

Name:

Matrikelnummer:

Bearbeitungszeit: 90 min.

Aufgabe 1. Gegeben ist das Interface

```
public interface RealFunction
{
    public double f(double x);
}
```

und die Klasse `Minimizer` mit einem parameterlosen Konstruktor und der Methode

```
public double findMin(RealFunction fun, double a, double b,
                    double error)
```

die das Minimum der übergebenen Funktion im Intervall $[a,b]$ mit maximalem Fehler `error` berechnet.

Schreiben Sie die Klasse `RealFunctionCount`, die das Interface `RealFunction` implementiert und die Anzahl der Funktionsaufrufe mitzählt. Dazu soll es den Konstruktor und die Methoden

```
public RealFunctionCount(RealFunction fun),
public double f(double x),
public int getFunCalls()
```

geben. Ein Aufruf von `f(x)` soll das Ergebnis für die Funktion `fun` zurückgeben, `getFunCalls()` soll die Anzahl der bisherigen Aufrufe von `f(x)` zurückgeben.

Implementieren Sie weiters die Unterklasse `MinimizerPlus` von `Minimizer`, die über die zusätzliche Methode

```
public double findMinCount(RealFunction fun, double a, double b,
                          int funCalls)
```

verfügt. Die Methode `findMinCount` soll `findMin` mit immer kleineren Werten für `error` aufrufen, bis die Gesamtanzahl an Funktionsaufrufen von `fun.f` den Wert von `funCalls` überschreitet. Konkret soll der erste Aufruf von `findMin` mit dem Wert `error=1` erfolgen, und in jedem weiteren Aufruf soll der Wert von `error` halbiert werden. Das Ergebnis des letzten Aufrufs von `findMin` soll zurückgegeben werden.

Aufgabe 2. Schreiben Sie die Klasse `DenseIntervals` mit einem parameterlosen Konstruktor und der Methode

```
public double findDense(double[] points, double width).
```

Der Aufruf `findDense(points,width)` soll einen Wert `x` zurückgeben, sodass die Anzahl der Werte in `points`, die im Intervall $[x,x+width]$ liegen, maximal ist.