**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

**Проект по дисциплине «Архитектура вычислительных систем»**

**Многопоточная программа на языке C++, моделирующая работу гостиницы**

Исполнитель

студент группы БПИ196-1

Махнач Ф. О.

13.12.2020 г.

**Оглавление**

[1. Основные требования 2](#_Toc58865501)

[2. Текст задания 2](#_Toc58865502)

[3. Ремарка об условии задачи 3](#_Toc58865503)

[4. Алгоритм решения задачи 3](#_Toc58865504)

[4.1. Ввод данных 3](#_Toc58865505)

[4.2. Исполнение в потоке 3](#_Toc58865506)

[4.3. Гостиница 3](#_Toc58865507)

[4.4. Приезд 4](#_Toc58865508)

[4.5. Отъезд 5](#_Toc58865509)

[5. Тестирование программы 5](#_Toc58865510)

[6. Используемые источники 7](#_Toc58865511)

# Основные требования

- Оформление титульного листа: ВУЗ, департамент, название разработки, данные о студента (ФИО, номер группы/подгруппы);

- Сообщения в консоли должны отражать состояние программы и соответствовать предметной области;

- Список источников должен быть осмысленным;

- Использование библиотеки Posix thread или стандартную библиотеку потоков (thread) языка C++;

- Предусмотреть ограничение по числу итераций / времени выполнения, т. к. стандартное завершение отсутствует.

# Текст задания

(Вариант 15)  
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.

# Ремарка об условии задачи

В условии проекта указана необходимость использования семафоров и (или) условных переменных, однако при данной формулировке задания мне не удалось найти им подходящее применение. В частности, предложение «Если для клиента не находится подходящего номера, он уходит искать ночлег в другое место» не оставляет места для семафора. Если бы клиент не уходил, а продолжал ждать (хотя бы некоторое время) того, что номер освободится, семафор мог бы найти себе применение.

Помимо этого, в задаче не сказано, что джентльмены и дамы когда-либо покидают гостиницу, что делает программу достаточно прямолинейной: сначала гостиница заполняется (что происходит достаточно быстро), после чего все приходящие дамы и джентльмены не находят для себя подходящего номера.

Эти моменты достаточно сильно смутили меня, так как я не нашел указания, что можно менять условие. Я позволил себе добавить логику покидания джентльменами и дамами отеля, чтобы вывод был немного интереснее (и не противоречит условию).

# Алгоритм решения задачи

## Ввод данных

При запуске программы пользователь может задать следующие параметры:

- количество исполняемых потоков (целое число в пределах [1; макс. кол-во потоков на компьютере]);  
- интервал между сообщениями в миллисекундах (целое положительное число);  
- количество итераций в каждом из потоков.

## Исполнение в потоке

В функции, которая выполняется каждым потоком, в цикле, ограниченном количеством итераций (см. п. 4.1.), случайно выбирается одна из операций:   
 1) попытаться заселить даму или джентльмена в гостиницу;   
 2) попытаться выселить даму или джентльмена из случайного номера.  
Первая операция имеет вероятность выполнения 0,66, вторая – 0,33 (значения выбраны эмпирически).

Если был задан интервал между сообщениями (см. п. 4.1.), то после выполнения каждой итерации поток засыпает на указанное число миллисекунд.

## Гостиница

Для имитации работы гостиницы было перепробовано достаточно много способов, но в конце концов я остановился на реализации, использующей счётчики:

1. Числа незанятых одноместных номеров;
2. Числа занятых дамами одноместных номеров;
3. Числа занятых джентльменами одноместных номеров;
4. Числа незанятых двухместных номеров;
5. Числа двухместных номеров с 1 жителем дамой;
6. Числа двухместных номеров с 2 жителями дамами;
7. Числа двухместных номеров с 1 жителем джентльменом;
8. Числа двухместных номеров с 2 жителями джентльменами;

Достаточно громоздко, однако другие способы (которые я пробовал) показались ещё более громоздкими. Из минусов: нельзя использовать номера комнат (комната #1, комната #10), что в принципе и не является требованием. Из плюсов: можно использовать малое число мьютексов.

Я использовал 4 мьютекса для доступа к перечисленным счётчикам:

1) для доступа к счётчикам 1-3;   
2) для доступа к счётчику 4;   
3) для доступа к счётчикам 5-6;   
4) для доступа к счётчикам 7-8.

Группировка основана на том, какие счётчики используются вместе.   
В коде это выглядит так:



## Приезд

По приезде дамы или джентльмена происходит последовательная проверка на существование подходящей комнаты. Последовательность проверки:

- Двухместные номера с уже имеющимся жителем того же пола;  
 - Пустые двухместные номера;  
 - Пустые одноместные номера.

