Clemson University – School of Computing

CPSC2150: Software Development Foundations

Lab 5

Written by:

Damion Anderson
Jeremy Holloway

Academic Year: 2018 - 2019

Contents

1. Requirement Analysis

User Story:

- As a user of MyQueue, I will be able to choose the data structure type to use the most beneficial type for my system.

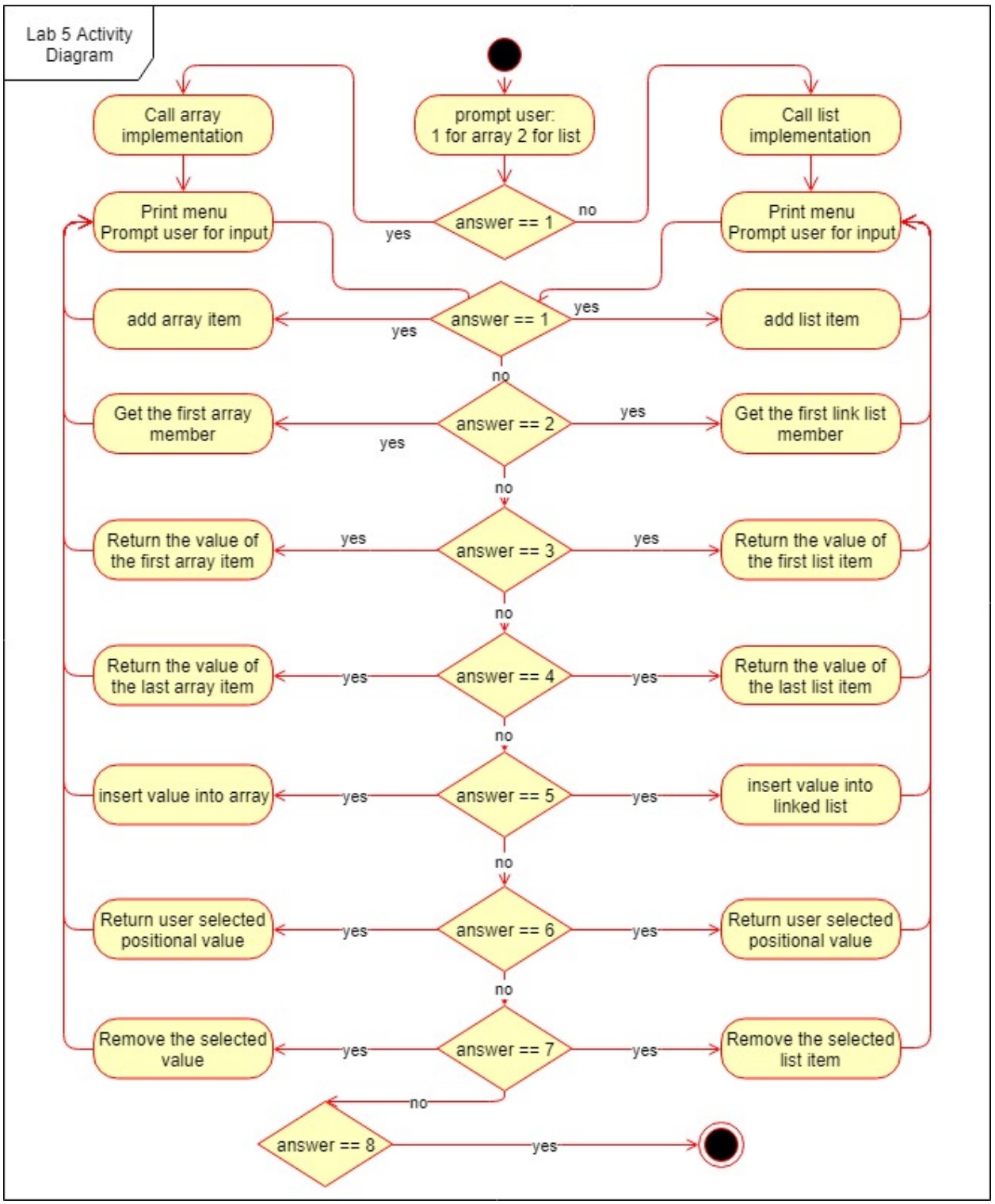
Functional Requirements:

