



Übersichtsblatt: Ausnahmenbehandlung

Grundidee:

C++ bietet die Möglichkeit Fehler in Programmen zu vermeiden indem man im Problemfall ein so genanntes Exception wirft. Das System basiert auf drei wesentlichen Komponenten:

- try
- throw
- catch

Funktionsprinzip:

Das eigentliche Programm(-fragment) steht in einem so genannten try-Block. Innerhalb des Programms wird nun eine Bedingung überprüft (z.B. $x > 0$). Falls diese Bedingung nicht erfüllt ist wird mit throw ein Exception geworfen. Hiefür kann jedes beliebige Objekt verwendet werden, in der Praxis bedient man sich allerdings spezieller Ausnahme-Klassen.

Außerhalb des try-Blockes wird das geworfene Exception dann von einem catch-Block aufgefangen. In diesem catch-Block wird dann auf die Ausnahme reagiert, bzw. wird diese Dokumentiert.

Beispielimplementierung:

Headerdatei der Ausnahmeklasse:

```
#ifndef AUSNAHME_H_
#define AUSNAHME_H_
#include<string>

class Ausnahme{
protected:
    std::string grund;
public:
    Ausnahme(std::string grund="");
    virtual ~Ausnahme();
    virtual std::string getGrund();
};

#endif /*AUSNAHME_H_*/
```

Implementierung der Ausnahmeklasse

```
#include "Ausnahme.h"
Ausnahme::Ausnahme(std::string grund)
:grund(grund){}

Ausnahme::~Ausnahme(){}

std::string Ausnahme::getGrund(){
    return grund;
}
```

Programmfragment der Klasse Punkt:

```
void Punkt::setX(int x){
    try{
        if(x < 0)
            throw Ausnahme("Ausnahme:
            negatives x!");
        else
            if(x==42)
                throw 42;
            else
                this->x=x;
    }
}
catch (Ausnahme a){
    std::cout << a.getGrund() << "\n";
    this->x=0;
}
catch (int){
    std::cout << "Haha 42 geht nicht\n"
    this->x=0;
}
std::cout << "x auf " << (this->x) << "gesetzt\n"
```

Originalausgabe in der Konsole:

```
Rancinger@HAUPTRECHNER ~
$ ausnahmen.exe

x auf 7 gesetzt
[7,5]
Ausnahme: negatives x
x auf 0 gesetzt
[0,5]
Haha 42 geht nicht
x auf 0 gesetzt
[0.5]
```

