Отчёт

Лабораторная работа №5

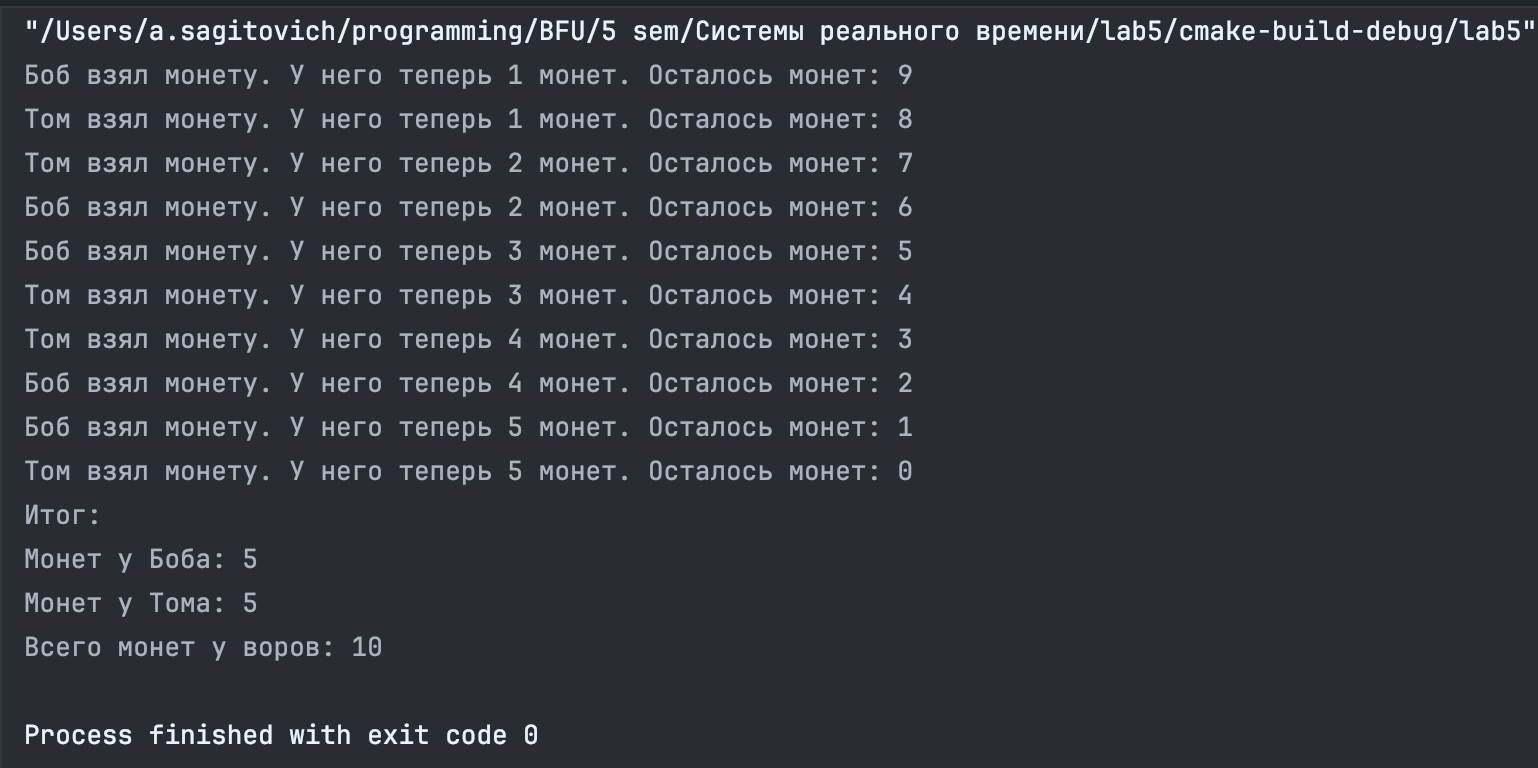
«100 и 1 монета»

