**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

Практические приемы построения многопоточных приложений. Вариант 18

**Исполнитель**

Студент группы БПИ192

Раджбхандари А. Д.

**Задание**

Задача о наследстве. У старого дона Энрике было два сына, у каждого из сыновей – еще по два сына, каждый из которых имел еще по два сына. Умирая, дон Энрике завещал все свое богатство правнукам в разных долях. Адвокат дон Хосе выполнил задачу дележа наследства в меру своих способностей. Правнуки заподозрили адвоката в укрывательстве части наследства. Требуется создать многопоточное приложение, которое при известных сумме завещания дона Энрике и доле каждого наследника, проверяет честность адвоката. При решении использовать принцип дихотомии.

Составление​​ программы.

С помощью принципа дихотомии проверяем количество денег у правнуков, сравнивая количество полученных денег с их долей в наследстве. В программе используются потоки из стандартной библиотеки C++.

Текст программы:

#include <iostream>

#include <thread>

using namespace std;

double fractions[8]; // доля каждого внука, согласно завещанию

double EachGreatGrandSonsMoney[8];// Сколько каждый внук получил на самом деле

void CheckTheFraction(int firstGreatGrandSon, int EndGreatGrandSon, double& result, int money) {

if (firstGreatGrandSon == EndGreatGrandSon - 1) {

result = fractions[firstGreatGrandSon] \* money;

if (result == EachGreatGrandSonsMoney[firstGreatGrandSon]) {

cout << "Адвокат не обманул внука номер " << firstGreatGrandSon + 1 << endl;

}

else

{

cout << "Адвокат обманул внука номер " << firstGreatGrandSon + 1<< endl;

cout << "Внук номер " << firstGreatGrandSon <<" должен был получить "<< result<< "денег" << ", а получил "<< EachGreatGrandSonsMoney[firstGreatGrandSon] <<endl;

}

return;

}

double leftGreatGrandSon, rightGreatgrandSon;

thread LeftGreatGrandSons(CheckTheFraction, firstGreatGrandSon, (EndGreatGrandSon + firstGreatGrandSon) / 2, ref(leftGreatGrandSon), money);

thread RightGreatGrandson(CheckTheFraction, (EndGreatGrandSon + firstGreatGrandSon) / 2, EndGreatGrandSon, ref(rightGreatgrandSon), money);

LeftGreatGrandSons.join();

RightGreatGrandson.join();

result = leftGreatGrandSon + rightGreatgrandSon;

}

int main() {

// money - сумма денег из завещания

// real\_money - деньги, которые получили правнуки

int money;

double real\_money;

setlocale(LC\_CTYPE, "rus");

do {

cout << "Введите сумму завещания дона Энрике" << endl;

cin >> money;

} while (money < 0 || money > INT32\_MAX);

cout << "Все доли могут быть размером от 0 до 1 (Доля это вещественное число от 0 до 1) " << endl;

for (int i = 0; i < 8; ++i) {

do {

cout << "Введите долю, которая положена правнуку №" << i + 1 << endl;

cin >> fractions[i];

} while (fractions[i] < 0 || fractions[i] > 1);

}

for (int i = 0; i < 8; ++i) {

do {

cout << "Введите сумму, которую плучил правнук №" << i + 1 << endl;

cin >> EachGreatGrandSonsMoney[i];

} while (EachGreatGrandSonsMoney[i] < 0 || EachGreatGrandSonsMoney[i] > DBL\_MAX);

}

// Считаем сумму денег у всех внуков с помощью потоков

thread Calculate(CheckTheFraction, 0, 8, ref(real\_money), money);

// Ждем окончания работы

Calculate.join();

return 0;

}

}