TU Berlin, Institut für Mathematik MA 4-5
Thomas Slawig, Uwe Prüfert

11. Übungsblatt

www.math.tu-berlin.de/Vorlesungen/WS06/ProgMa

Theoretische Aufgaben: Abgabe in der Vorlesung am 23.1.2007

Bei späterer Abgabe werden die erreichten Punkte nur zu 50 % angerechnet.

Hinweis: In objektorientierter Notation schreibt man bei Anwendung

- einer Methode auf das Objekt z.B. `a.display()` (und nicht `display(a)` !)
- und in der Signatur eines überladenen Operators z.B. `bruch::operator+(bruch b)`,

d.h das Objekt selbst tritt nicht als Parameter auf. In der **standardisierten UML-Notation** wird das Objekt selbst ebenfalls nicht als Parameter aufgeführt.

1. Aufgabe (5 Punkte)

Zeichnen Sie ein UML-Diagramm für eine Klasse `Complex` inklusive Attribute und Methoden mit Parametern, Rückgabewerten und deren Datentypen. Es sollen für $w, z \in \mathbb{C}, x \in \mathbb{R}$ die vier Grundrechenarten und die folgenden Operationen möglich sein:

$$\begin{aligned} w &= \bar{z}, \\ x &= \mathrm{Re}(z), \\ x &= \mathrm{Im}(z), \\ r &= \|z\|, \\ x &= \phi \quad \text{für } z = re^{i\phi}. \end{aligned}$$

Das UML-Diagramm soll der o.g. standardisierten UML-Notation entsprechen.

2. Aufgabe (5 Punkte)

Schreiben Sie ein UML-Diagramm (in der o.g. standardisierten UML-Notation) für eine Klasse für Polynome beliebigen Grades. Es sollen folegnde Operationen möglich sein:

- Ausgabe eines Polynoms auf den Bildschirm.
- Eingabe eines Polynoms beliebigen Grades über die Tastatur.
- Bestimmung des Grades eines Polynoms.
- Addition, Subtraktion und Multiplikation von zwei Polynomen.
- Ein Vergleichsoperator $<$, den wir für zwei Polynome p, q mit

$$p(x) = \sum_{i=0}^{m} a_i x^i, a_m \neq 0 \qquad q(x) = \sum_{i=0}^{n} b_i x^i, b_n \neq 0$$

  wie folgt definieren:

$$p < q \quad :\Longleftrightarrow \quad \Big(m < n \ \text{ oder } \ (m = n \text{ und } a_m < b_m)\Big). \tag{1}$$

3. Aufgabe (3+1+1+1 Punkte)

Wir betrachten die Gleitpunktzahlen im Binärsystem mit vierstelliger Mantisse und dem Exponentenbereich $-1 = e_{min} \leq e \leq e_{max} = 1$.

(a) Geben Sie alle darstellbaren normalisierten Gleitpunktzahlen und alle subnormalen Zahlen an.

(b) Vergleichen Sie jeweils

- die größte positive normalisierte Gleitpunktzahl,
- die kleinste positive normalisierte Gleitpunktzahl
- und die kleinste positive subnormale Zahl

mit derjenigen, die man bei nur dreistelliger Mantisse erhält.

4. Aufgabe (2+1+1 Punkte)

Wir betrachten die normalisierten Gleitpunktzahlen im Binärsystem mit dreistelliger Mantisse und dem Exponentenbereich $-2 = e_{min} \leq e \leq e_{max} = 2$.

(a) Geben Sie alle darstellbaren normalisierten Gleitpunktzahlen an. Vergleichen Sie jeweils

- die größte positive normalisierte Gleitpunktzahl
- und die kleinste positive normalisierte Gleitpunktzahl

mit derjenigen, die man beim Exponentenbereich $-1 = e_{min} \leq e \leq e_{max} = 1$ erhält.

Programmieraufgabe: Vorführen bis zum 31.1.2007

(a) Implementieren Sie in C++ eine Klasse `polynom` für reelle Polynome beliebigen Grades. Es sollen vorhanden sein:

- ein Konstruktor

  ```
  polynom(double *x, int n)
  ```

  der dynamisch Speicherplatz für die Koeffizienten bereitstellt. Dabei ist `x` ein Zeiger auf einen Vektor mit Koeffizienten und `n` der Grad des Polynoms.
- den sog. *Copy*-Konstruktor mit der Signatur

  ```
  polynom(const polynom& p)
  ```

  der eine Kopie des Objektes `p` erzeugt. Achten Sie darauf, dass Sie darin wirklich die Koeffizienten und nicht zur den Zeiger darauf kopieren.
- ein *Destruktor*

  ```
  ~polynom()
  ```

  in dem der im Konstruktor bereitgestellte Speicherplatz mit dem `delete`-Befehl wieder freigegeben wird,
- der überladene Zuweisungsoperator. Achten Sie auch hier darauf, dass Sie darin wirklich die Koeffizienten kopieren.
- der überladene Additionsoperator,
- der überladene Vergleichsoperator < wie in (1) definiert,
- eine Methode `display`, die die Koeffizienten des Polynoms in der Form

  ```
  a_{0} = ...
  a_{1} = ...
  ...
  a_{n} = ...
  ```

  untereinander auf dem Bildschirm ausgibt.

(b) Testen Sie Ihre Klasse mit dem Hauptprogramm, das in der Datei `prog11.cc` auf der Homepage gegeben ist. Dieses Hauptprogramm soll unverändert bleiben.