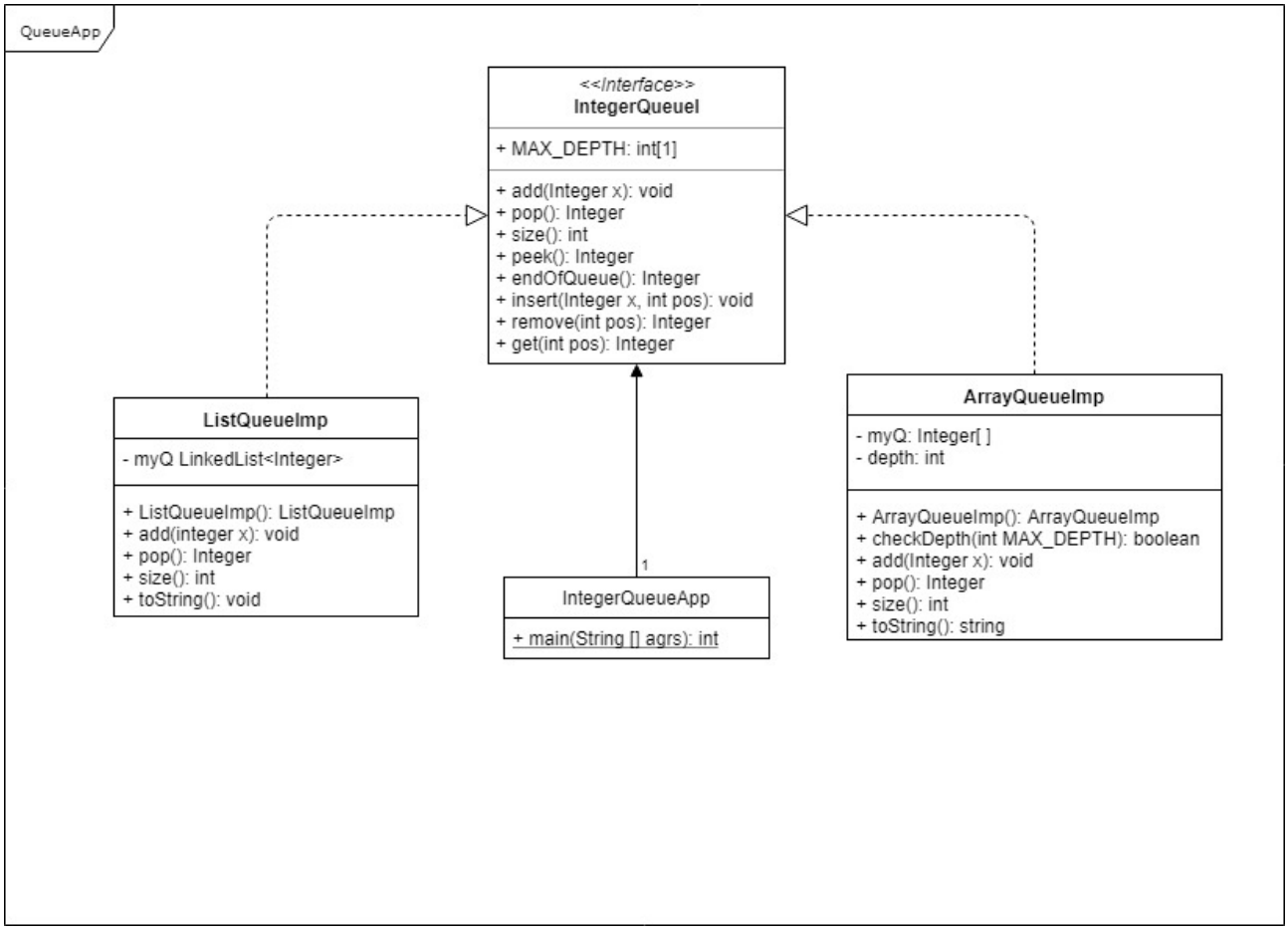
- A user will be able to choose a preferred method implementing a queue.
- A user will be able to add and remove items to or from their preferred storage method.
- A user will be able to display the number of items in their storage structure.
- A user will be able to see but not change the first or last item of the queue.
- A user will be able to insert or remove items at a specific position.
- A user will be able to get any value by specifying its position.
- A user will be able to view all items in the data structure after every use.

Non-Functional Requirements:

- The program is modular. If the client desires to remove certain functions, they may be commented out. These methods will work on any system with JDK8 installed.
- The program is written in Java. The system must have JDK8 installed. The program was designed for a Unix based environment.
- All interfaces between the user and the app will be displayed via the terminal window.
- The app is delivered via one main.java file. The user must compile via the javac Main.java command and run the app via the java main command.

