

Laboration 3 - Uppgift

Mallklasser och containerfunktioner
DT079G - Programmeringsmetodik

Syfte

Syftet med labben är att du ska lära dig skriva enklare klasser och funktioner som är parametriserade med typer (generiska klasser).

Priority queues

En priority queue är en kö där det finns en prioritet associerad med varje element. Det element som har högst prioritet ligger först i kön.

Du skall definiera en klass som kan representera en prioritetskö. De operationer som krävs för kön är:

`pop():T` - Hämta och ta bort det element som har högsta prioritet.

`push(T e)` - Lägg till ett element (med godtycklig prioritet).

`size():int` - Antal element i kön.

`empty():bool` - Töm kö.

Operationerna för en priority queue är snarlika som för en stack.

Du får på egen hand definiera vilken containerstruktur som håller reda på kön. Förslagsvis en dynamisk array, eller kanske hellre en länkad lista.

Containerstrukturen bör i alla lägen vara sorterad (efter prioritet, högsta först i strukturen, en invariant).

Du skall typparametrisera klassen (lämpligt namn är `p_queue`) med värdetyp `T`, dvs vilken elementtyp kön ska arbeta med, och jämförelsefunktion. Jämförelsefunktionen skall ta två instanser av elementtypen `T` och returnera `true` om argument ett är mindre än argument två. För typen `int` skulle funktionen kunna skrivas

```
bool less(int a1, int a2){  
    return a1<a2;  
}
```



Se föreläsningssanteckningar föreläsning 7 (predikat, sortering) för ytterligare exempel.

`p_queue` klassen ska naturligtvis använda predikatet/jämförelsefunktionen för att hålla kön sorterad.

Uppgift

Skapa klassen beskriven ovan i separat headerfil `p_queue.h`. Definiera kompilatordirektiv så att headerfilen kan användas på ett korrekt sätt. Fråga din handledare om du är osäker. Beskriv med kommentarer i headerfilen varför implementation måste ligga i headerfilen.

I filen `main.cpp` inkluderas mallklassen för användning. Uppgiften som ska lösas är matchning av köp och säljordrar för en aktiebörs med en aktie och tre aktiva mäklare. Utökningen av antalet aktier är faktiskt väldigt enkel, men tjänar inget syfte för labben.

Definiera två köer. En för säljordrar, en för köpordrar. En säljorder innehåller namnet på säljaren (använd gärna Textklassen från tidigare labb) och det pris som säljaren vill ha. En köporder innehåller namnet på köparen och det pris som köparen maximalt är beredd att betala. Köerna ordnas så att de ordrar som har lägst pris har högst prioritet.

Ett köp/sälj görs om säljaren efterfrågar ett lägre pris än vad köparen bjuder. Om köparen bjuder mindre än säljorden blir köporden ogiltig och kastas. Annars genomförs köpet och köparens namn och säljarens namn och överenskommet pris skrivs ut på skärmen.

I `main.cpp` behöver du således definiera klasser för köp och säljordrar, och jämförelsefunktioner eller funktionsobjekt (valet är ditt om du ska implementera en funktion eller ett funktionsobjekt. Motivera ditt val).

I mainfunktionen sätter du upp 21 köp och säljordrar med mäklarna 'Erik Pendel', 'Jarl Wallenburg' och 'Joakim von Anka'. Varje mäklare får sju sälj och köpordrar vardera. Priset på sälj och köpordrarna slumpas mellan säg 15 och 30 kronor. Då programmet körs ska det skriva ut resultatet av alla försäljningar. Det kommer alltså troligtvis bli färre än 21 köp/sälj utskrivna på skärmen.

Tips, börja med att implementera kön och felsök den. Testa den med heltal eller någon annan enkel datatyp för att fånga syntaxmissar. (Om du känner dig väldigt osäker kan du till och med börja med att implementera kön utan mallstöd, för ex typen `int`). Därefter håller du köp och säljklasser så enkla som möjligt för uppgiften. Detsamma med jämförelsefunktioner/funktionsobjekt.