Kompetenznachweis Modul 226B

Die Kompetenzentwicklung wird im Modul 226B mit einzelnen Aufgabenstellungen zu Beginn und einem durchgängigen Projekt bis zum Schluss des Quartals nachgewiesen. Als Grundlage der Bewertung dient ein Kompetenzraster, welches einer Matrix von 5 Zeilen und 3 Spalten entspricht.

Nachweis einer Kompetenz

Der einzelne Lernende hat die beschriebenen Kompetenzen im Verlaufe des Moduls nachzuweisen. Ist die entsprechende Kompetenz nachgewiesen worden, so wird dies durch das Visum der Lehrperson und des Lernenden im dazugehörigen Feld bestätigt.
Bewertet wird grundsätzlich nur die vom Lernenden/der Gruppe erbrachte Eigenleistung. Fremde Quellen sind zu kennzeichnen und zu kommentieren.

Notengebung

Jede Zelle der ersten Spalte im Kompetenzraster ist mit 0.5 Notenpunkten, alle übrigen Zellen sind mit 0.25 Notenpunkten, bewertet. Für jede erreichte Kompetenzzelle steigt die Note um den entsprechenden Wert.

Die Kompetenzen werden während des Quartals in Fachgesprächen mit der Lehrperson überprüft. Im Fokus steht immer der Nachweis der entsprechenden Teilkompetenz anhand von Eigenleistungen. Es können auch mehrere Teilkompetenzen auf einmal erreicht werden.

Form

Sie werden die Projektarbeit in einer 2er Gruppe realisieren. Bei ungerader Lernenden Anzahl kann es ausnahmsweise eine 3er Gruppe geben. Alle Lernenden müssen in einem Gespräch jeweils nachvollziehbar zeigen, dass Sie über die Kompetenzen verfügen.

ePortfolio

Der persönliche Lernfortschritt und die eingebrachten Nachweise sind in einem ePortfolio zu sammeln, zu strukturieren und der Lehrperson zugänglich zu machen (Google Drive Dokument). Zu allen Kompetenzen des Rasters, welche mit einem **P** versehen sind, sollen im ePortfolio folgende Punkte dokumentiert werden:

- eine Beschreibung (wie wird der Kompetenznachweis erbracht)
- ein Verweis auf das Lernprodukt (Dokument, Diagramm, Programmteil, Programm)
- eine persönliche Reflexion

Sie dürfen Kompetenzen auch in einem Portfolio Eintrag kombinieren, wenn eine Kombination sinnvoll erscheint.

Leitfragen für die Reflexion:

1. Das habe ich gemacht.
2. Das habe ich gelernt.
3. Das ist mir gut gelungen.
4. Damit hatte ich Schwierigkeiten
5. So habe ich auf die Schwierigkeiten reagiert.
6. Kritische Betrachtung der vorliegenden Lösung und mögliche Verbesserungsvorschläge

Projekt

Es gelten folgende Rahmenbedingungen für das Projekt:

- Applikation mit mindestens 3 fachlichen Klassen. Am Beispiel einer Bibliothek wären fachliche Klassen z.B.: Buch, Autor / Person, Bibliothek
- Sie haben eine grafische Benutzeroberfläche zur Bedienung ihres Programms
- Sie arbeiten mit Java oder mit C# .Net (keine Mobile-Anwendungen)
- Als Entwicklungsumgebung setzen Sie Eclipse, IntelliJ oder im Falle von C# MS Visual Studio ein
- Sie dürfen Frameworks / fremde Libraries einsetzen
- Sie dürfen externe Dienste verwenden
- Ihr Projekt wird im Rahmen des Kompetenzrasters bewertet

