Xcode/Cocoa/Objective-C Crashkurs

Programmieren unter Mac OS X

SwissMacMeeting #1
26. Juni 2004
Messeturm Basel

http://mac.naepflin.com

Was ist das Ziel dieses Kurses?

- Starthilfe
- Einblick in die Möglichkeiten, die Apples Developer Tools bieten
- Beispielapplikation
- Verweise auf weiterführende Quellen

Ablauf

- Was ist Xcode?
  - Was sind die Developer Tools?
  - Was ist Cocoa?
  - Was ist Objective-C?
  - Grundkurs Objektorientiertes Programmieren

Ablauf

- Ein Programmbeispiel
- Wie weiter?

Was ist Xcode?

- Ein Programm, das in den Developer Tools enthalten ist
- “Xcode” im Titel dieser Präsentation meint eigentlich die ganzen Developer Tools

Was sind die Developer Tools?

- Sammlung verschiedener Tools zur Erzeugung, zum Test und zur Fehlerbehebung von Applikationen
- Bei Mac OS X mitgeliefert

Was sind die Developer Tools?

- Die wichtigsten dieser Tools:

Interface Builder

  - Erzeugung des Userinterfaces (Nib-Files)

Xcode

  - Bearbeiten und Kompilieren des Codes
  - Zusammenfügen der einzelnen Teile der zu erzeugenden Applikation

Was ist Cocoa?

- Cocoa ist die Standard-Programmierumgebung unter Mac OS X

Was ist Objective-C?

- Objective-C ist die Standard-Programmiersprache unter Cocoa
- Einfach zu erlernen
- Objektorientiert
- Baut auf C auf (jeder C-Code funktioniert auch in Objective-C)

Grundkurs: Objektorientiertes Programmieren (OOP)

- Wichtig im OOP sind Klassen und Objekte
  - Klasse: “abstrakter Oberbegriff für die Beschreibung der gemeinsamen Struktur [...] von Objekten”
  - Objekt: “eine konkrete Ausprägung einer Klasse”; auch Instanz genannt

Quelle: http://de.wikipedia.org

Beispiel

- Wir definieren die Klasse “Quader”

Grundkurs: Objektorientiertes Programmieren (OOP)

- Klassen haben verschiedene Eigenschaften:

1) Sie stellen Variablen zur Speicherung von Daten zur Verfügung

Beispiel

- “Quader” speichert in den Variablen "Hoehe", "Breite" und "Laenge" verschiedene Zahlen

Grundkurs: Objektorientiertes Programmieren (OOP)

2) Klassen stellen Methoden zur Verfügung

- Methoden sind Funktionen der Klasse, die man abrufen kann. Einige Methoden verlangen Übergabeparameter, und einige geben Variablen zurück

Beispiel

- Die Klasse “Quader” soll vier Methoden bereit stellen:
- Die Methoden "HoeheAendern", "BreiteAendern" und "LaengeAendern" verlangen einen Übergabeparameter, der in die entsprechenden Variablen gespeichert wird

Beispiel

- Die Methode “Volumen” soll “Hoehe”, “Breite” und “Laenge” multiplizieren und das Produkt zurückgeben

Grundkurs: Objektorientiertes Programmieren (OOP)

- Nun besteht aber noch kein konkretes Objekt, sondern erst eine Definition davon

Beispiel

- Ein Objekt der Klasse “Quader” ist nun dadurch definiert, dass es eine “Hoehe”, “Breite” und “Laenge” hat, die man verändern kann.
Zudem kann man mittels “Volumen” das Produkt dieser Variablen erfahren

Grundkurs: Objektorientiertes Programmieren (OOP)

- Um aus der Klasse nun etwas Konkretes zu gewinnen, muss man ein Objekt davon erzeugen

