**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 С++»**

По дисциплине

«Архитектура вычислительных систем»

Работу выполнил:

Студент группы БПИ-193 Тимканов И.А.

Вариант 21

Москва 2020

# 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.

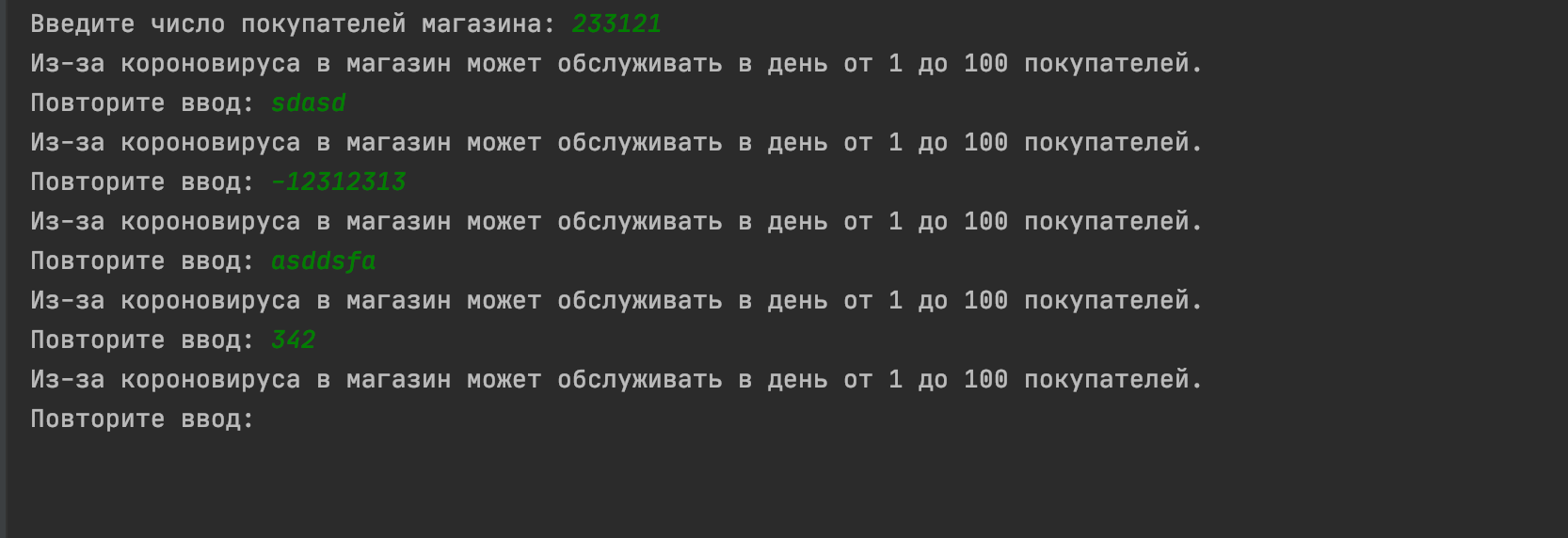
1. **Реализация**

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

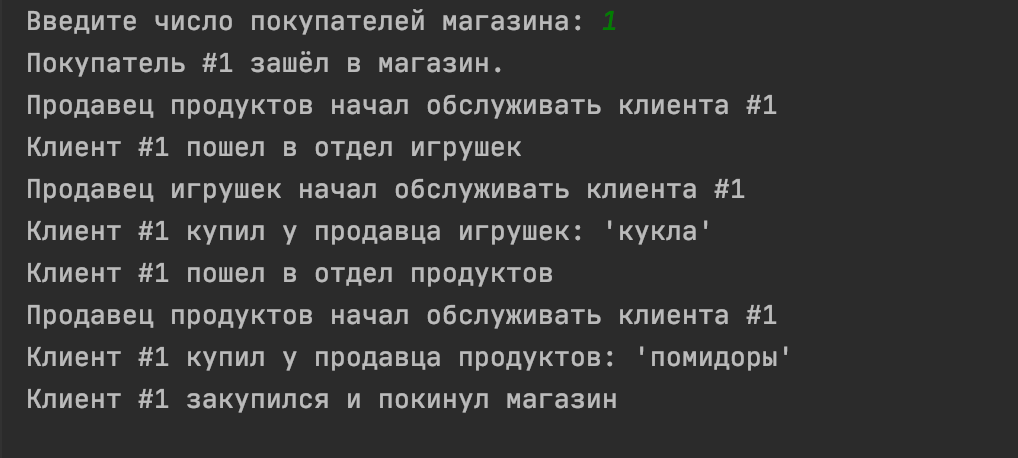
1. *«В магазине работают два отдела, каждый отдел обладает уникальным ассортиментом».* В исходном коде программы был реализовал метод **ServeClient**, который выполняет роль магазина с двумя отделами. В классе **Client** в списках **food\_list** и **toys\_list** реализованы товары в отделах.
2. *«В каждом отделе работает один продавец».* Для каждого клиента реализован отдельный поток, однако с методом **ServeClient** одного отдела единовременно 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 ∈ **[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 сшаблон «Producer – Consumers», реализация которой приходится на очередь. А для того, чтобы обеспечить безопастное получение данных из очереди – использовались мьютексы.