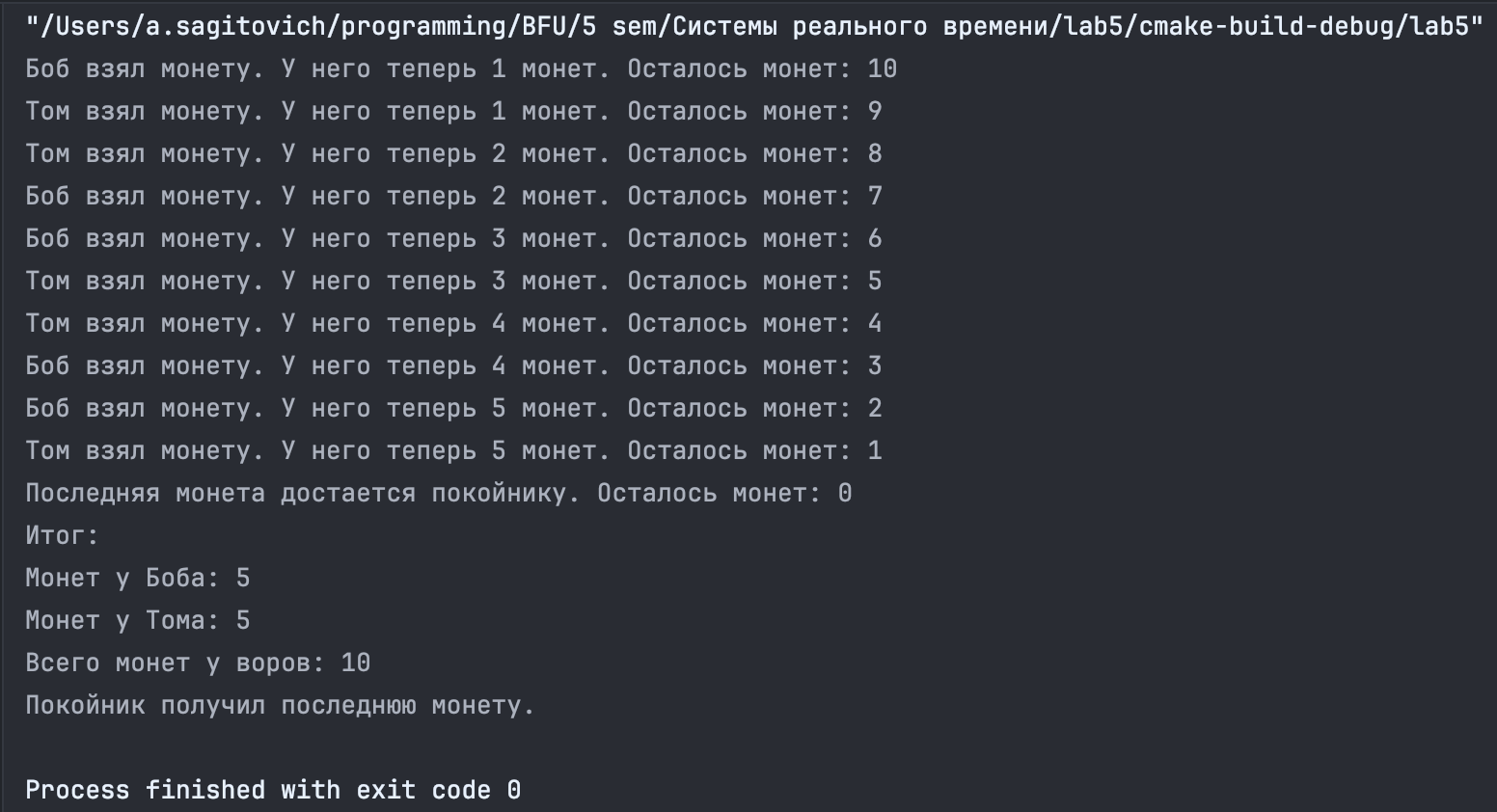
Системы Реального Времени

*Выполнил студент 3 курса МО  
Сагитов Александр Александрович*

**Результат выполнения программы**

P. S.: чтобы всё поместилось на один скриншот, уменьшил количество всех монет до 10 и 11 соответственно скриншотам.

****

****

**Приложение**

*main.cpp*

#include <iostream>  
#include <thread>  
#include <string>  
  
int coins = 10; // общее количество монет  
bool flag = false;  
int Bob\_coins = 0; // монеты Боба  
int Tom\_coins = 0; // монеты Тома  
std::mutex mtx; // мьютекс для синхронизации доступа к монетам  
  
  
  
void coin\_sharing(const std::string& name, int& thief\_coins, const int& companion\_coins) {  
 while (true) {  
 std::lock\_guard lock(mtx); // блокируем доступ coins  
  
 if (coins == 1 && Bob\_coins == Tom\_coins) {  
 coins--;  
 std::cout << "Последняя монета достается покойнику. Осталось монет: " << coins << std::endl;  
 break;  
 }  
  
 if (coins > 0 && thief\_coins <= companion\_coins) {  
 thief\_coins++;  
 coins--;  
 std::cout << name << " взял монету. У него теперь "  
 << thief\_coins << " монет. Осталось монет: "  
 << coins << std::endl;  
 } else if (coins == 0) {  
 break;  
 }  
 }  
}  
  
int main() {  
 if (coins % 2 != 0)  
 {  
 flag = true;  
 }  
  
 std::thread bob\_thread(coin\_sharing, "Боб", std::ref(Bob\_coins), std::ref(Tom\_coins));  
 std::thread tom\_thread(coin\_sharing, "Том", std::ref(Tom\_coins), std::ref(Bob\_coins));  
  
 bob\_thread.join();  
 tom\_thread.join();  
  
 std::cout << "Итог:\n"  
 << "Монет у Боба: " << Bob\_coins << "\n"  
 << "Монет у Тома: " << Tom\_coins << "\n"  
 << "Всего монет у воров: " << (Bob\_coins + Tom\_coins) << "\n";  
 if (flag) {  
 std::cout << "Покойник получил последнюю монету." << std::endl;  
 }  
  
 return 0;  
}