BERGISCHE UNIVERSITÄT
GESAMTHOCHSCHULE WUPPERTAL
GAUSS-STRASSE 20
42097 WUPPERTAL
(Korrespondenzanschrift)
42119 WUPPERTAL
(Lieferanschrift)
TELEX 8 592 262 bughw
TELEFAX (0202) 439-2901
TELEFON (0202) 439-1

Fachbereich 7

MATHEMATIK

Prof. Dr. Hans-Jürgen Buhl
Praktische Informatik / Numerik

e-mail: Juergen.Buhl@math.uni-wuppertal.de

Einführung in die Informatik und Programmierung (Informatik I)

WS2000/2001 – Übungsblatt 13

31. Januar 2001
Bearbeitungstermin: 6. KW

Aufgabe 1. *Primzahlsuche: noch weniger Teilerkandidaten, 5 Punkte*
Anstatt die Teilerkandidaten aus der Menge

$$\{2,3,5\} \cup \{30 \cdot k + a \mid k \in \mathbb{N}_0,\ a \in \{1,7,11,13,17,19,23,29\}\} \setminus \{1\}$$

zu wählen, kann man mit Hilfe der Speicherung der im Algorithmenfortgang schon errechneten Primzahlen in einem statischen Feld die zu überprüfenden Teilerkandidaten noch weiter vermindern.
Realisieren Sie das in einem Programm. Welchen Zeitgewinn bringt diese Modifikation (Zeittests)?

Aufgabe 2. *Redundante Attribute, 4 Punkte*
Zur effektiven Implementierung benutzt man häufig redundante Attribute: Z.B. ist der Operator + mit Hilfe von Realteil und Imaginärteil einfach zu berechnen, während der Operator * mit Hilfe von Winkel und Länge einfacher berechnet werden kann. Ergänzen Sie das folgende Beispiel um die beiden friend-Operatoren + und * und testen Sie:

```
////////////////////////////////
// Datei:   comp4.cc
// Version: 1.0
// Zweck:   redundante Attribute
// Autor:   Hans-Juergen Buhl
// Datum:   17.09.1998
////////////////////////////////

#include <iostream>
#include <cmath>

using namespace std;
```

```
class comp {

  double re;
  double im;
                       // re == cos(Winkel) * Laenge
                       // im == sin(Winkel) * Laenge
  double Winkel;       // -Pi <= Winkel < +Pi,   {read only}
  double Laenge;       // >= 0.0                {read only}

public:

  comp() : re(0.0), im(0.0), Winkel(0.0), Laenge(0.0) {};

  comp( const double r, const double i ) :  re(r), im(i),
                                            Winkel( atan2(im, re) ),
                                            Laenge( hypot(re, im) ) {};

  // void print() const { ... };

  void printPolar() const { cout << "Winkel= " << Winkel
                            << " Laenge= " << Laenge; };

};

int main()
{
  comp x;
  comp x2(-2.0, -0.000000001);

  double d(3.5);
  comp x3(d, 4);

  x.printPolar();
  cout << endl;
  x2.printPolar();
  cout << endl;
  x3.printPolar();
  cout << endl;

  return 0;
}
```

Aufgabe 3. *Bisektionsmethode, 6 Punkte*
Ergänzen Sie die Klasse `Function` um eine Methode `getZero`, die eine Nullstellennäherung mit der folgenden Methode berechnet:

Zunächst wird aus den Teilintervallen des betrachteten Intervalls eines herausgesucht, in dem ein Vorzeichenwechsel vorliegt.

Dann wird in diesem Teilintervall $[x_l, x_r]$ eine Näherungsnullstelle mit Hilfe der Bisektionsmethode bestimmt:

Finde ein neues Teilintervall von $[x_l, x_r]$,

$$
\begin{aligned}
&x_m = \text{midpointOf}(x_l, x_r);\\
&\text{if}(\text{fkt}(x_l) * \text{fkt}(x_m)) < 0.0)\\
&\quad x_r = x_m;\\
&\text{else}\\
&\quad x_l = x_m;
\end{aligned}
$$

das nur halb so breit wie das ursprüngliche $[x_l, x_r]$ ist, und in dem ebenfalls ein Vorzeichenwechsel von `fkt` vorliegt. Wiederhole diesen Schritt, bis eine vorgegebene Genauigkeit erreicht ist.

Realisieren Sie die Methode `getZero` mit Hilfe der „lazy evaluation“. Treffen sie Vorkehrungen dafür, dass der Algorithmus keine Nullstelle finden kann (was soll die Methode dann tun?).
Programmieren Sie `midpointOf`(x_l, x_r) so, dass weder $+\infty$ noch $-\infty$ als Ergebnis vorkommen können.

Aufgabe 4. *Quadratische Gleichungen, 5 Punkte*
Modifizieren Sie Ihre Klasse `quadGleichung` (Aufgabe 3 / Übungsblatt 7) so, dass die beiden Lösungen im Falle $b \neq 0$ statt mittels

$$ax^2 + bx + c = 0 \quad \Leftrightarrow \quad x_{1/2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

durch

$$x_1 = \frac{-b + \text{sign}(-b)\sqrt{b^2 - 4ac}}{2a}$$

und

$$x_2 = \frac{c}{a \cdot x_1}$$

(wegen $x_1 x_2 = \frac{c}{a}$ und $x_1 + x_2 = -\frac{b}{a}$) berechnet werden. Was ist der Vorteil dieser Modifikation?