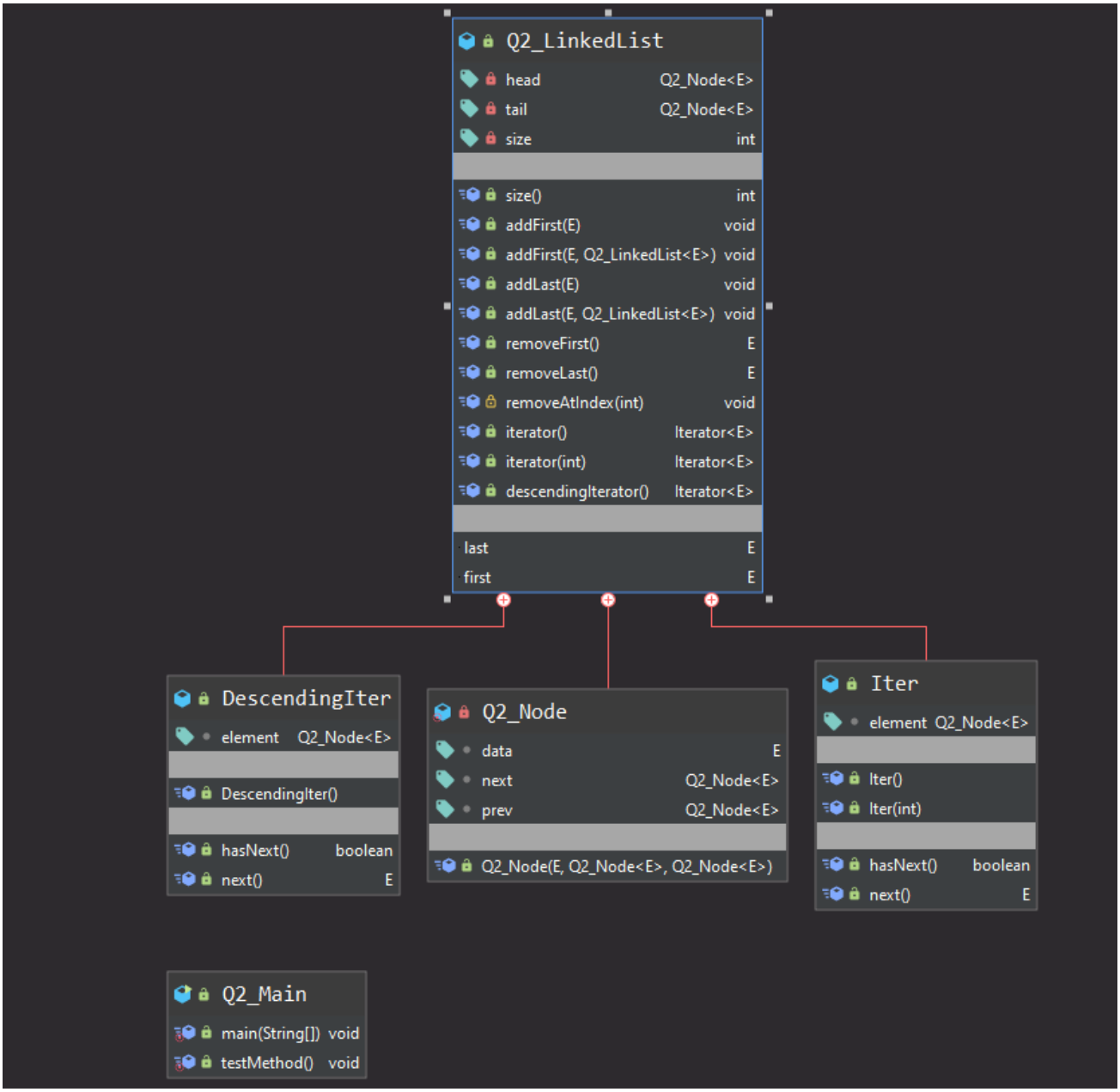
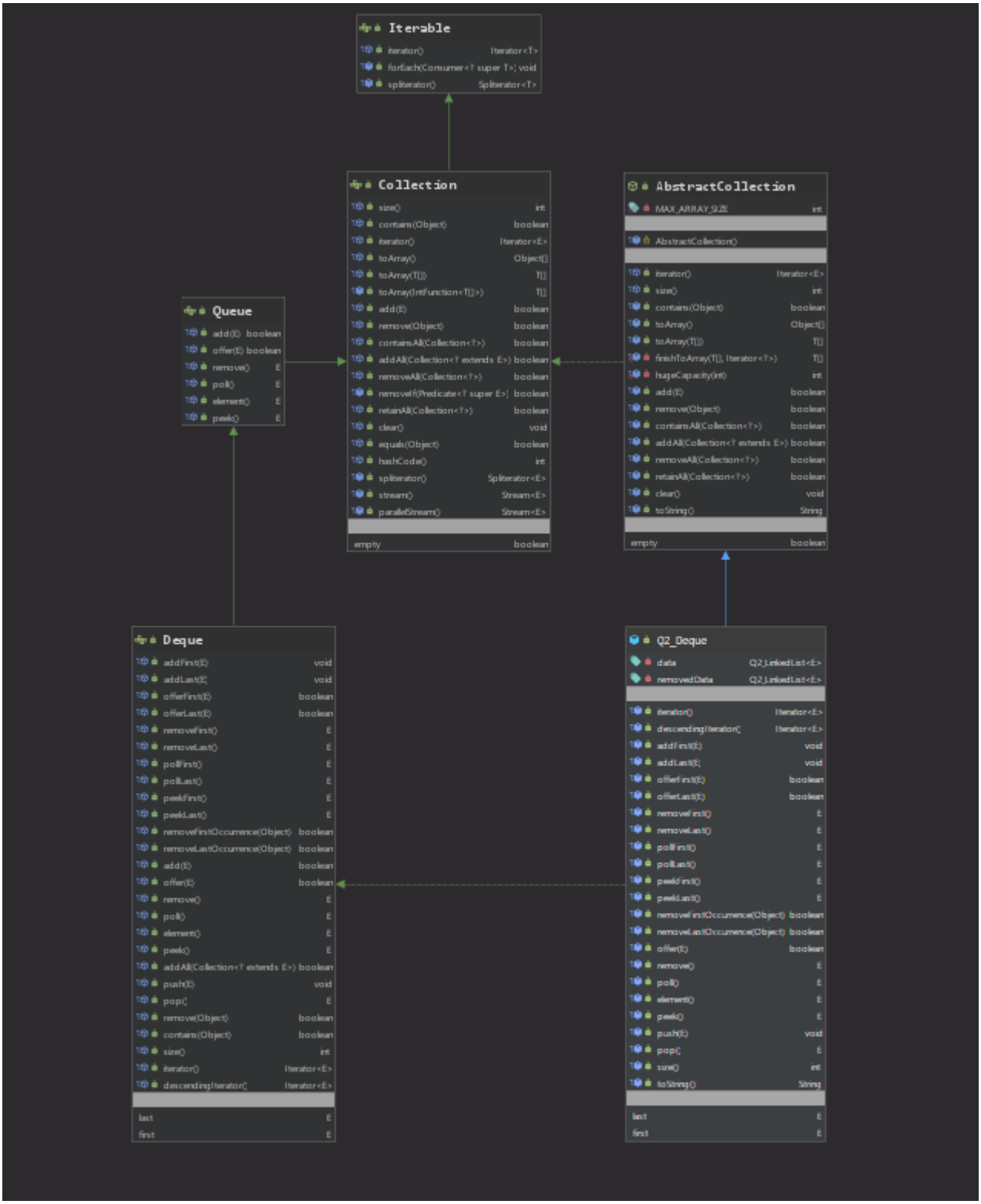
GIT Department of Computer Engineering
CSE 222/505 - Spring 2020
Homework #4 Report

