

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа бакалавриата «Программная инженерия»

Отчет к Микропроекту №2

**«Многопоточное приложение с использованием стандартной
библиотеки C++»**

По дисциплине
«Архитектура вычислительных систем»

Работу выполнил:

Студент группы БПИ-193 Тимканов И.А.

Вариант 21

Москва 2020

1. Условие

В магазине работают два отдела, каждый отдел обладает уникальным ассортиментом. В каждом отделе работает один продавец. В магазин ходят исключительно забывчивые покупатели, поэтому каждый покупатель носит с собой список товаров, которые желает купить. Покупатель приобретает товары точно в том порядке, в каком они записаны в его списке. Продавец может обслужить только одного покупателя за раз. Покупатель, вставший в очередь, засыпает пока не дойдет до продавца. Продавец засыпает, если в его отделе нет покупателей, и просыпается, если появится хотя бы один. Создать многопоточное приложение, моделирующее работу магазина.

2. Реализация

Для реализации программы использовалась стандартная библиотека C++ для работы с потоками. Ниже приведен разбор условия

1. *«В магазине работают два отдела, каждый отдел обладает уникальным ассортиментом».* В исходном коде программы был реализован метод **ServeClient**, который выполняет роль магазина с двумя отделами. В классе **Client** в списках **food_list** и **toys_list** реализованы товары в отделах.
2. *«В каждом отделе работает один продавец».* Для каждого клиента реализован отдельный поток, однако с методом **ServeClient** одного отдела одновременно может взаимодействовать только один клиент, что достигается путем применения мьютексов.
3. *«В магазин ходят исключительно забывчивые покупатели, поэтому каждый покупатель носит с собой список товаров, которые желает купить».* Для каждого покупателя, которых ровно n штук (задаётся через консольный ввод) случайным образом генерируется его список покупок из K товаров, где $K \in [1, 9]$.
4. *«Покупатель приобретает товары точно в том порядке, в каком они записаны в его списке».* Покупатель в отдельном потоке проходится по всем элементам коллекции списка товаров и делает запросы в нужный отдел и ожидает ответа от продавца нужного отдела, пока тот не ответит, покупатель не пойдёт за другим товаром.
5. *Продавец может обслужить только одного покупателя за раз».* Использован шаблон «Producer – Consumers», реализация которой приходится на очередь. А для того, чтобы обеспечить безопасное получение данных из очереди – использовались мьютексы.

Тестовое покрытие

1. Ввод невалидных данных

```
Введите число покупателей магазина: 233121
Из-за коронавируса в магазин может обслуживать в день от 1 до 100 покупателей.
Повторите ввод: sdsd
Из-за коронавируса в магазин может обслуживать в день от 1 до 100 покупателей.
Повторите ввод: -12312313
Из-за коронавируса в магазин может обслуживать в день от 1 до 100 покупателей.
Повторите ввод: asdsfsa
Из-за коронавируса в магазин может обслуживать в день от 1 до 100 покупателей.
Повторите ввод: 342
Из-за коронавируса в магазин может обслуживать в день от 1 до 100 покупателей.
Повторите ввод:
```

2. Ввод числа покупателей $n = 1$

```
Введите число покупателей магазина: 1
Покупатель #1 зашёл в магазин.
Продавец продуктов начал обслуживать клиента #1
Клиент #1 пошел в отдел игрушек
Продавец игрушек начал обслуживать клиента #1
Клиент #1 купил у продавца игрушек: 'кукла'
Клиент #1 пошел в отдел продуктов
Продавец продуктов начал обслуживать клиента #1
Клиент #1 купил у продавца продуктов: 'помидоры'
Клиент #1 закупился и покинул магазин
```

3. Ввод числа $n = 1$ (Уже было, но в потоках всегда работает по-разному)

```
Введите число покупателей магазина: 1
Покупатель #1 зашёл в магазин.
Продавец продуктов начал обслуживать клиента #1
Клиент #1 пошел в отдел игрушек
Продавец игрушек начал обслуживать клиента #1
Клиент #1 купил у продавца игрушек: 'пластилин' 'кукла' 'шахматы'
Клиент #1 пошел в отдел продуктов
Продавец продуктов начал обслуживать клиента #1
Клиент #1 купил у продавца продуктов: 'печенье'
Клиент #1 закупился и покинул магазин
```

