Cup O’ Java’s Deliverable Three:

Twenty-One Card Game

Arianna Biernacki, Devi Sharma

Department of Applied Science and Technology, Sheridan College

SYST17796 Fundamentals of Software Design and Development

Amandeep Sidhu

April 20, 2020

Table of Contents

Class Diagram 3

Test + Scripts 3

GitHub Source Code 3

GitHub Test Code 3

Manual Test Instructions 4

Test Report 4

**Class Diagram**

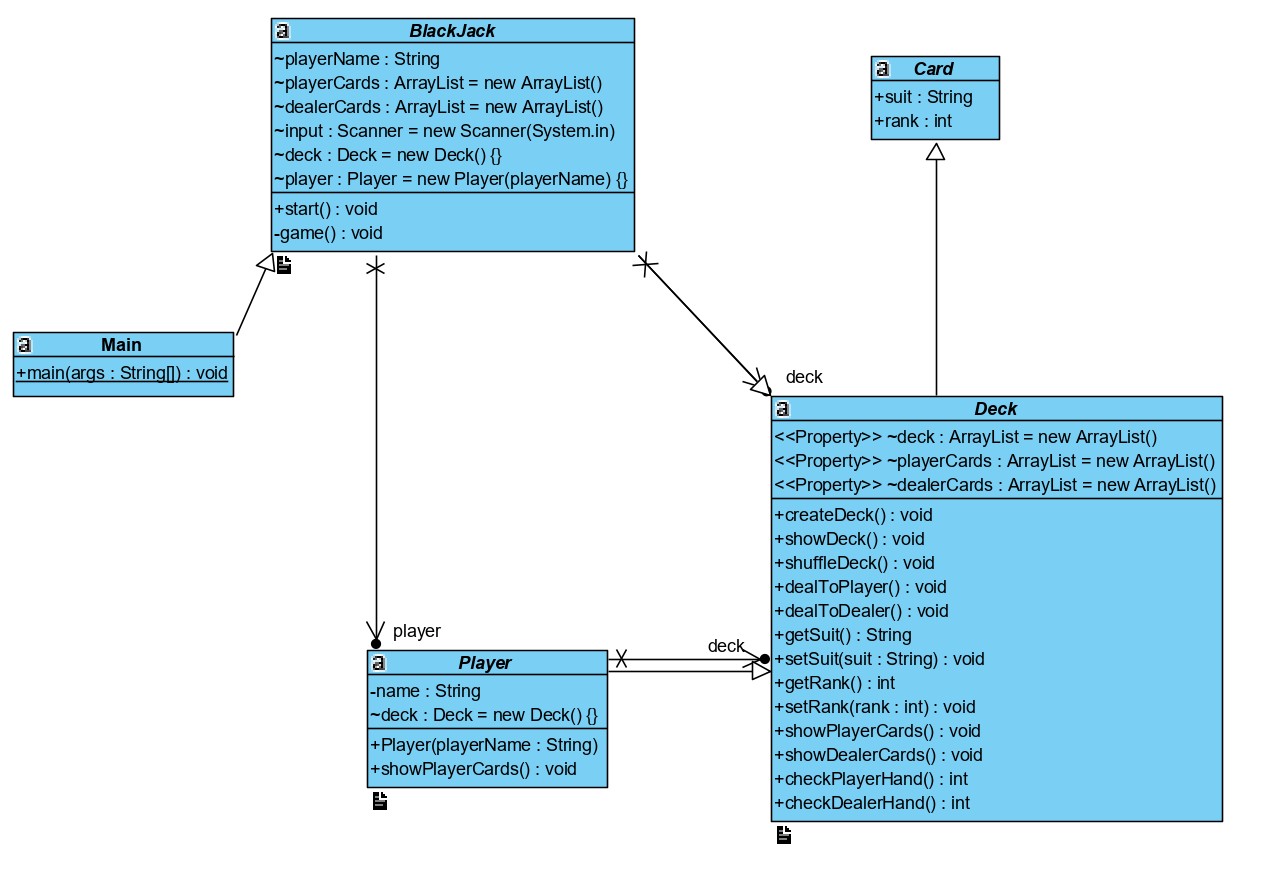


Figure 1: Updated Class Diagram

**Test + Scripts**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | Use Case | Test Method (ClassName.methodName) | Status (Date) |
| Player enters if they want to hit or stay | “Regular play” | BlackJackTest.userInputGood() | Pass (April 19, 2020) |
| Player enters if they want to hit or stay | “Regular play” | BlackJackTest.userInputBad() | Pass (April 19, 2020) |
| Player enters if they want to hit or stay | “Regular play” | BlackJackTest.userInputBoundary() | Pass (April 19, 2020) |

**GitHub Source Code**

<https://github.com/Dsharma624/Project1/tree/master/src/ca/sheridancollege/project>

**GitHub Test Code**

<https://github.com/Dsharma624/Project1/tree/master/test/ca/sheridancollege/project>

**Manual Test Instructions**

We did manual testing on the createDeck() method by using the showDeck() method to print out the deck. We tested the shuffleDeck method by using the showDeck() method to print the deck after it has been shuffled to know that it is working properly.

We tested the dealToPlayer()/dealToDealer() method by using the showPlayerCard()/showDealerCards() methods to print the hand, so we can see that the cards are being dealt from the deck and entering the arrayList of the player’s hand and dealer’s hand.

The checkPlayerHand()/checkDealerHand() is used to calculate the ranks of each card to come to a total. This is manually tested by adding the rank in each hand and ensure the checkPlayer/DealerHand() method correctly calculates the total.

**Test Report**

Our Junit testing tested the input from a user to determine if they wanted to hit or stay. We used good, bad, and boundary methods to determine the validity of the test. For “good”, it took the string “Hit”. The expected result is true, and the test passed. For “bad”, the user entered a special character and so the test failed. For “Boundary”. The user entered “s” for stay, and the test passed as expected.

Our manual testing also passed the expected requirements. With many practice rounds under a controlled environment, we tested the many different outcome the game can have and compared it to our expected outcome.