**Тестовое покрытие**

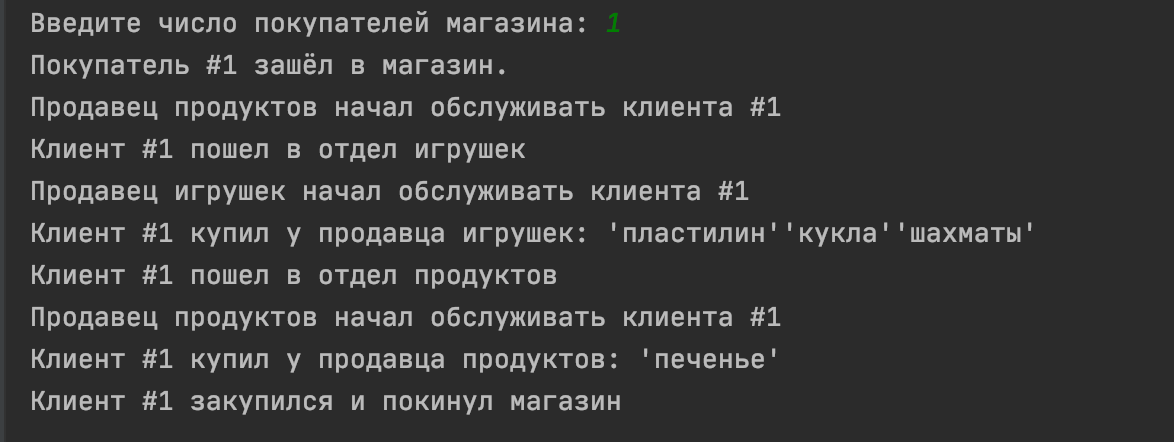
1. Ввод невалидных данных



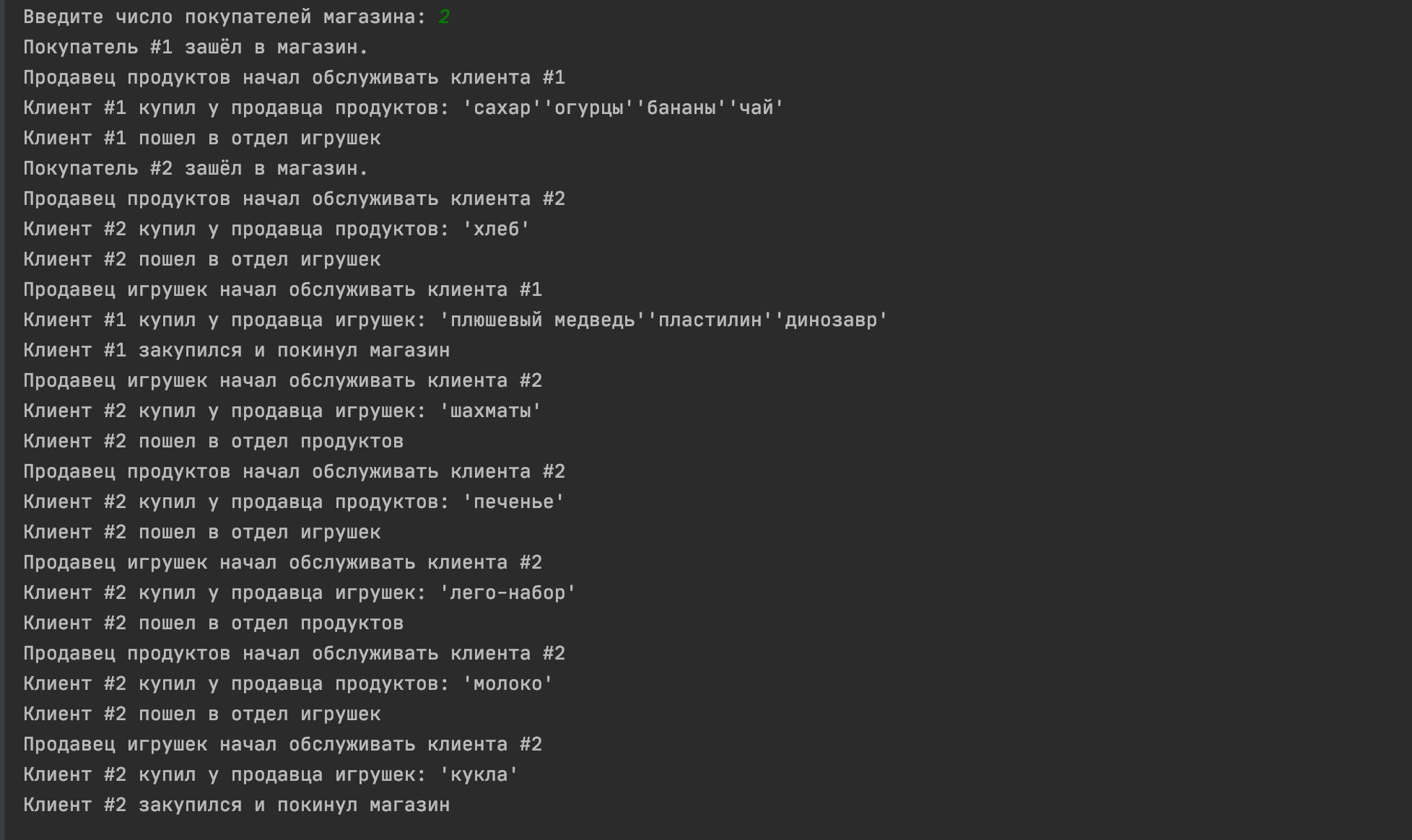
1. Ввод числа покупателей n = 1



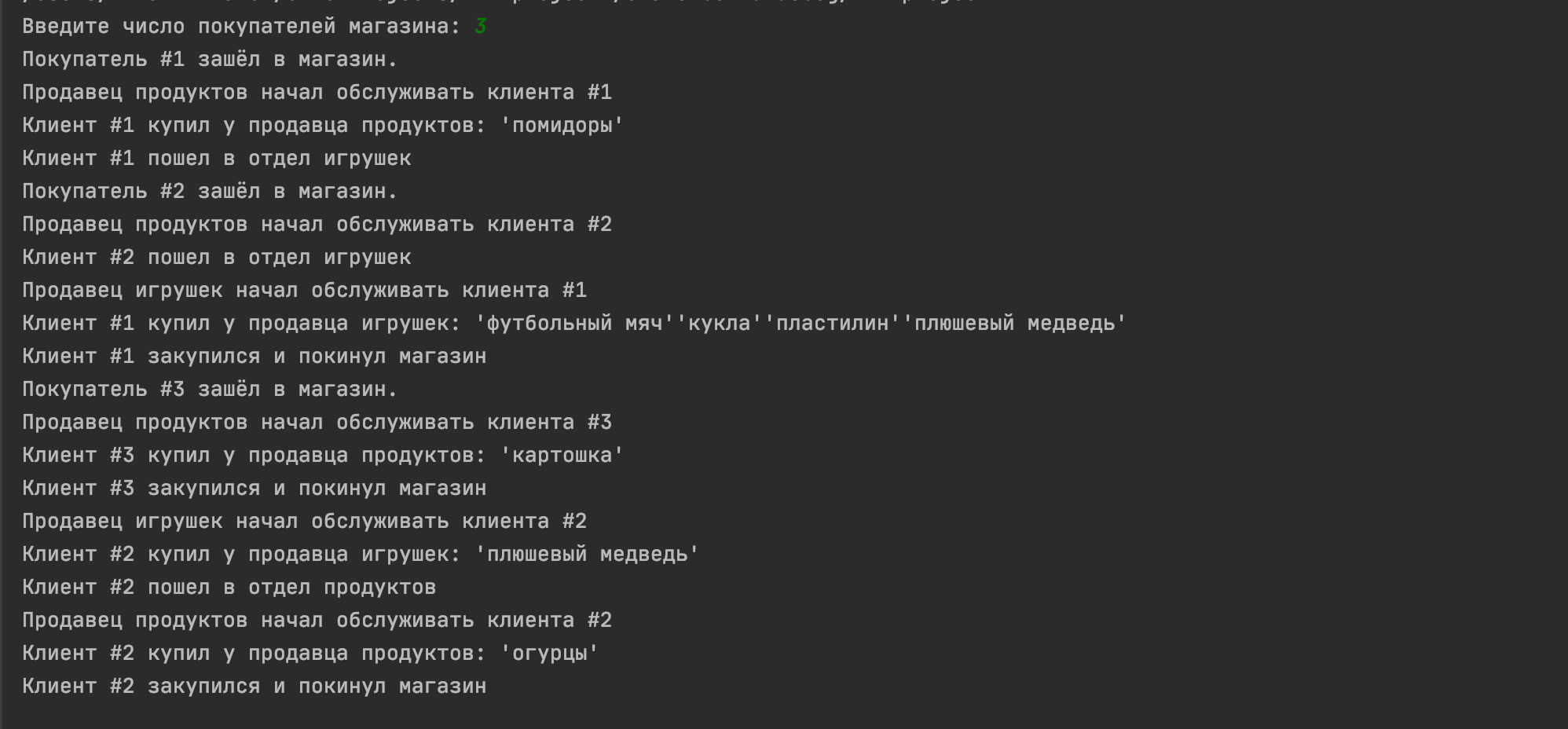
1. Ввод числа n = 1 (Уже было, но в потоках всегда работает по-разному)



1. Ввод числа n = 2



1. Ввод числа n = 3



**Список используемых источников**

1. Cppreference (2020) «Документация по С++: std::condition\_variable::wait» (<https://en.cppreference.com/w/cpp/thread/condition_variable/wait>). Просмотрено 03.12.2020
2. Cppreference (2020) «Документация по С++: 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