

Lab 5 – Standardbiblioteket och generiska algoritmer

Objektorienterad programmering i C++

Användning av generiska algoritmer och funktionsobjekt.

Lab 5 – Standardbiblioteket och generiska algoritmer

Syftet med laborationen är att få praktiskt erfarenhet av funktionsobjekt, mallar, list, iteratorer och några generiska funktioner från STL. Kursiverad text är namn på komponenter som ska användas i uppgiften.

Uppgift

Du ska konstruera ett program som slumpar och bearbetar slumpade 20 tal i intervallet [1000,2000]. Typen på talen ska antingen vara av typen `int` eller `double`. Talen ska lagras i en `list`. Val av datatype (`int/ double`) ska användaren göra först i programmet.

Efter programmet har startas ska listan vara tom.

Innehållet i listan ska kunna behandlas enligt följande:

a) fylla listan med nya slumpstal (*generate* med lämpligt *funktionsobjekt*. Slumpningen ska göras med komponenter från header-filen `<random>`).

b) beräkning av summan av alla tal (*accumulate*)

c) beräkning av medelvärdet av alla tal (*iteratorer och funktionsmall*)

d) söka efter det första talet som ligger i intervallet [1500, 1900] (*find_if* med lämpligt *funktionsobjekt*)

e) dividera alla tal med 2 (*transform* med lämpligt fördefinierat *funktionsobjekt*)

f) byta plats på första och sista elem, näst första och näst sista elem osv. (*iter_swap*)

g) söka efter största och minsta talet (*max_element* och *min_element*)

h) sortera talen i stigande ordning (lämplig form av *sort*)

i) skriva innehållet i listan till fil (*copy* och *ostream_iterator*)

j) tömma listan på innehållet

k) läsa värden från fil till listan (*copy*, *istream_iterator* och *back_inserter*)

l) skriva ut alla tal (*for_each*)

Ovanstående operationer ska alltså kunna tillämpas på den valda typen, `int` eller `double`. När `double` är vald ska decimaler vara synliga.

Programmet ska vara menystyrt.

Testprogrammet ska utnyttja funktionsmallar.

Koden får inte 'dubbleras' på så vis att 'samma' funktion skrivs i en *int-version* och en *float-version*.

Tips: Sedan användaren valt `int` eller `float` skapas en `list<int>` eller `list<float>`. Låt denna vara argument till en funktionstemplate från vilken resten av arbetet görs:

```
template<typename T>
void myMainFunc<list<T> &theList) {
    Här är T den valda typen.
}
```