Zu Blatt 11, Einführung in die Informatik

17. Januar 2002

Aufgabe 11-1, Klasse `IntListElem`

```java
class IntListElem {
  private int data;
  private IntListElem next;

  IntListElem (int i) {
    data = i;
    next = null;
  }

  int getData() {
    return data;
  }

  void setData(int newData) {
    data = newData;
  }
```

```java
  IntListElem getNext() {
    return next;
  }

  void setNext(IntListElem elem) {
    next = elem;
  }

  IntListElem copy() {
    IntListElem result = new IntListElem(data);
    if (next != null)
      result.next = next.copy();
    return result;
  }
}
```

Aufgabe 11-1, Klasse `IntList`

```java
class IntList {
  private IntListElem head;

  public IntList() {
    head = null;
  }

  public IntList (IntListElem _head) {
    head = _head;
  }

  public IntList(int elem) {
    head = new IntListElem(elem);
  }

  public void clear() {
    head = null;
  }
```

```java
  public void addFirst(int value) {
    IntListElem elem = new IntListElem(value);
    elem.setNext(head);
    head = elem;
  }

  public int first() {
    return head.getData();
  }

  public void setFirst(int data) {
    head.setData(data);
  }

  public void rest() {
    head = head.getNext();
  }

  public boolean isEmpty() {
    return head == null;
  }
```

```java
  public int size() {
    IntListElem elem = head;  int i = 0;
    while (elem != null) {
      elem = elem.getNext();
      i = i + 1;
    }
    return i;
  }

  public IntList shallowCopy() {
    return new IntList(head);
  }

  public IntList copy() {
    IntList result = new IntList();
    if (head != null)
      result.head = head.copy();
    return result;
  }
```

```java
  public void append(IntList list) {
    if (head == null)
      head = list.head.copy();
    else {
      IntListElem last = head;
      while (last.getNext() != null)
      last = last.getNext();
      last.setNext(list.head.copy());
    }
  }

  public void reverse() {
    if (head != null) {
      IntList newList = new IntList();
      while (head != null) {
        newList.addFirst(this.first());
        this.rest();
      }
      head = newList.head;
    }
  }
```

```java
  public void print() {
    System.out.print("(");
    for (IntList list = this.shallowCopy(); !list.isEmpty(); list.rest()) {
      System.out.print(list.first());
      if (list.head.getNext() != null)
        System.out.print(", ");
    }
    System.out.print(")");
  }

  public boolean contains(int v) {
    IntListElem elem = head;
    while (elem != null && elem.getData() != v)
      elem = elem.getNext();
    return (elem != null);
  }

  public int min() {
    int min = first();
    for (IntList list = this.shallowCopy(); !list.isEmpty(); list.rest()) {
      if (list.first() < min)
        min = list.first();
    }
  return min;
  }
}
```

Aufgabe 11-2, Klasse `Call`

```
public class Call {
  private double cost;

  public Call() {
    cost = Math.random() * 1.9 + 0.1;
  }

  public double getCost() {
    return cost;
  }
}
```

Aufgabe 11-2, Klasse `CallListElem`

```
public class CallListElem {
  private CallListElem next;
  private Call data;

  public CallListElem (Call data) {
    this.data = data;
    this.next = null;
  }

  public void setNext (CallListElem next) {
    this.next = next;
  }

  public Call getData() {
    return data;
  }

  public CallListElem getNext() {
    return next;
  }
}
```

Aufgabe 11-2, Klasse `CallList`

```
public class CallList {
  private CallListElem head;

  public CallList() {
    head = null;
  }

  public boolean isEmpty() {
    return (head == null);
  }

  public Call getFirst() {
    return head.getData();
  }

  public void rest() {
    head = head.getNext();
  }

  public void addFirst (Call c) {
    CallListElem newHead = new CallListElem(c);
    newHead.setNext(head);
    head = newHead;
  }

  public int size() {
    int l = 0;
    CallListElem e = head;
    while (e != null) {
        l++; e = e.getNext();
    }
    return l;
  }

  public CallList shallowCopy() {
    CallList newList = new CallList();
    newList.head = head;
    return newList;
  }
}
```

Aufgabe 11-2, Klasse `CellPhone`

```
public class CellPhone {
  private CallList calls;

  public CellPhone() {
    calls = new CallList();
  }

  public void simulateCalls() {
    double total = 0.0;
    while (total <= 20.0) {
      Call c = new Call();
      calls.addFirst(c);
      total += c.getCost();
    }
  }

  public void printBill() {
    for (CallList l = calls.shallowCopy(); ! l.isEmpty(); l.rest()) {
      System.out.println("Anruf: " + l.getFirst().getCost() + " Euro.");
    }
    System.out.println(calls.size() + " Gespraeche.");
  }

  public static void main(String[] args) {
    CellPhone c = new CellPhone();
    c.simulateCalls();
    c.printBill();
  }
}
```