



Seminar 3

Ausnahmen (Exceptions)

Wenn eine Programmsituation keine sinnvolle Fortführung erlaubt, tritt eine Ausnahme ein.

Das ermöglicht eine Unterscheidung zwischen Rückgabewerten und Fehlersituation.

Vorhersehbaren und unvorhersehbaren Fehlersituationen können in C++ mit dem Konstrukt der **Exceptions** behandelt werden.

Ausnahmen können verursacht werden (**THROW**). Codeabschnitte können angeben dass sie Ausnahmen erwarten (**TRY**), abfangen und behandeln (**CATCH**).

SOURCE CODE	OUTPUT
<pre>#include <iostream> using namespace std; class Test { public: Test() { cout << "Constructor of Test " << endl; } ~Test() { cout << "Destructor of Test " << endl; } }; int main() { try { Test t1; throw exception(); } catch (exception& e) { cout << "Caught exception" << endl; } }</pre>	<pre>Constructor of Test Destructor of Test Caught 10</pre>

Ausnahmen sind Objekte, und können Informationen über die Art des Fehlers beinhalten.

Die Klasse `exception` aus der Standardbibliothek dient als Basisklasse. Spezielle Fehler können als Unterklassen von `exception` definiert werden.

Wenn für eine Ausnahme keine Behandlung definiert ist, dann wird ein **Default-Handler** aktiviert, der das Programm mit einer Fehlerausgabe terminiert.

Übung

Eine Autovermietung Firma braucht ein Programm um ihre Kunden- und Mietwagen-Listen zu verwalten.

Das Programm soll in Klassen strukturiert werden. Als Kontroller diene eine Klasse Rental mit der folgenden Funktionalität:

- Kunden
 - Kunden einfügen (Kunden-ID, Name)
 - Kundendaten aktualisieren
 - Kunden löschen
- Wagen
 - Wagen einfügen (Automarke, Modell, Typ [LKW/PKW], Auto-ID)
 - Wagen löschen

Zu schreiben sind sowohl die Klassendeklarationen mit Konstruktoren, Destruktoren und Methoden als auch ein Testprogramm das die Klasse Rental verwendet.

