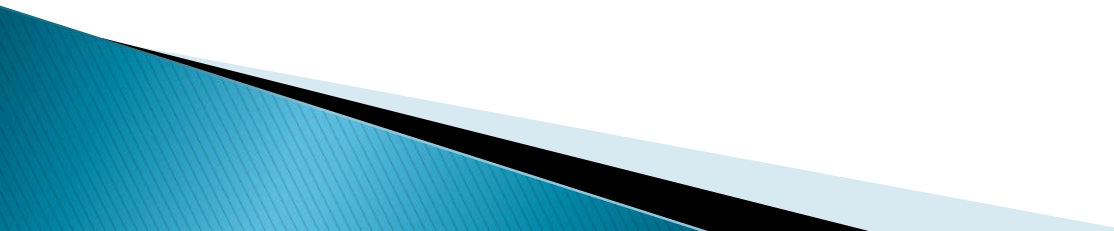


Code-Optimierungen

Thomas Möller

www.Team-Moeller.de

Feststellung:

- ▶ Code wird in der Regel einmal geschrieben und sehr oft gelesen.
 - ▶ Warum geben wird uns dann beim Schreiben des Codes nicht entsprechend Mühe?
- 

Woran erkennt man guten Code?

- ▶ Man hat guten Code vor sich, wenn man merkt, dass derjenige, der den Code erstellt hat, sich damit sehr viel Mühe gemacht hat.

Lesbarkeit

Vittaulin, mit diesem Namen
wurden seit 1865 bis in
19. und 20. Jahrhundert
gebührenden Handschriften
bezeichnet. Die Bezeichnung leitet
sich von dem Namen
Ludwig Vittaulin (1865-1917)
ab, der die vorstehende
Handschrift entwarf. Diese
Entwurf wurde 1924
verbindlich an den preussischen
Landesarchiv und fand seit
1930 in den meisten deutschen
Archiven Anwendung.

Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

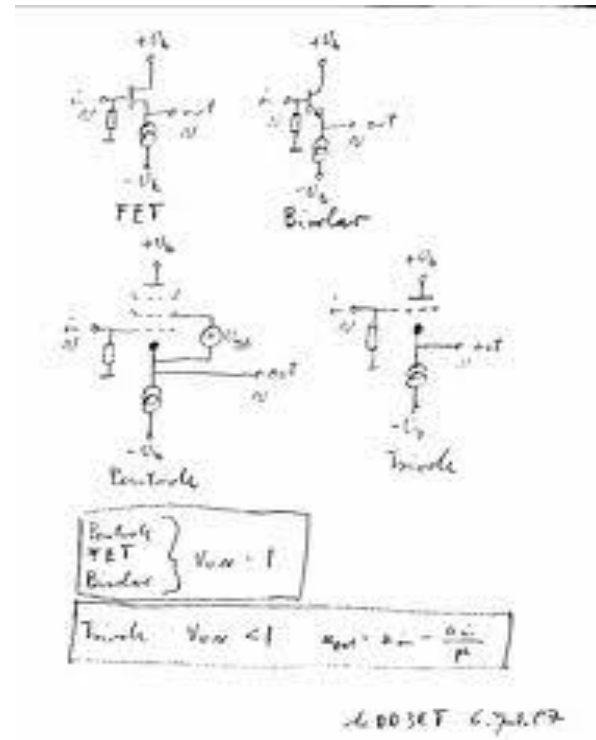
Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

