

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”  
КАФЕДРА ИИТ

ОТЧЁТ  
по лабораторной работе №7  
«Синхронизация потоков.»

Выполнил:

студент 3 курса  
группы ПО-9  
Мисиюк Алексей Сергеевич

Проверил:

Козик И. Д.

Брест 2023

**Цель работы:** научиться синхронизировать работу с потоками в приложениях.

### Вариант №3

Необходимо написать программу, используя знания, полученные в лабораторной работе №5. Потоки должны быть синхронизированы для выполнения поставленной задачи или код должен выполнять задачи последовательно.

Создать симулятор рынка с минимум 3 торговцами. Торговцы обслуживают по одному человеку в очереди, а после того, как очередь закончилась – начинаю обслуживать людей в очередях других торговцев.

### Код программы

#### main.cpp

```
#include <iostream>
#include <thread>
#include <mutex>
#include <queue>

const int PEOPLE = 30;

std::mutex coutMutex;

class Trader {
private:
    std::string name;
    std::queue<std::pair<int, int>> customers;
public:
    std::mutex mtx;

    Trader(std::string name) : name(name) {}

    void serveCustomers();

    void addCustomer(int id, int customerDifficulty)
    {
        std::unique_lock<std::mutex> key(mtx);
        customers.push({ id, customerDifficulty });
        key.unlock();
    }

    std::pair<int, int> handOverCustomer()
    {
        std::unique_lock<std::mutex> key(mtx);
        if (!customers.empty()) {
            std::pair<int, int> customer = customers.front();
            customers.pop();
            return customer;
        }
        key.unlock();
        return { -1, -1 };
    }

    void serveCustomer(std::pair<int, int> customer)
    {
        std::unique_lock<std::mutex> key(coutMutex);
        std::cout << name << " обслужет клиента #" << customer.first << " через " << customer.second
        << " ms." << std::endl;
        key.unlock();

        std::this_thread::sleep_for(std::chrono::milliseconds(customer.second));

        key.lock();
        std::cout << name << " обслужил клиента #" << customer.first << std::endl;
        key.unlock();
    }
};

Trader trader1("Торговец 1");
Trader trader2("Торговец 2");
Trader trader3("Торговец 3");
```

```

void Trader::serveCustomers()
{
    while (true) {
        if (!customers.empty()) {
            serveCustomer(this->handOverCustomer());
        }
        else
        {
            std::pair<int, int> customer = { -1, -1 };

            customer = trader1.handOverCustomer();
            if (customer.second < 0) customer = trader2.handOverCustomer();
            if (customer.second < 0) customer = trader3.handOverCustomer();

            if (customer.second < 0) {
                break;
            }

            serveCustomer(customer);
        }
    }
}

int main() {
    for (int i = 1; i <= PEOPLE; ++i) {
        int rand = std::rand() % 300;
        std::cout << "Человек #" << i << " - " << rand;

        if (i <= PEOPLE / 3) {
            trader1.addCustomer(i, rand);
            std::cout << " - 1";
        }
        else if (i <= PEOPLE / 3 * 2) {
            trader2.addCustomer(i, rand);
            std::cout << " - 2";
        }
        else {
            trader3.addCustomer(i, rand);
            std::cout << " - 3";
        }

        std::cout << std::endl;
    }

    std::thread thread1(&Trader::serveCustomers, &trader1);
    std::thread thread2(&Trader::serveCustomers, &trader2);
    std::thread thread3(&Trader::serveCustomers, &trader3);

    thread1.join();
    thread2.join();
    thread3.join();

    return 0;
}

```

## Пример работы

```

Человек #1 - 41 - 1
Человек #2 - 167 - 1
Человек #3 - 34 - 1
Человек #4 - 100 - 1
Человек #5 - 269 - 1
Человек #6 - 124 - 1
Человек #7 - 78 - 1
Человек #8 - 258 - 1
Человек #9 - 262 - 1
Человек #10 - 164 - 1
Человек #11 - 5 - 2
Человек #12 - 245 - 2
Человек #13 - 181 - 2
Человек #14 - 27 - 2
Человек #15 - 61 - 2
Человек #16 - 191 - 2
Человек #17 - 295 - 2

