

Пояснительная записка к микропроекту №2

НИУ ВШЭ Департамент программной инженерии

Формулировка задания: «Задача о гостинице-2 (умные клиенты). В гостинице 10 номеров с ценой 200 рублей, 10 номеров с ценой 400 рублей и 5 номеров с ценой 600 руб. Клиент, зашедший в гостиницу, обладает некоторой суммой и получает номер по своим финансовым возможностям, если тот свободен. Если среди доступных клиенту номеров нет свободных, клиент уходит искать ночлег в другое место. Создать многопоточное приложение, моделирующее работу гостиницы.»

Ловчикова Юлия Васильевна, БПИ192, вариант 14

1. Задание

Задача о гостинице-2 (умные клиенты). В гостинице 10 номеров с ценой 200 рублей, 10 номеров с ценой 400 рублей и 5 номеров с ценой 600 руб. Клиент, зашедший в гостиницу, обладает некоторой суммой и получает номер по своим финансовым возможностям, если тот свободен. Если среди доступных клиенту номеров нет свободных, клиент уходит искать ночлег в другое место. Создать многопоточное приложение, моделирующее работу гостиницы.

2. Математическое обоснование и алгоритм

Поскольку задача довольно проста, в основе её решения лежит весьма простой алгоритм: есть три счётчика `first`, `second`, `third`, соответствующие верним границам числа занятых номеров каждого типа (10, 10, 5 соответственно). Факту «жильец поселился в номер типа x » соответствует уменьшение счётчика под номером x на единицу. Факту «жильец не может нигде поселиться» соответствует одно из двух происшествий: `budget` потенциального жильца меньше допустимой минимальной суммы бронирования (200 рублей), либо все подходящие ему по цене заняты, то есть соответствующие счётчики равны нулю.

3. Программа

3.1. Глобальные переменные

```
std::mutex _lock;
```

Вынесена с целью использования в методе

3.2. Список методов:

Метод, реализующий процесс бронирования комнаты. Алгоритм описан в пункте 2. Часть, работающая с общими данными (`first`, `second`, `third`), закрыта «локами» для аккуратной работы потоков:

```
void book_a_room(int budget, int& first, int& second, int& third) {
    _lock.lock();
    std::cout << "На данный момент в гостинице есть " << first << " номеров по цене
200 рублей, " << second << " номеров по цене 400 рублей, " << third <<
    " номеров по цене 600 рублей. \n";
    int number_of_type = 0;
    if (budget >= 600) {
        if (third > 0) {
            third--;
            number_of_type = 3;
        }
        else if (second > 0) {
            second--;
            number_of_type = 2;
        }
        else if (first > 0) {
            first--;
            number_of_type = 1;
        }
        else number_of_type = -1;
    }
    else if (budget >= 400) {
        if (second > 0) {
            second--;
            number_of_type = 2;
        }
        else if (first > 0) {
            first--;
            number_of_type = 1;
        }
        else number_of_type = -1;
    }
}
```

```

    }
    else if (budget >= 200) {
        if (first > 0) {
            first--;
            number_of_type = 1;
        }
        else number_of_type = -1;
    }
    else number_of_type = -1;
    _lock.unlock();
    if (number_of_type > 0)
        std::cout << "Клиент с бюджетом " << budget << " занял комнату типа " <<
number_of_type << "\n";
    else
        std::cout << "Клиент с бюджетом " << budget << " не смог занять никакую
комнату\n";
}

```

3.3. Смысловые части

Формирование вектора потоков:

```

std::vector<std::thread> threads;

for (int i = 0; i < 50; i++)
{
    std::cout << "Введите имеющуюся сумму клиента:\n";
    std::cin >> budget;
    threads.emplace_back(std::thread(book_a_room, budget, std::ref(first),
std::ref(second), std::ref(third)));
}

```

3.4. Формат выходных данных

Выводится одна из двух строк формата "Клиент с бюджетом " << budget << " занял комнату типа " << number_of_type << "\n" или "Клиент с бюджетом " << budget << " не смог занять никакую комнату\n"