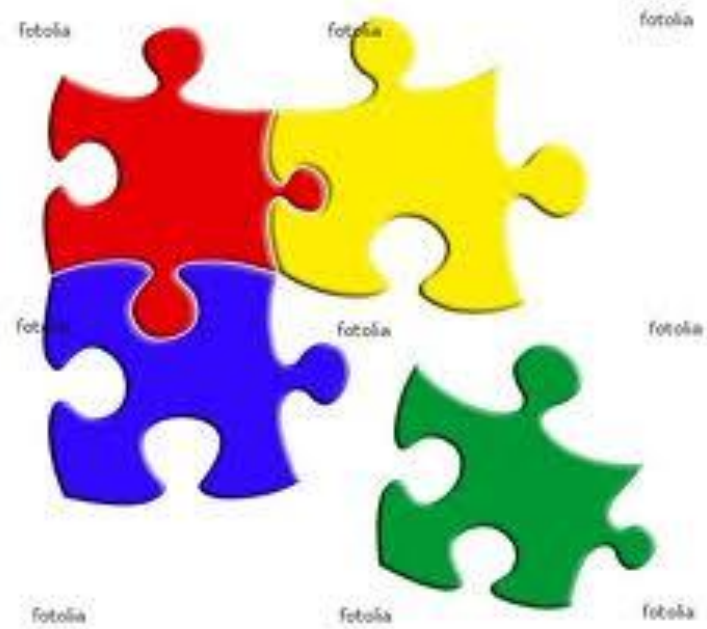
Eine gute Lesbarkeit ist wichtig

Eine gute Lesbarkeit ist wichtig

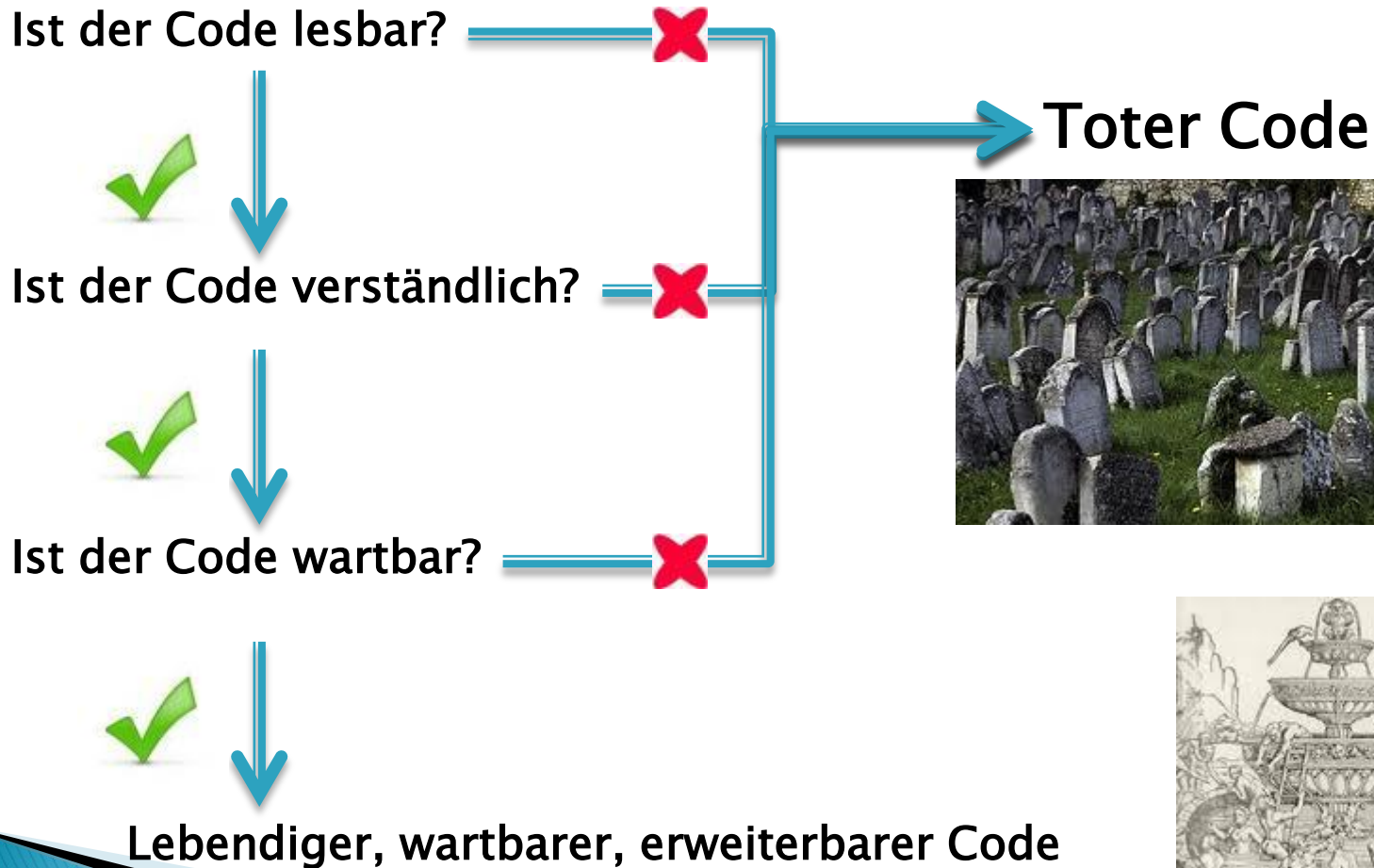
Verständlichkeit



Wartbarkeit / Erweiterbarkeit



Das Ziel: Langlebige Software



Unterstützung durch Tools

- ▶ **Smart-Indenter**

Unterstützt beim automatischen Einrücken des VBA-Codes

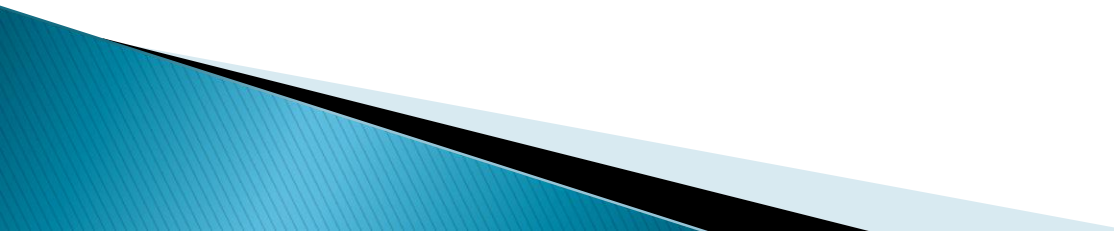
<http://www.oaltd.co.uk/Indenter/Default.htm>

- ▶ **TM VBA-Inspector**

Hilft bei der Suche nach verborgenen Fehlern im VBA-Code

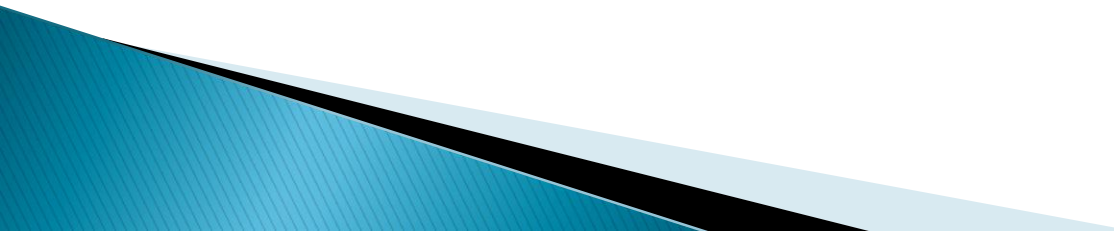
http://www.team-moeller.de/?Add-Ins:TM_VBA-Inspector

Einfache Maßnahmen

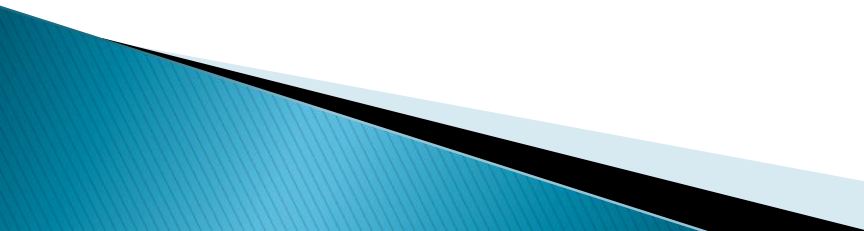
- ▶ Nur ein Befehl pro Zeile
 - ▶ Einrückungen
 - ▶ Leerzeilen zur Strukturierung
- 

Kommentare

- ▶ Nicht zu viele
 - ▶ Nicht zu wenige

 - ▶ Erklären das Vorgehen
 - ▶ Wiederholen nicht den Code
- 

Sinnvolle Benennung von Variablen und Parametern

- ▶ Aussagekräftig, bedeutungsvoll
 - ▶ Keine (unnötigen) Abkürzungen
 - ▶ Klare Aussage, was Variable bzw. Parameter beinhaltet
 - ▶ Schreibweise: CamelCase
 - ▶ Berücksichtigung Namenskonvention
- 

Keine magischen Zahlen

- ▶ Statt dessen Konstante verwenden
- ▶ Name der Konstante gibt der Zahl eine Bedeutung
- ▶ Verwechslung im Code wird vermieden:
Steht die 19 jetzt für den aktuellen MWSt-Satz oder für 19 Zoll?

Enumerationen für Parameter und Select-Case

- ▶ Vermeiden magische Zahlen
- ▶ Zusätzliches Bonbon: Intellisense

Kein DoMenuItem

- ▶ Code unverständlich
- ▶ Besser DoCmd.RunCommand verwenden
- ▶ Übersetzung hier:
<http://www.accessruncommand.com/domenuitem.htm>

Methoden auslagern

- ▶ Codeteile als Methode auslagern
 - ▶ Hält Code übersichtlich
 - ▶ Kleine, handhabbare Einheiten
 - ▶ Eine Aufgabe pro Methode
- 