

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

Факультет компьютерных наук Департамент
программной инженерии

**Самостоятельная работа по дисциплине:
"Архитектура вычислительных систем"**

Исполнитель

Студент группы БПИ195

Кенесбек Ерасыл

Почта: ekenesbek@edu.hse.ru

Задание:

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.

Описание решения:

Программа принимает в качестве входных параметров количество посетителей. Пол распределяется случайно.

Посетители в данной программе и являются потоками:

```
for (int i = 0; i < count; ++i){
    threads.push_back(thread(human, i + 1, rand() % 2));
}
```

Синхронизированный вывод(для безопасного вывода):

```
mutex report;
mutex ready;

void print(stringstream& msg)
{
    report.lock();
    cout << msg.str();
    msg.str("");
    report.unlock();
}
```

Для вывода пола:

```
string getSex(int sex)
{
    if (sex == 0) return "Male";
    else if (sex == 1) return "Female";
    else return "Empty";
}
```

Метод, для заселения/распределения человека в комнату. В качестве входных параметров берет номер человека и пол, выводит результат заселения: в какую комнату человек был заселен, в противном случае человек покидает отель:

```
void human(int index, int sex)
{
    stringstream ss;
    srand(time(nullptr));
    this_thread::sleep_for(chrono::milliseconds(1000 + rand() % 10000));

    ss << getSex(sex) << " (" << index << ") came to the hotel.\n";
    print(ss);

    this_thread::sleep_for(chrono::milliseconds(1000 + rand() % 10000));

    ready.lock();

    int number;
    pair<bool, int> trySingle = searchSingle(sex);
    pair<bool, int> tryDouble = searchDouble(sex);

    if (trySingle.first) {
        number = trySingle.second;
        ss << getSex(sex) << " (" << index << ") get room single " << number+1 << ".\n";
        print(ss);
    }
    else if (tryDouble.first) {
        number = tryDouble.second;
        ss << getSex(sex) << " (" << index << ") get room double " << number+1 << ".\n";
        print(ss);
    }
    else {
        ss << getSex(sex) << " (" << index << ") left the hotel.\n";
        print(ss);
    }

    ready.unlock();
}
```

Код программы:

```
#include <iostream>
#include <thread>
#include <mutex>
#include <vector>
#include <utility>
#include <chrono>
#include <random>
#include <string>
#include <sstream>

using namespace std;

mutex report;
mutex ready;

vector<int> singleRooms(10, -1);
vector< pair<int, int> > doubleRooms(15, make_pair(-1, -1));

void print(stringstream& msg)
{
    report.lock();
    cout << msg.str();
    msg.str("");
    report.unlock();
}

string getSex(int sex) {
    if (sex == 0) return "Male";
    else if (sex == 1) return "Female";
    else return "Empty";
}

pair<bool, int> searchSingle(int sex) {
    for (int i = 0; i < singleRooms.size(); ++i) {
        if (singleRooms.at(i) == -1) {
            singleRooms.at(i) = sex;
            return make_pair(true, i);
        }
    }
    return make_pair(false, -1);
}

pair<bool, int> searchDouble(int sex) {
    for (int i = 0; i < doubleRooms.size(); i++) {
        if (doubleRooms.at(i) == make_pair(-1, -1)) {
            doubleRooms.at(i) = make_pair(-1, sex);
            return make_pair(true, i);
        }
        else if (doubleRooms.at(i) == make_pair(-1, sex) || doubleRooms.at(i) ==
make_pair(sex, -1)) {
            doubleRooms.at(i) = make_pair(sex, sex);
            return make_pair(true, i);
        }
    }
    return make_pair(false, -1);
}

void human(int index, int sex) {
    stringstream ss;
    srand(time(nullptr));
    this_thread::sleep_for(chrono::milliseconds(1000 + rand() % 10000));
```

```

ss << getSex(sex) << " (" << index << ") came to the hotel.\n";
print(ss);

this_thread::sleep_for(chrono::milliseconds(1000 + rand() % 10000));

ready.lock();

int number;
pair<bool, int> trySingle = searchSingle(sex);
pair<bool, int> tryDouble = searchDouble(sex);

if (trySingle.first) {
    number = trySingle.second;
    ss << getSex(sex) << " (" << index << ") get room single " << number+1 << ".\n";
    print(ss);
}
else if (tryDouble.first) {
    number = tryDouble.second;
    ss << getSex(sex) << " (" << index << ") get room double " << number+1 << ".\n";
    print(ss);
}
else {
    ss << getSex(sex) << " (" << index << ") left the hotel.\n";
    print(ss);
}

ready.unlock();
}

int main() {
    cout << "Input amount of human(gender assigns randomly)"<<endl;
    int count;
    cin>> count;
    vector<thread> threads;

    for (int i = 0; i < count; ++i) {
        threads.push_back(thread(human, i + 1, rand() % 2));
    }

    for (int i = 0; i < count; ++i) {
        threads.at(i).join();
    }

    cout << "\nSingle Rooms:\n";
    for (int i = 0; i < singleRooms.size(); ++i) {
        cout << "Room number " << i+1 << ": " << getSex(singleRooms.at(i)) << endl;
    }

    cout << "\n\nDouble Rooms:\n";
    for (int i = 0; i < doubleRooms.size(); ++i) {
        cout << "Room number " << i+1 << ": " << getSex(doubleRooms.at(i).first) << ", "
<< getSex(doubleRooms.at(i).second) << endl;
    }
    return 0;
}

```