BAUINFORMATIK

SS 2013 Vorlesung V

Johannes Lange

Allgemeines

- NEU Allgemeine Punkte? - Fragen?
- Hausarbeit (zu zweit)
  - Kleine Projekte zum Programmieren
  - Eigene Ideen gewünscht

VBA (Visual Basic for Applications)

- Was machen wir?
  - Wiederholung
    - Objektorientierte Programmierung
  - Und dann:
    - Objektorientierte Programmierung
      - Mehrere Objekte, Zusammenspiel
    - Software Entwurf und Abwicklung

1. Klasse erstellen

- Einfügen / Klassenmodul
- Öffentliche Variable erzeugen

```
Public OpenValue As String
```

- Öffentliche Funktion erzeugen

```
Public Function OpenFunction()
    MsgBox ("Public Funcion")
    OpenFunction = "ok"
End Function
```

2. Objekt der Klasse erzeugen

- Aufruf aus einem anderen Modul
- Objekt-Variable wird deklariert

```
Dim Obj_Klasse1 As Klasse1          'Nur Deklaration
```

- Objekt der Klasse *New* erzeugen
- Objekt der Variablen mit *Set* zuweisen

```
Set Obj_Klasse1 = New Klasse1
```

3. Mit Objekt arbeiten

- Mit dem „Punkt“ können jetzt Funktionen und Variablen aufgerufen werden

```
Obj_Klasse1.OpenValue = "Hallo Welt"
Testfkt = Obj_Klasse1.OpenFunction()
```

Initialize und Terminate

- Initialize wird beim Erzeugen eines Objekts aufgerufen (Konstruktor)

  → Voreinstellungen
- Terminate wird beim Zerstören eines Objekts aufgerufen (Destruktor)

  → Speichern, Zerstören…

Public / Private

- Variablen und Funktionen sind aufrufbar (gültig):
  - Public: in allen Modulen
  - Private: nur im aktuellen Modul

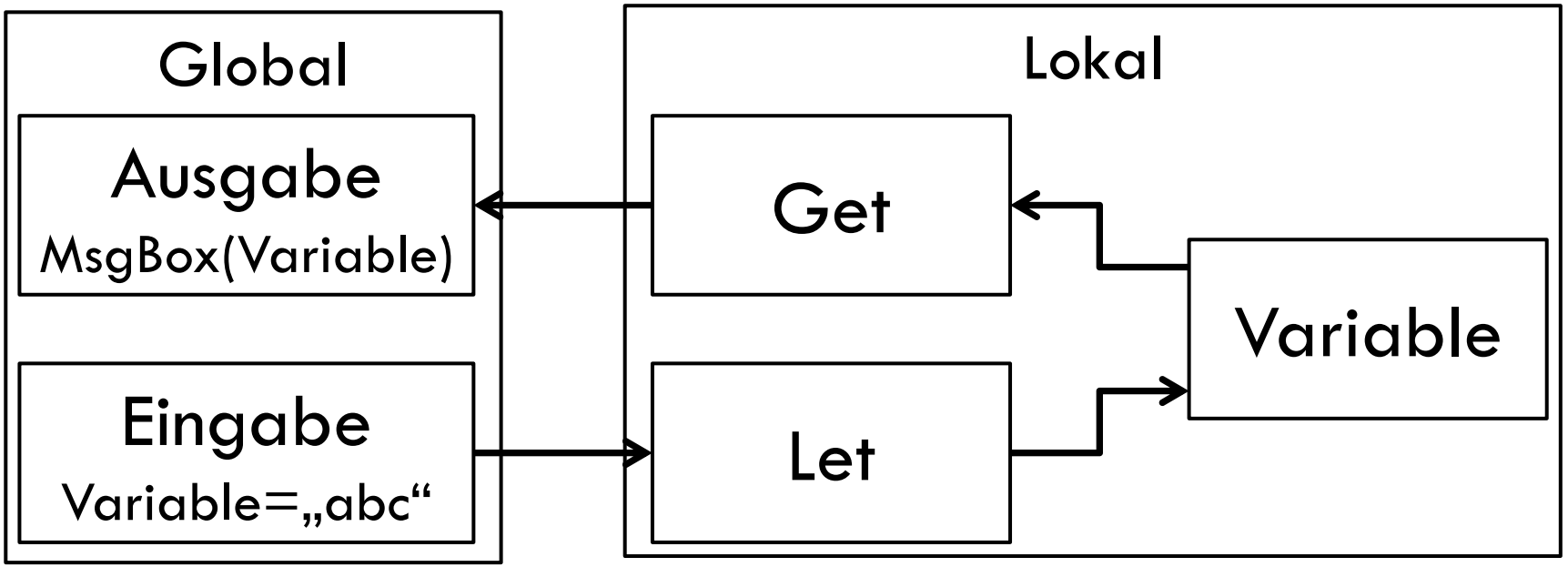
- Wozu?
  - Kapselung →Zugriff nur auf offizielle Schnittstellen
  - Gerade in der Objektorientierung wichtig

