**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное** **учреждение высшего образования «Национальный исследовательский университет «Высшая школа экономики»**

Факультет компьютерных наук

Департамент программной инженерии

# Отчет к микропроекту 2 заданию по дисциплине

**«Архитектура вычислительных систем»**

Работу выполнил:

Студент группы БПИ-194 Остапенко С.В.

**Москва 2020**

**Задание**

15. Задача о гостинице - 3 (дамы и джентльмены). В гостинице 10 номеров рассчитаны на одного человека и 15 номеров рассчитаны на двух человек. В гостиницу приходят клиенты дамы и клиенты джентльмены, и конечно они могут провести ночь в номере только с представителем своего пола. Если для клиента не находится подходящего номера, он уходит искать ночлег в другое место. Создать многопоточное приложение, моделирующее работу гостиницы.

**Решение**

Для решения данной задачи используются следующие 3 класса:

1. **Customer**

Данный класс хранит информацию о посетителе отеля, а именно:

* Имя
* Пол
* Количество дней, на которое человек хочет заселиться в отель
* Bool значение has\_left, показывающее, находится ли человек в отеле или уже покинул его

Имя, пол и дни до отъезда заполняются случайными значениями в конструкторе класса. Изначально у всех объектов класса Customer значение, указывающее на присутствие человека в отеле, установлено как false. Оно меняется в методах класса Hotel и DoubleRoom. В классе так же содержится метод Print, выводящий информацию в формате: «Пол: Имя».

1. **DoubleRoom**

Данный класс представляет из себя комнату для двоих посетителей отеля. Класс содержит два объекта Customer и три публичных метода:

* Void Settle (Customer newCustomer, int days). Данный метод заселяет посетителя в двойную комнату. Значение has\_left у newCustomer становится true. Выводится сообщение о дне, в который посетитель заселился, живет ли он один (второе место в комнате пока пустует) или его/ее сожителя, а также количество дней, планируемых провести в отеле.
* Bool CanSettle (Customer newCustomer). Данный метод возвращает bool значение, показывающее, может ли посетитель newCustomer заселиться в данной комнате.
* Void MoveOut(Customer customer). Данный метод выселяет customer из комнаты для двоих. Значение has\_left становится true.

1. **Hotel**

Данный класс представляет из себя отель. В нем содержится vector<Customer> RoomsForOne – 10 комнат для одного посетителя и vector<DoubleRoom> RoomsForTwo – 15 комнат для двух посетителей. В данном классе также находятся два метода GetFreeRoomIndex и GetFreeDoubleRoomIndex, возвращающий индекс свободной комнаты для одного или для двух человек. В случае, если свободных комнат нет, возвращается -1. Присутствуют два метода для отладки, выводящие информацию о постояльцах в комнатах: PrintOneRoom и PrintTwoRoom. 1 – постоялец есть, 0 – постояльца нет.

Основной метод класс: void Settle(Customer newcomer). В самом начале метода вызывается задержка на случайное число секунд. Это делается для того чтоб все посетители не попробовали заселиться в отель одновременно, так как данный метод будет вызван не один раз в нескольких потоках одновременно. После этого у объекта Mutex mutex вызывается метод lock() и происходит последовательный поиск свободных комнат в отеле. Сначала в одиночных, потом в двойных. При наличии свободной комнаты метод выводит соответствующее сообщение в консоль, вызывает mutex.unlock() и происходит выход из метода. При отсутствии свободных комнат выводится соответствующее сообщение в консоль и происходит выход из метода.

В main() вызывается запрос на ввод числа посетителей отеля в промежутке (1, 750). После этого создается данное количество экземпляров класса Customer и для каждого из них в отдельном потоке вызывается метод hotel.Settle(Customer newcomer). После у каждого из потоков вызывается метод join(), после чего в консоль выводится сообщение о том, что все посетители уехали из отеля.

**Тестирование**

При вводе некорректного числа посетителей будет выведено соответствующее сообщение. (рисунок 1)

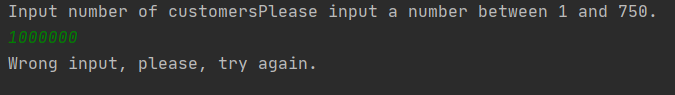


Рисунок 1

Заселение первых постояльцев (рисунок 2).

Выселение, заселение новых людей и сообщение о полной заполненности отеля. (рисунок 3)

