

Вариант 9. *Задача о магазине - 1.* 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. Сразу считаю должным заметить, что условие понято мной в том смысле, что покупатель приходит в магазин и по очереди посещает интересующие его отделы, стоя там в отдельных очередях. При этом в отличие от супермаркета по торговому залу не ходит, а магазин представляет собой некоторое количество прилавков, куда посетители сразу становятся в очередь.

1. Структура модели

Сложно однозначно приписать модель к какому-то конкретному виду, однако наиболее выраженное сходство я нахожу с клиентами и серверами: покупатель приходит в магазин, становится в очередь, извещает о себе продавцу. Продавец уведомляет покупателя, когда подходит его очередь, покупатель тратит какое-то время у прилавка (в зависимости от количества покупок), после чего благодарит продавца, уведомляя его о том, что тот может принять следующего покупателя.

После этого покупателю может понадобиться посетить и другие отделы.

При этом вся система работает не так как в жизни: к примеру, оказалось удобным, чтобы о том, что рабочий день для продавцов закончен и покупателей в торговом зале больше нет, уведомлял как раз последний покупатель.

2. Более подробное описание структуры.

Прежде всего есть основной поток, который работает с пользователем, считывая его команды, и распределяя работу между потоками, ответственными за покупателей.

Есть класс магазина, в котором постоянно в цикле трудятся три потока-продавца (3 по условию задачи, в реализации предусмотрена масштабируемость).

Основной поток поручает потоку-покупателю “сходить в магазин”, то есть выполнить некоторые методы класса магазина (прийти, желательно до закрытия, встать в очередь, оплатить покупки, возможно, встать в и в другие очереди, после чего уйти)

В начале пользователь поручает основному потоку открыть магазин (к этому моменту все продавцы уже на месте ждут покупателей), потом добавляет покупателей и, наконец, закрывает магазин. Когда магазин закрыт, это означает лишь то, что покупатели не могут в него зайти, однако те, что на

данный момент находятся в магазине, могут завершить свои покупки. Таким образом по настоящему магазин закрывается только тогда, когда уходит последний покупатель, а продавцы заканчивают свою работу.

3. Детали реализации.

Программа разработана на языке C++ (20-й стандарт). Используются библиотеки `std::thread`, `std::mutex` и `std::condition_variable` (с помощью них реализована вся синхронизация).

4. Формат ввода:

Всего есть три пользовательские команды - `open`, `add` и `close`. `open` открывает магазин, `close` закрывает (также это маркер конца диалога с пользователем). Команда `add` добавляет покупателя, с ней также необходимо указать параметры: `n` - количество отделов, которые желает посетить покупатель, далее `n` чисел - номера этих отделов (с нуля), а также ещё `n` чисел - количество покупок в каждом отделе (оно влияет на количество времени, который тратит каждый покупатель у прилавка).

5. Формат вывода:

Логи действий покупателей и администрации магазина.

Ввод и вывод реализован через консоль для интерактива, для больших тестовых файлов (как `input.txt`) лучше использовать перенаправление ввода/вывода в файлы.