Property

- Variablen über get, let indirekt ansteuern

→ Kapslung der Variablen

→ Prüfung/Veränderung der Manipulation



Hilfsmittel „With“

- With (*klasse*)

→ Attribute und Methoden von Klasse beginnen mit „.“

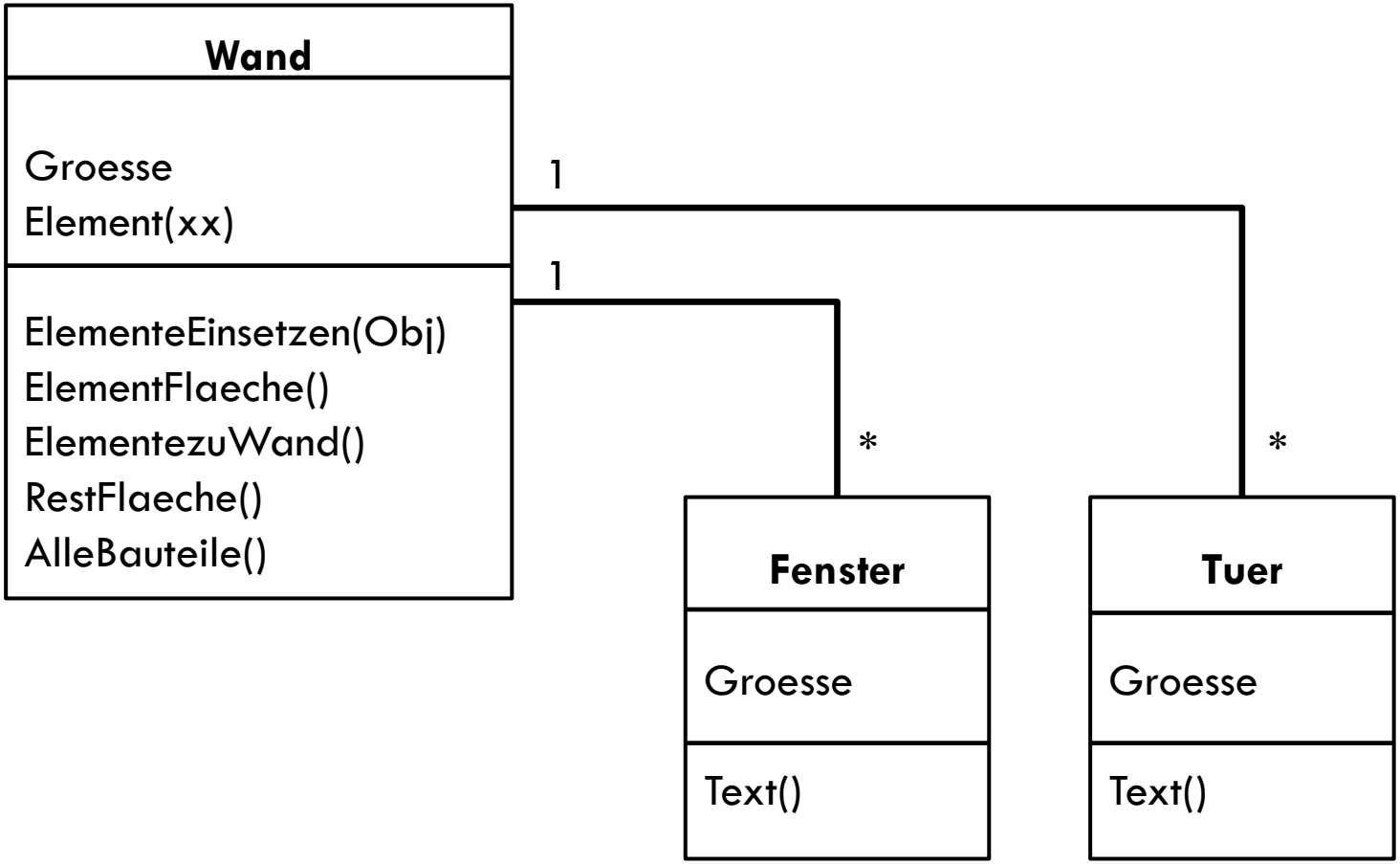
```
'Immer vollständig:
Rad_HintenLinks.Beschleunigen (10)
Rad_HintenLinks.Bremsen (5)
Rad_HintenLinks.Bremsen (3)

'Oder schöner:
With Rad_HintenLinks
     .Beschleunigen (10)
     .Bremsen (5)
     .Bremsen (3)
End With
```

