

Assistierte Bewertung von Freitextaufgaben in Moodle

Ulrike Pado

Cornelia Kiefer, Verena Meyer

7.4.2016

Freitextfragen - Beispiel

Frage 1

Bisher nicht beantwortet










Erreichbare Punkte: 1,00

🚩 Frage markieren

⚙️ Frage bearbeiten

Erklären Sie kurz anhand der gegebenen Methode, wie Rekursion funktioniert.

```
static int summe(int n) {  
    int wert = ( n == 0 ) ? 0 :  
        n + summe(n - 1);  
    return wert;  
}
```



- Offene Fragen, mit vom Prüfling selbst formulierten Antworten
- Antworten liefern gute **Einblicke** in Verständnis und Missverständnis: **Raten** wird weitgehend ausgeschlossen
- Auch **Analysefähigkeit** und kreative Prozesse gut prüfbar

Geschwindigkeit und Konsistenz steigern

- **Sortieren** der Studentenantworten nach Güte (v1.0)
 - Gut = ähnlich zur Referenzantwort
 - Bewerter kann sich auf Zweifelsfälle konzentrieren; gut/schlecht können schnell abgehakt werden
 - Deutlicher Nutzen für genau die Fragen, die langsam zu bewerten sind (Pado & Kiefer 2015)
 - Bewerter kann gute Studentenantworten als Referenz hinzufügen
- **Konsistenzprüfung** beim Bewerten (v2.0)
 - Ist die gerade vergebene Bewertung konsistent mit den Punkten, die ähnliche Antworten schon erhalten haben?

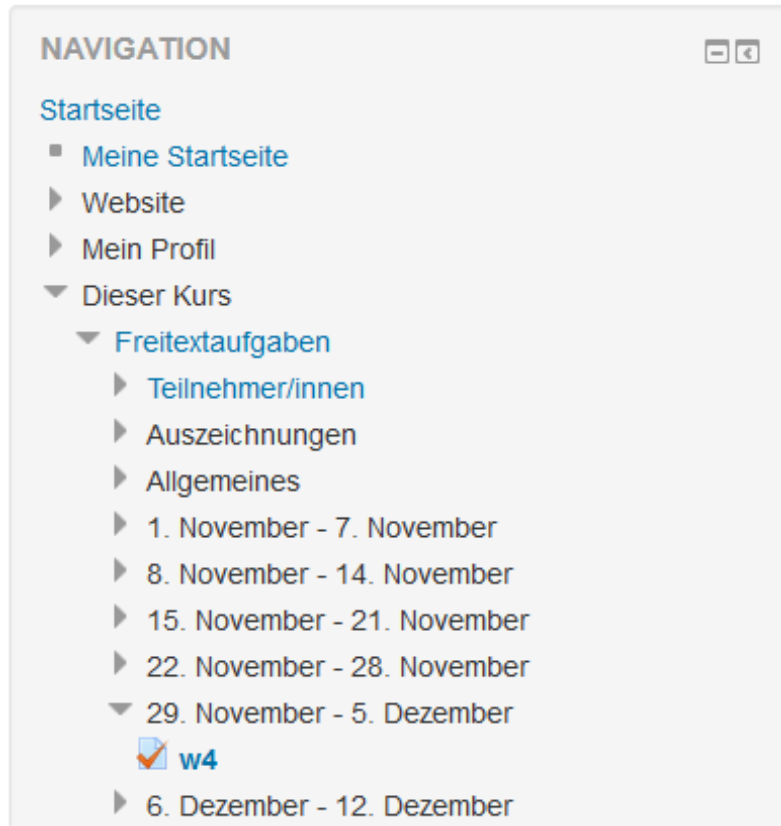
Moodle-Plugin: Ähnlichkeitsbasierte Sortierung

Webservice ermittelt Ähnlichkeit
zu Referenzantwort(en)
durch ling. Normalisierung und
String Matching (schlägt dt.
state of the art
auf unseren Daten)

