CS 1302 Generics, Stack, and Queue

7/21/2015

Fares

1. You were asked to use generics and any of the data structures covered in chapters 20 and 21 to implement your own generic stack and queue. Use the UML diagrams below as a guidance.

|  |  |
| --- | --- |
| MyStack <E> |  |
| -data structure | Select your own data structure to implement stack |
| +MyStack()  +MyStack (c :E[])  +isEmpty():Boolean  +push(item:E):void  +pop():E  +clear():void | Creates an empty stack  Creates a stack with items in an array  Returns true if empty, otherwise, returns false  Pushes an item on top of stack  Returns and removes an item from the top of stack  Reset Stack to empty |

|  |  |
| --- | --- |
| MyQueue <E> |  |
| -data structure | Select your own data structure to implement queue |
| +MyQueue()  +MyQueue (c :E[])  +isEmpty():Boolean  +enqueue(item:E):void  +dequeue():E  +clear():void | Creates an empty queue  Creates a queue with items in an array  Returns true if empty, otherwise, returns false  Inserts an item at end of queue  Returns and removes an item from the front of queue  Reset queue to empty |

1. Create a driver to test the stack and queue.
2. Test your stack and queue with the attached drivers.
3. Attach your Stack and Queue only (drivers are not needed) via Blazeview.