

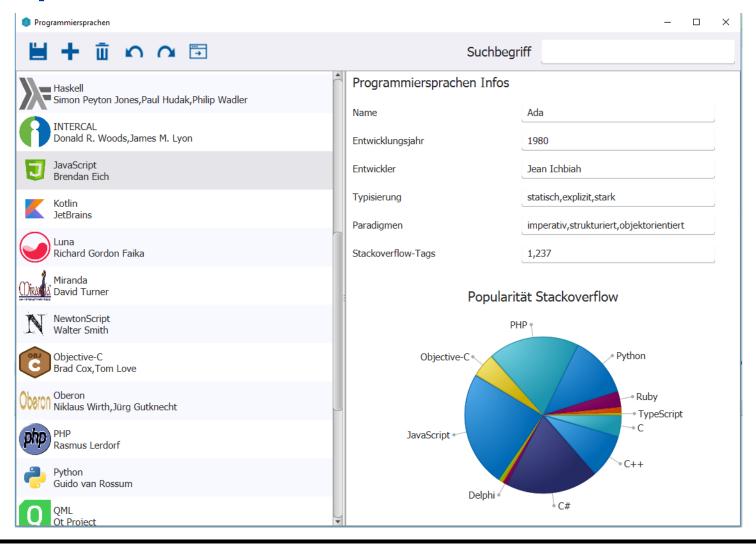
## **Objektorientierte Programmierung 2**

**Programmierprojekt** 

Yves Senn



## **Beispiel**





### **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 'Programmiersprache'
- Löschen bestehender Einträge
- Alle Basis-Features implemenVert: mindestens 4.0
- Nicht alle Basis-Features implemenVert: im besten Fall 4.0



#### **Features**

- Bewährte, Programmstruktur Einsatz des PresentationModel
  - klare Trennung von "Was" und "Wie"
  - unabhängige UI-Elemente
- Undo/Redo
  - Tooltipp 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**

			•			4		
	H	asi	16-	<b>⊢</b> ∠	2	tu	ır۵	C
_	u	ası	13-	, ,	5 CI	LU	пс	J

	→ "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