Beispiel Wand und Fenster

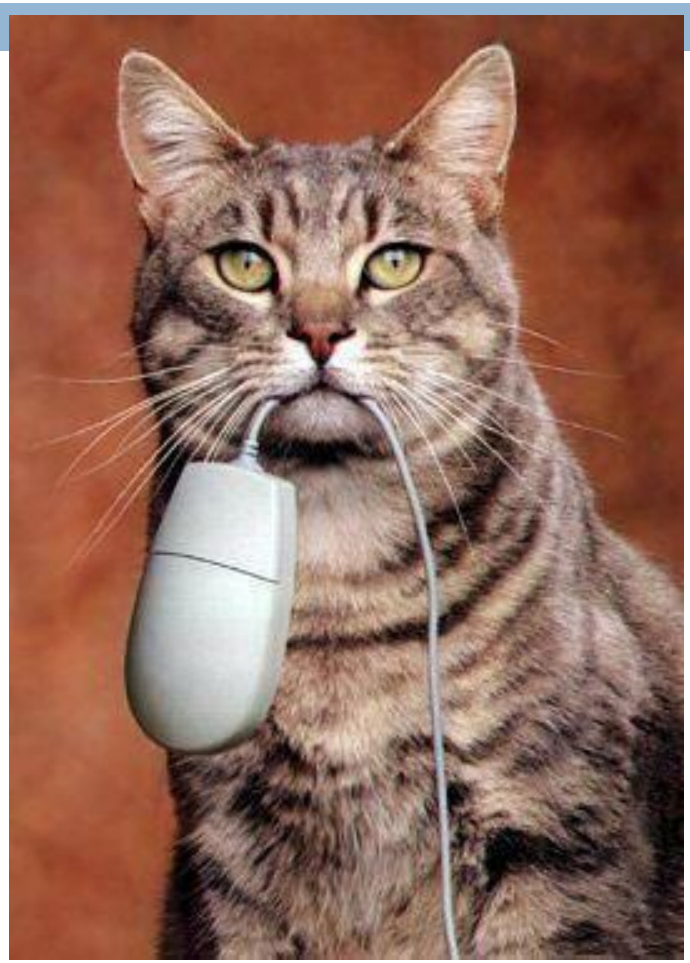
- Klassen Wand, Fenster, Tuer
- Fenster und Tuer werden dynamisch in Wand gespeichert
- Wand kann für Berechnungen (z.B. Fenster-/ Türenfläche in %) aufgerufen werden.
- Dialogbox zu Anschauung

Klassendiagramm

Excel



And now to …



Quelle: www.loslachen.ch

Was ist Software?

- Sammelbegriff für
  - ausführbare Programme
  - zugehörigen Daten

Nicht physikalische Funktionsbestandteile eines Computers als Gegenstück zu Hardware

www.Wikipedia.de

Lehrbuch der Softwareentwicklung, Prof. Dr. Balzert

Warum ist Software schwer zu entwickeln?

