
Semesteraufgabe Informatik 2

Abgabe spätestens bis: 07.07.2024

Allgemeine Aufgabenstellung

Sie können als freiwillige Zusatzaufgabe zu Hause **eine** der unteren Systembeschreibungen implementieren. Die Aufgabe wird durch die in den Übungsgruppen vorhandenen Teams bearbeitet.¹ Die Bewertung erfolgt durch eine Abnahme. Die Teams mit dem besten Programm zu einer Aufgabe erhalten eine Urkunde. Eine Teilnahme mit gutem Erfolg ist empfohlen, wenn man Tutor für Informatik 1 oder Informatik 2 werden möchte.

Abgabe

Die Abgabe muss in folgender Form erfolgen:

- Upload der Materialien in den Semesteraufgabe-Abgabe-Ordner der Digicampus-Veranstaltung **Informatik 2** als **eine ZIP-Datei**. Die Benennung der ZIP-Datei muss von der Form **AufgabeX.Nachname1Vorname1.Nachname2Vorname2.Nachname3Vorname3** sein, wobei **X** durch den Buchstaben der gewählten Systembeschreibung (M/R/O) und **NachnameVorname** jeweils durch **Ihre** Namen ersetzt wurde. Aus der Benennung der Datei geht also hervor, welche der Aufgaben bearbeitet wurde und von wem.
- Die Upload-Materialien müssen enthalten:
 - Quellcode.
 - Ausführbare JAR-Datei.
 - Hinweise zur Ausführung und Benutzung.
 - Auflistung der Personen, die an der Aufgabe gearbeitet haben.
 - Angabe von zeitlichen Überschneidungen am Tag der Abnahme (23.07.2024)

Falls Sie nicht möchten, dass die Gruppe, die Ihre Abgabe bewertet, Ihre Namen erfährt, packen Sie eine anonymisierte Fassung mit ein.

Allgemeine Hinweise

- Jedes Team bewertet die Lösung eines anderen Teams:
 - Die Lösung des anderen Teams wird vom Lehrstuhl am 08.07.2024 zugeschickt.
 - Die Bewertung muss bis 15.07.2024 in den Semesteraufgabe-Bewertung-Ordner der Digicampus-Veranstaltung **Informatik 2** hochgeladen werden (als normale Textdatei oder PDF).
 - Die Benennung der Datei mit Ihrer Bewertung muss von der Form **Nachname1Vorname1.Nachname2Vorname2.Nachname3Vorname3** sein, wobei **NachnameVorname** jeweils durch **Ihre** Namen ersetzt wurde.

¹Eine Zusammenarbeit mit Personen außerhalb des eigenen Teams ist erlaubt, wenn die eigenen Teammitglieder kein Interesse an der Bearbeitung der Semesteraufgabe haben. Jedoch darf eine Gruppe auch bei der Semesteraufgabe aus maximal drei Personen bestehen.

-
- Abnahme am 23.07.2024:²:
 - Bei der Abnahme müssen alle Teammitglieder anwesend sein.
 - Die Teammitglieder führen ihr Programm vor und beantworten Fragen zur Funktionalität und zur Programmierung.
 - Die Abnahme erfolgt in Präsenz.
 - In der Vorlesung wurden noch nicht alle notwendigen Anforderungen besprochen. Sie sollten bereits jetzt mit der Programmierung der bereits behandelten Konzepte (Datenklassen, grafische Benutzerschnittstelle) beginnen, und diese dann nach und nach um die weiteren Anforderungen (z. B. persistente Datenhaltung) ergänzen und verbessern, sobald diese in der Vorlesung besprochen wurden.

Technische Mindestanforderungen

- Das Programm muss in der Programmiersprache Java erstellt worden und fehlerfrei ausführbar sein.
- Es dürfen bei der Implementierung keine anderen Bibliotheken als die Standardbibliothek benutzt werden.
- Der Code muss sich im Wesentlichen an den Vorgaben der Vorlesung orientieren.
- Der Code folgt einer 3-Schichten-Architektur mit grafischer Benutzerschnittstelle, Fachkonzeptschicht und persistenter Datenhaltung Ihrer Wahl (z. B. per Datenbankbindung oder Speicherung in Textdateien).
- Vorgaben für die grafische Benutzerschnittstelle: Layout und Ereignisbehandlung in AWT, grafische Komponenten in Swing.
- Sie müssen alle Designentscheidungen begründen können.
- Die öffentlichen Schnittstellen ihrer Module sind mit JavaDoc dokumentiert.
- Die Anforderungen der spezifischen Aufgabenstellung müssen erfüllt sein.
- Der Code muss eigenständig erstellt worden sein.

