Практичне заняття

Алгоритми і структури даних STL

Мета: знасвоєння навичок створення та роботи із складними структурами даних на прикладі шаблонів мови С++ , а також навичок застосування високорівневих підходів до програмування.

Контейнери дають можливість зберігати обєкти одного типу з можливістю доступу, пошуку впорядкування, тощо. Різні контейнери забезпечують різну швидкість доступу, створення пошуку і т.д., а також мають різні вимоги до оперативної пам’яті. Доступ до елементів контейнерів здійснюється а допомогою ітераторів, які є аналогами вказівників. Різіні контейнери підтримують різні типи ітераторів для різних операцій. Ітератори можуть підтримувати операції введення, виведення, напрямку зміни, порівняння, тощо. Алгоритми дають можливість виконувати різні операції з контейнерами використовуючи функціональний підхід. Стандартні алгоритми, такі як сортування, пошук вимагають передачі функцій, або функторів користувача для виконання дій, а також ітераторів початку і кінці контейнера.

Приклад програми, яка використовує стандартні контейнери, алгоритми та ітератори для структур даних користувача

#include <iostream>

#include <iterator>

#include <sstream>

#include <string>

#include <vector>

#include <algorithm>

using namespace std;

//define you hash tables for historgams

//all data columns are strings

struct data\_column:public string{};

// user's data structure

struct my\_data{

int num; // number of element

//all columns

data\_column c[26];

};

//input operator for single data column

istream& operator >> (istream& is, data\_column &f){

char c;

int num=0;

f.clear();

while(is.get(c)){

switch(c){

case '"': num++; /\*f+=c\*/; break;

case ',': if( !(num&1) ) return is;

default: f+=c;

}

}

return is;

}

// input operator for user data structure

istream& operator >> (istream& is, my\_data& d){

string l;

getline(is,l); // read line from stream

istringstream tmpi(l); // create inpot stream from string

//input number and first comma

tmpi>>d.num>>d.c[0];

//input all columns

for (int i=0;i<26; i++){

tmpi>>d.c[i];

}

//add your data to histogram here

return is;

}

int main(){

vector <my\_data> dd; //create vector of my structures

//read stdin into this vector

copy( istream\_iterator <my\_data>(cin),

istream\_iterator <my\_data> (),

back\_inserter<vector <my\_data> >(dd));

//create your complex histogram here

// apply lambda-function to each vector element

for\_each(dd.begin(), dd.end(),

//output gender

[&] (my\_data& d){cout << d.num<<" "<<d.c[9]<<endl;}

);

}

—----------------------------------------------

Для компіляції цієї на кластері необхідно використовувати компілятор icc, або gcc-5.2 із підтримкою стандарту с++11

[saa@plus7 tmp]$ ml gcc/5.2.0

[saa@plus7 tmp]$ g++ -std=c++11 file.cpp

або

[saa@plus7 tmp]$ ml icc

[saa@plus7 tmp]$ icc -std=c++11 file.cpp

приклад запуску

[saa@plus7 tmp]$ ./a.out < lab3\_stl\_dataset\_1.csv.csv

1 Female

2 All

3 All

4 All

5 All

6 All

7 All

8 Female

9 All

10 All

11 All

…

Дані доступні на гугл-диску і мають формат

"Rank","NCT Number","Title","Acronym","Status","Study Results","Conditions","Interventions","Outcome Measures","Sponsor/Collaborators","Gender","Age","Phases","Enrollment","Funded Bys","Study Type","Study Designs","Other IDs","Start Date","Primary Completion Date","Completion Date","First Posted","Results First Posted","Last Update Posted","Locations","Study Documents","URL"

Тобто дані кожного об’єкту починаються з нового рядка, поля розділені комами (csv формат).

