

Erweiterung der eLearning- Plattform Prolog

Ausgangssituation

Ziel der Veranstaltung „Einführung in Prolog“ ist das Erlernen der logischen Programmierung in Prolog. Hierfür ist es unerlässlich, dass die Studierenden das eigenständige Entwickeln von Programmen einüben und sich nicht auf das Nachvollziehen gegebener Programme beschränken. Viele Studierende stehen jedoch regelmäßig vor dem Problem, dass sie nicht wissen, wie sie gestellte Übungsaufgaben angehen können.

Ziele & Zielgruppen

Durch den Ausbau der Veranstaltung mit verschiedenen eLearning-Elementen soll den Studierenden ein zeit- und ortsunabhängiges Lernen ermöglicht werden. Angestrebt wird, dass das Tempo der Veranstaltung allen Studierenden trotz unterschiedlichen Vorwissens gerecht wird. Außerdem sollen die begrenzten zeitlichen Kapazitäten der Dozierenden in den Übungssitzungen optimal genutzt und Leerlaufzeiten der Studierenden, die sich aus dem Warten auf individuelle Hilfe ergeben, vermieden werden. Die Lehrveranstaltung richtet sich jedes Wintersemester an ungefähr 50 - 60 Studierende.

Umsetzung

Die eLearning-Plattform basiert auf einer WordPress-Instanz, in der die Aufgaben direkt implementiert, Vorlesungsinhalte hinterlegt und Musterlösungen abgerufen werden können. Die Studierenden werden Schritt für Schritt durch die Übungsaufgaben geleitet und haben dabei die Möglichkeit, für ein besseres Verständnis entsprechende Quizseiten zu bearbeiten, welche an den Lösungsweg heranführen. Für jede Aufgabe wurde außerdem eine zugehörige interaktive Experimentierumgebung (SWISH-Seite) erstellt, welche das selbstständige Implementieren und Überprüfen von Lösungen ermöglicht. Als weiteres Tool werden Kahoots genutzt, welche es sowohl den Studierenden als auch den Dozierenden erlauben, spielerisch den aktuellen Lernstand zu überprüfen.

Ergebnisse

Es wurde eine interne Evaluation über das ILIAS durchgeführt, bei der herausgefunden werden sollte, wie zufrieden die Studierenden mit den verschiedenen eLearning-Elementen sind. Untersucht wurde die Nutzung der Quizseiten, sowie die der darin eingebundenen Hilfestellungen und SWISH-Seiten. Insgesamt wurden all diese Elemente häufig genutzt und überwiegend als sehr hilfreich eingestuft. In den Freitextantworten lobten die Studierenden die Möglichkeit, in eigenem Tempo die Inhalte zu erarbeiten, sowie die Übungsaufgaben und den übersichtlichen Aufbau der eLearning-Plattform. Insgesamt zeigt das positive Feedback auf, dass gerade in der aktuellen Distanzlehre der Aufbau und die Erweiterung der Plattform die Studierenden inklusiv und erfahrungsbasiert im Lernen unterstützt.

02.06.03. Deutsche Grammatik – Vertiefung

Um sich mit dem Aufbau der Wissensbasis genauer vertraut zu machen, können Sie zuerst eine **Grammatik erstellen**, die nur einen **einzigsten Satz** generieren kann. Diese Grammatik kann dann erweitert werden. Die folgenden Aufgaben helfen dir eine einfache Wissensbasis zu erstellen, die folgenden Satz generiert:

die katze klaut

Zur Hilfestellung hier nochmal die englische Grammatik:

```

1 % word/2% word(X): X ist ein Satzglied
2 word(determiner,d).
3 word(determiner,every).
4 word(noun,criminal).
5 word(noun,'big kahuna burger').
6 word(verb,eats).
7 word(verb,likes).
8 % sentence/1% sentence(X): X ist ein Satz
9 % seq/5% seq(X,Y,Z,A,B): seq ist eine Folge von Wörtern
10 sentence(seq(Word1,Word2,Word3,Word4,Word5):-
11   word(determiner,Word1),
12   word(noun,Word2),
13   word(verb,Word3),
14   word(determiner,Word4),
15   word(noun,Word5)).

```

Im einfachsten Fall können wir den Satz (die katze klaut) bereits als Fakt in unsere Wissensbasis schreiben. **Vervollständige** die folgende Klausel, indem Du die drei Punkte ersetzt:
sentence(seq(...)).

Gib hier deine Antwort ein.

Überprüfen

Welche Probleme können entstehen, wenn Du deine Wissensbasis weiter nach diesem Schema aufbaust?

- Alle möglichen Sätze müssen einzeln eingegeben werden – das ist eine Menge Arbeit.
- Es ist schwieriger auf Fehler zu prüfen, da diese vermutlich nicht systematisch sind.
- Es ist unmöglich um Pluralformen und Maskulina zu erweitern.

Überprüfen

interaktive Quizseite mit Freitext- und Multiple Choice-Aufgaben

SWISH File Edit Examples Help

hhu_prolog_03_04_greater_than

greater_than

Definiere ein Prädikat **greater_than/2**, das zwei natürliche Zahlen in der succ/1-Notation nimmt und überprüft, ob die erste Zahl größer ist als die zweite:

```

?- greater_than(succ(succ(succ(0))),succ(0)).
true.
?- greater_than(succ(succ(0)),succ(succ(succ(0)))).
false.
?- greater_than(succ(0),succ(0)).
false.

```

```

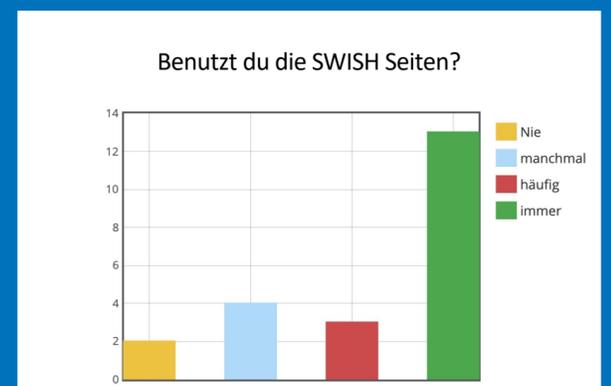
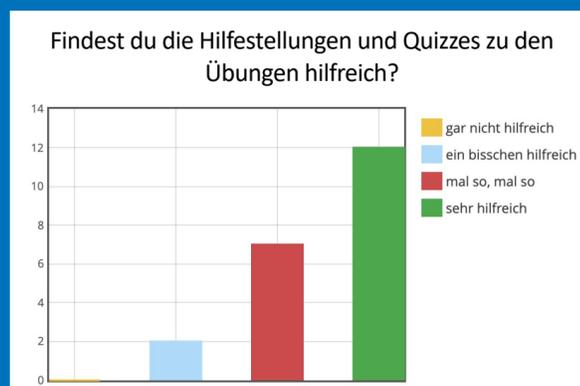
1 # greater_than/2
2 # greater_than(X,Y)
3
4 # Füge in der nächsten Zeile deinen Code ein
5
6
7
8
9
10
11
12
13 #

```

?- Your query goes here ...

Examples History Solutions table results Run!

SWISH-Seite zum selbstständigen Implementieren und Testen von Code



interne Evaluationsergebnisse (durchgeführt im ILIAS)