Software (ist) (im Vergleich zu einem Bauprojekt)

- ein immaterielles Produkt,
- physikalisch nicht begrenzt,
- schwer zu vermessen,
- keinem Verschleiß unterlegen,
- altert.

www.Wikipedia.de

Lehrbuch der Softwareentwicklung, Prof. Dr. Balzert

Softwareentwicklung - Anforderungen

Anforderungen an Software (ISO 9126)

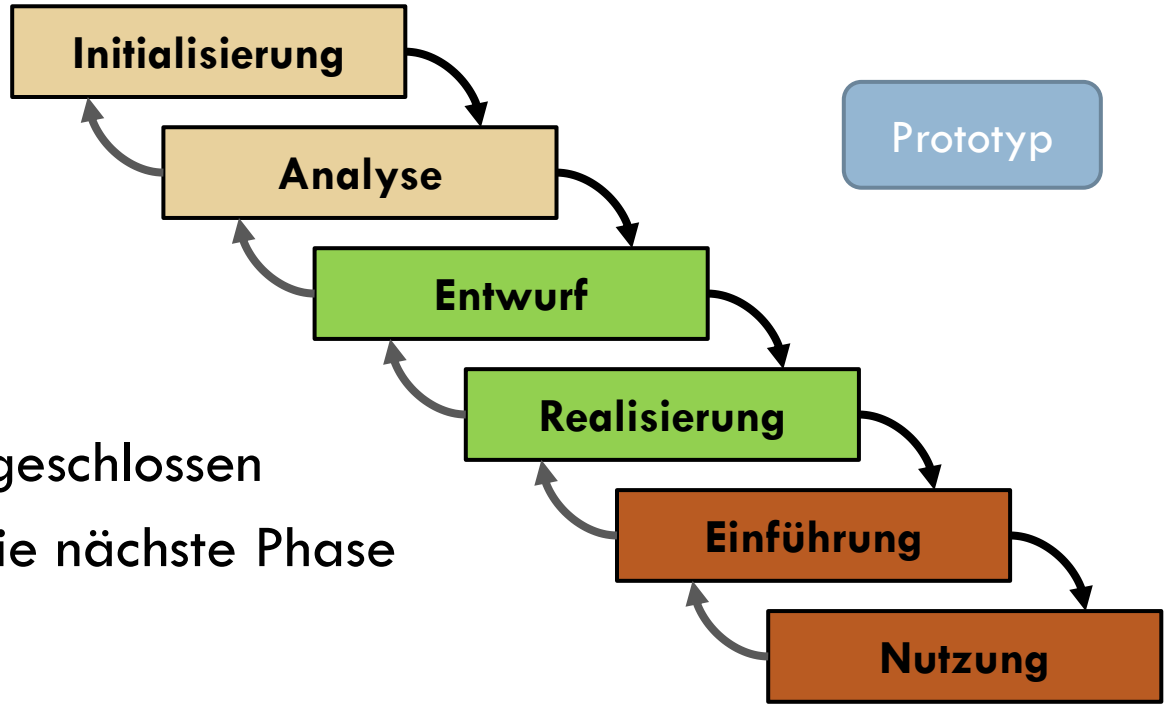
- Funktionalität (functionality)
- Zuverlässigkeit (reliability)
- Benutzbarkeit (useability)
- Effizienz (efficiency)
- Änderbarkeit (maintainability)
- Übertragbarkeit (portability)

Prinzip: Präzise Spezifikation der Anforderungen

Wie entwickelt man ein Programm?

! Nicht „Quick and Dirty“ losprogrammieren !

