STL Basqa containerler (Lesson 24 - vector, list kecmisdik)

1) set

2) map

3) multiset

4) multimap

To use:

#include <set>

#include <map>

1) Set

Set in arxasinda binary tree durur

ve setde elementler tekrarlanmir, her elementi unikal

olaraq saxlamaga calisir

Set elementleri sort edib saxlamaga calisir

2) Map

Map in arxasinda binary tree durur

En cox islenen map dir - key(.first) value(.second) mentiqi ile isleyir

Ve key lere gore sort edib saxlayir

Keyler unikal olur her dilde

ypx["10-KY-248"] = "Range Rover";

"10-KY-248" mapda varsa valuesini deyisir, yoxdursa value kimi "Range Rover"-i

key kimi "10-KY-248"-i elave edir

structda map saxlamaqla onun ozellik sayini deyisken saxlaya bilerik

3) Multiset

Setle eynidir lakin unikalliq yoxdur

4) Multimap

melumat saxlamaq ucundur

indeksle catmaq olmur : //cout << multimap["AUDI"] << endl; // error

unikalliq yoxdur

////////////////////////////////////////////////////////////////////////

#include "list" (next, previous (double linked list))

#include <forward\_list> (next (single list))

////////////////////////////////////////////////////////////////////////

Predicates - qisa bool qaytaran funksiya

listin sort metodunda istifade oluna biler prediktalar

sorta predikatin (funksiyanin) adresini gonderirik

// Predicates - qisa bool qaytaran funksiya

bool byNameAsc(const Kitty& k1, const Kitty& k2) // - Predicate

{

return k1.GetName() < k2.GetName();

}

bool byAgeAsc(const Kitty& k1, const Kitty& k2) // - Predicate

{

return k1.GetAge() < k2.GetAge();

}

void main()

{

list<Kitty> kitties;

kitties.push\_back(Kitty("Toplan", "Toplan123", 35));

kitties.push\_back(Kitty("Mestan", "Qara", 12));

kitties.push\_back(Kitty("Black", "Jav Jav", 67));

kitties.push\_back(Kitty("Garfield", "Hungry Cat", 16));

PrintKitties(kitties);

kitties.sort(byNameAsc);

cout << endl;

PrintKitties(kitties);

kitties.sort(byAgeAsc);

cout << endl;

PrintKitties(kitties);

}