

Komponenten- und ereignisorientierte Softwareentwicklung am Beispiel von Borland-Delphi

Dr. Henry Herper

Otto-von-Guericke-Universität Magdeburg
Institut für Simulation und Graphik

Lisa-Weiterbildung - Aug. 2002
(angepasst Aug. 2004)

Software – Begriffe und Definitionen

Software: Sammel-Bez. für Programme, die für den Betrieb von Rechensystemen (z. B. Computern) zur Verfügung stehen, einschließlich der dazugehörigen Dokumentation.

/Lexirom 4; Meyers Lexikonverlag/

Software, **Computer**programme. Anweisungen, die dafür sorgen, daß die **Hardware**, also der Rechner, läuft. Software kann anhand der Art der durchzuführenden Aufgaben in verschiedene Kategorien eingeteilt werden. Die Hauptkategorien umfassen einerseits die **Betriebssysteme** (Systemsoftware), die die Arbeiten des Computers steuern, und andererseits die **Anwendungssoftware**, die eine Vielzahl von Funktionen übernimmt, die Menschen von einem Computer erwarten.

...

/"Software," Microsoft® Encarta® 97 Enzyklopädie. © 1993-1996 Microsoft Corporation/

Software – Begriffe und Definitionen

Unter Software ("weiche Ware") versteht man die Gesamtheit aller Programme, die zur Steuerung der Hardware ("harte Ware", Gerätetechnik) eines Computers und zum Lösen vielfältiger Aufgaben und Probleme notwendig sind. Software ist eine im Gegensatz zur Hardware leicht veränderbare, weil nicht stoffliche Komponente des Computers. Man unterscheidet grob in 2 Kategorien:

1. Systemsoftware (das Betriebssystem) zur Steuerung der Hardware, zur Ausführung von Anwendungsprogrammen und zur Erstellung von Programmen (sogenannte nicht-anwendungsbezogene Software),
2. Anwendersoftware wie Textverarbeitungs-, Grafik-, Tabellenkalkulations-, Datenbankprogramme, Computerspiele, etc.

© Bibliographisches Institut & F.A. Brockhaus AG Mannheim und paetec Gesellschaft für Bildung und Technik mbH Berlin. Alle Rechte vorbehalten. www.schuelerlexikon.de

Komponentenbasierte Softwareentwicklung

(Software-) **Komponenten** sind Bausteine zur Erstellung von Applikationen bzw. größerer Komponenten durch einfaches Verschalten. Die Komponenten verfügen über standardisierte Schnittstellen.

Mit komponentenorientierter Softwareentwicklung wird das „**Rapid Application Development**“ unterstützt.

Komponentenbasierte Softwareentwicklung

Zur komponentenorientierten Softwareentwicklung wird ein **Komponentenframework** verwendet. Als Komponentenframework wird eine Bibliothek von Komponenten und die zugehörige Softwarearchitektur bezeichnet. Im Framework werden die Regeln für die Interaktion und Kommunikation der Komponenten definiert.

Komponentenbasierte Softwareentwicklung

Die zugrundeliegende **Komponentenarchitektur** stellt eine gemeinsame Sprache zur Verfügung, mit der zwei Komponenten miteinander kommunizieren.

Das Laufzeitsystem stellt sicher, dass gesendete Nachrichten bei der Komponente ankommen, für die sie bestimmt sind.

Komponentenbasierte Softwareentwicklung

traditionelle Softwareentwicklung

- aus wenigen Sprachelementen werden komplexe Strukturen erstellt (Borland-Pascal7 : 65 reservierte Wörter und Standardbezeichner)
- Applikationen können sehr effektiv implementiert werden, aber sehr großer Aufwand, da alle Elemente neu erstellt werden müssen

komponentenbasierte Softwareentwicklung

- aus vielen komplexen Komponenten werden die geeigneten ausgewählt (Delphi 7: bis zu 300 Standardkomponenten+ca. 70 reservierte Wörter)
- Softwareprodukte werden größer, aber komplexe Applikationen können schneller entwickelt werden

Komponentenbasierte Softwareentwicklung

**wieder verwendbare
Bausteine für Applikationen**

```
graph TD; A[wieder verwendbare Bausteine für Applikationen] --> B[White-Box]; A --> C[Black-Box];
```

White-Box

Komponentenquelltext
steht zur Verfügung
und kann ggf.
verändert werden

Black-Box

es stehen nur die
Schnittstellen zur
Verfügung

Ereignisorientierte Softwareentwicklung

Moderne Anwendungen auf graphischen Oberflächen arbeiten **ereignisorientiert**. **Anwendungen reagieren auf Ereignisse**. Diese Ereignisse können vom Betriebssystem, von anderen Anwendungen oder von der Anwendung selbst erzeugt werden.

