

Übersicht des Moduls

Vorlesungen:

- Freitag, 07.30 – 09.00 Uhr, FOR-0270
- Freitag, 14.00 – 15.30 Uhr, FOR-0270



Übungen:

- 4 Übungsgruppen mit je 2 Terminen wöchentlich
- Start in der dritten Woche (16. KW)

- Prüfung: Klausur, 120 min
- Organisation: Opal → **Bitte einschreiben!**



Vorstellung

Dipl.-Math. Martin Reinhardt

martin.reinhardt@tu-freiberg.de

+49 3731 / 39 – 3114

Humboldt Bau, Zimmer 115



Ziele des Moduls

- Verstehen verschiedener Programmierparadigmen
- Beherrschen von prozeduraler und objektorientierter Programmierung in C# .NET Core
- Einführung in die Werkzeuge der Softwareentwicklung
- Nutzung moderner und aktueller Programmier Techniken
- Erstellen von plattformunabhängigen grafischen Oberflächen

Was ist Softwareentwicklung?

Unter dem Begriff Softwareentwicklung versteht man die Konzeption und standardisierte Umsetzung von Softwareprojekten und die damit verbundenen Prozesse.

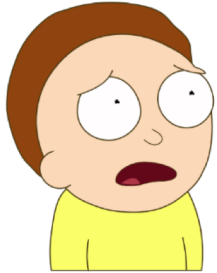
Gemäß ISO 9126 gibt es die sechs folgenden Qualitätsmerkmale für Softwareprodukte:

- *Funktionalität*
- *Zuverlässigkeit*
- *Benutzbarkeit*
- *Effizienz*
- *Wartungsfreundlichkeit*
- *Übertragbarkeit.*

Nachfolger ISO 25010: Zusätzlich

- Kompatibilität
- Sicherheit

Die Norm kann als eine Art Checkliste verstanden werden.



Warum sind diese Normen so wichtig? Warum eine Checkliste?

iPhones stürzen ab – iOS 11.2 soll helfen

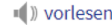
02.12.2017 10:36 Uhr – Leo Becker



Security > 7-Tage-News > 03/2018 > Ernstzunehmende Lücken im LTE-Protokoll aufgedeckt, kaum Hoffnung auf...

Erstzunehmende Lücken im LTE-Protokoll aufgedeckt, kaum Hoffnung auf Absicherung

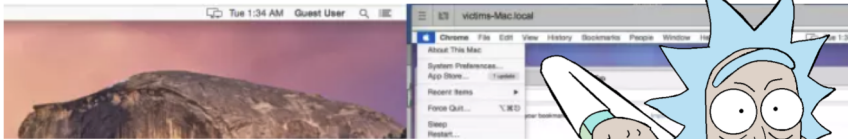
06.03.2018 06:59 Uhr – Fabian A. Scherschel



Mac & i > News > 03/2018 > Schwachstelle in Chrome RDP für macOS: Gast kann vollen...

Schwachstelle in Chrome RDP für macOS: Gast kann vollen Remote-Zugriff erhalten

15.03.2018 18:14 Uhr – Leo Becker



heise online > News > 09/2017 > Lufthansa, Qantas, British Airways ...: Technisches Problem sorgt für...

Lufthansa, Qantas, British Airways ...: Technisches Problem sorgt für Verzögerungen

28.09.2017 16:00 Uhr – Martin Holland



Mac & i > News > 03/2018 > Datenschutzpanne: Siri plaudert versteckte Nachrichten aus

Datenschutzpanne: Siri plaudert versteckte Nachrichten aus

UPDATE

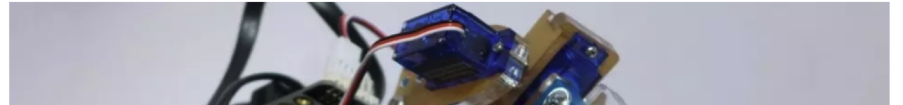
21.03.2018 18:04 Uhr – Leo Becker



Mittwoch 21. März

Autonomes Fahrzeug fordert erstes Menschenopfer

22. März 2018 – Gunter Laßmann





1. Funktionalität - Functional-Suitability
2. Effizienz - Performance-Efficiency
3. Kompatibilität - Compatibility
4. Benutzbarkeit - Usability



