Föreläsning 3

# Agenda

Containrar och Standardbibliotek

Vector

Deque

Grid

# Containrar och STL (standard library)

Varför inte bara använda array? I c++ har de fix storlek och kan inte enkelt ändra storlek, arrayen vet inte ens om sin egen storlek (ingen .length).

Du kan även lägga något utanför array utan exception som i Java vilket kan leda till stora problem

Du kan heller inte göra vissa nödvändiga operationer som att stoppa in saker mitt i, söka efter värde och dylikt.

**Container** är ett objekt (en datastruktur) som lagrar element med mer användbarhet. Förstå hur den fungerar, inte bara hur den används!

**STL** har flera delar som består av komponenter som kan interagera med varandra.

Containrar, t.ex. vector.

Iteratorer, objekt som modifierar givna intervall av data. Gemensamt gränssnitt gör att de kan verka på godtyckliga containrar

Algoritmer är funktioner i STL som opererar på data specifierat av iteratorer.

Adaptrar är objekt som transformerar ett objekt på en form till en annan.

Funktionsobjekt för att hjälpa till med callback-funktioner som ofta krävs.

# Vector

Vector är en samling objekt med 0-baserade index. Den har dynamisk kapacitet som utökas vid behov av operationen push\_back(), som sätter in sist.

Har iteratorer som pekar på nästa element och tar sig dit, likt java

Liknar arraylist i java

Gör du en for-loop med t.ex.

For (int i = 0, i < v.size(), i++)

Kan kompilatorn klaga på att du jämför en int ( i ) med en size\_t som returneras av v.size()

Int är signed int (med tecken) medan v.size som är size\_t typ är unsigned då den ej kan vara negativ.

Typdeklarationen auto kan användas när kompilatorn vet sedan tidigare vilken typ det är den jobbar med, istället för att skriva ut hela.

Auto kan vara väldigt lämpligt att använda för smidighetsskull om en variabel används på många ställen. Då räcker det att ändra variabeldeklarationen.

# Deque

Vector är go to för sekvenser, men deque (double ended queue) är lämplig när man alltid sätter in saker först och sist. Effektivitet är anledning till att vector är att föredra annars. En deque håller reda på ett antal vector för att kunna placera saker i bägge ändar på ett effektivt sätt (ej behöva flytta bak alla andra element, utan endast i en liten del), vilket gör den olämplig för de flesta områden.

# Grid