Функции, проверяющие возможность въезда имеют тип возвращаемого значения bool. Помимо этого, в эти функции передаётся экземпляр stringstream, хранящий сообщение, которое будет в итоге выведено на экран. Это выглядит следующим образом:



Сами функции проверки достаточно прямолинейны:



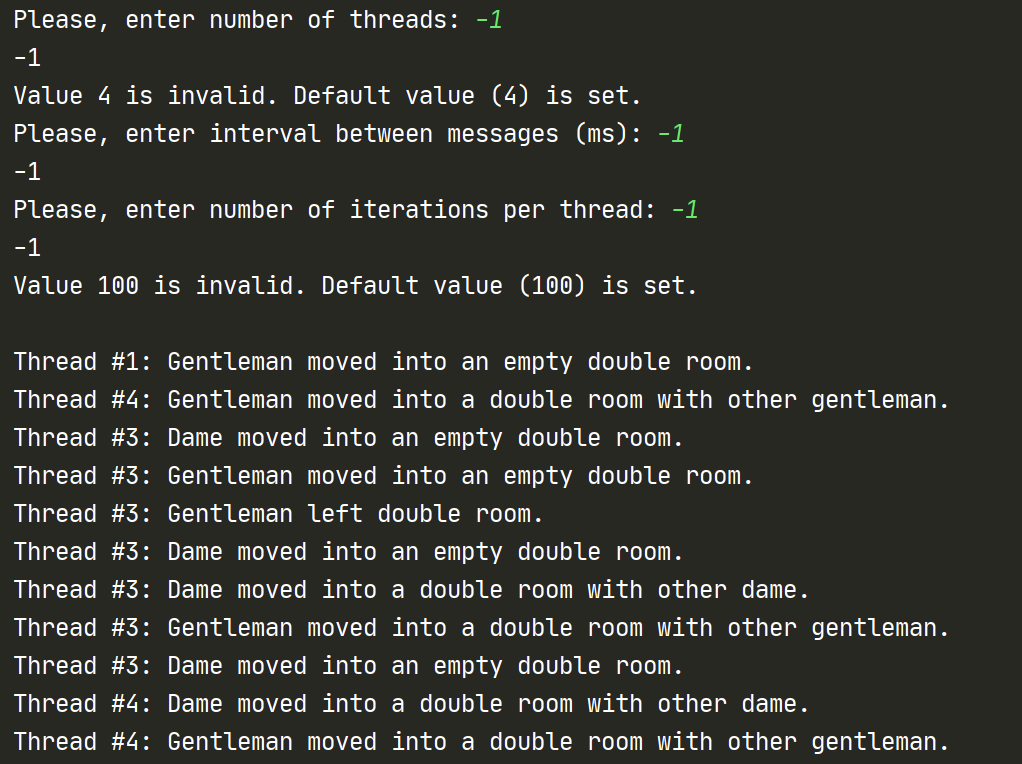
## Отъезд

Отъезд основан на случайном выборе следующих параметров:  
 - дама или джентльмен;  
 - одноместная или двухместная комната;  
 - если двухместная – комната с одним проживающим или с двумя.

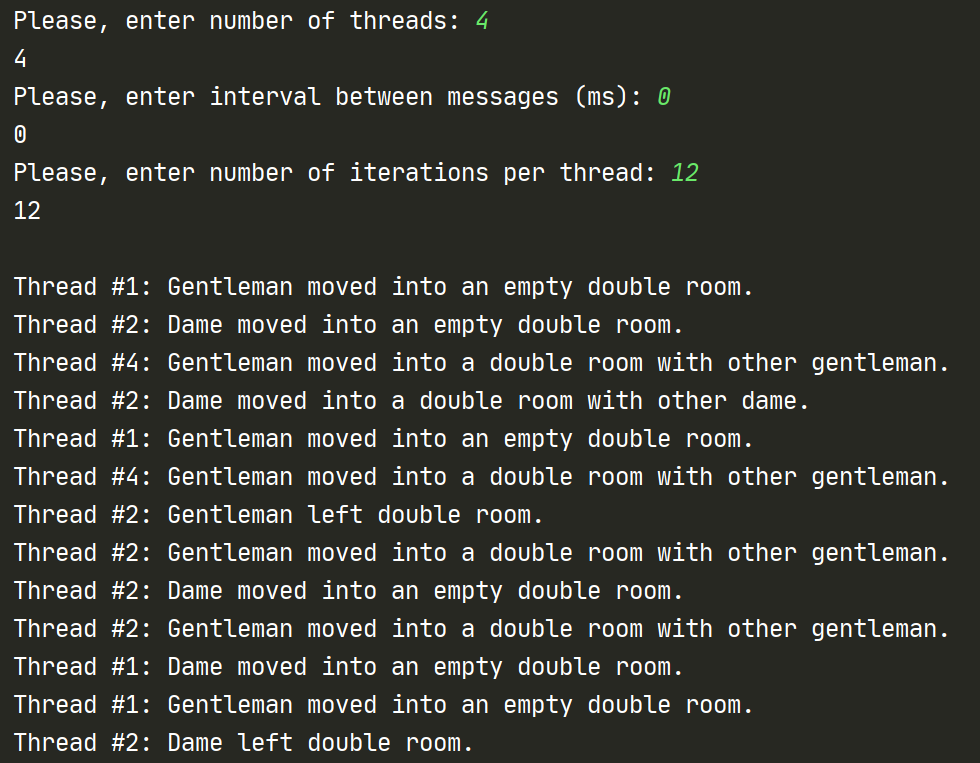
Если оказывается, что подходящая комната не находится (например, нет одноместной комнаты, в которой живёт дама), то ничего не происходит.

