

LABORATION 6

AFFÄREN

Rapport av

Henrik Bäck, 850611-6253
Mathias Andersson, 850424-6292

Handledare

Kerstin Anderson
Nils Däverhög

KARLSTADS UNIVERSITET
2007-09-03

Innehållsförteckning

INNEHÅLLSFÖRTECKNING	2
1. ANTAGANDEN.....	3
2. ANVÄNDARBESKRIVNING.....	3
<i>Lägga in personer i affären</i>	<i>3</i>
<i>Visa kunder i affären.....</i>	<i>3</i>
<i>Skicka en kund i affären till kassa.....</i>	<i>3</i>
<i>Öppna en ny kassa</i>	<i>3</i>
<i>Betjäna en kund.....</i>	<i>3</i>
<i>Stäng en kassa.....</i>	<i>3</i>
<i>Titta på kassa-köer.....</i>	<i>3</i>
<i>Avsluta.....</i>	<i>3</i>
3. SYSTEMÖVERSIKT.....	3
<i>Filrelationer</i>	<i>3</i>
4. DETALJERADE MODULBESKRIVNINGAR.....	3
<i>Klassen Affare</i>	<i>3</i>
<i>Klassen Queue.....</i>	<i>3</i>
<i>Klassen List.....</i>	<i>3</i>
<i>Klassen Node.....</i>	<i>3</i>
<i>Drivrutiner</i>	<i>3</i>
5. SAMMANFATTNING.....	3
6. PROBLEM	3
7. REFERENSLISTA.....	3
BILAGOR	3

1. Antaganden

Det antogs att det kan finnas ett oändligt antal personer i affären, och ett oändligt antal köer. Det kan även finnas ett oändligt antal personer i varje kö.

2. Användarbeskrivning

Programmet är utvecklat för att simulera en affär, användargränssnittet är utformat i en CLI¹-miljö. Genom en meny kan användaren välja olika funktioner för att hantera affären och dess kunder. Det finns möjligheter att öppna kassor, flytta folk till kassa-kön, mm.

```
Välkommen in till den virtuella butiken
-----
                Skriven av: Henrik Bäck
                   Mathias Andersson

-----
                Meny
-----
1. Någon kliver in
2. Någon går till kassan
3. Någon betjänas
4. En ny kassa öppnas
5. En kassa stängs
6. Titta på kassorna
7. Mingla med kunderna
0. Stäng affären

Ange ditt val:
```

Fig 1. Programmets huvudmeny

Lägga in personer i affären

För att lägga till en person till butiken ange menyalternativ 1 i huvudmenyn. En text kommer att visas på skärmen som ombeder användaren att mata in personens namn och ålder. Efter att detta är gjort är personen tillagd i affären.

Visa kunder i affären

Det är möjligt att se vilka personer som finns inne butiken, alltså de som inte står i någon kassakö. Från huvudmenyn välj menyalternativ 7. En lista över samtliga kunder som finns i butiken kommer att visas.

¹ Command Line Interface

```
Minglar med kunderna
Kundnummer  Namn      Ålder
1            Rut       76
2            Pelle    18
3            Lisa     19
```

Fig 3. Exempel på kunder i butiken

Skicka en kund i affären till kassa

Genom menyalternativ 2 i huvudmenyn (Se Fig 1.) finns möjlighet att skicka någon av de kunder som finns inne i butiken till kassan. Ange vilket kundnummer kunden du vill skicka till kassan har samt vilken kassa du vill skicka denne till. Kunden kommer att hamna sist i den kassa-kö som du anger att denne skall gå till. För att visa samtliga kundnummer för kunder se funktionen för att visa kunder i butiken. Läs mer under rubriken **Visa kunder i affären** i denna manual.

Öppna en ny kassa

Om vilja skulle finnas eller att exempelvis trycket på de befintliga kassaköerna kan ännu en kassa öppnas. Genom att ange menyalternativ 4 i huvudmenyn öppnas ännu en kassa.

Betjäna en kund

Genom menyalternativ 3 i huvudmenyn är det möjligt att betjäna en kund som står i kö. Genom att ange vilken kö som skall betjänas kommer den första personen i denna kö att betjänas.

Stäng en kassa

Det är när som helst möjligt att stänga en kassa såvida det är mer än en kassa öppen. Genom att ange menyalternativ 5 i huvudmenyn kan du stänga en kassa. Ange endast vilken kassa du vill stänga. Skulle det finnas kunder i kö till kassan kommer dessa att omplaceras i en annan kö.

