

Unit Tests

Es wurden 3 Testklassen geschrieben.

MoneyTest: Testet die Addition, Subtraktion und Multiplikation der Money Klasse

EvaluationHandlerTest: Testet alle Pfadmöglichkeiten der EvaluationHandler Methoden. Es wird das Evaluieren des Anfangs des Spieles, das Evaluieren eines normalen Spieles und das Evaluieren eines split Spieles getestet. Für die jeweiligen Evaluierungen werden alle möglichen Pfade und Pfadeingänge getestet. Dadurch besitzen die Tests aus EvaluationHandlerTest eine 100% Code Coverage des EvaluationHandlers

CardHandlerTest: Testet alle Pfade und Methoden der CardHandler Klasse. Für diese Testklasse wurde für die Tests zum Überprüfen der checkAce() Methode eine Mockklasse geschrieben, welche es erlaubt checkAce() zu testen. Die Tests aus CardHandlerTest decken alle Methoden ab. Des Weiteren wurden für Methoden mit mehreren Pfaden mehrere Tests geschrieben, um eine 100% Code Coverage zu erreichen.

Mockklasse CardHandlerMockForCheckAce: Diese Mockklasse dient dem einfachen Zugang zu der checkAce() Methode des CardHandlers. checkAce() ist eine private Methode und ist daher nicht direkt in JUnit ansprechbar. Des Weiteren baut checkAce() auf den aktuellen Status des deck Fields der Klasse CardHandler auf. Um dies zu umgehen und checkAce() testbar zu machen wurde in CardHandlerMockForCheckAce die Methode givePlayingCard() von CardHandler überschrieben. In der überschriebenen Methode wird das Deck so präpariert, dass als nächste Karte ein Ass gezogen wird. Dadurch kann nun checkAce() über givePlayingCard() getestet werden.

Die Klasse Money wurde auf ihre Subtraktion, Addition und Multiplikation getestet, um sicherzustellen, dass die Beträge für den Spieler richtig berechnet werden. Würden die Beträge durch Änderungen oder Bugs nicht mehr richtig berechnet werden, so wäre dies ein großer Fehler. Durch die Tests von Money ist nun sichergestellt, dass bei allen Änderungen getestet werden kann, ob die Beträge noch richtig berechnet werden.

Die Klasse CardHandler wurde umfassend mit Tests abgedeckt, da dies eine Klasse ist, welche sehr wichtig für den Spielablauf ist. Funktioniert die Ausgabe der Karten an den Spieler und den Croupier nicht mehr, so fehlt die Grundlage eines Kartenspiels wie BlackJack. Deshalb wurde CardHandler mit sehr großer Testabdeckung versehen, um zu gewährleisten, dass die Grundlagen der Anwendung nicht durch Änderungen zerstört werden können.

EvaluationHandler wurde ebenfalls mit einer großen Testabdeckung versehen, da dies eine Kernklasse der Anwendung ist. EvaluationHandler wertet den Ausgang und den Anfang einer BlackJack runde aus. Wird nicht mehr richtig ausgewertet, wie ein Spiel ausging, so bekommt der Spieler bei einem Sieg eventuell kein Geld ausgezahlt, was ein fataler Fehler in der Anwendung wäre. Gegenteilig könnte ein Spieler bei einer Niederlage eventuell Geld nicht abgezogen bekommen. Um diese Fehler zu verhindern, wurde eine große Testabdeckung für EvaluationHandler geschrieben.