## Завдання

1. Скомпілюйте програму і перевірте її працездатність.
2. Розберіть, що ця програма робить і як реалізована. Зверніть увагу, як визначені типи даних для кожного поля таблиці (data\_column), рядка таблиці (my\_data) і всієї таблиці (vector <my\_data> dd). Як вводяться дані у кожне поле (istream& operator >> (istream& is, data\_column &f)), рядок (istream& operator >> (istream& is, my\_data& d)) і всі рядки таблиці (copy( istream\_iterator <my\_data>(cin), istream\_iterator <my\_data> (), back\_inserter<vector <my\_data> >(dd));). Як відбувається виведення даних за допомогою лямбда-виразу
3. Відповідно до індивідуального завдання (номер відповідає номеру студента у списку), вискориствуючи контейнери STL на основі наданої програми створіть програму, яка зчитує складні структури даних із стандартного вводу, виконує з ними вказані дії і виводить результат на стандартний вивід. У процесі створення програм необхідно використовувати лише високорівневі конструкції: контейнери, ітератори, алгоритми, функтори, або лямбда-функції. По можливості не використовуйте низькорівневі конструкції, зокрема цикли. Наведена вище програма виконує зчитування даних, необхідно лише побудувати гістограми, використовуючи STL контейнери. Для гістограми (хеш-таблиці) використайте контейнери map або multimap. Подумайте, як створити ключі до хеш-таблиці.

### Індивідуальні завдання

1. Створіть хеш-таблички даних за статтю і віком і їх комбінаціями. Підрахуйте кількість пацієнтів різних статей і і віку. Виведіть отриману гістограму розподілів за віком
2. Створіть хеш-таблички даних за назвою обстеження і віком і їх комбінаціями. Підрахуйте кількість пацієнтів з різними назвами і віком. Виведіть отриману гістограму розподілів за назвами від віку
3. Створіть хеш-таблички даних за локаціями і віком і їх комбінаціями. Підрахуйте кількість пацієнтів з різними титулами і віком. Виведіть отриману гістограму розподілів за локаціями від віку
4. Створіть хеш-таблички даних за типом обстеження і віком і їх комбінаціями. Підрахуйте кількість пацієнтів з різними типами обстеження і віком. Виведіть отриману гістограму розподілів за типами обстежень від віку
5. Створіть хеш-таблички даних за спонсорами-колабораторами і статтю і їх комбінаціями. Підрахуйте кількість пацієнтів з різними колабораторами і статтю. Виведіть отриману гістограму розподілів за колабораторами і статтю
6. Створіть хеш-таблички даних за тривалістю обстеження і статтю і їх комбінаціями. Підрахуйте кількість пацієнтів з різними тривалостями обстежень і статтю. Виведіть гістограму розподілів за тривалостями обстеження і статтю
7. Створіть хеш-таблички даних за результатми і віком і їх комбінаціями. Підрахуйте кількість пацієнтів з різними результатми обстежень і выком. Виведіть отриману гістограму розподілів за результатми обстеження і выком
8. Створіть хеш-таблички даних за interventions і віком і їх комбінаціями. Підрахуйте кількість пацієнтів з різними interventions і выком. Виведіть отриману гістограму розподілів за результатми обстеження і віком
9. Створіть хеш-таблички даних за outcome measures і віком і їх комбінаціями. Підрахуйте кількість пацієнтів з різними interventions і выком. Виведіть отриману гістограму розподілів за outcome measures і віком
10. Створіть хеш-таблички даних за Study designs і віком і їх комбінаціями. Підрахуйте кількість пацієнтів з різними interventions і выком. Виведіть отриману гістограму розподілів за study design і віком

**Контрольні питання**

1. Що таке контейнери, ітератори, алгоритми, функтори, лямбда-вирази?
2. Які переваги і недоліки має застосування STL?

**Література**

1. Огляд стандартної бібліотеки шаблонів <https://learn.microsoft.com/en-us/cpp/standard-library/cpp-standard-library-overview?view=msvc-170>
2. Підручник з STL <https://www.geeksforgeeks.org/cpp-stl-tutorial/>
3. Огляд C++ <https://en.cppreference.com/w/>