Dr. Matthias Grabert
Johannes Mayer
Abteilung Angewandte Informationsverarbeitung
13. November 2002
Blatt 4

Objektorientierte Softwareentwicklung mit C++ (WS 2002/2003)

Abgabetermin: 21. November 2002

1 Vektoren ... die letzte! ;-) (3 Punkte)

Erzeugen Sie eine Klasse `vector` basierend auf der `vector`-Struktur aus den Übungen. Diese Klasse soll – analog zur Aufgabe 2 des vorigen Blattes – eine Template-Klasse sein. Achten Sie beim Design der Klasse besonders auf „Information Hiding“. Achten Sie bei der Implementierung außerdem darauf, dass das Kopieren korrekt funktioniert (Kopierkonstruktor und Operator =). Aber das war's dann jetzt wirklich mit den selbst gebastelten Vektoren.

2 Primzahlen (7 Punkte)

The Prime Pages sind eine wahre Fundgrube für alles mögliche rund um Primzahlen. Da bietet es sich nun geradezu an, eine eigene Klasse `prime` für Primzahlen zu schreiben. Diese soll einen Konstruktor besitzen, dem eine Integer übergeben wird. Ist die Integer nicht prim, so kommt es zu einer Exception. Außerdem soll es folgende Operationen auf dieser Klasse geben:

- `value()`: liefert die Primzahl als Integer

- `operator++(int)`: springt zur nächsten Primzahl (die nun vom Objekt repräsentiert wird) und gibt eine Referenz auf das Objekt zurück;
  Bsp.: Wenn p die Primzahl 3 repräsentiert, dann repräsentiert p nach p++ die Primzahl 5.

- `operator--(int)`: springt zur vorigen Primzahl (die nun vom Objekt repräsentiert wird) und gibt eine Referenz auf das Objekt zurück
  Bsp.: Wenn p die Primzahl 3 repräsentiert, dann repräsentiert p nach p-- die Primzahl 2.

- `operator+(int i)`: gibt die „i-nächste“ Primzahl zurück; das Objekt ändert sich dabei nicht!
  Bsp.: Wenn p die Primzahl 3 repräsentiert, dann repräsentiert p+3 die Primzahl 11.

- `operator-(int i)`: gibt die „i-vorige“ Primzahl zurück; das Objekt ändert sich dabei nicht!
  Bsp.: Wenn p die Primzahl 7 repräsentiert, dann repräsentiert p-3 die Primzahl 2.

Damit Primzahltests für eine Zahl wiederholt werden müssen, soll es für alle Instanzen der Klasse `prime` einen (!) vector aller bisher bestimmten Primzahlen geben. Dieser enthält lückenlos von 2 bis zur zuletzt bestimmten Primzahl alle Primzahlen. Damit ist es nun auch leicht möglich eine Methode `by_index(int i)` der Klasse (!) `prime` zu implementieren, die die i-te Primzahl liefert. (Die Zählung soll bei 0 beginnen!)

Bestimmen Sie nun mit Ihrer Primzahl-Klasse die 10000-ste Primzahl und jeweils die ersten 100 Primzahlzwillinge, -drillige und -vierlinge. Nähere Infos dazu und Kontrollergebnisse siehe *Primzahlzwillinge* und *Primzahldrillinge und -vierlinge*.

Viel Spass!