Titta på kassa-köer

För att se om eller vilka personer som står i en kö till en kassa kan du ange ange menyalternativ 6 i huvudmenyn. Ange vilken kassa du vill titta på. Den totala kön för kassan kommer att skrivas ut på skärmen.

Avsluta

Detta menyalternativ, 0, gör det möjligt att avsluta programmet. OBSERVERA! Vid avslutning kommer all inmatad data bli oåterställbar.

3. Systemöversikt

Programvaran består av fyra stycken klasser. Dessa klasser samarbetar på ett sådant sätt

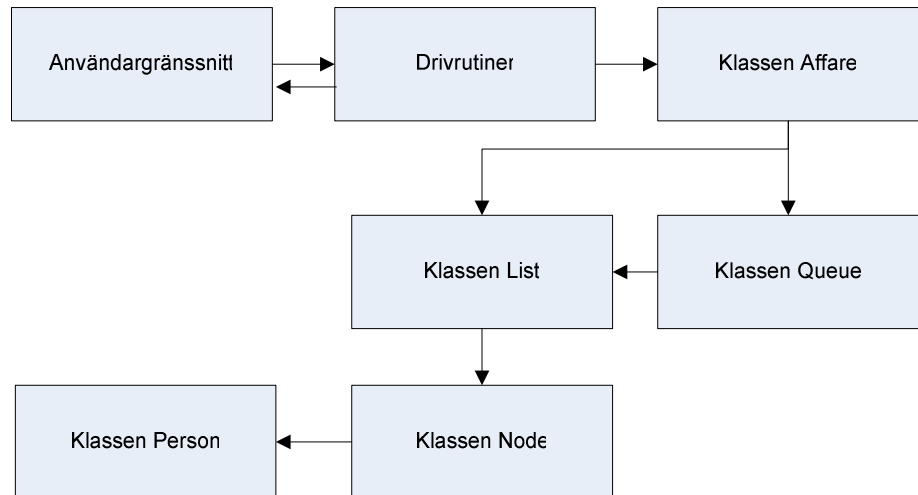


Fig 4. Systemöversikt

Filrelationer

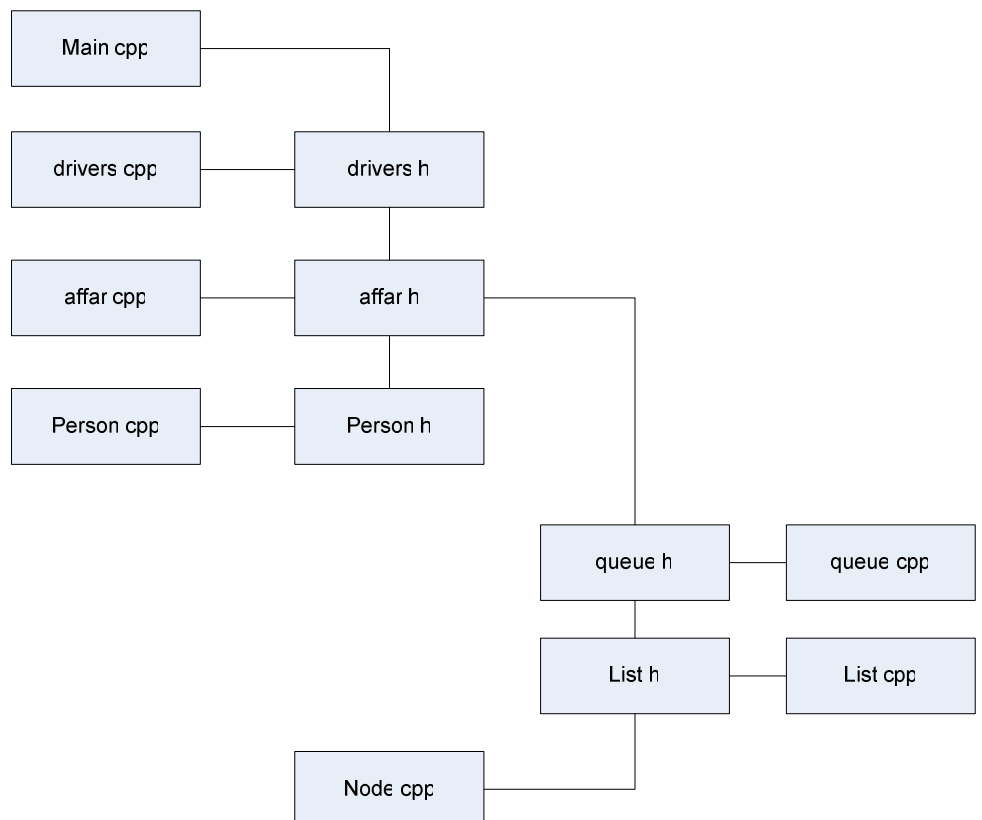


Fig 5. Filrelationer

Filen *main.cpp* har hand om själva main filen för programmet. Härifrån startas och anropas programmets funktioner. Filen inkluderar *drivers.h*

drivers.cpp innehåller affärens drivfunktioner. Även denna fil inkluderar *drivers.h* eftersom *main.cpp* och *drivers.cpp* separatkompileras. I och med att *drivers.h* inkluderas kommer även resterande .cpp och .h-filer att inkluderas. Se figur 5.

affar.h innehåller klassen affär. *Affar.h* inkluderas av *affar.cpp*. *affar.h* inkluderar även *person.h* som innehåller klassen person. *affar.h* inkluderar *queue.h*.

queue.h innehåller klassen Queue. Denna fil inkluderar *queue.cpp* som innehåller kö-klassens funktioner. *queue.h* inkluderar även *List.h* som innehåller klassen List.

List.h inkluderar *List.cpp* som innehåller List-klassens funktioner. *List.h* inkluderar även *Node.cpp* eftersom klassen List använder sig av klassen Node.

4. Detaljerade modulbeskrivningar

Klassen Affare

Klassen Affar ger tillgång till att skapa ett objekt av typen Affar

Villkor och beskrivningar för klassen Affar finns i Bilaga A sida 3.

Klassen Queue

Klassen Queue ger tillgång till att skapa ett objekt vars egenskaper är som hos en kö. Klassen Queue har följande funktioner/operationer:

- enqueue
- dequeue
- isEmpty
- getSize
- display

klassen har också en constructor och en destructor som körs när man skapar en instans av ett objekt respektive när objektet går ur scope². Dock har inte klassen Queue någon specifik destructor då klassen endast är ett skal för att hantera en sorts lista med specifika egenskaper.

