Objektorientierte Programmierung: Aufgabenblatt 8

Florian Ludewig (185722)

7. Juli 2020

Aufgabe 1

Node.java

```java
public class Node {
  public int value;
  public Node next;

  public Node(int val, Node next) {
    value = val;
    this.next = next;
  }

  protected Node clone() {
    if (next == null)
      return new Node(value, null);
    return new Node(value, next.clone());
  }
}
```

SimpleList.java

```java
public class SimpleList {
  private Node head = null;

  SimpleList() {
  }

  SimpleList(Node initialHead) {
    head = initialHead;
  }

  public void add(int i) {
    head = new Node(i, head);
  }

  protected SimpleList clone() {
    return new SimpleList(head.clone());
  }
}
```

Aufgabe 2

1)

ActionObject.java

```java
public interface ActionObject {
  void action(Node n);
}
```

CountStringAction1.java

```
public class CountStringAction1 implements ActionObject {
  private int counter = 0;
  private String search;

  CountStringAction1(String str) {
    search = str;
  }

  public void action(Node n) {
    if (n.data instanceof String && n.data == search)
      counter++;
  }

  public int getCounter() {
    return counter;
  }
}
```

List.java

```
public class List {
  private Node head = null;

  void add(Object obj) {
    this.head = new Node(obj, head);
  }

  public void traverseAndApply(ActionObject p) {
    for (Node cursor = head; cursor != null; cursor = cursor.next) {
      p.action(cursor);
    }
  }

  void print() {
    for (Node cursor = head; cursor != null; cursor = cursor.next) {
      System.out.println(cursor.data.toString());
    }
  }
}
```

Node.java

```
public class Node {
  public Object data;
  public Node next;

  Node(Object obj, Node node) {
    data = obj;
    next = node;
  }
}
```

Main.java

```
public class Main {
  public static void main(String[] args) throws Exception {
    List list = new List();
    list.add("string");
    list.add(42);
    list.add("string");
    list.add("something else");
    list.add("string");
    countOccurences1(list, "string");
    countOccurences2(list, "string");
  }

  public static void countOccurences1(List list, String str) {
    CountStringAction1 cs1 = new CountStringAction1(str);
```

```
15      list.traverseAndApply(cs1);
16      System.out.println(str + " kommt " + cs1.getCounter() + " mal in der
            Liste vor");
17    }
18
19    public static void countOccurences2(List list, String str) {
20      CountStringAction2 cs2 = new CountStringAction2(str);
21      list.traverseAndApply(cs2);
22      System.out.println(str + " kommt " + cs2.getCounter() + " mal in der
            Liste vor");
23    }
24  }
```

2)

CountStringAction2.java

```
1   public class CountStringAction2 implements ActionObject {
2     private int counter = 0;
3     private String search;
4
5     CountStringAction2(String str) {
6       search = str;
7     }
8
9     public void action(Node n) {
10      if (n.data instanceof String && n.data == search) {
11        counter++;
12        if (counter >= 5)
13          n.data = "XXX";
14      }
15    }
16
17    public int getCounter() {
18      return counter;
19    }
20  }
```