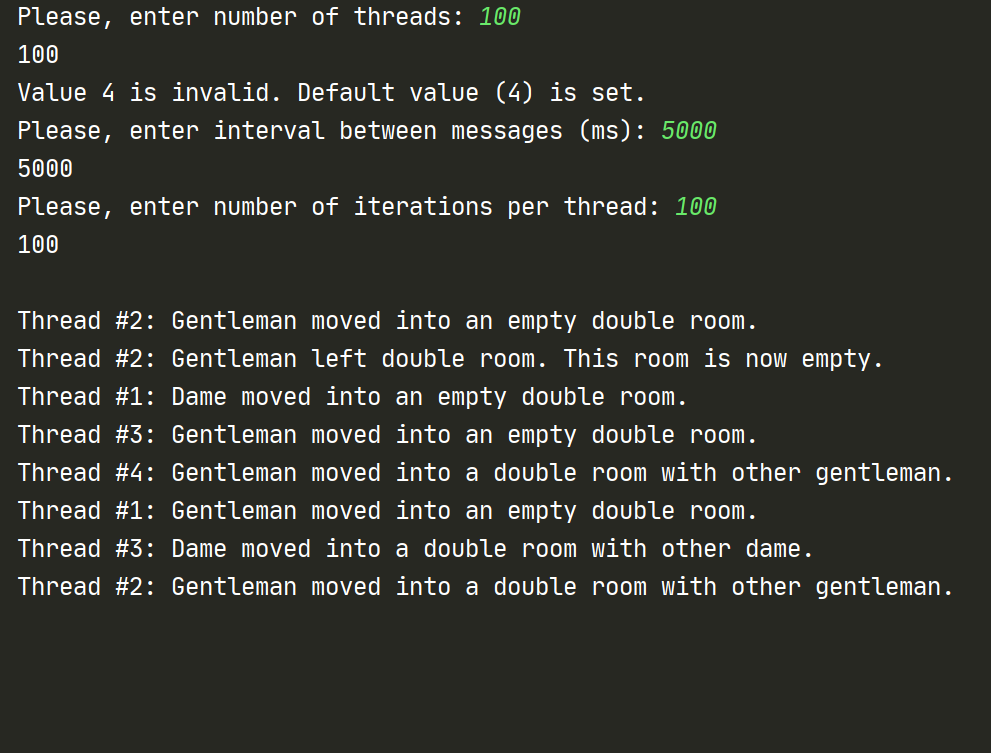
# Тестирование программы

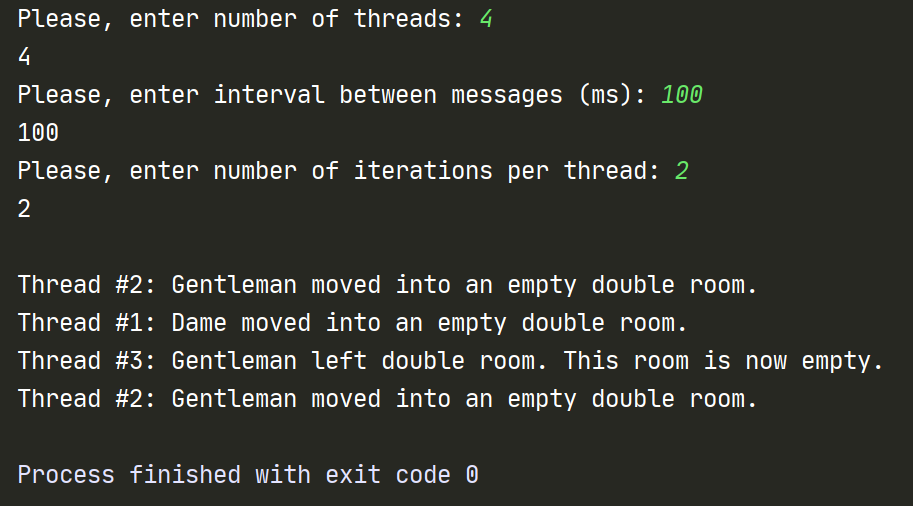
Примеры работы программы при различных входных параметрах:











# Используемые источники

1. <http://softcraft.ru/edu/comparch/practice/thread/02-sync/> – Примеры многопоточных приложений с использованием синхронизации.
2. <https://en.cppreference.com/w/cpp/thread/mutex> – Мьютекс стандартной библиотеки C++;
3. <https://en.cppreference.com/w/cpp/thread/lock_guard> – Lock guard стандартной библиотеки C++;
4. <https://ravesli.com/urok-71-generatsiya-sluchajnyh-chisel-funktsii-srand-i-rand/> – генерация случайных чисел;
5. <https://en.cppreference.com/w/cpp/thread/sleep_for> – ожидание потоком истечения заданного времени;