Vorgehensweise bei ereignisorientierter Softwareentwicklung

Abweichend von der klassischen Technik der Programmentwicklung ist für die Entwicklung von Delphi-Anwendungen folgende Vorgehensweise zweckmäßig:

- Visueller Entwurf der Bedienoberfläche,
- Zuweisen der Objekteigenschaften und
- Verknüpfen der Objekte mit Ereignissen.

Die Entwicklung wird von der IDE (IntegratedDevelopment Environment) unterstützt.

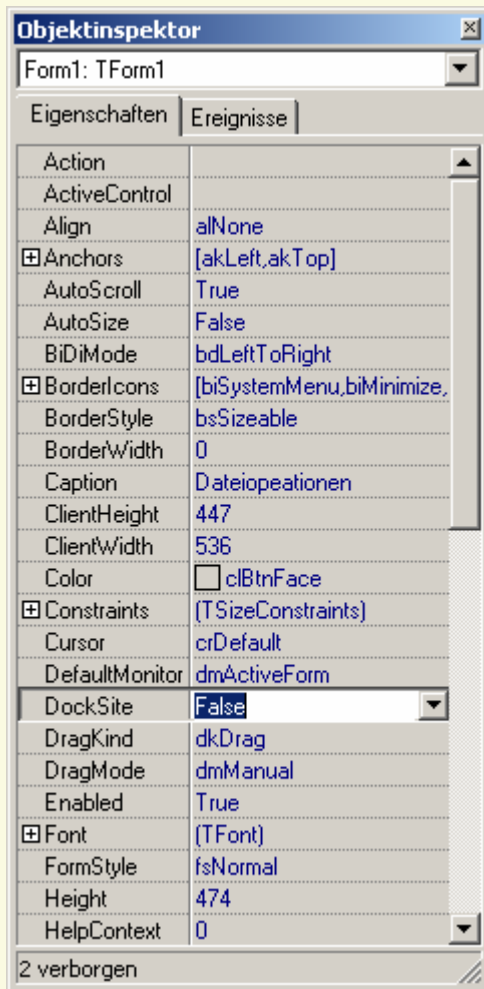
Ereignisorientierte Softwareentwicklung mit Borland Delphi

Ereignistypen

- Mausereignisse
- Tastaturereignisse
- Timer-Ereignisse
- Befehlsereignisse
- Eingabeereignisse
- u.v.m.

Die Ereignisse werden in Borland-Delphi als Events bezeichnet.

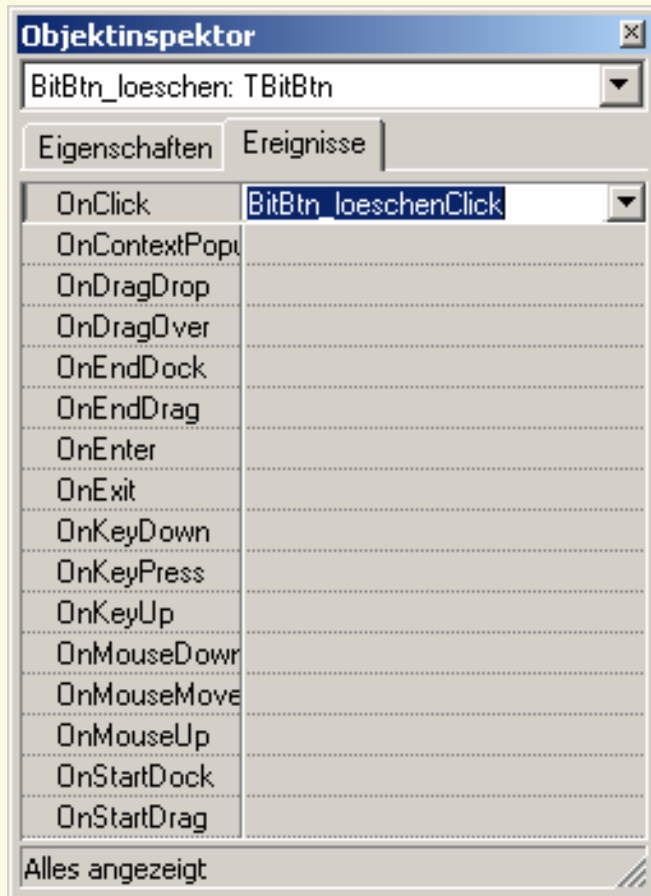
Delphi – Objektinspektor



Mit dem Objektinspektor werden die Eigenschaften des Objektes definiert. Diese Eigenschaften beschreiben die Anfangseigenschaften und können zur Laufzeit verändert werden.

