



## Übung 4

### Aufgabe 1: Quadrat

Schreiben Sie ein Programm, das die Quadrate der Zahlen 1 bis 20 ausgibt. Benützen Sie dazu ihre eigene Funktion `square` um das Quadrat einer Zahl zu berechnen.

### Aufgabe 2: Grösster gemeinsamer Teiler

In dieser Aufgabe geht es um Rekursion: Eine Möglichkeit, den grössten gemeinsamen Teiler (`ggT`) zweier ganzer Zahlen  $a \geq 0$  und  $b \geq 0$  zu berechnen, ist die folgende:

- Falls  $a > b$ , subtrahiere  $b$  von  $a$  und berechne `ggT(a - b, b)` anstelle von `ggT(a, b)`.
- Falls  $b > a$ , subtrahiere  $a$  von  $b$  und berechne `ggT(a, b - a)` anstelle von `ggT(a, b)`.
- Falls  $a = 0$  oder  $b = 0$ , gilt `ggT(a, 0) = a` respektive `ggT(0, b) = b` und man ist fertig.

#### Aufgabe:

- Überlegen Sie sich, warum dieser Algorithmus korrekt ist.
- Sie haben sicher bemerkt, dass dieser Algorithmus rekursiv ist. Wo ist in diesem Algorithmus die rekursive Vorschrift, wo die Abbruchbedingung?
- Schreiben Sie ein C++-Programm, das die beiden Zahlen  $a$  und  $b$  von der Tastatur einliest, im Anschluss die von Ihnen zu implementierende rekursive Funktion  

```
int ggT(int a, int b)
```

mit diesen Zahlen als Parametern aufruft und das Ergebnis ausgibt. Testen Sie Ihr Programm an einigen (auch speziellen) Beispielen (z.B.  $a = 0$  und  $b > 0$ ,  $a = b$ ,  $a < b$ , ... ).
- Ersetzen Sie die rekursive Funktion im Programm aus Aufgabe c) durch eine iterative Funktion (ohne Rekursion, mit Schleifen). Testen Sie Ihr Programm wiederum an einigen (auch speziellen) Beispielen (z.B.  $a = 0$  und  $b > 0$ ,  $a = b$ ,  $a < b$ , ... ).
- Der oben beschriebene rekursive Algorithmus zur Berechnung des grössten gemeinsamen Teilers (`ggT`) zweier Zahlen kann noch optimiert werden. Finden Sie einen ähnlichen Algorithmus, der jedoch mit viel weniger Schritten ans Ziel führt.

### Aufgabe 3: Polynom auswerten

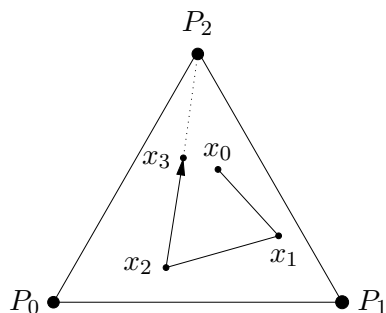
In dieser Aufgabe sollen Sie die Syntax von Feldern richtig anwenden lernen. Schreiben Sie ein Programm, das den Wert eines Polynoms  $p(x) = \sum_{i=0}^g a_i \cdot x^i$  berechnet. Es soll dabei folgende Schritte in der angegebenen Reihenfolge ausgeführt werden:

1. Es liest den ganzzahligen Grad  $g$  des Polynoms von der Tastatur ein.
2. Es liest die Koeffizienten  $a_i$ ,  $i = 0 \dots g$ , des Polynoms von der Tastatur und speichert sie in einem Feld.
3. Es liest den Wert der Variable  $x$  von der Tastatur und setzt ihn in das Polynom ein.

### Aufgabe 4: Chaos-Spiel

Wir haben uns folgendes *Chaos-Spiel* überlegt:

- Man markiert drei beliebige Bezugspunkte  $P_0$ ,  $P_1$  und  $P_2$  in der Ebene so, dass sie ein Dreieck bilden, und wählt weiters einen beliebigen Startpunkt.
- Pro Spielschritt wählt man zufällig einen der Bezugspunkte  $P_i$  aus und bewegt sich um die halbe Distanz auf ihn zu.



Beispiel:

1. zufällige Wahl des Startpunktes  $x_0$ .
2. zufällige Wahl von  $P_1$ ,  $x_1 = (x_0 + P_1)/2$ .
3. zufällige Wahl von  $P_0$ ,  $x_2 = (x_1 + P_0)/2$ .
4. zufällige Wahl von  $P_2$ ,  $x_3 = (x_2 + P_2)/2$ .
5. ...

#### Aufgabe:

Programmieren Sie dieses Chaos-Spiel: Führen Sie etwa 50000 Schritte durch und markieren Sie dabei in jedem Schritt den jeweiligen Aufenthaltsort  $x_i$  durch einen Punkt im Graphikfenster.