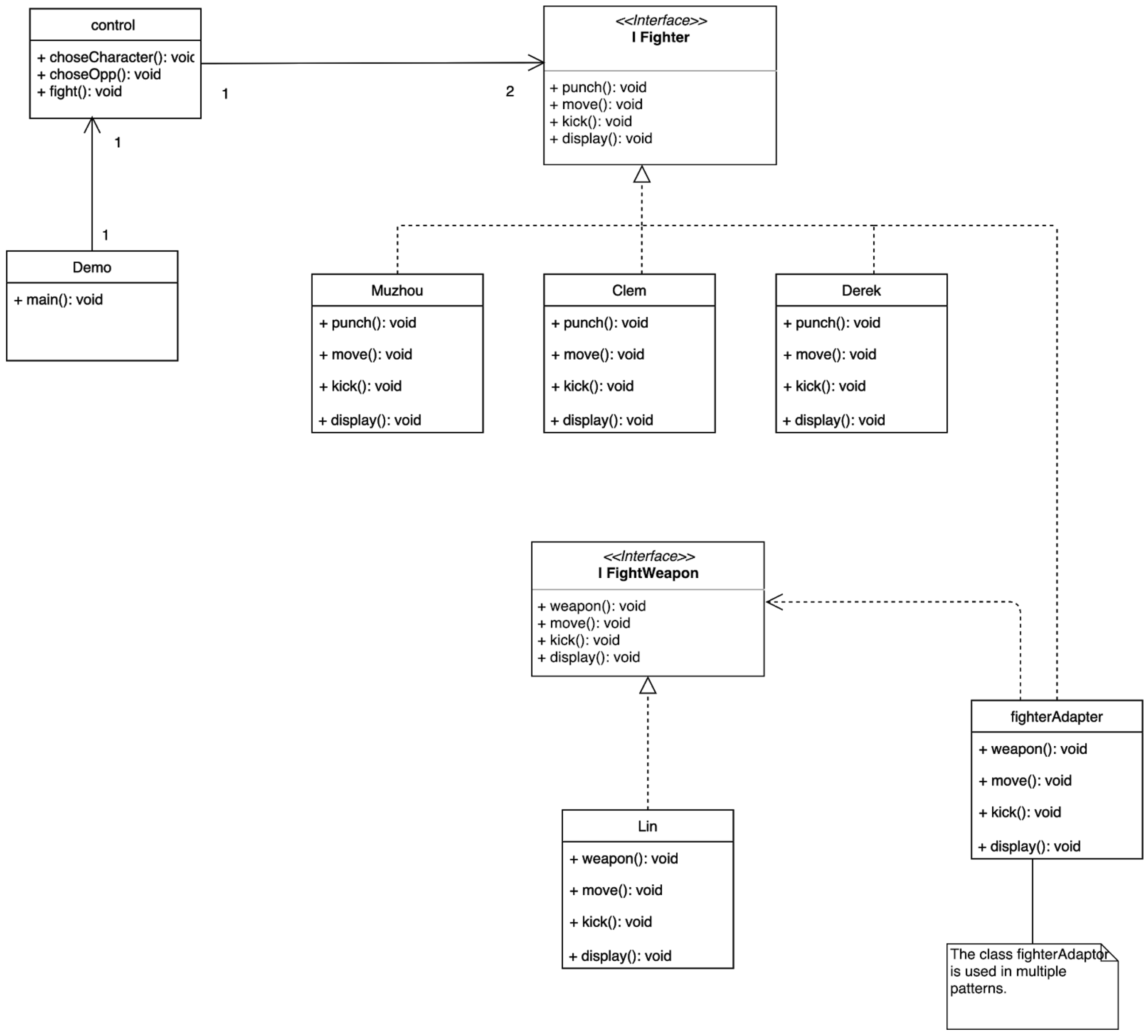
Lin Shi & Muzhou Chen

Homework 3

