

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

Факультет компьютерных наук

Департамент программной инженерии

**Микропроект №2**

**Тема:**

**Задача о магазине**

**Вариант №11**

**Пояснительная записка**

Исполнитель  
Студент группы БПИ 199  
\_\_\_\_\_/Кочик Д.А./  
«\_\_»\_\_\_\_\_2020 г.

Москва

2020

## Содержание

Содержание .....	2
1. Постановка задачи .....	3
2. Применяемые расчетные методы.....	4
3. Входные и выходные данные .....	5
3.1. Входные данные .....	5
3.2. Выходные данные.....	5
4. Использованные источники .....	6

## **1. Постановка задачи**

В магазине работают три отдела, каждый отдел обслуживает один продавец. Покупатель, зайдя в магазин, делает покупки в произвольных отделах, и если в выбранном отделе продавец не свободен, покупатель становится в очередь и засыпает, пока продавец не освободится. Создать многопоточное приложение, моделирующее рабочий день магазина.

Приложение должно быть выполнено на языке C++.

## 2. Применяемые расчетные методы

Для решения поставленной задачи используется стандартная библиотека `thread`[1]. При запуске программы происходит инициализация параметров (см. п.3), после чего происходит наполнение магазина объектами. Магазин представляет из себя три продавца, каждый из которых описывают свой отдел. При выводе в консоль каждый продавец именуется как “Reseller #{N+1}”, где N – порядковый номер продавца отдела (нумерация начинается с 1). Все продавцы являются владельцами собственного бизнеса – скупкой и перепродажей лимитированных кроссовок. Поэтому объекты продажи являются кроссовки – пара значений типа `string` и `double`, где первый элемент пары – название модели кроссовок[2], а второй – размер модели в британской размерной сетке (UK). При создании каждого продавца создается с уникальным склад товара, со случайным числом объектов, после чего выводится сообщение с номером продавца и информации о количестве товаров.

После создания трех продавцов происходит создание покупателей, которые описываются потоками, выполняющими метод покупки товара. В этом методе каждый пользователь случайным образом выбирает продавца. После чего, если у продавца еще остался товар, происходит блокировка потоков, планирующих работать именно с этим продавцом. Таким образом текущий поток реализует покупку (вызывая дополнительно метод продавца, удаляющий товар с его склада), в то время как другие потоки, описывающие покупателей, выбравших этого же продавца, блокируются и ожидают в очереди. Стоит отметить, что при этом не происходит блокировки потоков, не взаимодействующих с конкретным продавцом. Для блокировки потоков используются экземпляры классов `mutex`[3]. Во время начала сделки и по ее завершении выводится соответствующая информация, сообщающая о покупателе, продавце и объекте сделки (модели обуви конкретного размера). При выводе информации используется отдельный экземпляр класса `mutex`, блокирующий все потоки. За счет этого обеспечивается синхронизированный вывод информации. После вывода информации покупатель с 50% вероятностью может совершить на одну покупку больше. При этом, если покупатель попадает к продавцу, у которого закончился товар, он расстраивается и выводит соответствующее сообщение (повторное посещение других продавцов при этом не гарантируется; оно также случается с 50% вероятностью). При выводе информации покупатель обозначается как “Consumer #{N+1}”, где N – номер потока покупателя.

Как только все покупатели закончили совершать покупки, рабочий день завершается и выводится соответствующее сообщение. После этого каждый из продавцов делает вывод по прошедшему дню: радуется своему успеху, если он смог продать весь товар или подсчитывает, сколько пар кроссовок у него осталось.

Для генерации случайных чисел используется стандартная библиотека `random`[4] так как использование встроенной функции `random()` не дает подходящих результатов при генерации случайных чисел в многопоточной среде[5].

### **3. Входные и выходные данные**

#### **3.1. Входные данные**

При запуске программы из консоли возможны следующие сценарии ввода аргументов:

- 1) Три целочисленных аргумента. При этом первый аргумент описывает число покупателей, второй – минимальное число пар обуви у продавца, третий – максимальное число пар обуви у продавца. Второй и третий аргументы задают границы генерации случайных чисел для заполнения склада продавца.
- 2) Два целочисленных аргумента. При этом первый аргумент описывает минимальное число пар обуви у продавца, второй – максимальное число пар обуви у продавца. Оба аргумента задают границы генерации случайных чисел для заполнения склада продавца. Число покупателей принимает значение по умолчанию.
- 3) Один целочисленный аргумент. Аргумент описывает число покупателей. Границы генерации пар обуви продавцов принимают значения по умолчанию.
- 4) Запуск без параметров. Всем аргументам задаются значения по умолчанию.

Ввод параметров осуществляется через пробел.

Значения по умолчанию для параметров запуска программы:

Число покупателей – 4 человек.

Минимальное количество единиц товара у продавца – 10 единиц.

Максимальное количество единиц товара у продавца – 30 единиц.

При вводе некорректного числа параметров (больше 3) и/или если один или несколько введенных параметров невозможно представить в виде целых чисел и/или минимальное число единиц товара у покупателя больше или равно максимальному числу единиц товара у покупателя выводится сообщение об ошибке и подсказка по вводимым параметрам.

#### **3.2. Выходные данные**

Вся информация о работе программы выводится в консоль. Результатом работы программы является история взаимодействия пользователей и покупателей (сообщения, содержащие информацию, кто у кого, что приобрел). Все сообщения разделяются пустой строкой для удобочитаемости.

#### **4. Используемые источники**

- 1) Официальная документация к библиотеке thread [Электронный ресурс]//URL: <https://en.cppreference.com/w/cpp/thread> (дата обращения 03.12.2020).
- 2) Торговая площадка лимитированных вещей[Электронный ресурс]//URL: <https://stockx.com/> (дата обращения 01.12.2020).
- 3) Официальная документация к классу mutex [Электронный ресурс]//URL: <https://en.cppreference.com/w/cpp/thread/mutex> (дата обращения 02.12.2020).
- 4) Официальная документация к библиотеке random [Электронный ресурс]//URL: <https://www.cplusplus.com/reference/random/> (дата обращения 03.12.2020)
- 5) Форум, содержащий ответ на вопрос о том, почему функция random() дает одинаковые значения в многопоточной среде [Электронный ресурс]//URL: <https://www.manongdao.com/q-32735.html> (дата обращения 03.12.2020)