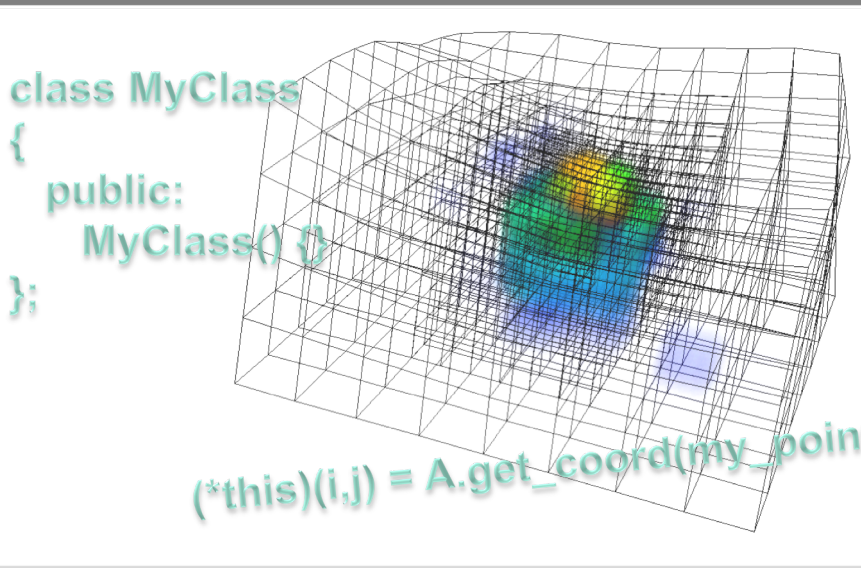
Karlsruher Institut für Technologie

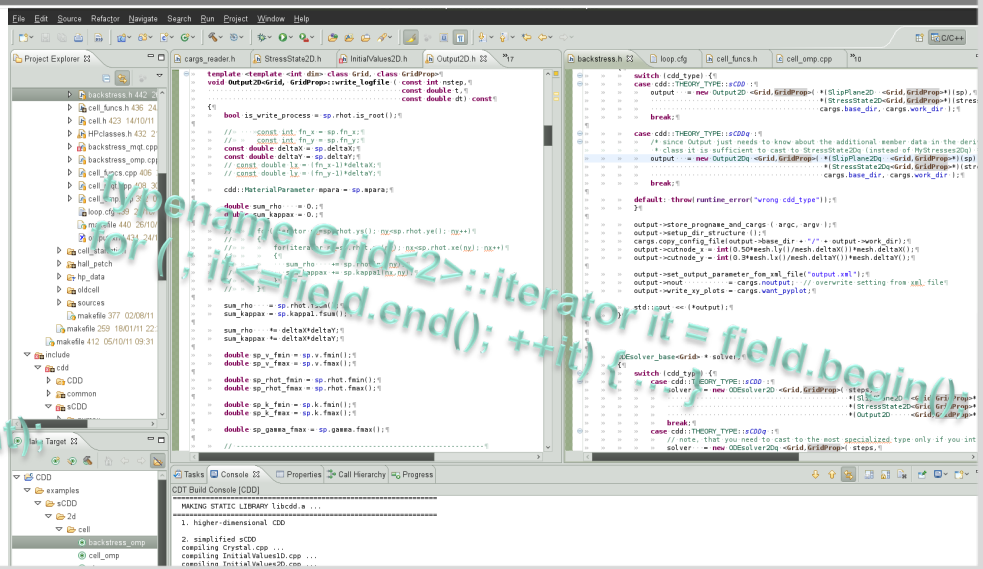
Wissenschaftliches Programmieren für Ingenieure

Dr. M. Stricker, A. Trenkle, T. El Achkar, Dr. D. Weygand

Übung 6:
Zahlenspielereien, Abstrakte Basisklassen, virtuelle Memberfunktionen
Lineare Gleichungssysteme: Feder-Gitter
Numerische Integration: MD Programm Planeten

Institute for Applied Materials – Computational Materials Science (IAM-CMS)
KIT – The Karlsruhe Institute of Technology – University of the State of Baden-Wuerttemberg, GERMANY





Zahlen – Rechengenauigkeit

Todo

- Übersetzen Sie den Quellcode „main_zahlen.cpp“ und erklären Sie die Ausgabe des Programms.

```
#include <iostream>
#include  <cmath>
using namespace std;


int main(){
    double v1,v2,v3;
    // Vergleich von Fliesszahlen

    for(int i=0;i<10;i++){

        v1=pow(10,i)*1e16;
        v2=v1;
        unsigned int nc { 0};
        v3 =0.;
        while(v2==v1){
            v3+=10.e2;  // Inkrement erhöhen
            v2+=v3;
            nc++;
            // was macht dies hier?
            if(nc==0){
                cout<<"break ";
                break;
            }
        }
        cout<<"nc="<<nc  <<" v2="<<v2<<" v1="<<v1<<endl;
    }
    return 0 ;
}
```

- Welche Schlussfolgerungen ziehen Sie aus dem Verhalten des Programms?

Vererbung: abstrakte Basisklassen + Polymorphie

Karlsruher Institut für Technologie

Info

- In der Vorlesung wurde die Rolle virtueller Memberfunktionen bei der Vererbung betrachtet.

