Lösung zur Aufgabe 7

```
package diffari;
public class Diff {
  private double f;  // Funktionswert
  private double df; // Ableitungswert

  private Diff (double f, double df) { // Konstruktor
    this.f = f;
    this.df = df;
  }
  public static Diff diffVar (double x) { // Erzeuge unabhängige Variable x
    return new Diff (x,1);
  }
  public static Diff diffConst (double c) { // Erzeuge Konstante
    return new Diff (c,0);
  }
  public double getF () {
    return f;
  }
  public double getDf () {
    return df;
  }

  public Diff plus (Diff v) {
    double  hf = f + v.f;
    double hdf = df + v.df;
    return new Diff(hf,hdf);
  }
  public Diff minus (Diff v) {
    double  hf = f - v.f;
    double hdf = df - v.df;
    return new Diff(hf,hdf);
  }
  public Diff times (Diff v) {
    double  hf = f * v.f;
    double hdf = df * v.f + f * v.df;
    return new Diff(hf,hdf);
  }
  public Diff divby (Diff v) {
    double  hf = f / v.f;
    double hdf = (df - f * v.df / v.f) / v.f;
    return new Diff(hf,hdf);
  }

  public static Diff sin (Diff u) {
    double  hf = Math.sin(u.f);
    double hdf = u.df * Math.cos(u.f);
    return new Diff(hf,hdf);
  }
}
```

```
import Prog1Tools.IOTools;
import static Prog1Tools.IOTools.*;
import diffari.Diff;
import static diffari.Diff.*;
public class DiffTest {
  public final static Diff d4 = diffConst(4);
  public final static Diff d3 = diffConst(3);

  public static Diff f (Diff x) {
    return sin(x.times((d4.plus(x)).divby(d3.minus(x))));
  }

  public static void main (String[] args) {
    double x = readDouble("x = ");
    Diff dx = diffVar(x);
    Diff fx = f(dx);
    System.out.println("f(x)  = " + fx.getF());
    System.out.println("f’(x)  = " + fx.getDf());
  }
}
```