

2214 Report Shashank Mondrati

Stack Implementations	push	pop	peek	isEmpty
LinkedStack	O(1)	O(1)	O(1)	O(1)
ArrayStack	O(n)	O(N)	O(1)	O(N)
Queue Implementation	enqueue	dequeue	isEmpty	size
LinkedQueue	O(N)	O(1)	O(1)	O(1)
ArrayCircularQueue	O(1)	O(1)	O(1)	O(1)

QueueADT has few variables in it. With the first queue of object E initialized but not declared yet. Void enqueue has E target which adds to the back of the target, and the boolean which return true or false is Empty tells us if the QueueADT is empty or not. For dequeue throws EmptyCollection Exception, and returns without removing the element that is located at the top of the queue, and returns the element at the top of the queue. Int Size tells us how many elements are in the StackADT.

StackADT has few variables in it they are not initialized, but they will be. Starting with isEmpty, tells us if the StackADT is empty or not with true or false. The E peek method, takes or gets or accesses the first element in the stack, but changes nothing, and throws an exception if its empty. The pop method removes the element and returns the top item in the stack, and throws an exception. The push method adds target to the top of the stackADT, and the size method returns how many elements are in the stackADT.

Card java file has few final int variables, like the suites and ranks of a card. Spades set to 0, hearts set to 1, clubs to 3 and diamonds to 2, and ACE is always set to 1. When it comes to higher cards like jack king and queen they are to set to 11, 12, and 13. The boolean equals methods has this and that variables, if this equals that it should return true for that rank and suit, else it should return false as it is written. And setters and getters for ranks and suits of the card.

Deck java file is different, the private variable Card has [] card in it, and private int size for the number of cards in the stack or this deck, and the default constructor has declared itself to 52 set elements, and size set to 0. There is a nested for loop to create all 52 cards in order without the joker card. Suit equaling card of spades, and suit is less or equal to clubs and incrementing, the same goes for ace and king, and setting them to cards of size equaling to a new card based on their rank and suite. The deal method decrements the size or the num of elem in the deck, and returns cards size. The isEmpty returns if the size equals 0 or not. The shuffle method shuffles the cards using the swap methods mixed in a for loop, and the swap method swaps the indexes i and j, i for a card, and j for another card.

In **WarByLinkedLists** java file, there are few methods. There are few imported java from Apps folder. Starting with scanner variable called INPUT, and Queue ADT called hand1 for players 1 pile of cards, and hand2 for players 2 pile of hands. In the default constructor one parameter passed in boolean isShuffled. Intialized and declared hand1 and hand2 as a linkedQueue as a card object, in the while loop while the deck isn't empty, both the hands will enqueue with dealing the deck, the give method gives all of the cards to the winning player. If winner gets the hand 1, the players wins, if not player2 get the winning hand. In a try-catch block, while the stacks aren't empty winner enques by popping stacks1 and 2. In the play method, while both the hands aren't empty, if the player1's hand is empty player2 wins, and reverse for the nextplayer. In the next play round, plays a single round, starting with stacks 1 and 2 as linked stack, and pushing both the stacks by dequeuing both hands. In a do while statement, if the card1's rank is better than the second the player wins, or card 1's rank is lesser than the second.

IN this assignment I learnt how to enqueue things and I learnt how to dequeue things, they were not easy to understand since I had help from all sorts like TA, tutoring from the UCAE. The most interesting in this assignment were understanding what the pop, push and peek methods are. I had a confusion what the pop, push and peek methods are. But now I have an understanding what those methods mean. Also, I have a clear understanding how to instantiate the array based data collection, and how to insert an element to the end of the queue. I learnt how to do the expand the capacity, and increment the size variable. I also learnt how dequeue, if the stack or list is empty throw an exception. I learnt how to double the length of a data by using the for loop and using the modulus symbol.

In this assignment, there were many challenges I faced. The biggest was understanding the instructions, they were hard for me to understand, I mean in the instructions it says some of the methods have errors, when opened they have no errors on the outside. If it shows, we're being fooled in to doing nothing. I don't blame them for anything, it just is different the instructions and study is whole new thing. The to-do's given in the instructions were easy, and sometimes harder to understand, some of the methods don't even any anything to say. Gives me the impression there's nothing for me to. The easiest way for me to understand to do what to do was read the other codes and do what the undone code needs to be done. This way, I can finish the codes, some to-do's are some to understand in the warbyLinked lists, that took me longest to finish. I had some help by seeing some videos online I was able to finish the project.