

aus

Oliver J. Bott, Peter Fricke, Uta Priss, Michael Striewe (Hrsg.)

Automatisierte Bewertung in der Programmierausbildung

Digitale Medien in der Hochschullehre Band 6

2017, 420 Seiten, br., 42,90 €, ISBN 978-3-8309-3606-0



Waxmann Verlag GmbH

www.waxmann.com info@waxmann.com

17 Steckbriefe und Featurematrix der Grader

JACK Grader-Steckbrief (siehe Kapitel 9)

Entwicklung seit:	2006
Aktueller Hauptentwickler:	Universität Duisburg-Essen Arbeitsgruppe Prof. Dr. Michael Goedicke
Kontaktmöglichkeit:	jack@paluno.uni-due.de
Online-Demo:	https://jack-demo.s3.uni-due.de/

Zielgruppe:	Veranstalter und Teilnehmer einführender und fortgeschrittener Lehrveranstaltungen zur objektorientierten Programmierung
--------------------	--

Praktomat Grader-Steckbrief (siehe Kapitel 10)

Entwicklung seit:	1999
Aktueller Hauptentwickler:	KIT Lehrstuhl Prof. Dr. Gregor Snelting
Kontaktmöglichkeit:	https://www.lists.kit.edu/sympa/info/praktomat-users
OpenSource:	https://github.com/KITPraktomatTeam/Praktomat

Zielgruppe:	Vor allem Programmierneinsteiger, aber auch fortgeschrittene Vorlesungen (Isabelle-Praktikum, Financial Economics)
--------------------	--

Graja Grader-Steckbrief (siehe Kapitel 11)

Entwicklung seit:	2012
Aktueller Hauptentwickler:	Hochschule Hannover Prof. Dr. Robert Garmann
Kontaktmöglichkeit:	robert.garmann@hs-hannover.de
Zielgruppe:	Einführende und fortgeschrittene Programmierung mit Lerninhalten von prozeduraler Programmierung über objektorientierte Konzepte bis hin zu Threads und Grafik

aSQLg Grader-Steckbrief (siehe Kapitel 12)

Entwicklung seit:	2010
Aktueller Hauptentwickler:	Hochschule Hannover Prof. Dr. Carsten Kleiner, Prof. Dr. Felix Heine
Zielgruppe:	Studierende in Datenbankvorlesungen mit SQL-Anteil, sowohl Anfänger als auch Fortgeschrittene

GATE Grader-Steckbrief (siehe Kapitel 13)

Entwicklung seit:	2009
Aktueller Hauptentwickler:	HU Berlin (Sven Strickroth) TU Clausthal (Oliver Müller)
Kontaktmöglichkeit:	svен@strickroth.net, oliver.mueller@tu-clausthal.de
OpenSource:	https://github.com/csware/si
Zielgruppe:	Java-Programmierübungen, Tutorinnen und Tutoren in Java-Programmierübungen mit mehreren hundert Studierenden, Studierende insbesondere in Einführungsveranstaltungen zur Java-Programmierung

VEA Grader-Steckbrief (siehe Kapitel 14)

Entwicklung seit:	2000
Aktueller Hauptentwickler:	Universität Osnabrück, Zentrum für Informationsmanagement und virtuelle Lehre
Kontaktmöglichkeit:	Tobias Thelen https://mvc.ikw.uni-osnabrueck.de/vips/vips.php?adm=1&mgc=!ViPS!&srv=prolog
Online-Demo:	
OpenSource:	ja / GPL
Zielgruppe:	Einführungskurse Prolog / Lisp

PABS Grader-Steckbrief (siehe Kapitel 15)

Entwicklung seit:	2010
Aktueller Hauptentwickler:	Universität Würzburg Alexander Dallmann, Lukas Iffländer lukas.ifflander@uni-wuerzburg.de, alexander.dallmann@uni-wuerzburg.de
Kontaktmöglichkeit:	
Zielgruppe:	Programmierpraktika und praktische Aufgaben zur Vorlesungsbegleitung

ASB Grader-Steckbrief (siehe Kapitel 16)

Entwicklung seit:	2006
Aktueller Hauptentwickler:	Hochschule Trier, Fachbereich Informatik, ASB-Projektteam (Britta Herres, David Schuster)
Kontaktmöglichkeit:	Rainer Oechsle (oechsle@hochschule-trier.de)
Zielgruppe:	Programmierpraktika und praktische Aufgaben zur Vorlesungsbegleitung

Feature	JACK	Praktomat	Graja	aSQLg	GATE	VEA	PABS	ASB
<i>Programmiersprachen</i>								
Java	✓	✓	✓		✓		✓	✓
C/C++	✓	✓						✓
Fortran		✓						
.NET	✓							
Haskell		✓						
Python	✓	✓						✓
R	✓	✓						
Prolog						✓		
Lisp						✓		
<i>Andere Sprachen</i>								
Isabelle		✓						
UML	✓				✓			
SQL				✓				
<i>Statische Prüfverfahren</i>								
Syntax	✓	✓	✓	✓	✓			✓
Stil	✓	✓		✓				✓
FindBugs/PMD			✓					✓
Metriken	✓							✓
andere	✓			✓	✓	✓		
<i>Dynamische Prüfverfahren</i>								
Ein-/Ausgabetests (z. B. Unit-Tests)	✓	✓	✓	✓	✓	✓	✓	✓
GUI-Tests								✓
Remote-Tests								✓
Weitere								✓
<i>Prüfverfahren erweiterbar</i>	✓	✓			✓			✓

Tabelle 17.1: Featurematrix (Teil 1) der Grader aus Teil II dieses Buches.

Feature	JACK	Praktomat	Graja	asSQLg	GATE	VEA	PABS	ASB
<i>Feedbackmöglichkeiten</i>								
Punkte/Noten	✓	✓	✓	✓	✓	✓		
Textuell	✓	✓	✓	✓	✓	✓	✓	✓
Codeannotation	✓	✓						
Grafisch	✓		✓					
<i>Manuelle Nachbewertung</i>	✓	✓	✓		✓			
<i>Plagiatserkennung</i>	✓	✓			✓	✓	✓	✓
<i>Integration in LMS</i>								
Für Studierende	✓	✓	✓	✓		✓		
Für Lehrende		✓	✓	✓		✓		
<i>Unterstützte Standards</i>								
IMS-LTI	✓							
ProFormA XML	✓	✓	✓	✓	✓			
Grappa Middleware			✓	✓				
<i>Didaktische Einsatzszenarien</i>								
Objektorientierte Programmierung	✓	✓	✓		✓		✓	✓
Deduktive/logische oder funktionale Programmierung		✓				✓		
Relationale Datenbankabfragen				✓				
UML-Modellierung	✓				✓			
App-Entwicklung								✓
Mathematikausbildung	✓	✓						

Tabelle 17.2: Featurematrix (Teil 2) der Grader aus Teil II dieses Buches.