Максимальное число потенциальных постояльцев – 50 (ограничение цикла for по итерациям).

```

Введите имеющуюся сумму клиента 1:
500
Введите имеющуюся сумму клиента 2:
На момент пребывания гостя с бюджетом 500 в гостинице есть 10 номеров по цене 200 рублей, 10 номеров по цене 400 рублей,
5 номеров по цене 600 рублей.
Клиент с бюджетом 500 занял комнату типа 2

350
Введите имеющуюся сумму клиента 3:
На момент пребывания гостя с бюджетом 350 в гостинице есть 10 номеров по цене 200 рублей, 9 номеров по цене 400 рублей,
5 номеров по цене 600 рублей.
Клиент с бюджетом 350 занял комнату типа 1

```

4. Код программы

```

#include <iostream>
#include <thread>
#include <string>
#include <vector>
#include <locale>
#include <mutex>

std::mutex _lock;
void book_a_room(int budget, int& first, int& second, int& third) {
    _lock.lock();

```

```

        std::cout << "На момент пребывания гостя с бюджетом " << budget << " в гостинице есть "
        << first << " номеров по цене 200 рублей, " << second << " номеров по цене 400 рублей, "
        << third << " номеров по цене 600 рублей. \n";
        int number_of_type = 0;
        if (budget >= 600) {
            if (third > 0) {
                third--;
                number_of_type = 3;
            }
            else if (second > 0) {
                second--;
                number_of_type = 2;
            }
            else if (first > 0) {
                first--;
                number_of_type = 1;
            }
            else number_of_type = -1;
        }
        else if (budget >= 400) {
            if (second > 0) {
                second--;
                number_of_type = 2;
            }
            else if (first > 0) {
                first--;
                number_of_type = 1;
            }
            else number_of_type = -1;
        }
        else if (budget >= 200) {
            if (first > 0) {
                first--;
                number_of_type = 1;
            }
            else number_of_type = -1;
        }
        else number_of_type = -1;

        if (number_of_type > 0)
            std::cout << "Клиент с бюджетом " << budget << " занял комнату типа " <<
            number_of_type << "\n\n";
        else
            std::cout << "Клиент с бюджетом " << budget << " не смог занять никакую
            комнату\n\n";
        _lock.unlock();
    }

    int main(int argc, char* argv[]) {
        setlocale(LC_CTYPE, "rus");

        int first = 10;
        int second = 10;
        int third = 5;
        int budget;
        std::vector<std::thread> threads;

        for (int i = 0; i < 50; i++)
        {
            std::cout << "Введите имеющуюся сумму клиента " << i + 1 << ":\n";
            std::cin >> budget;
            threads.emplace_back(std::thread(book_a_room, budget, std::ref(first),
            std::ref(second), std::ref(third)));
        }
    }

```

```
for (int i = 0; i < 50; i++)
{
    threads[i].join();
}
```

5. Испытания

5.1. Проверка заполненности одного из типов номеров:

C:\Users\yuvlo\source\repos\MicroProject2\Debug\MicroProject2.exe

Введите имеющуюся сумму клиента 2:
На момент пребывания гостя с бюджетом 600 в гостинице есть 10 номеров по цене 200 рублей, 10 номеров по цене 400 рублей, 5 номеров по цене 600 рублей.
Клиент с бюджетом 600 занял комнату типа 3

700
Введите имеющуюся сумму клиента 3:
На момент пребывания гостя с бюджетом 700 в гостинице есть 10 номеров по цене 200 рублей, 10 номеров по цене 400 рублей, 4 номеров по цене 600 рублей.
Клиент с бюджетом 700 занял комнату типа 3

800
Введите имеющуюся сумму клиента 4:
На момент пребывания гостя с бюджетом 800 в гостинице есть 10 номеров по цене 200 рублей, 10 номеров по цене 400 рублей, 3 номеров по цене 600 рублей.
Клиент с бюджетом 800 занял комнату типа 3

