

Objektorientierte Programmierung 2

Programmierprojekt

Yves Senn

Beispiel

Programmierersprachen

Suchbegriff

Haskell

Simon Peyton Jones,Paul Hudak,Philip Wadler

INTERCAL

Donald R. Woods,James M. Lyon

JavaScript

Brendan Eich

Kotlin

JetBrains

Luna

Richard Gordon Faika

Miranda

David Turner

NewtonScript

Walter Smith

Objective-C

Brad Cox,Tom Love

Oberon

Niklaus Wirth,Jürg Gutknecht

PHP

Rasmus Lerdorf

Python

Guido van Rossum

QML

Ot Project

Programmierersprachen Infos

Name

Ada

Entwicklungsjahr

1980

Entwickler

Jean Ichbiah

Typisierung

statisch,explizit,stark

Paradigmen

imperativ,strukturiert,objektorientiert

Stackoverflow-Tags

1,237

Popularität Stackoverflow

Language	Popularity (Estimated)
JavaScript	22%
Python	18%
PHP	12%
Objective-C	8%
Ruby	5%
TypeScript	3%
C	2%
C++	10%
Delphi	1%
C#	15%

Programmierprojekt Basis

- **Einlesen der Daten von File**
- **Abspeichern der Änderungen**
- **Darstellen aller Programmiersprachen in Tabelle / Liste**
- **Editor-Bereich**
 - Editor-Bereich arbeitet stets auf dem in der Tabelle selektierten Eintrag
 - Änderungen führen zu unmittelbarem Update der Tabelle und der Überschriften
 - Änderung von 'StackoverflowTags' und 'Paradigmen' berechnen die Werte in Tabellen oder Diagrammen neu
- **Header-Bereich mit Titel und Name der Programmiersprache**
- **Footer-Bereich mit Auswertungen zu den Kategorien und Tags (Diagramm, Tabelle, oder ähnliches)**
- **Informationen werden immer aktuell gehalten**
- **Layout mit SplitPane inklusive sinnvollem Resizing-Verhalten**
- **Anlegen einer neuen 'Programmierprache'**
- **Löschen bestehender Einträge**
- **Alle Basis-Features implementiert: mindestens 4.0**
- **Nicht alle Basis-Features implementiert: im besten Fall 4.0**

Features

- **Bewährte, Programmstruktur - Einsatz des PresentationModel**
 - klare Trennung von "Was" und "Wie"
 - unabhängige UI-Elemente
- **Undo/Redo**
 - Tooltip auf Undo/Redo Button
- **Animationen**
 - beim Aufstarten (Daten werden geladen)
 - beim Wechsel der Programmiersprache
- **Überarbeitung des UI**
 - eigener Style, andere Übersicht als Tabelle, ...
- **neue Funktionen**
 - Input-Validierung
 - Mehrsprachigkeit

Bewertung

- **Basis-Features**
 - “Make it run” 4.0
 - “Make it right” 5.0
- **Jedes Feature: +1**
 - Features müssen „verhandelt“ werden
- **Gruppenarbeit: -1**
- **Präsentation der Lösung, ca.15 min. +1**

Zusatzregeln

- **Teamgrösse: max. 2 Personen**
- **Maximale Differenz zwischen Erfahrungsnote und MSP: 3.0**
 - bei grösserer Abweichung wird MSP mit 1.0 bewertet
- **Bei Zweifel an selbstständiger Arbeit:**
 - gemeinsame Code-Walkthrough Session
 - detaillierte Fragen zur gewählten Lösung
- **Bei Präsentationen: Alle Teammitglieder stellen einen Teil der Lösung vor**
 - nur die Demo machen reicht nicht

Beispiel

- **Basis-Features “gut” programmiert**
 - klare Programm-Struktur
 - gute Erweiterbarkeit um weitere Features
 - angemessene Datenstrukturen
 - sinnvolle Design Pattern
- **1 Feature, z.B. undo/redo**
- **2-er Team**
- **Präsentation**
- **$5.0 + 1 - 1 + 1 = 6.0$**

Organisatorisches

- **Abgabetermin: 13.06.2017**
- **Anmeldung Präsentation bis: 31.05.2017**
- **Projektname = 'VornameNachname-VornameNachname'**
- **Entweder als zip-File oder via Source-Code-Repository**
- **Startbar aus IDE**
- **Bewertung Modul**
 - 50% Erfahrungsnote
 - 25% *Zwischenprüfung*
 - **75% Projekt**
 - 50% Modulabschlussprüfung