4. Zuverlässigkeit - Reliability

5. Sicherheit - Security

6. Wartbarkeit - Maintainability

7. Portabilität - Portability



Werkzeuge in der Softwareentwicklung

- Programmiersprache und Compiler
- Editor, IDE
- Build-System
- Debugger
- Projektverwaltung
- Dokumentation
- Code-Base Management und Versionierung
- Deployment
- Support
- **...Ihren Kopf !**
- Softskills und Koffein



Programmierparadigma

- Beschreibt den fundamentalen Programmierstil bzw. Eigenschaften von Programmiersprachen
- Unterscheidung durch die Repräsentation der statischen und dynamischen Programmelemente
- Paradigmen und Stile sind nicht komplementär
- Paradigma beruht auf Sprache, aber auch auf individuellem Stil.

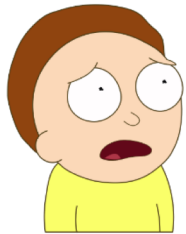


Die wichtigsten Paradigmen sind:

- Imperativ
- Strukturiert
- Prozedural
- Modular
- Deklarativ
- Funktional
- Logisch
- Objektorientiert
- ...

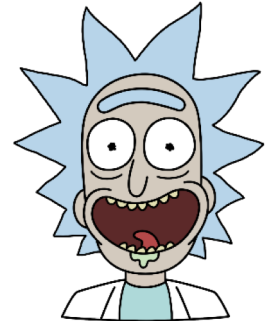
Imperative Programmierung

Quellcode besteht aus einer Folge von Befehlen die in einer festen Reihenfolge abgearbeitet werden.



Ist nicht jede Sprache eine imperative Sprache?

Richtig! Die meisten Programmiersprachen unterstützen die imperative Programmierung!
Aber denke doch mal an die SQL Abfragesprache.



Beispiele:

ALGOL	C, C++, C#	AVR-C
Assembler	Pascal	COBOL
FORTRAN	JAVA	BASIC

Strukturierte Programmierung

- Weiterentwicklung des imperativen Paradigmas
- Verzicht bzw. starke Einschränkung des Goto-Statements
- Nutzung von Kontrollstrukturen:
 - If...then...else
 - While...Do
 - Repeat...until...

Wegweiser: Niklaus Wirth, Erfinder von **PASCAL** (1968-1972)

„Software expands to fill the available memory.“

“Software is getting slower more rapidly than hardware becomes faster.”



Prozedurale Programmierung

- Zerlegung von Programmen in überschaubare Teile, die durch eine definierte Schnittstelle aufrufbar sind
- Funktion, Prozedur, Routine, Unterprogramm
- Bilden eine wesentliche Abgrenzung zwischen Assemblersprachen und Hochsprachen
- Weiterentwicklung: Modulare Programmierung

Die meisten modernen Sprachen **zum Schreiben von Programmen** unterstützen die Paradigmen **imperativ, strukturiert, prozedural und modular**.

Viele der Konzepte sind erst später in bestehende Programmiersprachen eingeflossen.



