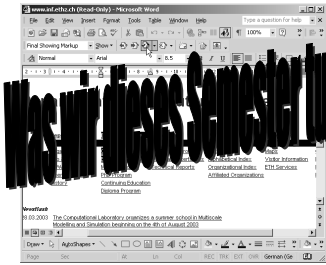


Click to add title

Programmieren im Grossen



Übungsgruppe Matthias Sala



4. April 2003

Informatik IV SS 03

4. April 2003

Meine Email
salam@student.ethz.ch

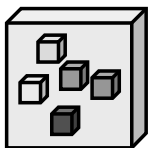
Informationen, Material, Übungsunterlagen
www.n.ethz.ch/student/salam

das Testat bekommt ihr, wenn ihr
80% Übungen abgibt

die Übungen gebt Ihr bitte jeweils
am Freitag per Email ab

Was wir dieses Semester tun

Programmieren im Grossen



Objektorientiertes
Programmieren

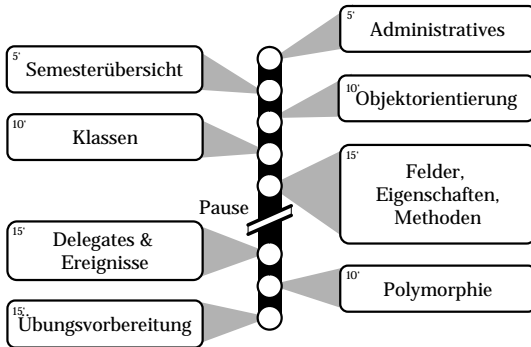


Parallele
Programmierung



Verteilte
Programmierung

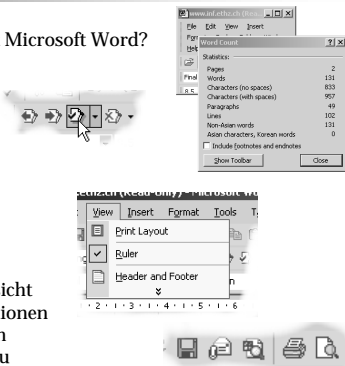
Was wir heute tun



Ein grosses Programm

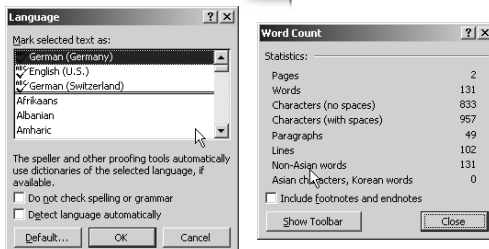
aus was besteht Microsoft Word?

- Fenstern
- Rahmen
- Toolbars
- Schaltflächen
- Popupmenüs
- Menüpunkte
- Dokumentansicht
- Speicherfunktionen
- Druckfunktion
- Druckvorschau



Funktionen in Fenstern

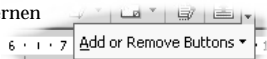
was passiert, wenn man auf drückt?




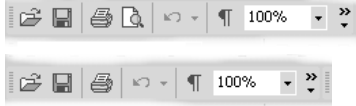
hat der Programmierer in beiden Fenstern das Schliessen zweimal programmiert?


Funktionen der Toolbar

in Toolbars kann man neue Schaltflächen hinzufügen / entfernen



was passiert, wenn ich die Schaltfläche  entferne?



sind zwei Toolbars mit und ohne  abgespeichert?

Dinge wiederverwenden - Objektorientierung

wenn man im Grossen programmiert, spart man sich Zeit, indem man Dinge immer wiederverwendet

Microsoft Word wäre sonst noch viel umfangreicher und fehlerhafter

diese wiederverwendbaren Dinge sind Objekte

Toolbar-Objekt, Fensterobjekt, ...