Jede Komponente verfügt über ihre speziellen Eigenschaften.

Delphi – Objektinspektor



Mit dem Objektinspektor werden die für das Objekt auswertbaren Ereignisse festgelegt. Diese Seite wird als Ereignisseite bezeichnet.

Über diese Ereignisse werden Methoden aufgerufen, die die Funktionalität des Programms realisieren. Diese müssen vom Anwender selbst erstellt werden.

Die zur Verfügung stehenden Eigenschaften sind für jede Komponente definiert.

Ereignisbearbeitung in Delphi

Delphi erzeugt zur Bearbeitung von Ereignissen eine Ereignisbehandlungsmethode. Diese wird vom Entwickler mit dem für das Ereignis notwendigen Quelltext versehen. Es besteht die Möglichkeit, den Absender des Ereignisses auszuwerten.

Beispiel: On-Click-Methode für Button 1

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
  
end;
```

Unterstützung des Programmtests

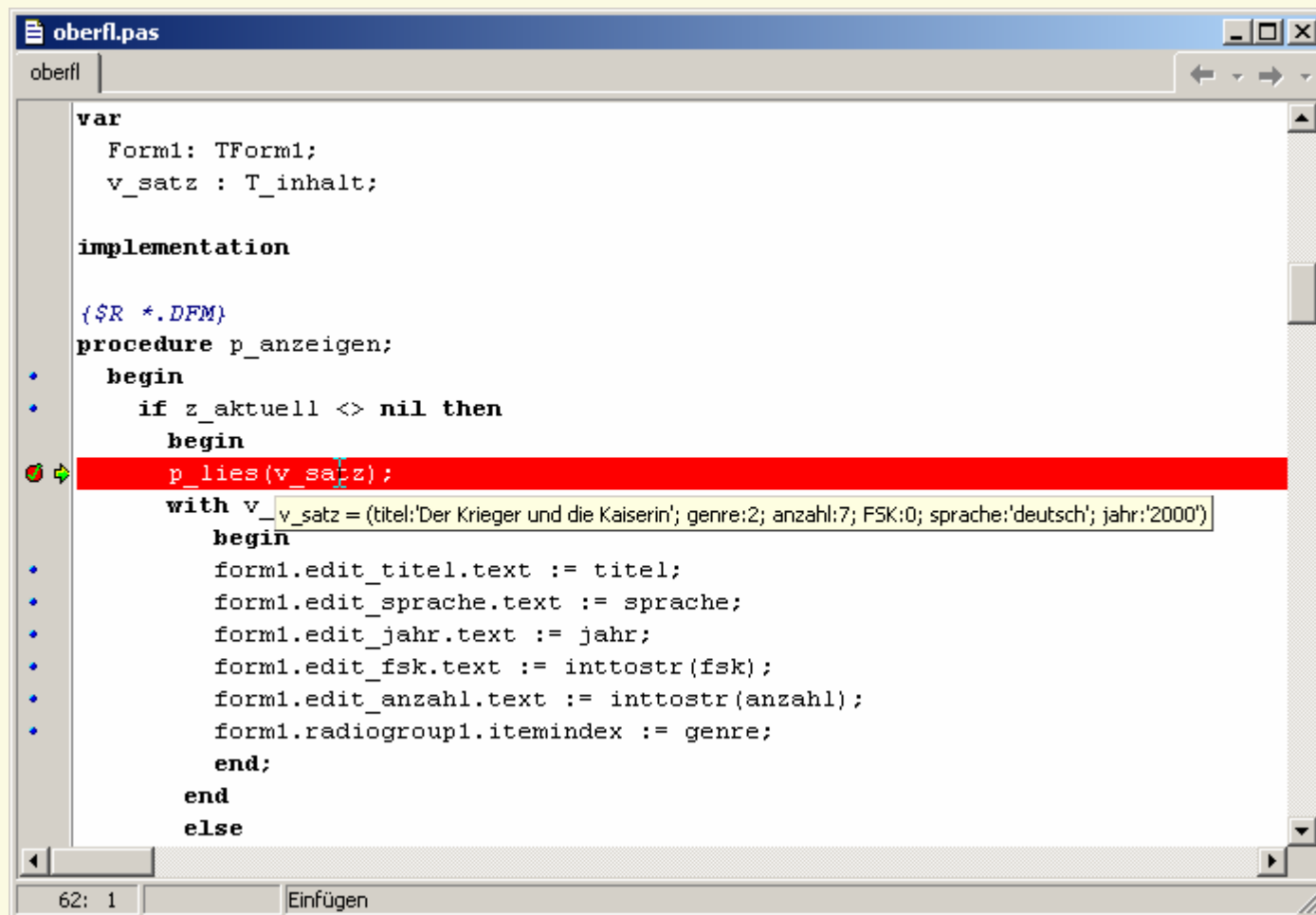
Nutzung des integrierten Delphi – Debugging systems

Möglichkeiten z.B.:

- Setzen von Haltepunkten zur Laufzeit des Programms
- Einzelschrittabarbeitung des Programms
- Überprüfen von Variablenwerten zur Laufzeit
- Kontrolle des Stackinhaltes

Sicherstellen, dass der Quelltext zum abgearbeiteten Programm gehört!

Unterstützung des Programmtests



