**Министерство образования и науки Российской Федерации**

**Федеральное государственное автономное образовательное учреждение**

**«Уральский федеральный университет имени первого Президента России Б.Н Ельцина»**

**ИРИТ-РТФ**

**Кафедра информационных технологий**

**О Т Ч Е Т**

**По практике по объектно-ориентированному программированию**

**Лабораторная работа №2**

**Студент Трефилов А.А.**

**Группа РИ-240002**

**Преподаватель Бабич Л.В.**

**Екатеринбург 2015**

**Задача №1**

Цель: Построить калькулятор времени, способный складывать или вычитать два значения времени и переводить из секунд в формат ЧЧ:ММ:СС и наоборот.

Алгоритм: пользователь вводит в командную строку время- операция- время (чч:мм:сс+(-)чч:мм:сс). Программа считывает первое и второе время, знак операции, затем с помощью отдельной функции переводит эти времена в секунды. После программа выбирает в зависимости от знака, какую функцию ей выполнять дальше: сложение или вычитание. Делает необходимое действие с секундами и не покидаю данную область функции делает перевод времени в нужный формат, с помощью дополнительно прописанной функции, и выводит результат в консоль.

Код:

#include <iostream>

#include <string>

using namespace std;

struct Time

{

int HH, MM, SS, sec;

};

int tts(Time clock);

int sum(int sec1, int sec2);

int supstr(int sec1, int sec2);

Time stt(Time clock3);

int main()

{

setlocale(LC\_CTYPE, "Russian");

Time clock1;

Time clock2;

Time clock3;

clock3.HH=clock3.MM=clock3.SS=0;

char c, znak;

cin >> clock1.HH >> c >> clock1.MM >> c >> clock1.SS;

cin >> znak;

cin >> clock2.HH >> c >> clock2.MM >> c >> clock2.SS;

clock1.sec = tts(clock1);

clock2.sec = tts(clock2);

clock3.sec=0;

if (znak == '+')

{

clock3.sec = sum(clock1.sec, clock2.sec);

}

if (znak == '-')

{

clock3.sec = supstr(clock1.sec, clock2.sec);

}

Time clock = stt(clock3);

cout << "Ответ: " << clock.HH << ":" << clock.MM << ":" << clock.SS << endl;

system ("pause");

return 0;

}

int tts(Time clock)

{

clock.sec = clock.HH \* 3600 + clock.MM \* 60 + clock.SS;

return clock.sec;

}

int sum(int sec1, int sec2)

{

int clock;

clock = sec1 + sec2;

return clock;

//stt(clock3);

}

int supstr(int sec1, int sec2)

{

int clock;

clock = sec1 - sec2;

return clock;

//stt(clock3);

}

Time stt(Time clock3)

{

clock3.HH = clock3.sec / 3600;

clock3.MM = clock3.sec / 60 - clock3.HH \* 60;

clock3.SS = clock3.sec - clock3.HH \* 3600 - clock3.MM \* 60;

return clock3;

}

**Задача №2**

Цель: Построить калькулятор дробей, который будет складывать, вычитать, умножать и делить их, а затем при возможности упрощать результат.

Алгоритм: пользователь записывает пример, составленный из двух дробей и знака, по которому определяется операция и выполняется соответствующая функция. Так же считывает числитель и знаменатель каждого числа. Функция сложения: мы должны получить новый числитель и знаменатель. Для числителя мы умножаем первый числитель на второй знаменатель и складываем с произведением второго числителя с первым знаменателем, а для знаменателя просто перемножаем оба заданных знаменателя. Функция вычитания аналогично, но вместо операции сложения выполняется операция вычитания. Функция умножения: перемножаем знаменатели- это у нас новый знаменатель и перемножаем числители- числитель. Для деления делаем умножения “накрест”. И для упрощения результата мы в конце каждой функции вставим еще одну функцию, в которой и будет происходить сокращение дроби по возможности. В этой функции мы будем подбирать число от 1 до того, пока она не достигнет значения числителя или знаменателя. И на каждое полученное число мы пробуем делить без остатка числитель и знаменатель. Если это выполняется, то обе переменные сокращаются на это число. После всего результат выводится в консоль.