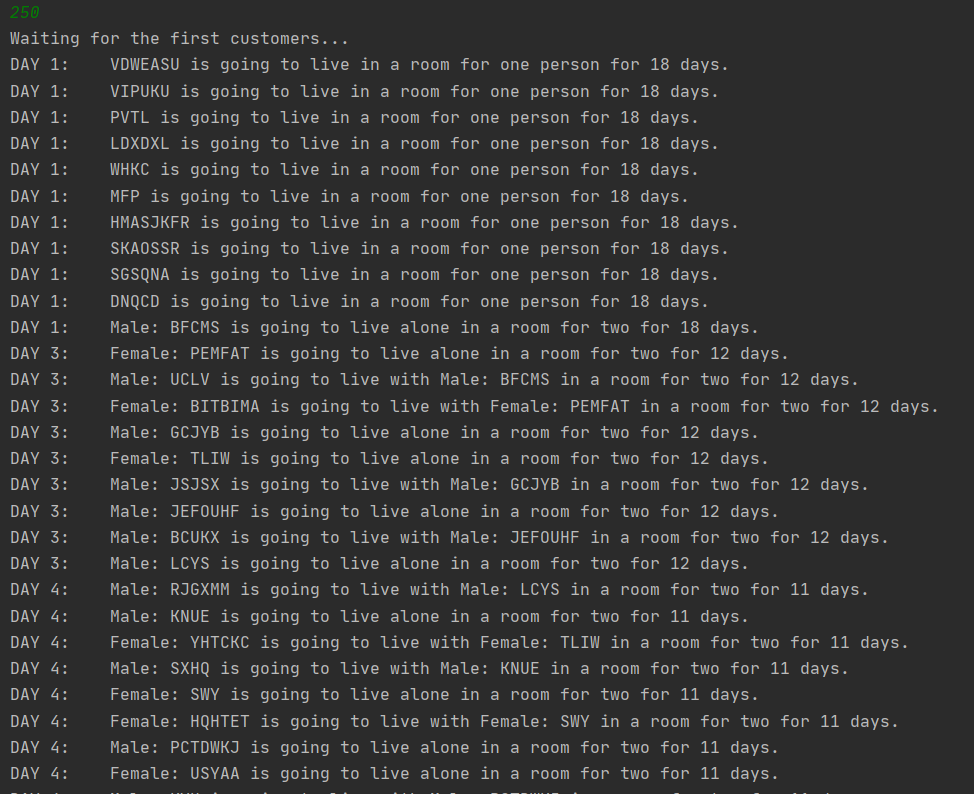


Рисунок 2

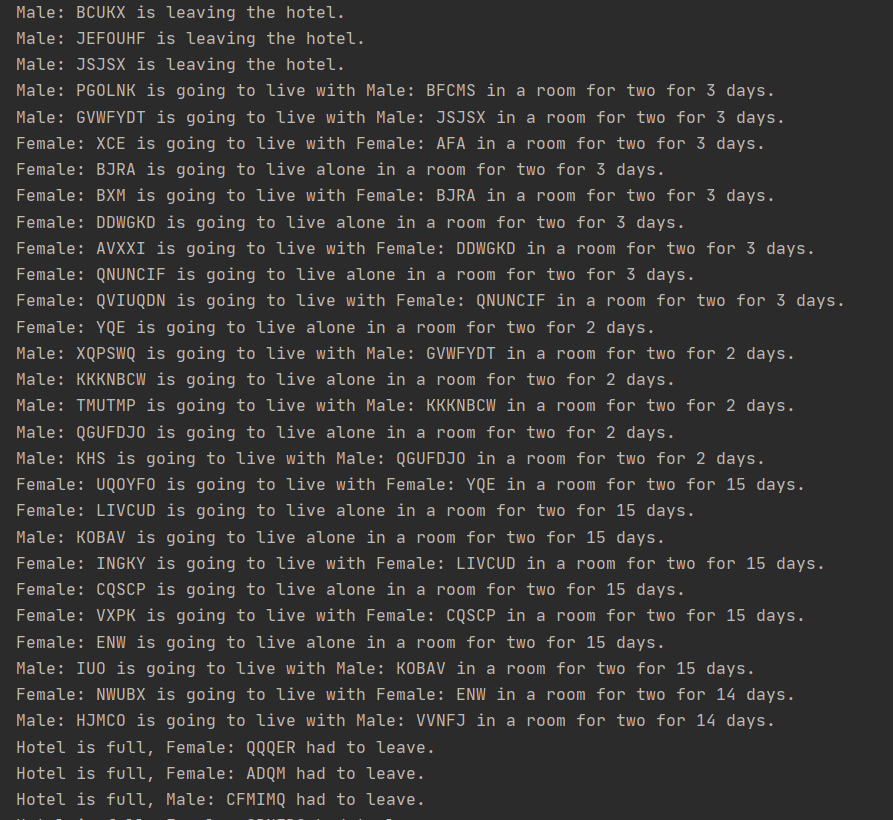


Рисунок 3

**Список используемой литературы**

1. Cppreference (2020) «Документация по С++: std::condition\_variable::wait» (<https://en.cppreference.com/w/cpp/thread/condition_variable/wait>).
2. Cppreference (2020) «Документация по С++: std::mutex» (<https://ru.cppreference.com/w/cpp/thread/mutex>).
3. Docs Microsoft (2020) «Creating Threads» (<https://docs.microsoft.com/en-us/windows/win32/procthread/creating-threads>)
4. Легалов А.И.(2020) «Архитектура параллельных вычислительных систем. Многопоточность » (<http://softcraft.ru/edu/comparch/lect/07-parthread/>)
5. Легалов А.И.(2020) «Многопоточность. Простая многопоточная программа. Основные функции» (<http://softcraft.ru/edu/comparch/practice/thread/01-simple/>).
6. Легалов А.И.(2020) «Многопоточность. Синхронизация потоков. Методы синхронизации» (<http://softcraft.ru/edu/comparch/practice/thread/02-sync/>).
7. Хабр «Такие удивительные семафоры» (<https://habr.com/ru/post/261273/>)
8. Learn.info «Семафоры: введение» (<https://learnc.info/c/pthreads_semaphores.html>)