Bewertung

- Die Bewertung erfolgt durch den Lehrstuhl auf Grundlage der Abnahme.
 - Die Abnahme besteht aus einer kurzen Präsentation des Programms (und zwar der Anwendung, nicht des Quellcodes, Dauer: maximal 10 Minuten) durch alle Teammitglieder, mit anschließender ca. 5-minütiger Diskussion. Jedes Teammitglied muss einen Teil der Präsentation übernehmen.
 - Erfüllt das Programm alle Mindestanforderungen, so erhält man 2 Punkte.
 - Für besondere Aspekte der Umsetzung, die über die Mindestanforderungen hinaus gehen, kann man bis zu 2 weitere Punkte erhalten, z. B. für
 - eine besonders schöne und durchdachte grafische Benutzerschnittstelle mit Benutzerführung.
 - zusätzliche sinnvolle Funktionalitäten, die deutlich über die Anforderungen der spezifischen Aufgabenstellung hinausgehen.
 - Berücksichtigung zusätzlicher sinnvoller Daten, die nicht in der spezifischen Aufgabenstellung erwähnt werden und einen erkennbaren Mehrwert für die Anwendung bedeuten.
 - besonders gelungene Strukturierung und Entkopplung des Quellcodes für eine bessere Wartung und Erweiterbarkeit des Programms.
- Solche besonderen Aspekte müssen in der Abgabe-Mail und in der Abnahme deutlich gemacht werden.
- Die Bearbeitung einer der freiwilligen Zusatzaufgaben bringt 1 Punkt.

²Voraussichtlicher Termin.

Systembeschreibung A: Deterministische endliche Automaten

Es soll ein Editor entwickelt werden, in dem deterministische endliche Automaten (siehe Vorlesung *Einführung in die theoretische Informatik* oder https://de.wikipedia.org/wiki/Deterministischer_endlicher_Automat) erstellt werden können. Dabei soll es möglich sein, mit Mausklicks Zustände zu erstellen und zu positionieren und diese auch mit Übergängen zu verbinden. Die vorhandenen Zustände sollen durch Verschieben mit der Maus neu positioniert werden können. Es soll die Möglichkeit geben, bestehende Beschriftungen zu ändern (z. B. sollen Zustände umbenannt werden können). Außerdem soll es die Möglichkeit geben, einzelne Elemente zu löschen oder mehrere Elemente zu markieren und diese dann alle zu löschen. Der Editor soll das Speichern und Laden von Automaten unterstützen.

Hinweis: Es bietet sich an, beim Speichern ein Format zu wählen, in dem die Position der einzelnen Elemente dokumentiert wird. So kann der endliche Automat beim Laden angezeigt werden ohne dass man eine übersichtliche Darstellung berechnen muss. Andernfalls muss man sich beim Laden mit der (aufwendigen) Positionierung der Zustände und Übergänge auseinandersetzen.

Freiwillige Zusatzaufgabe

Es soll möglich sein, Wörter einzugeben, für die überprüft wird, ob sie in der Sprache des Automaten enthalten sind.

Technische Anforderungen

Das System soll modular aufgebaut und leicht erweiterbar sein.

Systembeschreibung R: Rezeptebuch

Entwickeln Sie ein System, in dem Rezepte verwaltet werden können. Rezepte sollen hinzugefügt, bearbeitet und gelöscht werden können. Die Rezepte sollen persistent gespeichert werden können. Die Rezepte sollen nach Namen und Zutaten durchsucht werden können.

Freiwillige Zusatzaufgaben

1. Suche nach Rezepten auf bestimmte Kategorien wie z. B. vegetarische oder vegane Gerichte einschränken
2. Zusammenfassen mehrerer Rezepte zu einem Wochenplan
3. Erstellen einer Einkaufsliste basierend auf mehreren Rezepten

Technische Anforderungen

Das System soll modular aufgebaut und leicht erweiterbar sein.

Systembeschreibung O: Othello

Entwickeln Sie ein Othello-Spiel, das von zwei menschlichen Spielern gespielt werden kann. Das Spiel soll über eine grafische Benutzeroberfläche verfügen, die es den Spielern ermöglicht, Züge auszuwählen und das Spielbrett zu sehen. Implementieren Sie die Spiellogik, die es den Spielern erlaubt, Züge zu machen und das Spiel zu beenden, wenn ein Spieler gewonnen hat oder das Spiel unentschieden ausgeht. Verwenden Sie geeignete Datenstrukturen, um das Spielbrett und den Zustand des Spiels zu speichern.

Link zum Regelwerk: [https://de.wikipedia.org/wiki/Othello_\(Spiel\)](https://de.wikipedia.org/wiki/Othello_(Spiel))

Freiwillige Zusatzaufgaben

Entwicklung einer einfachen KI (random/greedy) mit der man das Othello-Spiel auch alleine gegen den Computer spielen kann.

Technische Anforderungen

Das System soll modular aufgebaut sein und dem Schema Model-View-Controller folgen.