Код:

#include <iostream>

#include <string>

using namespace std;

struct vvod

{

int a, b;

};

vvod summ(vvod chislo1, vvod chislo2);

vvod razn(vvod chislo1, vvod chislo2);

vvod umn(vvod chislo1, vvod chislo2);

vvod del(vvod chislo1, vvod chislo2);

vvod sokr(vvod otvet);

int main()

{

setlocale(LC\_CTYPE, "Russian");

char znak, f;

vvod chislo1;

vvod chislo2;

cin >> chislo1.a >> f >> chislo1.b >> znak >> chislo2.a >> f >> chislo2.b;

if (znak == '+')

{

vvod otvet = summ(chislo1, chislo2);

vvod vivod = sokr(otvet);

cout << " = " << vivod.a << "/" << vivod.b << endl;

}

if (znak == '-')

{

vvod otvet = razn( chislo1, chislo2);

vvod vivod = sokr(otvet);

cout << " = " << vivod.a << "/" << vivod.b << endl; }

if (znak == '\*')

{

vvod otvet = umn( chislo1, chislo2);

vvod vivod = sokr(otvet);

cout << " = " << vivod.a << "/" << vivod.b << endl; }

if (znak == '/')

{

vvod otvet = del( chislo1, chislo2);

vvod vivod = sokr(otvet);

cout << " = " << vivod.a << "/" << vivod.b << endl; }

system ("pause");

return 0;

}

vvod summ(vvod chislo1, vvod chislo2)

{

vvod otvet;

otvet.b = chislo2.b \* chislo1.b;

otvet.a = chislo1.a\*chislo2.b + chislo1.b\*chislo2.a;

return otvet;

}

vvod razn(vvod chislo1, vvod chislo2)

{

vvod otvet;

otvet.b = chislo2.b \* chislo1.b;

otvet.a = chislo1.a\*chislo2.b - chislo1.b\*chislo2.a;

return otvet;

}

vvod umn(vvod chislo1, vvod chislo2)

{

vvod otvet;

otvet.b = chislo1.b \* chislo2.b;

otvet.a = chislo1.a \* chislo2.a;

return otvet;

}

vvod del(vvod chislo1, vvod chislo2)

{

vvod otvet;

otvet.b = chislo1.b \* chislo2.a;

otvet.a = chislo1.a \* chislo2.b;

return otvet;

}

vvod sokr(vvod otvet)

{ vvod vivod;

vivod.a =otvet.a;

vivod.b=otvet.b;

for(int i=1; i<=otvet.a && i<=otvet.b; i++)

{

int aa=otvet.a%i;

int bb=otvet.b%i;

if(!(otvet.a%i) && !(otvet.b%i))

{

vivod.a=otvet.a/i;

vivod.b=otvet.b/i;

}

}

return vivod;

}

**Задача №3**

Цель: Рассчитать количество площади под покраску стен в английской системе мер.

Алгоритм: В начале программы пользователя прося ввести количество комнат. Затем через указатель создается массив, размер которого равен введенному числу. После пользователь вводит запрашиваемые данные каждой комнаты. Последняя функция делает расчет площадей каждой комнаты и сразу переводит в нужную систему измерения. И в конце выводит общий размер площади и отдельно пола.

Код:

#include <stdio.h>

#include <iostream>

using namespace std;

struct Distance

{

int feet;

float inches;

};

struct Room

{

Distance x;

Distance y;

Distance z; //x- ширина, y- высота, z- длина.

};

void vvod(int i, Room\* box);

int sum(Room \*box, int i);

int sum1(Room \*box, int i);

int main()

{

setlocale(LC\_CTYPE, "Russian");

int i;

//char plosh;

cout << "Введите количество комнат, которые вы хотите перекрасить в новый цвет:" << endl;

cin >> i;

Room\* box = new Room[i];

vvod(i, box);

int s=sum(box, i), pol = sum(box, i);

cout << "Общая площадь стен будет " << s << " футов." << endl;

cout << "И общая площадь пола составляет " << pol << " Футов." << endl;

delete []box;

system ("pause");

return 0;

}

void vvod (int i, Room\* box)

{

int j;

for(j=0; j<i; j++)

{

int numkomn = j+1;

cin.sync();

cout << "Введите ширину " << numkomn << " комнаты в футах: " << endl;

cin >> box[j].x.feet;

cout << " И дюймов: " << endl;

cin >> box[j].x.inches;

cout << "Введите высоту " << numkomn << " комнаты в футах: " << endl;

cin>>box[j].y.feet;

cout << " И дюймов: " << endl;

cin>>box[j].y.inches;

cout << "Введите длину " << numkomn << " комнаты в футах: " << endl;

cin >> box[j].z.feet;

cout << " И дюймов: " << endl;

cin >> box[j].z.inches;

cout << endl;

}

}

int sum(Room \*box, int i)

{

float s=0, pol=0;

for(int j=0; j<i; j++)

{

float plst = 2 \* (box[j].y.feet + box[j].y.inches/144) \* (box[j].x.feet + box[j].x.inches/144 + box[j].z.feet + box[j].z.inches/144);

s +=plst;

}

return s;

}

int sum1(Room \*box, int i)

{

float s=0, pol=0;

for(int j=0; j<i; j++)

{

float plpol = (box[j].x.feet + box[j].x.inches/144) \* (box[j].z.feet + box[j].z.inches/144);

pol +=plpol;

}

return pol;

}

Выводы:

В данной работе я научился работать со структурами, передавать и возвращать переменные из одной функции в другую.