicht mehr.

Bei manchen Fehlern kann man nichts machen, sie führen unvermeidlich zum Absturz (z.B. Fehler in der Speicherverwaltung), andere Fehler führen erst zum Absturz, wenn man bestimmte Aktionen ausführt: Eine Division durch 0 lässt das Programm abstürzen, aber man könnte vorher den Divisor testen und dadurch den Absturz verhindern. Man könnte bei Dateien überprüfen, ob sie auch geöffnet werden konnten usw. Solche Tests erlauben es dem Programm, den Fehler zu bemerken, bevor es zum Absturz kommt.

### Wir haben einen Laufzeitfehler bemerkt, und nun?

Mögliche Reaktionen sind:

- 1) Programm mit Fehlermeldung beenden (nicht besonders benutzerfreundlich).
- 2) Fehler ignorieren und weitermachen (die dümmste aller Reaktionen).
- 3) Aktuelle Aktion, die den Fehler entdeckte und deshalb nicht mehr weitermachen kann, abbrechen und den Fehler einer übergeordneten Stelle melden.

Offensichtlich ist 3) die beste Möglichkeit, es gibt aber noch ein Paar offene Fragen:

- 1) Was ist eine übergeordnete Stelle? Jemand, der etwas in Auftrag gibt, ist dem Ausführenden meist übergeordnet. Also ist der Aufrufer einer Funktion (und dessen Aufrufer usw.) dieser Funktion übergeordnet.
- 2) Wie meldet man den Fehler zurück? Hier unterscheiden sich die einzelnen Programmiersprachen völlig. Manche verwenden einen globalen Fehlerspeicher (in C: `errno`), manche signalisieren mit speziellen Rückgabewerten eine Fehlfunktion (C: Nullpointer etc.). C++ (und Python) hat schon seit langem einen eigenen Mechanismus für die Fehlerrückmeldung: Die Exception = das Werfen eines Objekts:  
`throw some_object;`

```
throw 5; // wir werfen einen int
throw 7.0; // wir werfen einen double
throw "Dies ist ein Fehler"; // wir werfen einen C-String
throw std::string{"Division durch 0"}; // einen std::string
throw Hund{"Bello"}; // Tierquäler
```