

Übung 06: UML

Abgabetermin: 30.04.2009, 8:15

Name: _____ Matrikelnummer: _____

Informatik: G1 (Prähofer) G2 (Prähofer) G3 (Würthinger) G4 (Prähofer)

WIN: G1 (Khalil) G2 (Khalil) G3 (Schwinger)

Aufgabe	Punkte	abzugeben schriftlich	abzugeben elektronisch	korr.	Punkte
Übung 6	24	Use-Cases Sequendiagramme Klassendiagramm Java-Klassen mit Methodenschnittstellen (ohne Implementierung)	Java-Klassen mit Methodenschnittstellen (ohne Implementierung)	<input type="checkbox"/>	

Übung 06:Poker

Poker

Entwerfen Sie das Kartenspiel „Poker“ in der populären Variante „Texas hold 'em“ (Regeln siehe http://en.wikipedia.org/wiki/Texas_hold_'em) und erstellen Sie Use-Cases für den Spielablauf, Sequenzdiagramme zu den Use-Cases, ein Klassendiagramm und die Java-Klassen mit den Methodenschnittstellen (Achtung: keine Implementierung der Methoden gefordert, Implementierung erfolgt erst in Übung 07).

Modellieren Sie den Spielablauf: Eine beliebige Anzahl von Spielern (≥ 3) erhalten am Anfang je 2 Karten. Danach werden 3 Karten aufgedeckt, dann 1 Karte, und am Schluss noch einmal 1 Karte. Dazwischen können die Spieler einen Einsatz setzen. Jeder Spieler hat die Wahl, bei einer Erhöhung des Einsatzes mitzugehen, aus dem Spiel auszusteigen oder zu erhöhen.

Das System unterscheidet zwischen menschlichen Spielern und Computer-Gegnern. Die Ein- und Ausgabe soll über die Konsole erfolgen.

Aufgabenstellungen

- Aufgabe 1) (6 Punkte)
Erstellen Sie Use-Cases für die Initialisierung und für die Spielrunden (vom Ausgeben der Karten bis zum Showdown des Spiels). Treffen Sie dazu sinnvolle Annahmen über die Anzahl der Spieler und den Spielverlauf. Fügen Sie den Hauptszenario auch einige alternative Spiel-Abläufe hinzu. Modellieren Sie eventuell mehrere mögliche Abläufe.
- Aufgabe 3) (6 Punkte)
Erstellen Sie Sequenzdiagramme für die Use-Cases aus Aufgabe 1.
- Aufgabe 2) (6 Punkte)
Erstellen Sie ein Klassendiagramm.
- Aufgabe 4) (6 Punkte)
Aus den Sequenzdiagrammen und den Klassendiagramm sollen Sie die Java-Klassen mit den Methodenschnittstellen definieren (keine die Implementierung der Methoden; diese erfolgt dann in Übung 07).

Vorgabe:

Für die Implementierung sind folgende Klassen bereits vorgegeben, die Sie verwenden sollen:

- Suit: Enumeration für die Farben (Spades, Hearts, Diamonds, Clubs)
- Kind: Enumeration für die Kartenart (Ace, King, Queen, Jack, Ten, ..., Two)
- Card: Enumeration aller Pokerkarten (*S_A(Ace, Spades)*, *H_A(Ace, Hearts)*, *D_A(Ace, Diamonds)*, *C_A(Ace, Clubs)*, *S_K(King, Spades)*, *H_K(King, Hearts)*, *D_K(King, Diamonds)*, *C_K(King, Clubs)*, ..., *C2(Two, Clubs)*)
- PokerHand: Enumeration aller Kombinationen (HighCard, OnePair, TwoPairs, ThreeOfAKind, Straight, Flush, FullHouse, FourOfAKind, StraightFlush). Diese Klasse enthält zwei wichtige Methoden, mit der Kartenhände bestimmt und verglichen werden können:
 - **public static** PokerHand findBestPokerHand(List<Card> cardsList) – Findet aus einer Liste von Karten die höchste Kombination
 - **public static int** compareHands(List<Card> hand1, List<Card> hand2) – Vergleicht zwei Listen von Karten

Einschränkung:

Es geht nur um den prinzipiellen Spielablauf (Karten geben, Einsätze erhöhen), nicht um eine vollständige Abdeckung der gesamten Regeln. Insbesondere müssen Sie folgenden Fall nicht abdecken:

- Side-Pot: Kann ein Spieler nicht mehr den Einsatz leisten, kann er „All in“ gehen und es wird für diesen ein Side-Pot erstellt.

Anleitungen:

- Achten Sie auf einen guten Entwurf. Achten Sie darauf, dass die Sequenzdiagramme die Use-Cases abdecken und dass die Klassen die entsprechenden Methoden zur Verfügung stellen. Der Entwurf dieser Übung soll in Übung 07 implementiert werden. Ein guter und möglichst vollständiger Entwurf sollte den Aufwand bei Übung 07 wesentlich reduzieren.
- Die Use-Case-Beschreibungen sollen möglichst detailliert den Spielablauf darstellen. Use-Case-Beschreibungen beziehen sich immer auf einen konkreten Ablauf und stellen daher nicht das Gesamtverhalten dar. Die müssen also einen sinnvollen Ablauf annehmen, der von der Ausgabe der Karten bis zum Ende des Spiels nach allen Wett-Runden reicht. Dieser stellt das Hauptszenario dar. Fügen Sie dem Hauptszenario einige alternative Abläufe hinzu, z.B. wenn ein Spiel früher endet, weil alle bis auf einen Spieler aus dem Spiel ausgestiegen sind. Modellieren Sie eventuell mehrere Use-Cases. Modellieren Sie auch die Initialisierung des Spiels als eigenen Use-Case. Modellieren Sie eventuell das Setzen als einen eigenen Unter-Use-Case.
- Das Sequenz-Diagramme soll den gleichen Ablauf wie die Use-Cases enthalten. Zeigen Sie in den Sequenzdiagrammen die Methodenaufrufe und die notwendigen Parameter zwischen den Objekten. Alle im Sequenzdiagramm verwendeten Methoden müssen auch im Klassendiagramm und in den Java-Klassen spezifiziert sein.
- Das Klassendiagramm soll aus einigen Klassen bzw. Interfaces bestehen. Beschränken Sie sich auf die öffentliche Schnittstelle der Klassen und Interfaces.
- Verwenden Sie zum Zeichnen der Diagramme entweder ein UML-Werkzeug oder ein anderes graphisches Werkzeug (keine handgezeichneten Diagramme abgeben). Achten Sie auch auf ein gutes Layout der Diagramme.