```
oberfl.pas
oberfl

var
  Form1: TForm1;
  v_satz : T_inhalt;

implementation

{$R *.DFM}
procedure p_anzeigen;
begin
  if z_aktuell <> nil then
  begin
    p_lies(v_satz; z);
    with v_satz = (titel:'Der Krieger und die Kaiserin'; genre:2; anzahl:7; FSK:0; sprache:'deutsch'; jahr:'2000')
    begin
      form1.edit_titel.text := titel;
      form1.edit_sprache.text := sprache;
      form1.edit_jahr.text := jahr;
      form1.edit_fsk.text := inttostr(fsk);
      form1.edit_anzahl.text := inttostr(anzahl);
      form1.radiogroup1.itemindex := genre;
    end;
  end
  else
end
```

62: 1 Einfügen

Unterstützung des Programmtests

The screenshot shows a CPU debugger window titled "CPU" with a thread ID of "#000002A0". The main window displays assembly code for a Pascal program. The current instruction is highlighted in blue: `004440C0 B87C694400 mov eax,$0044697c`. Below it, the code continues with `004440C5 E87EF9FFFF call p_lies`, followed by a function call `004440CA 8D45FC lea eax,[ebp-$04]`, `004440CD BA7C694400 mov edx,$0044697c`, `004440D2 E821FAFBFF call @LStrFromString`, `004440D7 8B55FC mov edx,[ebp-$04]`, `004440DA 8B03 mov eax,[ebx]`, `004440DC 8B80D4020000 mov eax,[eax+$000002d4]`, `004440E2 E8C9F3FDFF call TControl.SetText`, `004440E7 8D45F8 lea eax,[ebp-$08]`, `004440EA BAB5694400 mov edx,$004469b5`, `004440EF E804FAFBFF call @LStrFromString`, and `004440F4 8B55F8 mov edx,[ebp-$08]`. At the bottom, a memory dump shows hex values and their ASCII representations: `00410000 45 08 8B 40 F4 8B 08 FF E.<@ó<.ÿ`, `00410008 51 10 8B D8 8B C3 5E 5B Q.<@<Ã^[`, `00410010 5D C3 8B C0 55 8B EC 53]Ã<ÀU<ìS`, `00410018 56 57 33 C0 55 68 FA 00 VW3ÀUhú.`, `00410020 41 00 64 FF 30 64 89 20 A.dw0dz`.

On the right side, a register window shows the state of various registers. The EAX register is highlighted with a value of 00446820. Other registers include EBX (00446978), ECX (00000000), EDX (00446820), ESI (0012F75C), EDI (0012F75C), EBP (0012F5E0), ESP (0012F5BC), EIP (004440C0), EFL (00000202), CS (001B), DS (0023), SS (0023), and ES (0023). Below the register window, a memory dump shows a list of memory addresses and their contents, with the address 0012F5BC highlighted.

Unterstützung des Programmtests

