Dr. Andreas F. Borchert   Institut für Angewandte Informationsverarbeitung   16. Dezember 2008
Wolfgang Kaifler   Blatt 8

Objektorientierte Programmierung mit C++ (WS 2008)

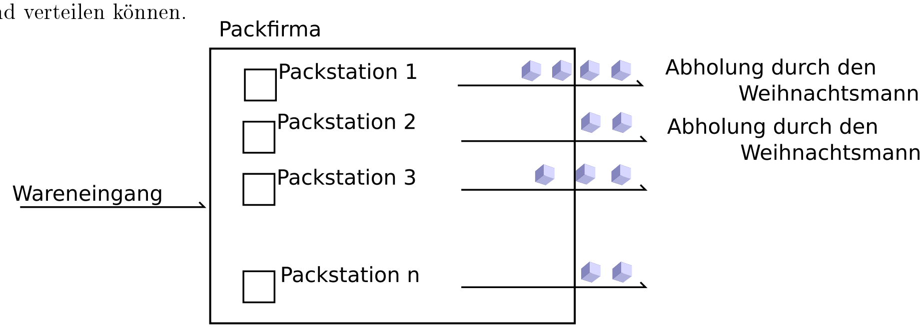
Abgabetermin: 13. Januar 2009

14 Weihnachtsfirma

Wie jedes Jahr um Weihnachten gilt es jede Menge Geschenke zu verteilen. Da es nur noch wenige Tage bis zum Feste sind, wurden die Wunsch-Zettel geschrieben und mittlerweile die Geschenke besorgt.

Natürlich will jedes Geschenk-Paket individuell zusammengestellt, verpackt und anschließend verteilt werden. Dazu hat das Unternehmen „Weihnachten“ eine Packfirma beauftragt. Diese besteht wie ein Speditions-Unternehmen aus mehreren Packstationen. (Sie können diese durch einen Vektor bestehend aus lauter Queues modellieren).

Die Geschenke werden also an die Packfirma geliefert, und auf die Packstationen verteilt. Natürlich bekommt die Packstation den nächsten Auftrag, die am wenigsten Pakete in der Queue hat. (Füllen sie die Queues mit beliebigen Werten). Sind die Geschenke verpackt, landen sie auf einer Lagerstätte hinter jeder Packstation, von wo aus sie die Weihnachtsmänner abholen und verteilen können.



Jeder Weihnachtsmann benötigt eine gewisse Zeit, um das Paket zuzustellen. Das kann einmal schneller gehen (der Weg ist kurz) - ein andermal dauert das natürlich länger. (Sie können das simulieren, indem sie per Zufall eine Packstation auswählen und das Paket entfernen).

Steht zufällig an einer Packstation gerade kein Päckchen zum verteilen mehr an, geht der Weihnachtsmann zur nächsten Station.

Nachdem 50 Pakete verteilt sind, stoppen Sie die Auslieferung, und lassen sich die restlichen - an der Packstation - befindlichen Päckchen anzeigen.

Vorgehensweise

Entwerfen Sie also eine Container-Klasse `Packfirma<T>` Der Konstruktor legt eine als Argument übergebene Anzahl leerer Queues im Vektor an. Weiterhin sind die Methoden `size()`, `empty()`, `push()`, und `pop()` zu deklarieren.

Die Methode `size()` wird zweifach überladen. Einmal liefert sie die Anzahl aller Elemente, und einmal (mit Argument int i) liefert sie die Anzahl der Elemente der i-ten Queue. Entsprechend sind die Methoden `empty()` bzw. `empty(int i)` zu definieren.

Die Methode `push()` fügt ein Element in der kürzesten Queue ein.

Abgeholt werden die Elemente durch die Methoden `pop()` bzw. `pop(i)`

Zum Testen definieren Sie in der main-Funktion einen Container vom Typ `Firma<int>`. Ein Paket wird durch eine Zahl dargestellt. In einer Schleife können Sie dazu Zufallszahlen zwischen 0 und 100 in den Container einfügen. Ein Teil der Pakete verteilt der Weihnachtsmann, nach 50 Paketen lassen Sie sich die restlichen Anzeigen. Verwenden Sie dafür die Methode `pop()`

```
Es wurden 4 Packstationen aufgestellt.

Die Paeckchen werden nun an die Packstationen verteilt.

Rentiere schwaemen aus und verteilen 50 Pakete, wobei die Dauer der Paketzustellung nic

Die restlichen, noch zu verteilenden Pakete:

1.Packstation:   66  52   5  94   8  10  88  56  72  67  66  32  51  96

2.Packstation:   39   7  46   5  43  21  68  97  97  26   9  71  11  82  99  65

3.Packstation:   23  17   8  96

4.Packstation:   65  33  72  76  51   4  63  83  84  22  76  72  61  92  61  84
```

Hinweis

Dieses Verfahren wird auch in der Datenkommunikation zwischen entfernten Rechnern angewendet. Auf dem Weg zum Zielrechner speichern dabei sogenannte *Router* die Nachrichten in Queues, bevor sie auf die nächste freie Leitung in Zielrichtung gesetzt werden. Die Router berechnen hier für jedes Paket das weitere Ziel, was meist über umfangreiche Adresstabellen geschieht.

Viel Erfolg, Frohe Weihnachten
und einen Guten Rutsch ins Jahr 2009!