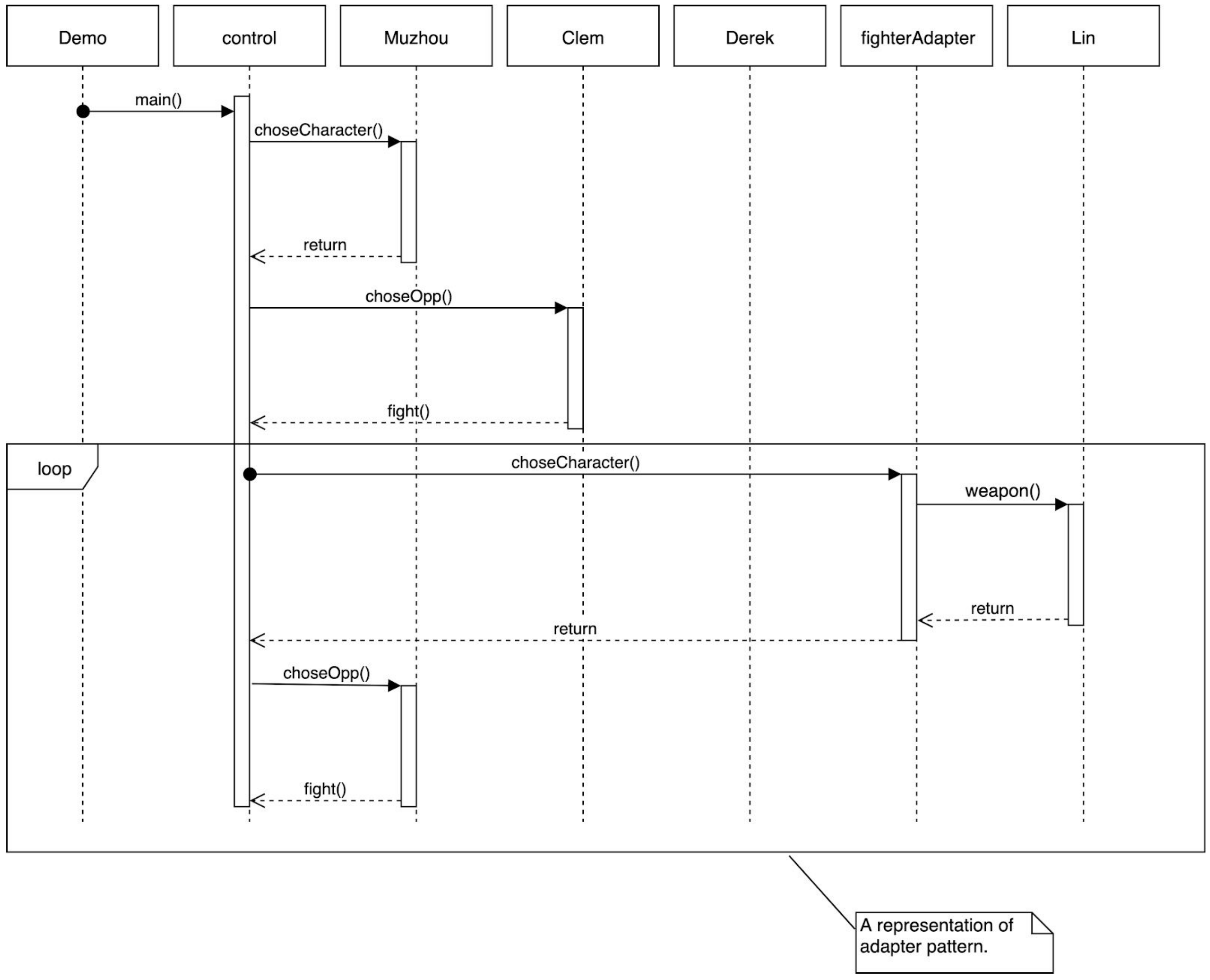
September 26,2019

Exercise 1:

part A:



part B:



Exercise 2:

1. Availability man-days

15

15

15

15

12

Total man-days: 72

Focus factor: 32/45 = 0.71

Estimated velocity = 72 * 0.71 = 51 story points

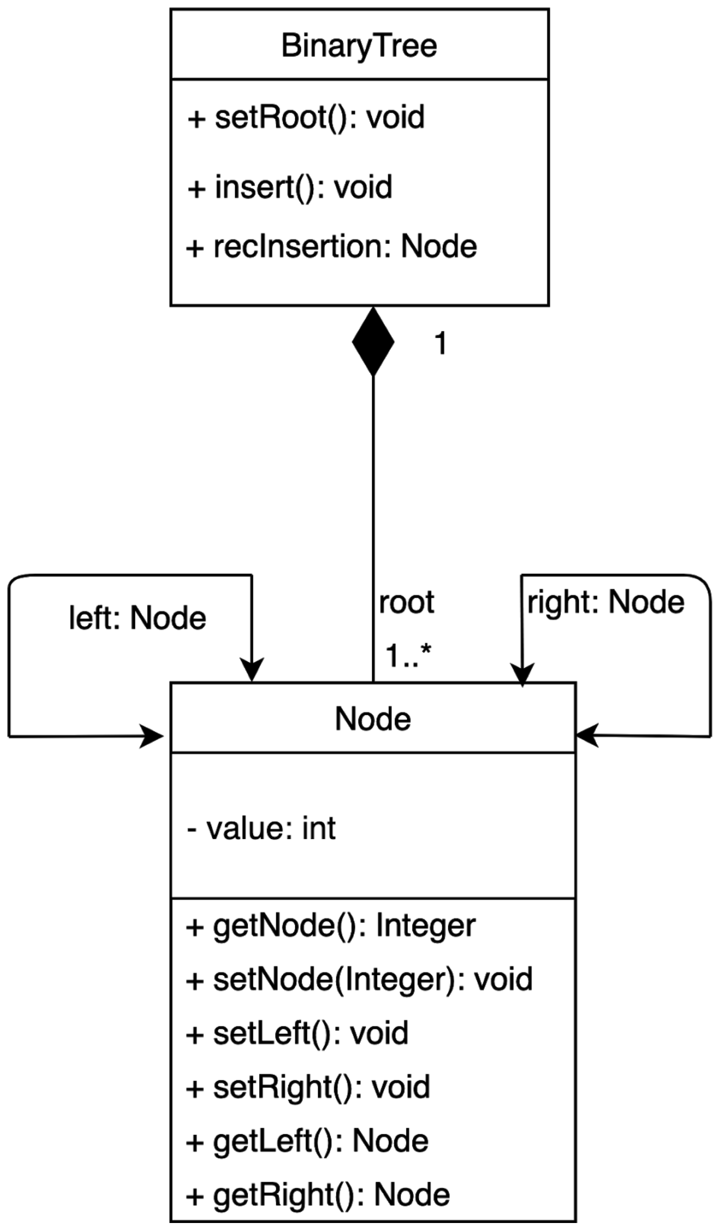
51 story points

2. Focus factor:

   51/72 = 0.71

   71%

3. An alternative to poker using semi-fibonacci sequence is by using t-shirt sizing. By using different sizes such as xs, s, m, l, xl, and etc.. For example, xs could represent ½ to 1 days, s could be 2 days. Without the numbers, it is clear that these are only estimates which could vary. It is better than poker by giving less pressure to meet the deadlines.