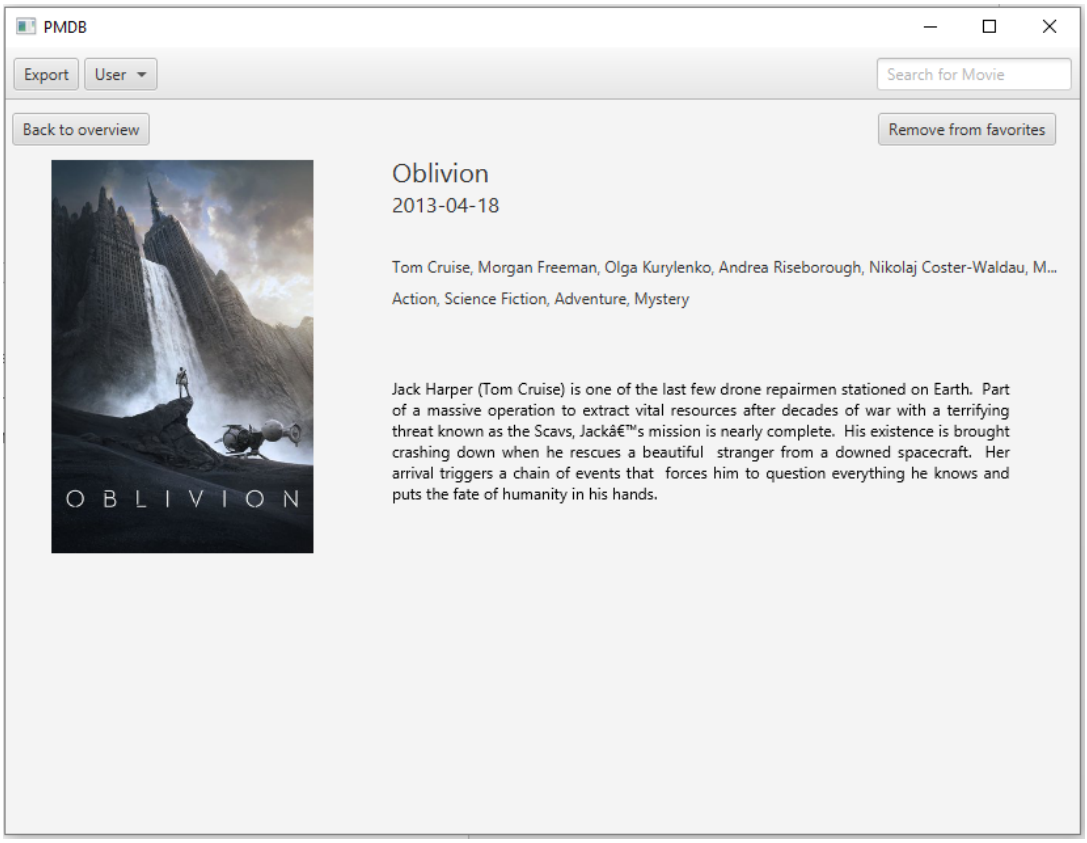
Kompetenzraster

Kompetenzbereich	A (+0.5 pro Zelle)	B (+0.25 pro Zelle)	C (+0.25 pro Zelle)
1 - Anforderungen und Design	Der Lernende kann die im Pflichtenheft definierten Anforderungen in ein Design umsetzen, welches auch Vererbung abbildet. Er kann das Design mittels UML-Klassendiagramm (OOA) darstellen. Bedingungen: OOA enthält Klassen, wichtigste Attribute, wichtigste Methoden (ohne getter/setter) sowie Beziehungen)	Der Lernende kann die Anforderungen in ein sinnvolles Design umsetzen. Dabei wurden Interfaces adäquat eingesetzt. Abstrakte Klassen werden an Stellen verwendet, wobei es Sinn macht. Er kann das Design mittels UML-Klassendiagramm (OOA) darstellen. **P** Bedingungen: Analog A1	Der Lernende hat die Kompetenz, ein ausgereiftes Design, orientiert an der 3-Schicht Architektur, zu entwickeln. Das Design wird in der Implementation widerspiegelt. Die Architektur wurde mit einem UML Klassendiagramm in OOD dargestellt, über alle Schichten hinweg. **P** Bedingungen: OOD darf generiert werden (z.B. ObjectAid) Schichten sind im OOD klar erkennbar Reflexion beinhaltet ein Vergleich zwischen OOD und geplantem OOA Diagramm.
2.1 – Implementierung	Der Lernende kann das Prinzip der Vererbung anhand von eigenen Beispielen aufzeigen. Er kann ebenfalls starke und schwache Polymorphie erklären.	Der Lernende implementiert das entwickelte Design und zeigt somit erweiterte Kenntnisse zu den Themen Vererbung, abstrakte Klassen, Polymorphie und Interfaces. Der Lernende versteht das Prinzip der dynamischen Bindung. **P**	Der Lernende implementiert das entwickelte Design, wobei die Anwendung ein vertieftes Verständnis der Konzepte der Vererbung und Polymorphie sowie die Anwendung von Interfaces, zeigt – oder es wird aufgezeigt / erklärt, weshalb gewisse Konzepte nicht verwendet wurden. **P**
2.2 - Implementierung	Der Lernende kann das Prinzip von Interfaces an eigenen Beispielen aufzeigen und die Vorteile des Einsatzes erläutern.	Der Lernende kann eine Benutzeroberfläche implementieren und berücksichtigt dabei die Usability im Sinne von Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Steuerbarkeit und Erwartungskonformität. Bedingungen (2.1 und 2.2): Das Programm ist startbar, es treten keine unerwarteten Exceptions auf, 80% der Features sind funktionsfähig.	Der Lernende kann eine 3-Schicht Architektur implementieren und achtet dabei auf die Unabhängigkeit der einzelnen Schichten **P** Bedingungen (2.1 und 2.2): Alle im Pflichtenheft geplanten Features sind vorhanden und funktionsfähig. Die 3-Schicht Architektur wurde eingehalten.
3 - Testing und Dokumentation	Der Lernende kann Anforderungen an eine Software in einem Pflichtenheft gemäss Vorgabe festhalten. Der Lernende kann Testfälle für sein Projekt definieren.	Die definierten Testfälle aus A3 wurden durchgeführt und protokolliert. Definierte Sourcecode-Elemente (Klassen, Konstruktoren und Methoden) sind mit den passenden Sprachmitteln dokumentiert.	Der Lernende kann sinnvolle Unittests (automatisiert) zu seinem Projekt aufweisen. Die Testfälle sind dabei unabhängig voneinander. Der Sourcecode ist sauber dokumentiert und gibt Auskunft über Funktionalität, Parameter sowie Rückgabewerte.
4 - Methoden-, Sozial- und Selbstkompetenz	Der Lernende ist in der Lage, eine Grobplanung zum Projekt zu erstellen. Er identifiziert dabei die Meilensteine und was an den Meilensteinen konkret vorliegen muss. Der Lernende kann sein Projekt in ein Sourcekontrollsystem einbinden.	Der Lernende ist in der Lage, seine Grobplanung in eine Detailplanung zu überführen (auch iterativ möglich). Der Lernende setzt ein Sourcekontrollsystem aktiv für die Entwicklung ein. **P** Bedingungen: Planungsdokumente sind während der Projektphase aktuell.	Der eigene Lernfortschritt wird selber geplant und auch überprüft. Das wird entsprechend dokumentiert. Ergebnisse werden kritisch hinterfragt.

