



Übungsblatt 2: Anwendungsschicht (Blackjack)

Praktikum

Hinweis: Diese Aufgaben werden bewertet. Im Praktikum können insgesamt **100 Punkte** erreicht werden. Zum Bestehen werden **75%** der Punkte benötigt. Für dieses Thema gibt es **20 Punkte**. Durch die Zusatzaufgaben können **20** zusätzliche Punkte erreicht werden

Hinweis: Die Lösungen zu diesem Übungsblatt müssen bis zum **18.05.2022 23:55 Uhr** **15.05.2022 23:55 Uhr** vorliegen.

- Erstellen Sie ein zip-Archiv mit dem Namen Ihrer Rhocode-Gruppe nach dem Muster (2022netzeXX) und laden es im read.MI unter der entsprechenden Abgabe hoch, die einige Tage zuvor angelegt wird. Enthalten sein soll a) die Beschreibung des Protokolls b) Wireshark-Traces, die belegen, dass das Protokoll funktioniert.
- Organisieren Sie den Quellcode Ihrer Lösung(en) in Git-Repositories mit aussagekräftigem Namen (z.B. Aufg1_Server, Aufg1_Client) auf Rhocode, sodass wir diese den Aufgaben zuordnen können. Die Repositories sollen jeweils eine README-Datei mit Installations- bzw. Startanweisungen enthalten. **Änderungen in den Repositories, die nach Ablauf der Frist erfolgen, werden nicht berücksichtigt.**

Sie sollen eine Netzwerkanwendung für das beliebte Kartenspiel Blackjack entwickeln. Ziel des Spiels ist es, mit zwei oder mehr Karten näher an 21 Punkte heranzukommen als der Dealer bzw. die anderen Mitspieler, ohne dabei die 21 Punkte zu überschreiten.

Die Wertigkeit der Karten entsprechen der Anzahl an Augen. Bildkarten, also Könige, Damen und Buben, zählen jeweils zehn Punkte. Das Ass zählt so, wie es für den Spieler am günstigsten ist (entweder ein oder elf Punkte). Der Wert wird festgelegt, sobald der Spieler keine weitere Karte mehr kauft.

Aufgabe 1 (Blackjack gegen den Dealer, 20 Punkte)

Im Rahmen dieser Aufgabe soll ein Spieler gegen einen computergesteuerten Dealer spielen können. Nachdem der Spieler die Bereitschaft zum Spielen signalisiert hat und eine entsprechende Bestätigung vom Dealer erhalten hat, fragt der Spieler nach einer Karte. Es soll aus einem zufällig aufgebauten (gemischten) Kartenstapel mit 4x13 Karten gezogen werden, sodass jede Wertigkeit genau vier mal vorkommen kann. Der Dealer zieht dann auch eine Karte für sich. Dieses Szenario wiederholt sich, bis der Spieler entscheidet, dass er genug Punkte hat. Damit fordert er den Dealer dazu auf, die Karten zu öffnen und das Spiel endet. Am Ende des Spiels wird dem Spieler mitgeteilt, ob dieser gewonnen oder verloren hat.

Das Spiel endet vorzeitig, wenn der Spieler oder Dealer die 21 Punkte überschreiten. In diesem Fall verliert die Partei, welche die 21 Punkte überschritten hat.

- Entwickeln Sie ein Protokoll für die Kommunikation zwischen Client und Server, welches auf TCP als Transportprotokoll basiert. Der Dealer wird vom Server gesteuert. Weiterhin soll pro Verbindung ein Spiel stattfinden, ein erneuter Spielstart nach Ende eines Spiels über die selbe Verbindung muss demnach nicht berücksichtigt werden.
- Implementieren Sie Client und Server mit der in der Vorlesung vorgestellten Socket-API in Java und ermöglichen Sie die Steuerung des Spiels **über die Kommandozeile**.
- Verbessern Sie Ihren Server so, dass mehrere Spieler gleichzeitig mit einem Dealer spielen können. Dabei sollen die Spieler nicht gegeneinander spielen, sondern unabhängig voneinander mit einem Dealer.

Zusatzaufgabe 1 (Blackjack mit anderen Spielern, 10 Punkte)

Traditionell wird Blackjack im Kasino mit mehreren Spielern gespielt. Sie sollen Ihre bisherige Lösung erweitern, sodass in einem Spiel mehrere menschliche Spieler gegeneinander spielen können. Der computer-gesteuerte Dealer nimmt zur Vereinfachung **nicht** am Spiel teil.

Nachdem sich alle Spieler zusammengefunden und signalisiert haben, dass sie bereit sind, erhalten sie nacheinander eine Karte. Anschließend entscheiden die Spieler nacheinander, ob sie eine neue Karte möchten oder stehen bleiben. Wenn ein Spieler mit seiner neuen Karte über die 21 Punkte kommt, scheidet dieser aus dem Spiel aus. Haben alle Spieler eine Entscheidung getroffen, wiederholt sich der Prozess mit den verbleibenden Spielern, die noch nicht stehen geblieben sind, so lange, bis alle stehen geblieben oder ausgeschieden sind. Anschließend werden die Karten aufgedeckt und es werden die Gewinner ermittelt. Es gewinnt der Spieler, der die meisten Punkte hat. Bei einem Gleichstand gibt es mehrere Gewinner. Es kann übrigens in einem bestimmten Szenario auch keinen Gewinner geben.

Überarbeiten Sie Ihr Protokoll und passen Sie die Implementierung entsprechend an. Es sollen dabei mehrere Spiele parallel stattfinden können. Nach Ende eines Spiels können Sie die Verbindung zu allen Teilnehmern des Spiels trennen.

Zusatzaufgabe 2 (Benutzungsoberfläche, 10 Punkte)

Als Alternative zur Kommandozeile soll eine Benutzungsoberfläche mit JavaFX für Aufgabe 1 entwickelt werden. Dabei sollen folgende Aspekte berücksichtigt werden:

- Anzeige der eigenen Karten mit ihren Wertigkeiten
- Verdeckte Anzeige der Karten des Dealers
- Interaktionsmöglichkeiten um eine neue Karte zu ziehen, stehen zu bleiben oder das Spiel zu beenden
- Anzeige des Ergebnisses nach Spielende sowie Anzeige der Karten des Dealers

In eventuell besuchten vergangenen Veranstaltungen (EIBO) haben Sie gelernt, dass eine gute Benutzungsoberfläche nicht „einfriert“, wenn blockierende Methoden aufgerufen werden. Weiterhin wissen Sie auch, dass asynchrone Programmierung einen koordinierten (synchronisierten) Zugriff auf gemeinsame Datenstrukturen erfordert. Bitte berücksichtigen Sie daher diese Qualitätsmerkmale.