900
Введите имеющуюся сумму клиента 5:
На момент пребывания гостя с бюджетом 900 в гостинице есть 10 номеров по цене 200 рублей, 10 номеров по цене 400 рублей, 2 номеров по цене 600 рублей.
Клиент с бюджетом 900 занял комнату типа 3

999
Введите имеющуюся сумму клиента 6:
На момент пребывания гостя с бюджетом 999 в гостинице есть 10 номеров по цене 200 рублей, 10 номеров по цене 400 рублей, 1 номеров по цене 600 рублей.
Клиент с бюджетом 999 занял комнату типа 3

999
Введите имеющуюся сумму клиента 7:
На момент пребывания гостя с бюджетом 999 в гостинице есть 10 номеров по цене 200 рублей, 10 номеров по цене 400 рублей, 0 номеров по цене 600 рублей.
Клиент с бюджетом 999 занял комнату типа 2


5.2. Проверка заполненности всех номеров:

200
Введите имеющуюся сумму клиента 26:
На момент пребывания гостя с бюджетом 200 в гостинице есть 1 номеров по цене 200 рублей, 0 номеров по цене 400 рублей, 0 номеров по цене 600 рублей.
Клиент с бюджетом 200 занял комнату типа 1

200
Введите имеющуюся сумму клиента 27:
На момент пребывания гостя с бюджетом 200 в гостинице есть 0 номеров по цене 200 рублей, 0 номеров по цене 400 рублей, 0 номеров по цене 600 рублей.
Клиент с бюджетом 200 не смог занять никакую комнату

200

5.3. Проверка отказа в брони для клиента с суммой, меньшей минимально допустимой:



```
C:\Users\yuvlo\source\repos\MicroProject2\Debug\MicroProject2.exe
Введите имеющуюся сумму клиента 1:
100
Введите имеющуюся сумму клиента 2:
100
На момент пребывания гостя с бюджетом 100 в гостинице есть 10 номеров по цене 200 рублей, 10 номеров по цене 400 рублей,
5 номеров по цене 600 рублей.
Клиент с бюджетом 100 не смог занять никакую комнату
```

6. Репозиторий исходного кода

<https://github.com/yuvlovchikova/ComputingArchitectureHW/tree/master/MicroProject>

7. Источники

- 1) <http://softcraft.ru/edu/comparch/tasks/mp02/> - формулировка задания
- 2) <https://habr.com/ru/post/182610/> – потоки, блокировки и условные переменные
- 3) <https://medium.com/nuances-of-programming/c-%D0%BC%D1%8C%D1%8E%D1%82%D0%B5%D0%BA%D1%81-%D0%BF%D0%B8%D1%88%D0%B5%D0%BC-%D0%BD%D0%B0%D1%88-%D0%BF%D0%B5%D1%80%D0%B2%D1%8B%D0%B9-%D0%BA%D0%BE%D0%B4-%D0%B4%D0%BB%D1%8F-%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%D0%BF%D0%BE%D1%82%D0%BE%>

[D1%87%D0%BD%D0%BE%D0%B9-%D1%81%D1%80%D0%B5%D0%B4%D1%8B-543a3d60ef30](http://www.cplusplus.com/reference/mutex/mutex/) – статья про мьютекс

4) <http://www.cplusplus.com/reference/mutex/mutex/> – статья про мьютекс

5) <https://nuancesprog.ru/p/5729/> - статья про мьютекс

6) <https://riptutorial.com/ru/cplusplus/example/26751/%D0%BC%D1%8C%D1%8E%D1%82%D0%B5%D0%BA%D1%81%D1%8B-%D0%B8-%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C-%D0%BF%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%B2> - мьютекс и

безопасность потоков

7) <https://ru.stackoverflow.com/questions/570042/%D0%A0%D0%B0%D0%B7%D0%BD%D0%B8%D1%86%D0%B0-%D0%BC%D0%B5%D0%B6%D0%B4%D1%83-push-back-%D0%B8-emplace-back/570080> – разница между push_back и emplace_back