4. Ввод числа $n = 2$

```
Введите число покупателей магазина: 2
Покупатель #1 зашёл в магазин.
Продавец продуктов начал обслуживать клиента #1
Клиент #1 купил у продавца продуктов: 'сахар' 'огурцы' 'бананы' 'чай'
Клиент #1 пошел в отдел игрушек
Покупатель #2 зашёл в магазин.
Продавец продуктов начал обслуживать клиента #2
Клиент #2 купил у продавца продуктов: 'хлеб'
Клиент #2 пошел в отдел игрушек
Продавец игрушек начал обслуживать клиента #1
Клиент #1 купил у продавца игрушек: 'плюшевый медведь' 'пластилин' 'динозавр'
Клиент #1 закупился и покинул магазин
Продавец игрушек начал обслуживать клиента #2
Клиент #2 купил у продавца игрушек: 'шахматы'
Клиент #2 пошел в отдел продуктов
Продавец продуктов начал обслуживать клиента #2
Клиент #2 купил у продавца продуктов: 'печенье'
Клиент #2 пошел в отдел игрушек
Продавец игрушек начал обслуживать клиента #2
Клиент #2 купил у продавца игрушек: 'лего-набор'
Клиент #2 пошел в отдел продуктов
Продавец продуктов начал обслуживать клиента #2
Клиент #2 купил у продавца продуктов: 'молоко'
Клиент #2 пошел в отдел игрушек
Продавец игрушек начал обслуживать клиента #2
Клиент #2 купил у продавца игрушек: 'кукла'
Клиент #2 закупился и покинул магазин
```

5. Ввод числа n = 3

```
Введите число покупателей магазина: 3
Покупатель #1 зашёл в магазин.
Продавец продуктов начал обслуживать клиента #1
Клиент #1 купил у продавца продуктов: 'помидоры'
Клиент #1 пошел в отдел игрушек
Покупатель #2 зашёл в магазин.
Продавец продуктов начал обслуживать клиента #2
Клиент #2 пошел в отдел игрушек
Продавец игрушек начал обслуживать клиента #1
Клиент #1 купил у продавца игрушек: 'футбольный мяч' 'кукла' 'пластилин' 'плюшевый медведь'
Клиент #1 закупился и покинул магазин
Покупатель #3 зашёл в магазин.
Продавец продуктов начал обслуживать клиента #3
Клиент #3 купил у продавца продуктов: 'картошка'
Клиент #3 закупился и покинул магазин
Продавец игрушек начал обслуживать клиента #2
Клиент #2 купил у продавца игрушек: 'плюшевый медведь'
Клиент #2 пошел в отдел продуктов
Продавец продуктов начал обслуживать клиента #2
Клиент #2 купил у продавца продуктов: 'огурцы'
Клиент #2 закупился и покинул магазин
```

Список используемых источников

1. Cppreference (2020) «Документация по C++: std::condition_variable::wait» (https://en.cppreference.com/w/cpp/thread/condition_variable/wait). Просмотрено 03.12.2020
2. Cppreference (2020) «Документация по C++: std::mutex» (<https://ru.cppreference.com/w/cpp/thread/mutex>). Просмотрено 03.12.2020
3. Docs Microsoft (2020) «Creating Threads» (<https://docs.microsoft.com/en-us/windows/win32/procthread/creating-threads>). Просмотрено 03.12.2020
4. Habr (2020) «Потоки, блокировки и условные переменные в C++11» (<https://habr.com/ru/post/182626/>). Просмотрено 03.12.2020
5. Intel (2011) «Шаблон проектирования Producer-Consumer» (<https://software.intel.com/content/www/ru/ru/develop/articles/producer-consumer.html>). Просмотрено 03.12.2020
6. Wikipedia (2014) «Producer–consumer problem» (https://en.wikipedia.org/wiki/Producer%E2%80%93consumer_problem). Просмотрено 03.12.2020
7. Легалов А.И.(2020) «Архитектура параллельных вычислительных систем. Многопоточность » (<http://softcraft.ru/edu/comparch/lect/07-parthread/>) Просмотрено 03.12.2020
8. Легалов А.И.(2020) «Многопоточность. Простая многопоточная программа. Основные функции» (<http://softcraft.ru/edu/comparch/practice/thread/01-simple/>). Просмотрено 03.12.2020
9. Легалов А.И.(2020) «Многопоточность. Синхронизация потоков. Методы синхронизации» (<http://softcraft.ru/edu/comparch/practice/thread/02-sync/>). Просмотрено 03.12.2020