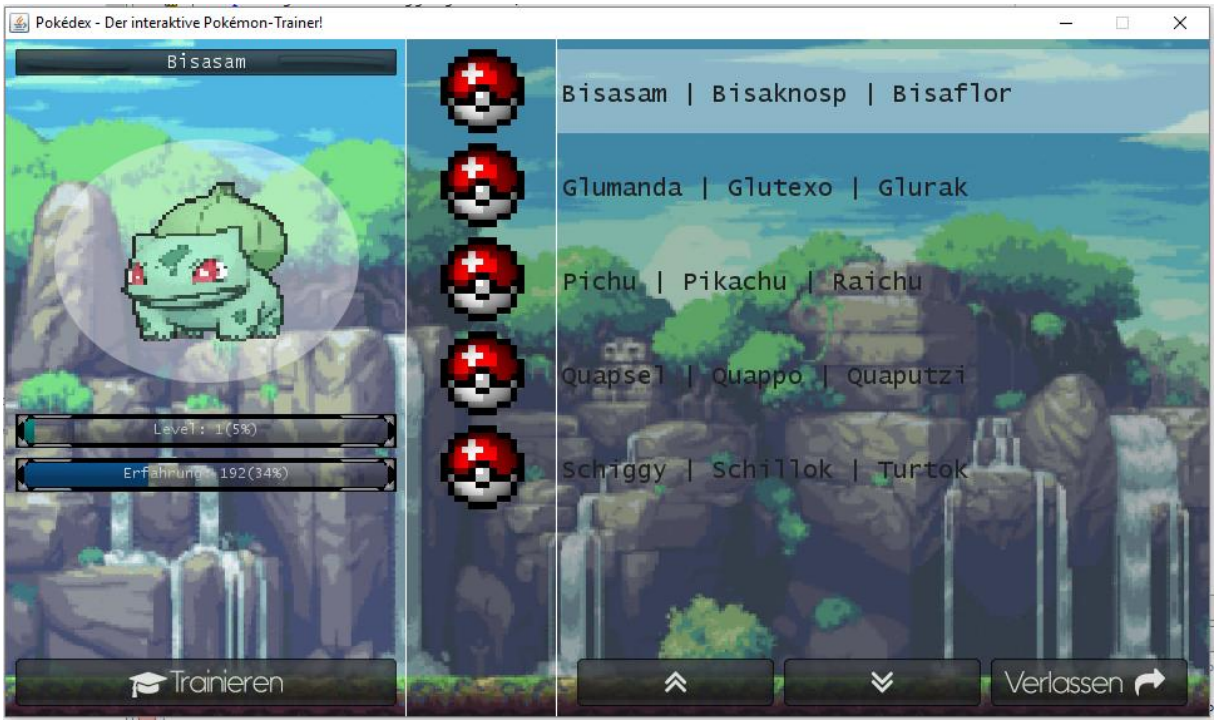
Kompetenzbereich	A	B	C
1 - Anforderungen und Design			
2.1 - Implementierung			
2.2 - Implementierung			
3 - Testing und Dokumentation			
4 - Methoden-, Sozial- und Selbstkompetenz			