- Im Beispielcode im Archiv `polymorphy.tgz` wird die Verallgemeinerung abstrakter Basisklassen vorgestellt.
  - In `polymorph_func.h` werden die verwendeten Klassen definiert, in `polymorph_func.cpp` implementiert.
  - Die Basisklasse trägt den Namen `BaseClass`.
  - Sie dient als Ausgangsklasse für die beiden abgeleiteten Klassen `Derived_A` und `Derived_B`.

Todo

- Entpacken und übersetzen Sie das Programm und beantworten Sie die Fragen auf der übernächsten Folie anhand der Bildschirmausgaben. (Jede Methode erzeugt eine Bildschirmausgabe, die es erlaubt die Aufrufe nachzuvollziehen.)

Überblick Baseclass und abgeleitete Klassen

Info

Die **Basisklasse** enthält die „virtual“ Memberfunktionen

- `another_function()`
- `just_a_function()`

Sowie die „pure virtual“ Memberfunktion

- `some_function()=0`

Die abgeleitete Klasse **Derived_A** implementiert

- `some_function()`

Und überlädt:

- `another_function()`
- `just_a_function()`

Die abgeleitete Klasse **Derived_B** implementiert

- `some_function()`

Hinweis: Sehen sie sich vor allem die `polymorph_funcs.h` genau an!

Vererbung: abstrakte Basisklassen + Polymorphie (1/2)

Karlsruher Institut für Technologie

Todo

Fragen:

- Wie wird eine Klasse von einer anderen Klasse (Basisklasse) abgeleitet ?
  - Welche Möglichkeiten der Vererbung gibt es ?
  - Wie sind die Zugriffsrechte einer mit **public** abgeleiteten Klasse? (Tabelle)

Zugriff auf	public	protected	private
Memberfunktion einer Instanz der selben Klasse			
Memberfunktion einer Instanz der abgeleiteten Klasse			
Keine Memberfunktion			

Vererbung: abstrakte Basisklassen + Polymorphie (2/2)

Karlsruher Institut für Technologie

Todo

- Wie werden Funktionen der Basisklasse angelegt, damit sie in abgeleiteten Klassen überladen werden **können**?
- Welche Funktionen **müssen** in abgeleiteten Klassen immer implementiert werden?
- Ein Objekt einer abgeleiteten Klasse kann auch über eine Referenz oder einen Pointer auf seine Basisklasse angesprochen werden.
  - Was geschieht beim Aufruf einer virtuellen Memberfunktion?
  - Was geschieht beim Aufruf einer nicht-virtuellen Memberfunktion?

Lineare Gleichungssysteme: Federgitter (1/2)

Karlsruher Institut für Technologie

Todo

Schreiben Sie ein Programm, um den Gleichgewichtszustand des Federgitters aus der Vorlesung zu berechnen

Verwenden Sie hierzu die Vektor und Matrix Klassen sowie die Funktionen zum Lösen von LGS aus Übung 5 als Vorlage.

Verwenden Sie zur Lösung des aufgestellten LGS

- Den vorhandenen CG Löser
- Optional: Cholesky-Verfahren (Implementierung nach Vorlesungsfolien)

Stellen Sie das lineare Gleichungssystem für das Kräftegleichgewicht

; mit

$$f_i + g_i = 0 \qquad g_i = -4u_i + u_{oben} + u_{unten} + u_{links} + u_{rechts}$$

Und f_i: äußere Kräfte senkrecht zur Ebene (rechte Seite)

u_i: Auslenkungen senkrecht zur Ebene des unbelasteten Federgitters.

Die Matrix muss, wie in der letzten Übung, symmetrisch sein.

Lineare Gleichungssysteme: Federgitter (2/2)

Karlsruher Institut für Technologie

Todo

Das Programm soll wie folgt benutzbar sein

- **Eingabe** durch Benutzer zur Laufzeit:
  - Größe des Gitters
  - externe Kräfte auf Knoten:
    (i) zum Beispiel alle Punkte erfahren die gleiche äußere Kraft
    (ii) die äußere Kraft liegt wirkt nur an einigen Knoten

- **Ausgabe**: Datei schreiben und mit Gnuplot visualisieren
  - Ausgabeformat (pro Zeile): Knotenindex und –verschiebung
    ```
    i  j  u_ij
    ```
  - Zur Darstellung in Gnuplot folgende Befehle verwenden:
    ```
    set dgrid3d              (3-dimensionales Gitter)
    set contour base         (Contourplot/“Höhenlinien“)
    splot 'dateiname' u 1:2:3
    ```

MD Programm: Planetenbewegung

Todo

Quellcode liegt auf ILIAS: `Planeten.tgz`

- Entpacken und analysieren Sie das Programm aus der Vorlesung (Seite 207 ff.) im Unterverzeichnis `./Planeten/src/`
- Übersetzen Sie das Programm (siehe „makefile“ Datei, die sich im Verzeichnis befindet)
- Testen Sie das Programm für das Planetensystem im Unterverzeichnis `./Planeten/data/`

- **Programmieraufgabe**
  Schreiben Sie zusätzlich eine Routine, die die Gesamtenergie des Systems berechnet (kinetische Energie und potenzielle Energie)
  - Tragen sie die Gesamtenergie gegen die Zeit auf!
  - Was beobachten Sie bei den verschiedenen Integrationsverfahren?