Vorgehensmodelle: Wasserfallmodell



- Besteht aus Phasen
- Jede Phase wird abgeschlossen
- Anforderungen für die nächste Phase

Initialisierung

Initialisierung – Aufgabenstellung der Hausarbeit

- Vokabeltrainer
- Erweiterung Biegeträger Durchbiegung mit Querschnitten
- Gerätekette (Bagger, Baugrube, LKW)
- Ampel (Kleine Kreuzung)

Mein Beispiel:

„Programmierung der Schnittgrößenermittlung am Einfeldträger“

- U-Wert-Rechner/ Temperatur/Feuchte in der Wand
- Durchflussrechner Geschwindigkeit+Form+Höhe…
- Berechnung eines Überfalls, Darstellung der Kurve
- Absenkungslinie eines Brunnens
- Geometrie
  - Gauss-Flächenberechnung Polygonzug
  - Klothoidenberechnung
  - Berechnung Widerstandsmoment
  - Interpolation von Punkten

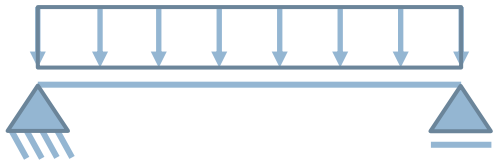
Analyse Lastenheft (Auftraggeber)

Zielbestimmung	Programm zur Berechnung der Schnittgrößen eines Einfeldträgers
Produkteinsatz	Präsentations-Beispiel(e) für Studierende
Produktfunktion	Detailliertere Aspekte des Programms und seines Ablaufs
Produktdaten	Informationen über die Datenhaltung z.B. Speichern, Laden
Benutzeroberfläche, Projektumgebung	GUI, Betriebssystem, Entwicklungsumgebung, Programmiersprache
Produktleistung	Leistungsdaten wie Geschwindigkeit, Stabilität
Ergänzungen	Erweiterungsmöglichkeiten, zu beachtende Normen, Anforderungen an Auftragnehmer u. Projektmanagement

Analyse Plichtenheft (Auftragnehmer)

Zielbestimmung	Programm zur Berechnung der Schnittgrößen eines Einfeldträgers
Produkteinsatz	Präsentations-Beispiel(e) für Studierende
Produktfunktion	Berechnung eines Einfeldträger mit einer Streckenlast, Eingabeparameter: Streckenlast, Länge, Darstellungsparameter Anzeige der Auflagerparameter Darstellung der Schnittgrößen M, Q, (N) Leicht erklärbar (Dokumentation)
Produktdaten	Keine Datenhaltung (Speichern, Laden)
Benutzeroberfläche, Projektumgebung	Windows GUI, Prototyp in Excel, Entwicklung mit VBA
Produktleistung	Keine Anforderung an Geschwindigkeit, hohe Stabilität
Ergänzungen	Erweiterbarkeit des Programms beachten

Entwurf



Zusammenfassung

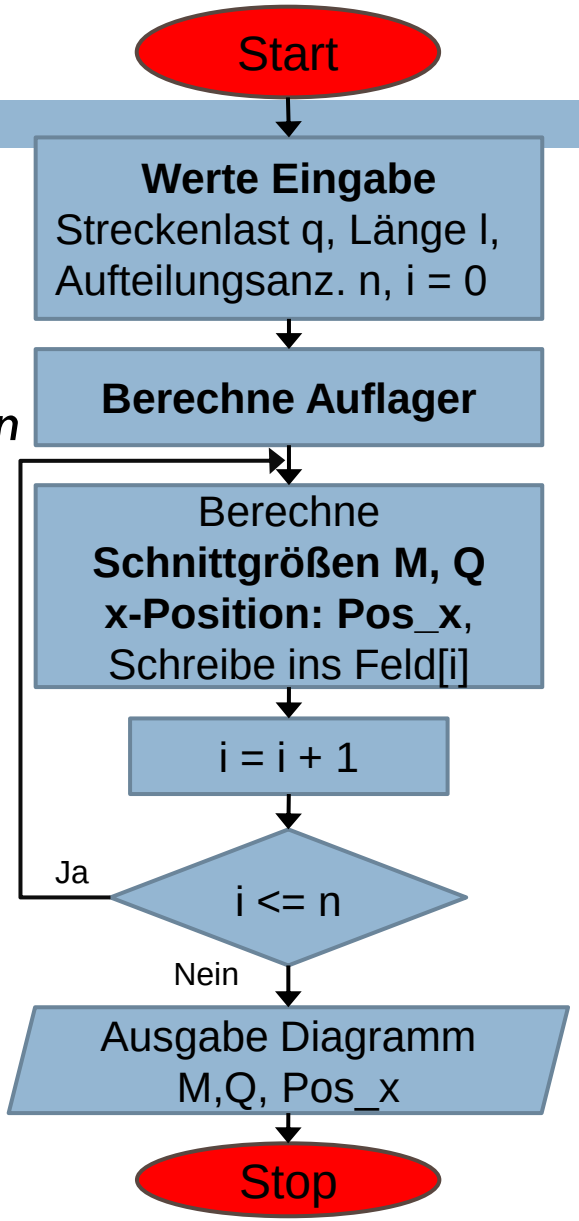
- Eingabe der Werte
  - *Streckenlast q, Länge l, Aufteilungsanz. n*
- Auflagerberechnung