Anzeige und Bewertung
nahtlos in Moodle

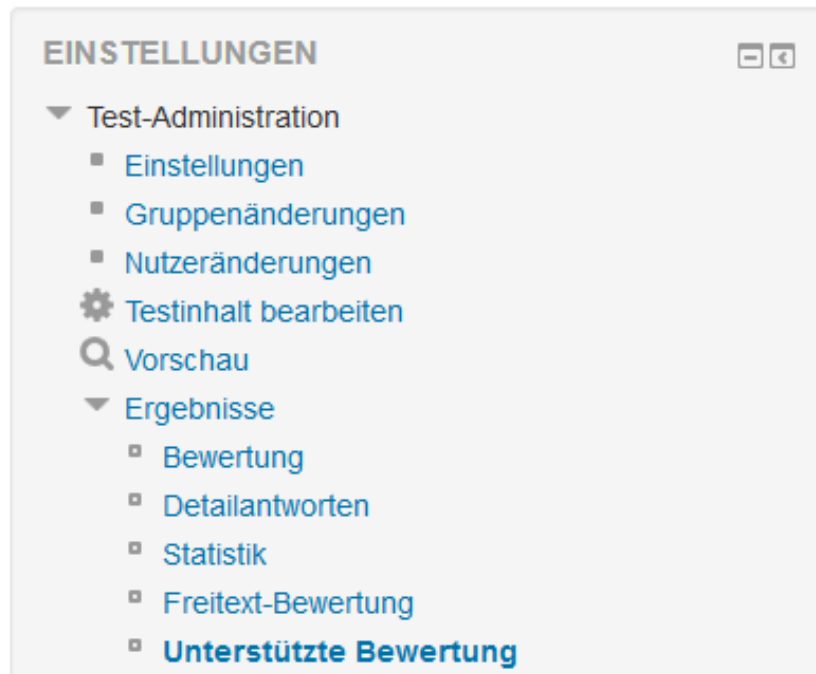


Anleitung – Schritt für Schritt



- Test auswählen

Anleitung – Schritt für Schritt



- Test auswählen
- Einstellungen > Ergebnisse > Unterstützte Bewertung

Anleitung – Schritt für Schritt

- Test auswählen
- Einstellungen > Ergebnisse > Unterstützte Bewertung
- Frage auswählen: Klick auf „Alles bewerten“

w4

Fragen, die eine Bewertung erwarten

F #	Fragename	Zu bewerten	Bereits bewertet	Summe
1	w4	0	36 Bewertungen aktualisieren	36 Alles bewerten

Hochschule für Technik Stuttgart

w4

Fragen bewerten 1: w4

[Zurück zur Fragenliste](#)

Optionen

Versuche zur Bewertung

Alles (36)

Webservice URL

localhost:8080/GA/webresources/

[[orderby]]

[[byscoredesc]]

Optionen ändern

Sortierung

Versuche von 1 bis 0 von 36

Versuch 1 von 7 Last7

Antwort als zusätzliche Referenzantwort verwenden

[[addanswer]]

Frage 1

Erreichte Punkte
1,00 von 1,00

Erklären Sie kurz anhand der gegebenen Methode, wie Rekursion funktioniert.

```
static int summe(int n) {  
    int wert = ( n == 0 ) ? 0 :  
        n + summe(n - 1);  
    return wert;  
}
```

Frage

Rekursion bedeutet, dass eine Methode sich selbst solange aufruft, bis z.B. eine Abbruchbedingung eintrifft. In diesem Fall wird eine Bedingungsabfrage über den ? Operator eingeleitet.

Solange der Wert n nicht 0 ist, wird der neue Wert gebildet aus: $n + \text{summe}(n-1)$.

Wenn n 0 ist wird die Rekursion verlassen und der wert endgültig zurück gegeben.

