

	Datum:	Fach:
Ausnahmebehandlung unter C++: Passwortüberprüfung		

Ausgangssituation:

„Es ist unmöglich, idiotensichere Programme zu schreiben, da Idioten so erfinderisch sind“.

Der Programmierer muss nicht nur den Algorithmus im Blick behalten, sondern alle Fehleingaben, Systemfehler und Unverträglichkeiten vorhersehen und ausschalten.

Ausgangspunkt ist das Beispiel „1_klassisch“. In der Funktion `calc()` soll der Quotient zweier positiver ganzer Zahlen berechnet werden.

Selbst in diesem einfachen Programm kann schon eine ganze Menge schiefgehen!

Das Beispiel „2_klassisch_fehler“ zeigt, wie man die Fehlerbehandlung „klassisch“ realisieren kann.

Das Beispiel „3_exception_1“ zeigt, wie man das etwas eleganter lösen kann. C++ bietet hierzu einen Mechanismus, der es ermöglicht, einen Block von Anweisungen gegen Abstürze zu sichern. Alle darin auftretenden Ausnahmefehler werden einem Behandlungsblock zugeleitet. Der gesicherte Anweisungsblock wird durch das Schlüsselwort **try** (engl. versuchen) eingeleitet. Der Fehlerbehandlungsblock beginnt mit **catch** (engl. fangen). Im Anweisungsblock wird also versucht, das Programm fehlerfrei abzuarbeiten. Der Fehlerbehandlungsblock fängt die Abstürze ab wie ein Netz. Mit dem Befehl **throw** kann man eigene Ausnahmen erzeugen. Er bricht die Verarbeitung sofort ab und springt direkt in die passende Ausnahmebehandlung.

Das Beispiel „4_exception_2“ zeigt wie man eigene Fehlerklassen deklarieren und verwenden kann.

Aufgabenstellung:

Das bereitgestellte Programm „Password_0“ überprüft ein eingegebenes Passwort „klassisch“ auf die Einhaltung bestimmter Kriterien. Bei Nichteinhaltung der Kriterien wird eine detaillierte Fehlermeldung ausgegeben.

1. Die Fehlerbehandlung soll nun mit **try** und **catch** realisiert werden. In der Funktion `CheckPassword()` wird bei Nichteinhaltung der Kriterien mit **throw** eine Exception vom Typ **string** ausgelöst. Das Fehlerobjekt soll auch hier eine detaillierte Fehlermeldung enthalten, die im `catch`-Block des Hauptprogramms ausgegeben wird.
2. Die Fehlerbehandlung soll nun um eine eigene Fehlerklasse **TPasswordError** erweitert werden, so wie es im Beispiel „4_exception_2“ gezeigt wird. In der Funktion `CheckPassword()` wird bei Nichteinhaltung der Kriterien nun mit **throw** eine Exception vom Typ **TPasswordError** ausgelöst. Das Fehlerobjekt soll auch hier wieder eine detaillierte Fehlermeldung enthalten, die im `catch`-Block des Hauptprogramms ausgegeben wird.

Hilfsmittel:

Eine ausführliche Beschreibung der Ausnahmebehandlung unter C++ findet sich im bereitgestellten Informationsblatt „Katastrophenschutz mit try und catch“.