Deklarative Programmierung

- Nicht das „Wie“ ist wichtig, sondern das „Was“
- Es wird kein Lösungsweg implementiert, sondern nur angegeben, was gesucht ist.
- Unterkategorien:
 - Funktionale Sprachen (LISP, Miranda, Haskell, F#)
 - Logische Sprachen (Prolog)
 - Mengen-orientierte Abfragesprachen (SQL und andere Datenbank-Sprachen)
 - Aufbausprachen (make, ANT)



Objektorientierte Programmierung (OOP)

Ideen der OOP:

- Objekte der realen Welt müssen sich in der Programmierung widerspiegeln
- Es geht nicht um das Manipulieren von Daten sondern um Zustandsänderungen von Objekten
- Im Zentrum der objektorientierten Programmierung stehen Objekte, die mittels Nachrichten miteinander kommunizieren

Alan Kay:

1. Alles ist ein Objekt
2. Objekte kommunizieren durch das Senden und Empfangen von Nachrichten (welche aus Objekten bestehen)
3. Objekte haben ihren eigenen Speicher (strukturiert als Objekte)
4. Jedes Objekt ist die Instanz einer Klasse (welche ein Objekt sein muss)
5. Die Klasse beinhaltet das Verhalten aller ihrer Instanzen (in der Form von Objekten in einer Programmliste)
6. Um eine Programmliste auszuführen, wird die Ausführungskontrolle dem ersten Objekt gegeben und das Verbleibende als dessen Nachricht behandelt

- Keine genaue Definition vorhanden
- OOP stellt die reale Welt und deren Objekte im Computer dar:
 - Objekte haben eine Identität
 - Objekte haben einen Zustand
 - Objekte haben ein Verhalten
- Gleichartige Objekte können zu einer Klasse von Objekten zusammengefasst werden
- *„Die Klasse bildet den Bauplan – das Objekt ist der fertige Bau“*



Die drei Säulen der OOP

- Kapselung
- Vererbung
- Polymorphie

Beispiel: Auto

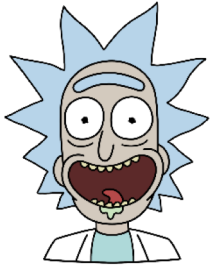
- Kapselung:
 - Ich kann nur lesen, welchen Kraftstoff ich benutzen muss
 - Ich kann nicht auf interne Details (Zündzeitpunkt des 3. Zylinder) zugreifen, damit das Auto weiterfährt
- Vererbung:
 - Kleinwagen und Kombis sind spezielle Arten von Autos
 - Ich muss das Auto nicht neu erfinden
- Polymorphie:
 - Ein Maserati Quattroporte und ein C Corsa können beide fahren. Trotzdem sind die Auswirkungen verschieden

Beispiel: Säugetier

- Kapselung:
 - Die genauen Vorgänge der Verdauung interessieren nicht. Nur das benötigte Futter ist wichtig
- Vererbung:
 - Eine Springmaus und eine Hausratte sind beide aus der Ordnung der Nagetiere
 - Beide Tiere teilen viele gleiche Eigenschaften
- Polymorphie:
 - Fortbewegen ist eine Eigenschaft jedes Säugetieres. Ein Wal schwimmt jedoch und ein Känguru hüpfte. Ein Pferd galoppiert.

Warum C#

Warum sollen wir C# lernen? Ich finde <generische OOP-Sprache> viel besser.



C#

- ist plattformunabhängig
- bietet eine breite Sammlung von Bibliotheken
- ist eine moderne und durchdachte Sprache
- ist eine gefragte Programmiersprache
- enthält Elemente vieler verschiedener Paradigmen
- kann mit anderen Sprachen über .NET interagieren
- besitzt ein Memory-Management
- wird durch eine IDE mit vielen Features unterstützt
- unterstützt Multi-Processing problemlos



Beispiele von OOP Sprachen

Smalltalk	C++	JAVA	Kotlin
Python	Objective-C	Ruby	PHP
Oberon	Turbo Pascal	MATLAB	Julia
Rust	Swift	Visual Basic .NET	JavaScript
Go	Delphi	COBOL	Lua
R	Modula-3

Und: ... C# .NET

Zusammenfassung

- Grober Fahrplan der Vorlesung
- Definition des Begriffes Softwareentwicklung
- Programmierparadigmen
- Objektorientierung



Alan Kay:
“The best way to predict the future is to invent it.”