

Welche Objekte wirft man am besten?

Während man prinzipiell jedes Objekt werfen kann, gibt es in der C++-Bibliothek vordefinierte Objekttypen, die besser dafür geeignet sind und die auch vom C++-Laufzeitsystem verwendet werden: `std::exception` aus `<exception>`

Hierbei handelt es sich um den allgemeinsten Typ von Exception-Objekten. In `<stdexcept>` definiert C++ speziellere Exception-Typen, die man eher verwenden sollte:

```
std::range_error, std::runtime_error, std::overflow_error ...
```

Alle diese Typen sind von `std::exception` abgeleitet. Sie haben alle einen Konstruktor mit einem Fehlertext (C-String oder `std::string`) und funktionieren prinzipiell gleich.

```
try {
    throw range_error("Index ist zu gross");
}
catch (runtime_error& e) // fängt nur std::runtime_error
catch (exception& e)    // fängt alle übrigen Exceptions ebenfalls
```

Man sollte diese **Typen IMMER per Referenz** auffangen, um unerwünschte Umwandlungen zu verhindern.

Bei größeren Projekten ist es eine gute Idee, eigene Exception-Typen zu definieren, die von den vorgegebenen Basistypen abgeleitet sind:

```
class my_exception : public std::runtime_error { // erbt davon
... };

try {
    ...
    throw my_exception{ " Fehler in Modul ..." };
    ...
} catch (my_exception& e) // ich kümmere mich nur um eigene Fehler
{...}
```