Harun ALBAYRAK
171044014

1)Class Diagrams

Q2)




Iterable
iterator() Iterator<T>
forEach(Consumer<? super T>) void
spliterator() Spliterator<T>
Collection
size() int
contains(Object) boolean
iterator() Iterator<E>
toArray() Object[]
toArray(T[]) T[]
toArray(IntFunction<T[]>) T[]
add(E) boolean
remove(Object) boolean
containsAll(Collection<?>) boolean
addAll(Collection<? extends E>) boolean
removeAll(Collection<?>) boolean
removeIf(Predicate<? super E>) boolean
retainAll(Collection<?>) boolean
clear() void
equals(Object) boolean
hashCode() int
spliterator() Spliterator<E>
stream() Stream<E>
parallelStream() Stream<E>
empty boolean
AbstractCollection
MAX_ARRAY_SIZE int
AbstractCollection()
iterator() Iterator<E>
size() int
contains(Object) boolean
toArray() Object[]
toArray(T[]) T[]
finishToArray(T[], Iterator<?>) T[]
hugeCapacity(int) int
add(E) boolean
remove(Object) boolean
containsAll(Collection<?>) boolean
addAll(Collection<? extends E>) boolean
removeAll(Collection<?>) boolean
retainAll(Collection<?>) boolean
clear() void
toString() String
empty boolean
Queue
add(E) boolean
offer(E) boolean
remove() E
poll() E
element() E
peek() E
Deque
addFirst(E) void
addLast(E) void
offerFirst(E) boolean
offerLast(E) boolean
removeFirst() E
removeLast() E
pollFirst() E
pollLast() E
peekFirst() E
peekLast() E
removeFirstOccurrence(Object) boolean
removeLastOccurrence(Object) boolean
add(E) boolean
offer(E) boolean
remove() E
poll() E
element() E
peek() E
addAll(Collection<? extends E>) boolean
push(E) void
pop() E
remove(Object) boolean
contains(Object) boolean
size() int
iterator() Iterator<E>
descendingIterator() Iterator<E>
last E
first E
Q2_Deque
data Q2_LinkedList<E>
removedData Q2_LinkedList<E>
iterator() Iterator<E>
descendingIterator() Iterator<E>
addFirst(E) void
addLast(E) void
offerFirst(E) boolean
offerLast(E) boolean
removeFirst() E
removeLast() E
pollFirst() E
pollLast() E
peekFirst() E
peekLast() E
removeFirstOccurrence(Object) boolean
removeLastOccurrence(Object) boolean
offer(E) boolean
remove() E
poll() E
element() E
peek() E
push(E) void
pop() E
size() int
toString() String
last E
first E

Q3)

2)Problem Solution Approach

Q1) Firstly, first infix expression was converted to prefix and postfix expressions. Then, these expressions was evaluated. Secondly, second infix expression was converted to prefix and postfix expressions, and same way these expressions was evaluated.

Q2) Firstly, I created a Double Linked List class, and created a Node class, a iterator and a descending iterator within it. And then, I created a Deque that consist of double-linked list i created. And I implemented the classic deque methods.

Q3) Firstly, I created a class that name is Question3.

In this class

1) I implemented a method that reverses the words of a sentence

2) I implemented a method that finds if a word is elfish.

3) I implemented the Selection sort algorithm.

4) I implemented a method that evaluates the Prefix expression.

5) I implemented a method that evaluates Postfix expression.

6) I implemented a method that prints the array as specified.

3) Test Cases

Q2)

Test Scenario	**Expected Results**	**Actual Results**	**Pass/Fail**
Adds a value(addFirst, addLast, offerFirst, offerLast)	Adds a value correctly	It has added correctly	**Pass**
Remove a value(removeFirst, removeLast, pollFirst, pollLast)	Removes a value correctly	It has removed correctly	**Pass**
Prints a last or first element(getFirst, getLast, peekFirst, peekLast)	Prints a value correctly	It has printed correctly	**Pass**
Removes a first or last occurence of object(removeFirstOccurence, removeLastOccurence)	Removes a value correctly	It has removed correclty	**Pass**

Q3)

Test Scenario	Expected Results	Actual Results	Pass/Fail
Reverses string of the sentence.	Reverses correctly.	They have reversed correctly	**Pass**
Checks the word is selfish	Checks the word correctly.	It has checked correctly	**Pass**
Selection sort alghorihm.	Sorts the values correctly.	They have sorted correctly	**Pass**
Evaluate Prefix expression	Evaluate the expression correctly	It has evaluated correclty	**Pass**
Evaluate Postfix expression	Evaluate the expression correctly	It has evaluated correclty	**Pass**
Prints the Array.	Prints the Array as specified	Printed approximately as specified.	**%75 Pass %25 Fail**

4) Running And Results

Q2)

```
Harun7 Harun6 Harun1 Harun2 Harun3 Harun4 Harun5
Harun5 Harun4 Harun3 Harun2 Harun1 Harun6 Harun7
Harun6 Harun1 Harun2 Harun3 Harun4
Harun6 Harun2 Harun3 Harun4
Harun2 Harun3
Harun2
Harun3
Harun2
Harun3
```

Q3)

```
reverse in sentence the writes function this
öğrencisidir 2.sınıf Üniversitesi Teknik Gebze Albayrak Harun
Yes it is Elfish word!
No it is not Elfish word.
-4 1 2 2 3 3 9 11
The evaluated Value(Postfix): 4
The evaluated Value(Prefix): 4
1 2 3 4 8 12 16 15 14 13 9 5 1 2 3
```