Nahom Haile

November 12, 2020

Programming III (420-G30-HR)

Assignment 3 – War Game

Deck Class:

Creating a deck of 52 cards

|  |  |  |
| --- | --- | --- |
| Rules/Constraints | Valid Equivalence Classes | Invalid Equivalent Classes |
| Deck size | 1. 52 card objects in the queue | 1. less than 52 cards in the queue 2. More than 52 cards in the queue |

|  |  |
| --- | --- |
| Test Values: | Test Equivalence # Mapping |
| Size = 52 | 1 |
| Size = 53 | 3 |
| Size = 51 | 2 |

Shuffling the deck

|  |  |  |
| --- | --- | --- |
| Rules/Constraints | Valid Equivalence Classes | Invalid Equivalent Classes |
| Cards are random in the queue | 1. The cards in the queue are not in order | 1. The cards in the queue are in order |

|  |  |
| --- | --- |
| Test Values: | Test Equivalence # Mapping |
| deckQueue is shuffled | 1 |
| deckQueue is in order | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Excepted Result |
| Let deck = new Deck() | Create a new Deck object | deckQueue = 52 cards | A new Deck object is created |
| deck.deckQueue.size() | Get the size of the | deckQueue = 52 cards | 52 |

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Excepted Result |
| Let deck = new Deck() | Create a new Deck object | deckQueue = 52 cards | A new Deck object is created |
| deck.deckQueue.shuffle() | To shuffle the deck | deckQueue = 52 cards |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Excepted Result |
| Let deck = new Deck() | Create a new Deck object | deckQueue = 52 cards | A new Deck object is created |
| deck.deckQueue.deal() | Dealing a card | deckQueue = 51 cards | a card object |

War Class

Getting the winning hand

|  |  |  |
| --- | --- | --- |
| Rules/Constraints | Valid Equivalence Classes | Invalid Equivalent Classes |
| Card rank is greater | 1. Card1 rank is higher than card2 rank | 1. Card1 rank is less than card2’s rank |

|  |  |
| --- | --- |
| Test Values: | Test Equivalence # Mapping |
| Card1 rank = 5  Card rank = 9 | 2 |
| Card1 rank = 9  Card rank = 5 | 1 |

Checking if it is a war

|  |  |  |
| --- | --- | --- |
| Rules/Constraints | Valid Equivalence Classes | Invalid Equivalent Classes |
| Comparing two card objects | 1. Two card object | 1. Not card objects |
| Ranks have to be equal | 1. Equal ranks | 1. Unequal ranks |

|  |  |
| --- | --- |
| Test Values: | Test Equivalence # Mapping |
| Card1 rank = 4  Card2 rank = 4 | 1, 3 |
| Card1 rank = 9  Card2 rank = 2 | 4 |
| Player and a card | 2 |

Test Case 1: starting up the game

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Excepted Result |
| Let deck = new Deck() | Create a new Deck object | deckQueue = 52 cards | A new Deck object is created |
| Let war = new War() | Create a new war object | Pile1: empty  Pile2: empty | A new war object is created |
| War.start() | The two players have their hands created | Pile1: 26 cards  Pile2: 26 cards |  |
| War.countCards(pile1) | Counting the cards in pile1 | Unchanged | 26 |
| War.countCards(pile2) | Counting the cards in pile2 | Unchanged | 26 |

Test Case 2 – playing the first round of the game regualry

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Excepted Result |
| Let deck = new Deck() | Create a new Deck object | deckQueue = 52 cards | A new Deck object is created |
| Let war = new War() | Create a new war object | Pile1: empty  Pile2: empty | A new war object is created |
| War.start() | The two players have their hands created | Pile1: 26 cards  Pile2: 26 cards |  |
| War.play() | Taking the first from the top of both piles and determining who one | Pile1: 25 cards  Pile2: 25 cards  Card1 = A of Clubs  Card2 = 5 of diamonds | Player 1 won this hand  Player 1 has 27 cards left  Player 2 has 25 card left |

Test Case 3 – hitting a tie

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Excepted Result |
| Let war = new War() | Create a new war object | Pile1: empty  Pile2: empty | A new war object is created |
| War.start() | The two players have their hands created | Pile1: 26 cards  Pile2: 26 cards |  |
| War.play() | Taking the first from the top of both piles and determining who one | Pile1: 25 cards  Pile2: 25 cards  Card1 = A of Clubs  Card2 = A of diamonds | Player 1 plays JS. Player 2 plays JC.  It's a tie! Each player lays 3 cards face down  Kitty has 8 cards.  Player 1 has 21 cards left.  Player 2 has 31 cards left. |

Test Case 4 – down to last card and the cards

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Purpose | Object State | Excepted Result |
| War.play() | Taking the first from the top of both piles and determining who one | Pile1: 51 cards  Pile2: 1 card  Card1 = 5 of Clubs  Card2 = 5 of diamonds | Player 1 plays 5C. Player 2 plays 5D. Player 2 losses because he has one card left. |