Hausaufgabe 2.1: a) Java-Class DoppeltVerketteteListe:

```
import java.lang.*;
import java.io.*;

 class ListElement{
    protected ListElement Nachfolger;
    protected ListElement Vorgaenger;
    protected int value;
    public ListElement( int v){ value = v;}
    int getValue(){return value;}
    void setValue(int v){value = v;}
    void setNachfolger( ListElement le){    Nachfolger = le;  }
    void setVorgaenger( ListElement le){    Vorgaenger = le; }
    ListElement getNachfolger(){    return Nachfolger;  }
    ListElement getVorgaenger( ){     return   Vorgaenger;  }
}

public class DoppeltVerketteteListe {
  protected ListElement Start;
  protected ListElement Ende;

 public DoppeltVerketteteListe()
 {
    Start = null;
    Ende = null;
  }

 void insertLast( int value)
{
   ListElement le = new ListElement(value);
  //Liste Leer
    if(Ende == null)
    {
      Start = le;
      Ende = le;
    }
    else
    {
      Ende.setNachfolger(le);
      le.setVorgaenger(Ende);
      Ende = le;
    }
  }
void insertFirst( int value)
{
    ListElement le = new ListElement(value);
    //Liste Leer
    if(Start ==null)
    {
      Start = le;
      Ende = le;
 }
else
{
   Start.setVorgaenger(le);
   le.setNachfolger(Start);
  Start = le;
 }
}

void deleteLast()
{
```

```
  // Liste nicht schon leer
  if(Ende != null)
  {
    Ende =Ende.getVorgaenger();
    //Falls Liste immer noch nicht leer
    if(Ende != null)
    {
      Ende.setNachfolger( null);
    }
    // Falls Liste leer rücksetzten des Starts
   else
    {
      Start = null;
    }
  }
 }

void deleteFirst()
{
  // Liste nicht schon leer
  if(Start != null)
  {
    Start = Start.getNachfolger();
    //Falls Liste immer noch nicht leer
    if(Start != null)
    {
      Start.setVorgaenger( null);
    }
    // Falls Liste leer rücksetzten des Starts
   else
    {
     Ende = null;
    }
  }
}

int getFirst()
{
  if(Start != null)
     return Start.getValue();
  else
     return -1;
 }

int getLast()
{
   if(Ende != null)
     return Ende.getValue();
   else
     return -1;
 }

//Main-Routine
public static void main(String args[])
  {
  DoppelVerketteteListe dvk = new DoppelVerketteteListe();
  dvk.insertFirst(1);
  dvk.insertLast(2);
  dvk.insertFirst(3);
  dvk.deleteLast();
  System.out.println( dvk.getFirst() + " | " + dvk.getLast());
  }
}
```