Einige Beispiele aus vergangenen Projekten

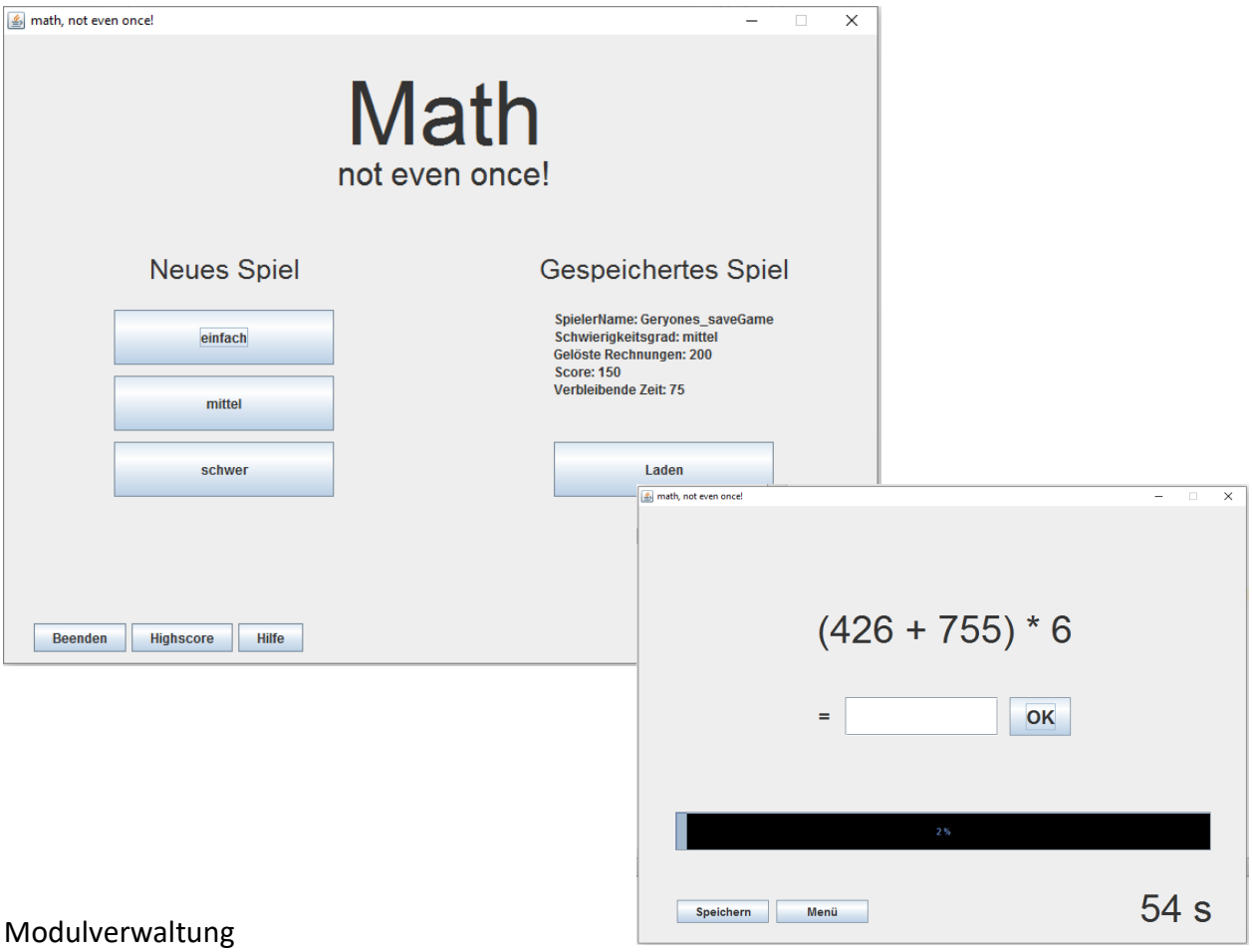
Filmeverwaltung mit Onlinesuche (imdb), Usermanagement, Favoritenverwaltung



Pokédex – Trainieren von Pokémons, Level/Erfahrungssystem, Threading



Mathe Quiz



Modulverwaltung