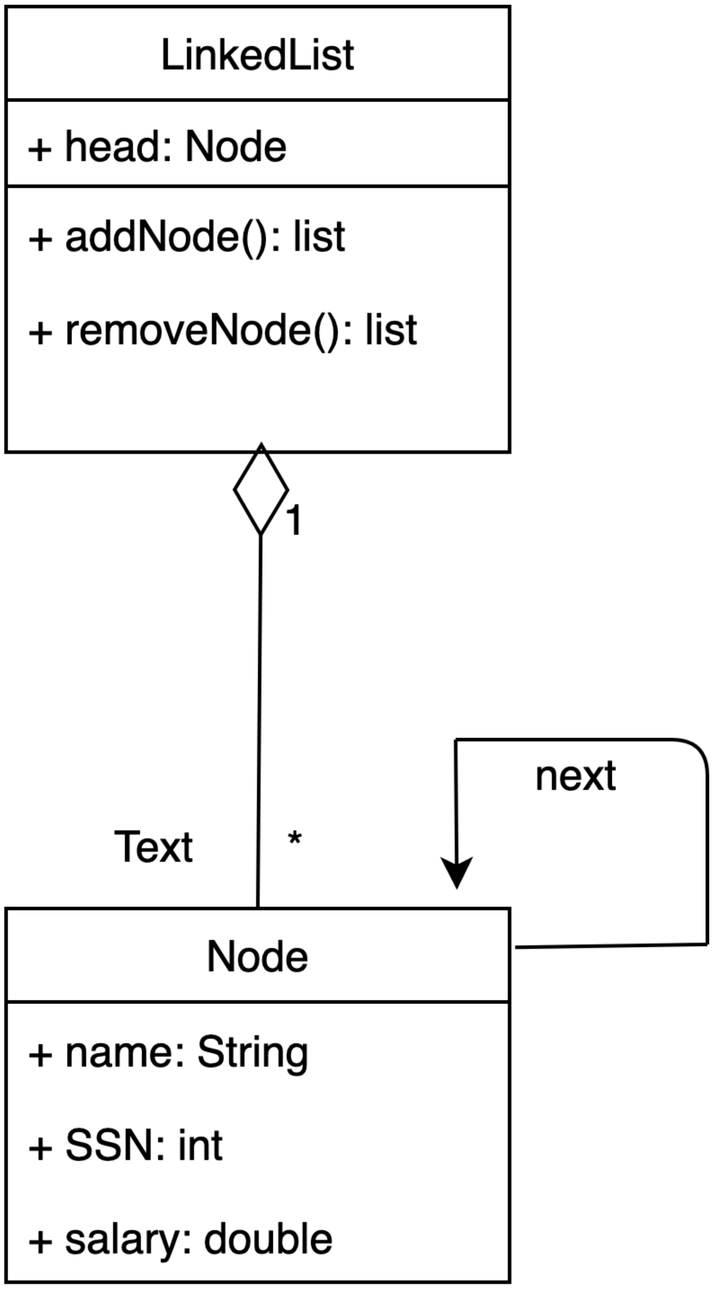


4.

5.

```
public class BinaryTree{

        private class Node{

                private Integer node;

                private Node left;

                private Node right;
                public void setNode(Integer n);

                public void setLeft(Node l);

                public void setRight(Node r);

                public Integer getNode();

                public Node getLeft();

                public Node getRight();

        }

        private Node roots;

        public void setRoot(Node roots);

        public void insert(int value){

                This.roots = this.recInsertion(roots, value);

        }

        public Node recInsertion(Node node, int value){

                if (node == null){

                        return new Node(value);

                }

                if (node.getNode() > value){

                        node.setLeft(this.recInsertion(node.getLeft(), value));

                        return node;

                }

                Else{

                        node.setRight(this.recInsertion(node.getRight(),value));
```

```
                    return node;

                }

            }

        }
```



6.

7.

```
public class LinkedList{
        Node head;
        private class Node{
                int ssn;
                String name;
                double salary;
        }
        public void addNode(){
                if (list.head == null){
                        list.head = newNode;
                }
                else {
                        Node last = list.head;
                        while (last.next != null){
                                last = last.next;
                        }
                        Last.next = newNode;
                }
                return list
        }
        public LinkedList removeNode(int position){
                if (head == null){
```

```
                return;

            }

            Node temp = head;

            if (position == 0){

                head = temp.next;

                return;

            }

            for (int i = 0; temp != null && i < position -1; i++){

                temp = temp.next;

            }

            if (temp == null || temp.next == null){

                return;

            }

            Node next = temp.next.next;

            temp.next = next;

        }


    }
```