2. Design


Lab 5 Activity Diagram
Call array implementation
prompt user: 1 for array 2 for list
Call list implementation
Print menu Prompt user for input
answer == 1
yes
no
Print menu Prompt user for input
add array item
answer == 1
yes
yes
add list item
no
Get the first array member
answer == 2
yes
yes
Get the first link list member
no
Return the value of the first array item
answer == 3
yes
yes
Return the value of the first list item
no
Return the value of the last array item
answer == 4
yes
yes
Return the value of the last list item
no
insert value into array
answer == 5
yes
yes
insert value into linked list
no
Return user selected positional value
answer == 6
yes
yes
Return user selected positional value
no
Remove the selected value
answer == 7
yes
yes
Remove the selected list item
no
answer == 8
yes


QueueApp
<<Interface>>
IntegerQueueI
+ MAX_DEPTH: int[1]
+ add(Integer x): void
+ pop(): Integer
+ size(): int
+ peek(): Integer
+ endOfQueue(): Integer
+ insert(Integer x, int pos): void
+ remove(int pos): Integer
+ get(int pos): Integer
ListQueueImp
- myQ LinkedList<Integer>
+ ListQueueImp(): ListQueueImp
+ add(integer x): void
+ pop(): Integer
+ size(): int
+ toString(): void
ArrayQueueImp
- myQ: Integer[ ]
- depth: int
+ ArrayQueueImp(): ArrayQueueImp
+ checkDepth(int MAX_DEPTH): boolean
+ add(Integer x): void
+ pop(): Integer
+ size(): int
+ toString(): string
1
IntegerQueueApp
+ main(String [] agrs): int

3. Testing Methods

1. Sample Input/Output

```
   Enter 1 for array implementation or 2 for List implementation
   1
   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 2
   Queue is empty!
   Queue is:

   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 3
   Queue is empty!
   Queue is:

   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 4
   Queue is empty!
   Queue is:
```

```
   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 6
   Queue is empty!
   Queue is:

   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 7
   Queue is empty!
   Queue is:

   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 1
   What number to add to the Queue?
   42
   Queue is:
   42

   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 1
```

```
   What number to add to the Queue?
   37
   Queue is:
   42
   37
   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 1
   What number to add to the Queue?
   36
   Queue is:
   42
   37
   36

   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
   6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 2
   Next number is 42
   Queue is:
   37
   36
```

```
   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
   5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 1
   What number to add to the Queue?
   42
   Queue is:
   37
   36
   42

   Select an option:
   1. Add to the Queue
   2. Get next number from the Queue
   3. Peek at the front of the Queue
   4. Peek at the end of the Queue
   5. Insert in the Queue
   6. Get a position in the Queue
   7. Remove from a position in the Queue
   8. Quit
   3
   Peek: 37
   Queue is:
   37
   36
   42

   Select an option:
1. Add to the Queue
2. Get next number from the Queue
   3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 4
   Peek at end: 42
   Queue is:
   37
   36
   42
```

```
   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 6
   What position to get from the Queue?
   7
   Not a valid position in the Queue!
   What position to get from the Queue?
   2
   36 is at position 2 in the queue
   Queue is:
   37
   36
   42

   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 5
   What number to add to the Queue?
   17
   What position to insert in?
   9
   Not a valid position in the Queue!
   What position to insert in?
   1
   Queue is:
   17
   37
   36
   42
```

```
   Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit 5
   What number to add to the Queue?
   19
   What position to insert in?
   5
   Queue is:
   17
   37
   36
   42
   19

   Select an option:
   1. Add to the Queue
   2. Get next number from the Queue
   3. Peek at the front of the Queue
   4. Peek at the end of the Queue
   5. Insert in the Queue
   6. Get a position in the Queue
   7. Remove from a position in the Queue
   8. Quit
   7
   What position to remove from the Queue?
   1
   17 was at position 1 in the queue
   Queue is:
   37
   36
   42
   19
```

```
Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit
7
What position to remove from the Queue?
5
Not a valid position in the Queue!
What position to remove from the Queue?
4
19 was at position 4 in the queue
Queue is:
37
36
42

Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit
7
What position to remove from the Queue?
2
36 was at position 2 in the queue
Queue is:
37
42
```

```
Select an option:
1. Add to the Queue
2. Get next number from the Queue
3. Peek at the front of the Queue
4. Peek at the end of the Queue
5. Insert in the Queue
6. Get a position in the Queue
7. Remove from a position in the Queue
8. Quit
9
Not a valid option!
Queue is:
37
42

Select an option:
2. Add to the Queue
3. Get next number from the Queue
4. Peek at the front of the Queue
5. Peek at the end of the Queue
6. Insert in the Queue
7. Get a position in the Queue
8. Remove from a position in the Queue
9. Quit
8

Process finished with exit code 0
```

4. Deployment

This program will include a make file which can be used for three operations make, make run, make clean. The makefile must be in the same folder as cpsc2150 and not inside of it or any other subfolder.

make:
Expected output:

javac cpsc2150/MyQueue/IntegerQueueApp.java

make run:
Expected output:

java cpsc2150.MyQueue.IntegerQueueApp
Would you like to use an array or a list?

make clean:
Expected output:

rm -f *.class