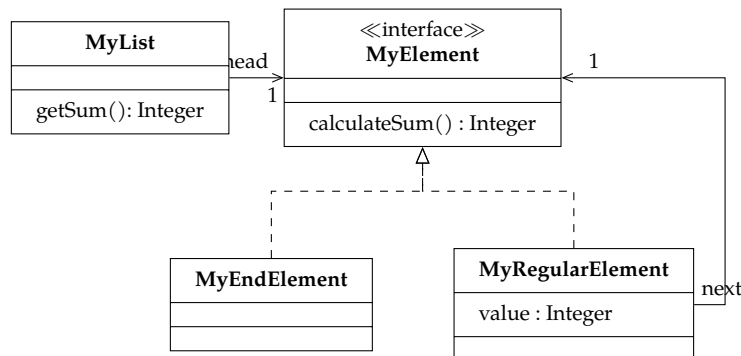


Aufgabe 5

Die folgende Abbildung stellt den Entwurf der Implementierung einer verketteten Liste dar, welche Integer-Werte als Elemente enthalten kann.



```

1 public class MyList {
2     private MyElement head;
3
4     public MyList() {
5         this.head = new MyEndElement();
6     }
7
8     public int getSum() {
9         // ..
10    }
11 }
  
```

Die Klasse `MyList` stellt die Methode `getSum()` zur Verfügung, welche die Summe über alle in einer Liste befindlichen Elemente berechnet. Ein Ausschnitt der Implementierung sieht folgendermaßen aus:

Gehen Sie im Folgenden davon aus, dass bereits Methoden existieren, welche Elemente in die Liste einfügen können.

- (a) Implementieren Sie in einer objektorientierten Programmiersprache Ihrer Wahl, z. B. Java, die Methode `calculateSum()` der Klassen `MyEndElement` und `MyRegularElement`, so dass rekursiv die Summe der Elemente der Liste berechnet wird. Als Abbruchbedingung darf hierbei nicht das Feld `MyRegularElement.next` auf den Wert `null` überprüft werden.

Hinweis: Gehen Sie davon aus, die Implementierung von `MyList` garantiert, dass `MyRegularElement.next` niemals den Wert `null` annimmt, sondern das letzte hinzugefügte `MyRegularElement` auf eine Instanz der Klasse `MyEndElement` verweist. Es gibt immer nur eine Instanz der Klasse `MyEndElement` in einer Liste.

Hinweis: Achten Sie auf die Angabe einer korrekten Methodensignatur.

```

3 public class MyElement {
4
5     MyElement next;
6
7     int value;
8 }
  
```

```

9     public MyElement(int value) {
10         this.value = value;
11     }
12
13     MyElement add(int value) {
14         next = this.next.add(value);
15         return this;
16     }
17
18     int calculateSum() {
19         return value + next.calculateSum();
20     }
21 }

```

github: raw

```

3     public class MyEndElement extends MyElement {
4
5         public MyEndElement () {
6             super(0);
7         }
8
9         MyElement add(int value) {
10             MyElement element = new MyElement(value);
11             element.next = this;
12             return element;
13         }
14
15         int calculateSum() {
16             return 0;
17         }
18
19     }

```

github: raw

```

3     public class MyList {
4         private MyElement head;
5
6         public MyList() {
7             this.head = new MyEndElement();
8         }
9
10        public int getSum() {
11            return this.head.calculateSum();
12        }
13
14        public void add(int value) {
15            head = head.add(value);
16        }
17
18        public static void main(String[] args) {
19            MyList myList = new MyList();
20            myList.add(1);
21            myList.add(2);
22            myList.add(3);
23            System.out.println(myList.getSum());
24        }
25    }

```

github: raw

- (b) Nennen Sie den Namen des Entwurfsmusters, auf welchem das oben gegebene Klassendiagramm basiert, und ordnen Sie dieses in eine der Kategorien von Entwurfsmustern ein.

Hinweis: Es genügt die Angabe eines Musters, falls Sie mehrere Muster identifizieren sollten.

Kompositium (Strukturmuster)