² Objektet raderas från minnet och är inte längre användbart.

Villkor och beskrivningar för klassen Queue finns i Bilaga A sida 20.

Klassen List

Klassen List ger används av klassen Queue.

Klassen List har följande funktioner/operationer:

- insert
- remove
- isEmpty
- isElement
- getPos
- size
- getElement

klassen har också en constructor och en destructor som sköter minnesallokering.

Villkor och beskrivningar för klassen List finns i Bilaga A sida 14.

Klassen Node

Klassen Node är den klass som innehåller den data som önskas lagras.

Klassen Node har inga operationer.

Drivrutiner

Programmet har följande drivrutiner för att ta hand om programmets kö.

- mataInInt
- kommerIn
- gaTillKo
- visaKunder
- oppnaKassa
- tittaKo
- stangKassa
- betjana

All information om och hur dessa drivrutiner fungerar och kan användas finns i Bilaga A sida 9.

5. Sammanfattning

Projektet har gett förståelse i hur man kan sammankoppla olika sorters klasser och hur man kan använda sig av en kö. Projektet har också gett en kunskap i hur man kan skapa en samling av objekt i ett annat objekt.

Tidsåtgången för projektet höll på att falla på grund av ett oväntat problem. Detta problem kunde inte lösas då det var avsett och fick vänta till dagen därpå. Detta resulterade i att lite tid gick förlorad

6. Problem

Ett stort problem uppstod då en person skall flyttas från listan över kunder i butik till en kassa-kö. Det som uppstod var att personen försvann från kassakön även fast den skulle ha blivit tillagd. Eftersom funktionen som sköter detta även tog bort (avsiktligt) personen från listan över kunder så försvann personen i tomma intet.

Problemet med de försvinnande kunderna mellan listan över kunder och kassan löstes genom hjälp från handledare. En ändring i klassen List behövde göras och returvärdet för funktionen getElement från T till T&

7. Referenslista

www.cplusplus.com – Internet Resurs

Bilagor

Bilaga A – Programkod

Bilaga B – Klassdiagram