Beispiel

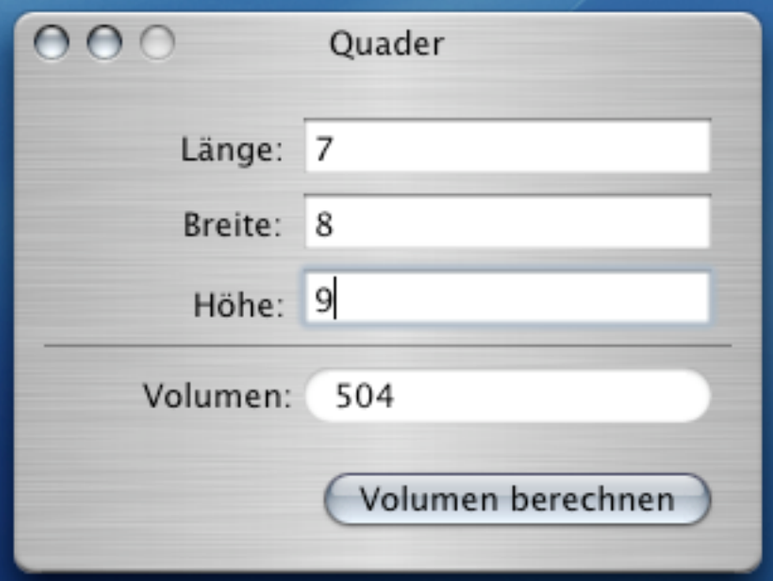
- Nun wird ein Objekt der Klasse “Quader” gebildet, welches z.B. “meinQuader” heisst. In “meinQuader” können nun “Hoehe”, “Breite” und “Laenge” belegt werden
- Dies wäre in “Quader” nicht möglich, da “Quader” nur eine Definition ist

Beispiel (Vergleich zur Realität)

- So könnte man z.B. “Bello” als Objekt der Klasse “Hund” bezeichnen

Ein Programmbeispiel

- Endprodukt:



Ein Programmbeispiel

Projekt erstellen:

- Menü File - New Project...
- Cocoa-Application
- Project Name: “Quader”
- Finish

Ein Programmbeispiel

MainMenu.nib anpassen:

- MainMenu.nib doppelklicken
- Fenster gemäss der Vorlage modellieren

Ein Programmbeispiel

- Klasse Controller definieren:
  - Fenster “MainMenu.nib” - Classes - NSObject wählen
  - Menü Classes - Subclass NSObject
  - Name: Controller

Ein Programmbeispiel

- Klasse Controller definieren:
  - Menü Tools - Show Info
  - neue Action: “VolumenBerechnen:”
  - neue Outlets: “BreiteFeld”, “LaengeFeld”, “HoeheFeld” und “VolumenFeld”

Ein Programmbeispiel

- Dokumente der Klasse Controller erstellen:
  - Menü Classes - Create Files for Controller
  - Choose

Ein Programmbeispiel

- Objekt der Klasse Controller erstellen:
  - Menü Classes - Instantiate Controller

Ein Programmbeispiel

- Outlets verbinden:
  - ctrl-Klick auf Controller Objekt und nach “Länge”, ”Breite”,“Höhe” ziehen
  - Zugehöriger Outlet-Name doppelklicken

Ein Programmbeispiel

Actions verbinden:

- ctrl-Klick auf den Button und nach Controller Objekt ziehen
- “VolumenBerechnen:” doppelklicken

Ein Programmbeispiel

- Klasse “Quader” erstellen:
  - Menü File - New File...
  - Objective-C Class
  - File Name: “Quader.m”
  - Finish

Ein Programmbeispiel

Der Code

- Den Code aus der Dokumentation kopieren und in die Files einfügen

Ein Programmbeispiel

Kompilieren und Testen:

- Menü Build - Build and Run

Wie weiter?

- http://developer.apple.com
- http://cocoadevcentral.com
- Tutorial: siehe Links Ordner
- http://mac.naepflin.com
- Tipp: Docoa Browser

Viel Spass!

Xcode/Cocoa/Objective-C Crashkurs

Programmieren unter Mac OS X

SwissMacMeeting #1
26. Juni 2004
Messeturm Basel

http://mac.naepflin.com