$$A_V = B_V = \frac{1}{2} lq$$

- *Berechnung Schnittgröße*

$$M_x = A_v \cdot x - q \cdot \frac{1}{2} \cdot x^2 \quad Q_x = A_v - qx$$

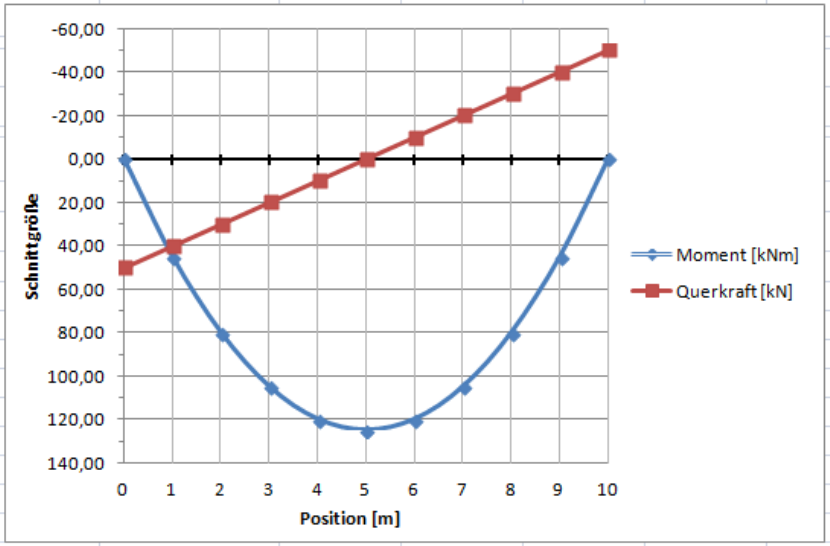
- *Schleife über Rasterpunkte*
- *Ausgabe im Diagramm*



