Федеральное государственное бюджетное образовательное учреждение высшего образования   
"Национальный исследовательский университет "МЭИ"

РАЗРАБОТКА ПРОГРАММНОГО ОБСЕПЕЧЕНИЯ СИСТЕМ УПРАВЛЕНИЯ

ЛАБОРАТОРНАЯ РАБОТА №1  
«Основы языка C++»

Студент:  
Никифорова Екатерина Владимировна  
Группа А-03-19

Преподаватель:   
Мохов Андрей Сергеевич

Москва 2020

Вариант 13

Постановка индивидуального задания

После вывода гистограммы запрашивайте у пользователя, доволен ли он результатом. Если ответ отрицательный, позвольте ввести новое количество столбцов и перестройте гистограмму. Процесс может повторяться сколько угодно раз.

Логика решения

* Создадим новую переменную res, которая будет отвечать за удовлетворенность результатом работы программы пользователем. И присвоим ей значение 1.
* Так как требуется изменить количество столбцов, то есть размер корзин(bin\_count), в случае неудовлетворенности результатом пользователя, нужно повторять действия с ввода нового размера.
* Для этого создадим цикл while (…) {…} и занесем туда все операторы, начиная с ввода размера корзин. Условием цикла выступит, созданная переменная res.
* В конец цикла добавим строки, в которых будет вопрос об удовлетворенности пользователя результатом и действия при согласии и несогласии.
* Для большей конкретики пусть ввод пользователем 1 является неудовлетворенностью, а 0 – удовлетворенностью.

Код программы

#include <iostream>

#include <vector>

const size\_t MAX\_WIDTH = 80;

const size\_t MAX\_AST = MAX\_WIDTH - 4;

using namespace std;

int main()

{

cerr << "Enter the quantity numbers: " << '\n';

size\_t number\_count;

cin >> number\_count;

vector <double> numbers(number\_count);

cerr << "Enter the number values: " << '\n';

for(size\_t i = 0; i < number\_count; i++)

{

cin >> numbers[i];

}

double res = 1;

while (res != 0)

{

{

cerr << "Enter the size of bin: " << '\n';

size\_t bin\_count;

cin >> bin\_count;

vector <size\_t> bins(bin\_count, 0);

double min = numbers[0];

double max = numbers[0];

for(double number : numbers)

{

if(number > max)

{

max = number;

}

else if(number < min)

{

min = number;

}

}

for(double number : numbers)

{

size\_t bin\_index = (size\_t)((number - min) \* bin\_count / (max - min));

if (number == max)

{

bin\_index--;

}

bins[bin\_index]++;

}

double koeff = 1;

size\_t max\_count = 0;

for(size\_t bin : bins)

{

if (max\_count < bin)

{

max\_count = bin;

}

}

if (max\_count > MAX\_AST)

{

koeff = static\_cast <double> (MAX\_AST) / max\_count;

}

for(size\_t bin : bins)

{

if(bin < 100)

{

cout << " ";

}

if (bin < 10)

{

cout << " ";

}

cout << bin << "|";

size\_t height = bin \* koeff;

for(size\_t i = 0; i < height; i++)

{

cout << "\*";

}

cout << endl;

}

cout << "Are you satisfied with the result? Yes(0) or No(1)? ";

cin >> res;

}

}

return 0;

}