Studentenantwort

Eine rekursive Methode ruft sich selbst im Methodenkörper wieder auf. Damit die Rekursion endet, muss das Argument bei dem neuen Aufruf verändert sein und es muss eine Abbruchbedingung geben, für die kein rekursiver Aufruf mehr stattfindet. Im Beispiel ist die Abbruchbedingung ($n=0$) und der Parameter beim rekursiven Aufruf wird jedes Mal reduziert (bis die Abbruchbedingung erreicht ist).

Methode ruft sich selber auf: 0,5
Abbruchbedingung 0,5

Referenzantwort

Kommentar

Rich text editor toolbar with icons for bold, italic, link, unlink, and other formatting options.

Bewertung

Punkte

1,00

von 1,00

Konsistenzprüfung

The screenshot shows the exam portal interface. At the top, a list of exams is displayed, with 'Versuch 1 von ws15-16pvl3026 e-klausur' highlighted in red. Below this, a detailed view of 'Versuch 1 von ws15-16pvl3031 e-klausur' is shown. The question asks why a program is faster with threads on a single processor. The answer field contains the text: 'Weil Aufgaben die parallel erledigt werden können parallel erledigt werden. Ohne Threads arbeitet ein Programm alle Aufgaben nacheinander ab.' The score for this question is 1.00 out of 1.00. A blue line points from the 'Ähnliche Antwort mit abweichender Punktzahl' text to the highlighted exam in the list. Another blue line points from the 'Bewertung kann entsprechend früherer Entscheidungen angepasst werden' text to the score box at the bottom of the question view.

Versuch 1 von ws15-16pvl3012 e-klausur

Versuch 1 von ws15-16pvl3026 e-klausur

Versuch 1 von ws15-16pvl3030 e-klausur

Versuch 1 von ws15-16pvl3023 e-klausur

Versuch 1 von ws15-16pvl3014 e-klausur

Versuch 1 von ws15-16pvl3004 e-klausur

Versuch 1 von ws15-16pvl3010 e-klausur

Versuch 1 von ws15-16pvl3002 e-klausur

Versuch 1 von ws15-16pvl3005 e-klausur

Versuch 1 von ws15-16pvl3018 e-klausur

Versuch 1 von ws15-16pvl3028 e-klausur

Versuch 1 von ws15-16pvl3039 e-klausur

Versuch 1 von ws15-16pvl3011 e-klausur

Versuch 1 von ws15-16pvl3029 e-klausur

Versuch 1 von ws15-16pvl3062 e-klausur

Versuch 1 von ws15-16pvl3031 e-klausur

Antwort der Referenzantwort hinzufügen

Frage 9

Vollständig

Erreichte Punkte
1,00 von 1,00

Warum wird Ihr Programm durch die Verwendung von Threads schneller, auch wenn Sie nur einen einzigen Prozessor zur Verfügung haben?

Weil Aufgaben die parallel erledigt werden können parallel erledigt werden. Ohne Threads arbeitet ein Programm alle Aufgaben nacheinander ab.

Die Ausführung wird schneller, weil der Prozessor zwischen verschiedenen Teilaufgaben hin- und herspringen kann und so z.B. Wartezeiten auf Daten in einem Thread zur Bearbeitung anderer Threads genutzt werden können.

Kommentar

Punkte 1,00 von 1,00

Ähnliche Antwort mit abweichender Punktzahl

Während der Bewertung wird überprüft, ob ähnliche Antworten gleich bewertet wurden. Wenn nicht, werden die konfligierenden Antworten markiert.

Die Markierung verschwindet, sobald zur nächsten Antwort gewechselt wird.

Bewertung kann entsprechend früherer Entscheidungen angepasst werden

Referenzen

Kiefer, C. und Pado, U. (2015). Freitextaufgaben in Online-Tests? Bewertung und Bewertungsunterstützung. HMD Praxis der Wirtschaftsinformatik, 52(1), 96-107.

Pado, U. and Kiefer, C. (2015). Short Answer Grading: When Sorting Helps and When it Doesn't. 4th NLP4CALL workshop at Nodalida.