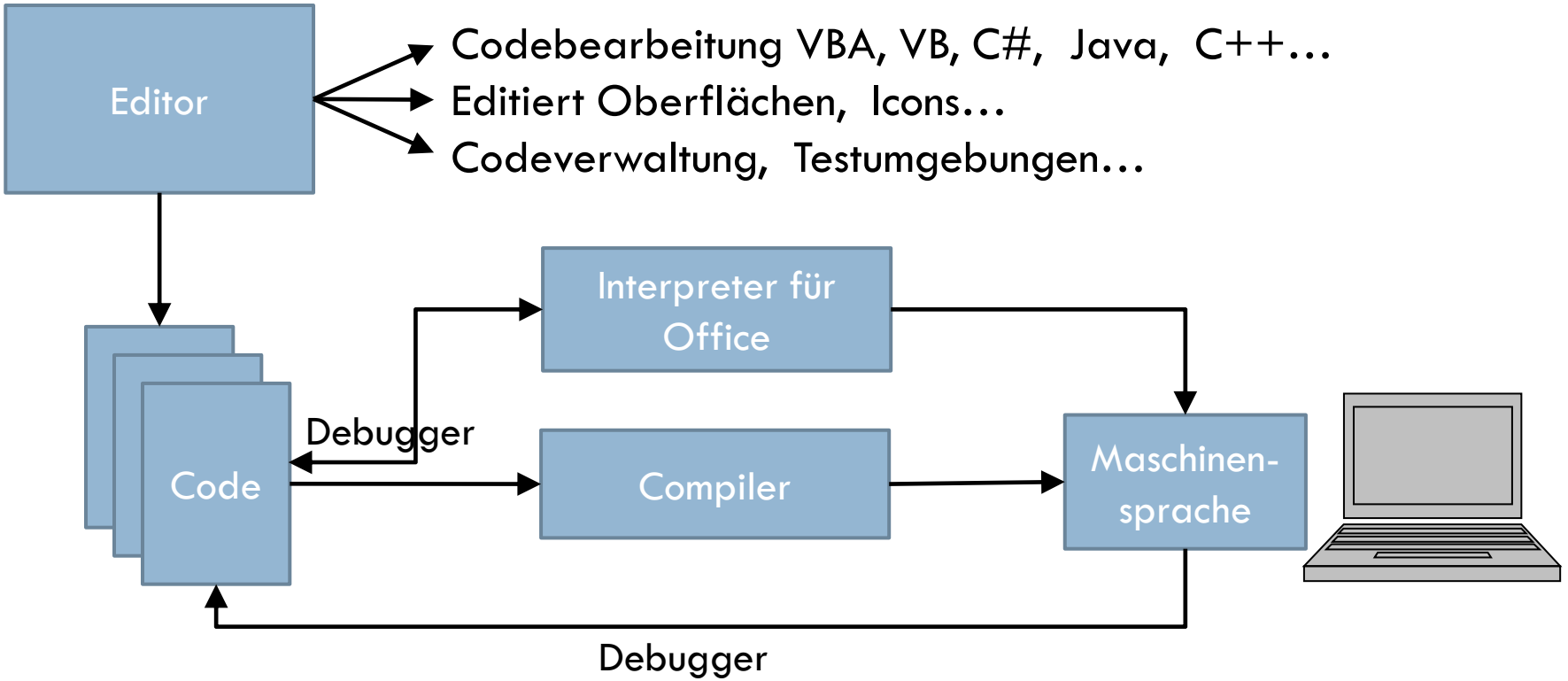
Entwurf Prototyp mit Excel

Realisierung mit VBA …

	B	C	D	E
1	**Eingaben**			
2		Länge l	10,00	m
3		Streckenlast q	10,00	kN/m
5	**Auflagerberechnung:**	$A_v = B_v = 0{,}5 * l * q$		
7		Auflager A_v [kN]	50	= 0,5 * D$2 * D$3
8		Auflager B_v [kN]	50	= 0,5 * D$2 * D$3
10	**Schnittgrößenberechnung**			
11		**Moment [kNm]**		**Querkraft [kN]**
12		$M_x = A_v * x - q * 0{,}5 * x^2$		$Q_x = A_v - q * x$
13	i	[kNm]		[kN]
14	0	0,00	= D$7 * B14 - D$3 * 0,5* B14 * B14	50,00	= D$7 - D$3 * B14
15	1	45,00	= D$7 * B15 - D$3 * 0,5* B15 * B15	40,00	= D$7 - D$3 * B15
16	2	80,00	…	30,00	…
17	3	105,00	…	20,00	…
18	4	120,00	…	10,00	…
19	5	125,00	…	0,00	…
20	6	120,00	…	-10,00	…
21	7	105,00	…	-20,00	…
22	8	80,00	…	-30,00	…
23	9	45,00	…	-40,00	…
24	10	0,00	…	-50,00	…



Realisierung Softwareentwicklung



Realisierung Testen

Testmöglichkeiten

- Auf Codebasis
  - Debugger
  - Reviews
  - Unittests, Modultests
- Auf Programmbasis
  - Integrationstests, Systemtests
- Bei Abschluss
  - Abnahmetest (Validierung), Stresstest

Haftung: Der Hersteller haftet für fahrlässige Fehler im Programm.

Der Ingenieur muss abschätzen, ob die Lösung stimmt.

Einführung und Nutzung

- Einführung
  - Installation beim Kunden (Setup)
  - Lizenz-Verwaltung
- Nutzung
  - Hilfe, Handbücher
  - Schulungen
  - Hotline → Problemlösung
- Wartung und Pflege
  - Stabilisierung / Korrektur
  - Optimierung
  - Änderungen / Erweiterungen