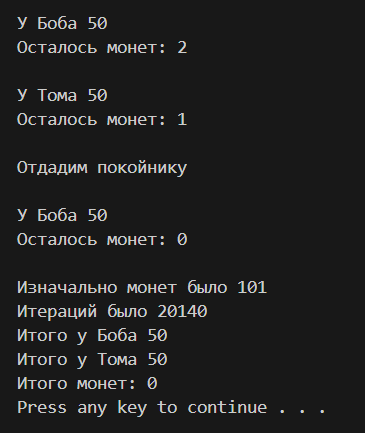
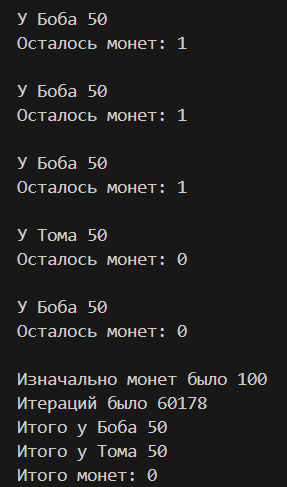
Лабораторная работа №5  
Соколов Арсений

Результат работы программы (см код в Приложении 1):



Два вора БЕЗ очереди пытаются взять монету из кучи, но берут, только в том случае, если сейчас их очередь. В данном случае монет было нечетное количество, поэтому последнюю отдали покойнику. Однако, видимо, Боб успел начать проверку, его ли сейчас очередь (псевдопараллельно с выдачей последней покойнику) и мы видим, что он не смог ничего взять. Судя по количеству запусков функции, мысли преследовали воров, но эти товарищи были быстрее.

Вот пример с четным количеством монет:



В этом запуске воры побили рекорд по количеству попыток взять монету, а покойнику ничего не досталось…

И мертвец, гремя костями, вдруг поднялся из земли:  
«Довели меня, проклятые, ей богу, довели!»  
Воры вмиг переглянулись, и помчались наутёк.  
А мертвец всё золото с собой в могилу уволок!

**ПРИЛОЖЕНИЕ 1**

***Программа «Event»***

#include <thread>

#include <mutex>

#include <string>

#include <iostream>

#include <cmath>

std::mutex m;

int Bob\_coins = 0;

int Tom\_coins = 0;

int coins = 101;

int count = 0;

void Func(std::string name)

{

    while (coins > 0) {

        m.lock();

        count++;

        if (name  == "Боб" && Bob\_coins == Tom\_coins && coins > 1) {

            Bob\_coins++;

            coins--;

        }

        else if (name  == "Том" && Bob\_coins > Tom\_coins) {

            Tom\_coins++;

            coins--;

        }

        std::cout << std::endl;

        std::cout << "У " << name << "а " << (name == "Боб" ? Bob\_coins : Tom\_coins) << std::endl;

        std::cout << "Осталось монет: " << coins << std::endl;

        if (Bob\_coins == Tom\_coins && coins == 1)

        {

            coins--;

            std::cout << std::endl;

            std::cout << "Отдадим покойнику" << std::endl;

        }

        m.unlock();

    }

}

int main()

{

int start = coins;

    std::cout << std::endl;

    std::thread thread1(Func, "Боб");

    std::thread thread2(Func, "Том");

    thread1.join();

    thread2.join();

    std::cout << std::endl;

std::cout << "Изначально монет было " << start << std::endl;

    std::cout << "Итераций было " << count << std::endl;

    std::cout << "Итого у Боба " << Bob\_coins << std::endl;

    std::cout << "Итого у Тома " << Tom\_coins << std::endl;

    std::cout << "Итого монет: " << coins << std::endl;

    return 0;

}