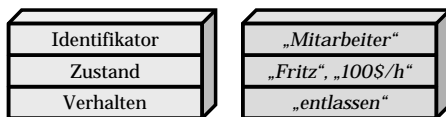
Grosse Programme bestehen aus vielen verschiedenen Objekten

Microsoft Word besteht aus einem Fensterobjekt, das weitere Objekte enthält wie die Toolbar

Objekte

Code wiederverwenden heisst nicht, dass der Code mit Copy & Paste immer überall wieder hinein kopiert wird

um Code wiederverwenden zu können, bilden wir Objekte



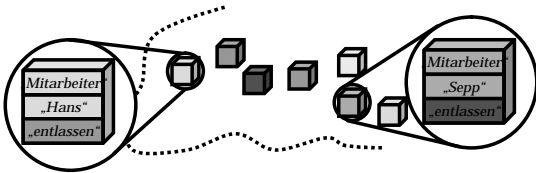
Ein Objekt ist eine Sammlung zueinander in Beziehung stehender Informationen und Funktionen

Objekte definieren

um Objekte zu erstellen oder mit Objekten zu arbeiten, müssen diese zuerst definiert werden

C# verwendet dafür Klassen

Eine Klasse definiert eine Menge von Objekten, die eine gemeinsame Struktur haben und ein gemeinsames Verhalten zeigen



Klassen

Klassen bestehen aus:

- Name

die Klasse wird benannt, um Objekte einer Klasse bilden zu können

„erstelle neuen Mitarbeiter“

- Zustand

das Objekt soll sich verändern können

Name, Alter, Lohn, Position, Begabung

- Verhalten

das Objekt soll Funktionen haben

„entlasse Mitarbeiter“

Klassen in C#

```
class Mitarbeiter {  
    String vorname;  
    int alter;  
    int stundenlohn; } Zustand  
  
    public void entlassen() {  
    }  
    public void befoerdern() {  
    }  
}
```

← Name der Klasse

} Verhalten

Felder in C#

halten den Zustand eines Objektes

```
[Modifikator] Datentyp Bezeichner [= Initialwert];
```

Modifikator definiert Sichtbarkeit & Verhalten,
`const, public, private, protected, static,`
`abstract, internal, virtual, readonly`

Datentyp definiert welchen Typ von Daten das
Feld aufnehmen kann,
`byte, short, int, long, char, string`

Bezeichner gibt dem Feld einen Namen, worüber
man darauf zugreifen kann

Initialwert gibt dem Feld einen Startwert

Methoden in C#

beschreiben das Verhalten eines Objektes

```
[Modifikator] Ergebnistyp Bezeichner ([Parameter])  
{  
    // Anweisungen  
}
```

Modifikator definiert Sichtbarkeit & Verhalten,
`public, private, protected, abstract`

Ergebnistyp definiert welchen Typ von Daten die
Methode zurückgeben wird,
`void, byte, short, int, long, char, string`

Bezeichner gibt dem Feld einen Namen, worüber
man darauf zugreifen kann

Parameter Parameter, welche die Methode bekommt

Eigenschaften in C#

sind dynamische Werte, welche mit zwei „Methoden“, dem getter und dem setter beschrieben werden

```
[Modifikator] Ergebnistyp Bezeichner {  
    get {  
        // Anweisungen  
        return Etwas;  
    }  
    set {  
        Etwas = value;  
        // Anweisungen  
    }  
}
```

return gibt bei Abfrage der Eigenschaft
einen Wert zurück

value enthält den Wert, welcher der Eigenschaft
zugewiesen wurde

Delegierte in C#

sind Verweise auf keine oder eine Methode

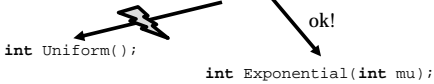
[Modifikator] delegate Methodenprototyp;

definiert den Rückgabewert und die Parameter,
welche die Methode, auf die verwiesen wird,
haben muss

zum Beispiel:

```
public delegate int Zufallszahl(int max);
```

```
int Uniform();  
int Exponential(int mu);
```



Material

C# Tutorial auf englisch

<http://www.softsteel.co.uk/tutorials/cSharp/contents.html>

gratis Buch zu C# in deutsch

<http://www.galileocomputing.de/openbook/csharp/>



komplettes Buch zu C# mit Übungen

<http://www.goepps.de/download/CSlernen.zip>

einige Links zu C# & .NET

<http://www.informatik.uni-rostock.de/~wlohmann/DotNET/>