```

Человек #18 - 242 - 2  
 Человек #19 - 27 - 2  
 Человек #20 - 36 - 2  
 Человек #21 - 291 - 3  
 Человек #22 - 204 - 3  
 Человек #23 - 2 - 3  
 Человек #24 - 153 - 3  
 Человек #25 - 292 - 3  
 Человек #26 - 82 - 3  
 Человек #27 - 21 - 3  
 Человек #28 - 116 - 3  
 Человек #29 - 218 - 3  
 Человек #30 - 95 - 3  
 Торговец 1 обслужет клиента #1 через 41 ms.  
 Торговец 2 обслужет клиента #11 через 5 ms.  
 Торговец 3 обслужет клиента #21 через 291 ms.  
 Торговец 2 обслужил клиента #11  
 Торговец 2 обслужет клиента #12 через 245 ms.  
 Торговец 1 обслужил клиента #1  
 Торговец 1 обслужет клиента #2 через 167 ms.  
 Торговец 1 обслужил клиента #2  
 Торговец 1 обслужет клиента #3 через 34 ms.  
 Торговец 1 обслужил клиента #3  
 Торговец 1 обслужет клиента #4 через 100 ms.  
 Торговец 2 обслужил клиента #12  
 Торговец 2 обслужет клиента #13 через 181 ms.  
 Торговец 3 обслужил клиента #21  
 Торговец 3 обслужет клиента #22 через 204 ms.  
 Торговец 1 обслужил клиента #4  
 Торговец 1 обслужет клиента #5 через 269 ms.  
 Торговец 2 обслужил клиента #13  
 Торговец 2 обслужет клиента #14 через 27 ms.  
 Торговец 2 обслужил клиента #14  
 Торговец 2 обслужет клиента #15 через 61 ms.  
 Торговец 3 обслужил клиента #22  
 Торговец 3 обслужет клиента #23 через 2 ms.  
 Торговец 3 обслужил клиента #23  
 Торговец 3 обслужет клиента #24 через 153 ms.  
 Торговец 2 обслужил клиента #15  
 Торговец 2 обслужет клиента #16 через 191 ms.  
 Торговец 1 обслужил клиента #5  
 Торговец 1 обслужет клиента #6 через 124 ms.  
 Торговец 3 обслужил клиента #24  
 Торговец 3 обслужет клиента #25 через 292 ms.  
 Торговец 2 обслужил клиента #16  
 Торговец 2 обслужет клиента #17 через 295 ms.  
 Торговец 1 обслужил клиента #6  
 Торговец 1 обслужет клиента #7 через 78 ms.  
 Торговец 1 обслужил клиента #7  
 Торговец 1 обслужет клиента #8 через 258 ms.  
 Торговец 3 обслужил клиента #25  
 Торговец 3 обслужет клиента #26 через 82 ms.  
 Торговец 2 обслужил клиента #17  
 Торговец 2 обслужет клиента #18 через 242 ms.  
 Торговец 3 обслужил клиента #26  
 Торговец 3 обслужет клиента #27 через 21 ms.  
 Торговец 3 обслужил клиента #27  
 Торговец 3 обслужет клиента #28 через 116 ms.  
 Торговец 1 обслужил клиента #8  
 Торговец 1 обслужет клиента #9 через 262 ms.  
 Торговец 3 обслужил клиента #28  
 Торговец 3 обслужет клиента #29 через 218 ms.  
 Торговец 2 обслужил клиента #18  
 Торговец 2 обслужет клиента #19 через 27 ms.  
 Торговец 2 обслужил клиента #19  
 Торговец 2 обслужет клиента #20 через 36 ms.  
 Торговец 2 обслужил клиента #20  
 Торговец 2 обслужет клиента #10 через 164 ms.  
 Торговец 1 обслужил клиента #9  
 Торговец 1 обслужет клиента #30 через 95 ms.  
 Торговец 3 обслужил клиента #29  
 Торговец 1 обслужил клиента #30  
 Торговец 2 обслужил клиента #10