

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

Факультет компьютерных наук  
Департамент программной инженерии

**МИКРОПРОЕКТ №2**  
**ПРОГРАММА-ИЛЛЮСТРАЦИЯ ЗАДАЧИ О БРАДОБРЕЕ**  
**НА ЯЗЫКЕ C++**

**Пояснительная записка**

Инв. №подл	Подп. и дата	Инв. № дубл.	Подп. и дата

**Исполнитель:**

Студент группы БПИ191\_1,  
Соколовский Вацлав Антонович

**Москва 2020**

## Оглавление

1.	ВВЕДЕНИЕ .....	3
	■ Текст задания .....	3
	■ Дополнительный реализованный функционал.....	3
	■ Расчетные методы .....	3
	■ Параметры ввода и их область значения.....	4
2.	ТЕСТИРОВАНИЕ.....	5
	■ Проверка некорректного ввода .....	5
	■ Простой тест.....	5
	■ Простой тест с заполнением комнаты ожидания .....	5
	■ Тест нормального размера .....	6
	■ Тест нормального размера с заполнением комнаты ожидания .....	7
	■ Большой тест.....	8
	ИСТОЧНИКИ: .....	9
	ПРИЛОЖЕНИЕ 1 .....	10

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 1. ВВЕДЕНИЕ

### ■ Текст задания

В тихом городке есть парикмахерская. Салон парикмахерской мал, ходить там может только парикмахер и один посетитель. Парикмахер всю жизнь обслуживает посетителей. Когда в салоне никого нет, он спит в кресле. Когда посетитель приходит и видит спящего парикмахера, он будит его, садится в кресло и спит, пока парикмахер занят стрижкой. Если посетитель приходит, а парикмахер занят, то он встает в очередь и засыпает. После стрижки парикмахер сам провожает посетителя. Если есть ожидающие посетители, то парикмахер будит одного из них и ждет пока тот сядет в кресло парикмахера и начинает стрижку. Если никого нет, он снова садится в свое кресло и засыпает до прихода посетителя. Создать многопоточное приложение, моделирующее рабочий день парикмахерской.

### ■ Дополнительный реализованный функционал

Было решено дополнительно реализовать зал ожидания в парикмахерской с заданным пользователем количеством мест, чтобы модель работы парикмахерской была чуть более приближена к реальности. Если все места в зале ожидания заняты, то новый клиент просто уходит по своим делам.

Дополнительно была добавлена очередь клиентов, представленная структурой `queue<thread::id>`, помогающая упорядочить поток клиентов из зала ожидания.

Добавлен метод генерации секунд на заданном интервале для случайной генерации времени между посещениями клиентов (для моделирования случайности поведения клиентов).

Также дополнительно реализован собственный семафор, ибо семафоры из `semaphore.h` отказывались работать на моей версии компилятора.

### ■ Расчетные методы

Во избежание бесконечного цикла в легенду было введено число – количество людей в городе. Парикмахер завершает рабочий день, если каждый человек в городе зашел в его парикмахерскую независимо от исхода (постригли в итоге его, или он ушел из-за заполненности зала ожидания)

Собственный семафор реализован с помощью счетчика типа `int`, мьютекса на этот счётчик и `condition_variable`, позволяющим ожидать выполнения некоего условия (в нашем случае мы ждем, пока счетчик семафора не поднимется выше нуля).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Метод парикмахера реализован с помощью цикла, который завершается, когда количество жителей города исчерпано. Каждая итерация цикла представляет из себя:

- Оповещение клиентов о том, что парикмахер освободился
- Сон в ожидании нового клиента
- Пробуждение после оповещения от клиента
- Работа

Метод Клиента реализован с помощью блока условного ветвления if/else, где ветка else представляет собой случай, когда очередной клиент не находит места в зале ожидания и просто уходит, а ветка if представляет собой действия клиента в зале ожидания:

- ID клиента добавляется в очередь
- Ожидание, пока начало очереди не будет иметь тот же ID, что и у данного клиента
- Ожидание оповещения от парикмахера о том, что он освободился
- Оповещение парикмахера о своем прибытии
- Удаление своего ID из начала очереди

### ■ Параметры ввода и их область значения

Пользователь должен ввести два параметра для запуска программы:

- chairs – количество мест в зале ожидания. Область значения для корректной работы: (0; 10000]
- limit – количество людей в городе, не считая парикмахера. Область значения для корректной работы: (0; 10000]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

## 2. ТЕСТИРОВАНИЕ

Для удобства тестирования в конце программы выводится количество пришедших в общем клиентов (их количество всегда должно быть равно количеству людей в городе).

### ■ Проверка некорректного ввода

```
Введите количество мест в зале ожидания (целочисленное значение из диапазона (0; 10000]: f
Неверный ввод! Попробуйте еще раз.
Введите количество мест в зале ожидания (целочисленное значение из диапазона (0; 10000]: check
Неверный ввод! Попробуйте еще раз.
Введите количество мест в зале ожидания (целочисленное значение из диапазона (0; 10000]: please do it
Неверный ввод! Попробуйте еще раз.
Введите количество мест в зале ожидания (целочисленное значение из диапазона (0; 10000]: -5
Введите количество мест в зале ожидания (целочисленное значение из диапазона (0; 10000]: 0
Введите количество мест в зале ожидания (целочисленное значение из диапазона (0; 10000]: 3
Введите количество людей в городе (целочисленное значение из диапазона (0; 10000]: f
Неверный ввод! Попробуйте еще раз.
Введите количество людей в городе (целочисленное значение из диапазона (0; 10000]: check
Неверный ввод! Попробуйте еще раз.
Введите количество людей в городе (целочисленное значение из диапазона (0; 10000]: please do it
Неверный ввод! Попробуйте еще раз.
Введите количество людей в городе (целочисленное значение из диапазона (0; 10000]: -10
Введите количество людей в городе (целочисленное значение из диапазона (0; 10000]: 0
Введите количество людей в городе (целочисленное значение из диапазона (0; 10000]: 10
Клиент 0x700001956000 проходит в зал ожидания.
```

### ■ Простой тест

```
Введите количество мест в зале ожидания (целочисленное значение из диапазона (0; 10000]: 3
Введите количество людей в городе (целочисленное значение из диапазона (0; 10000]: 3
Клиент 0x700000e18000 проходит в зал ожидания.
Клиент 0x700000e18000 проходит на стрижку.
Парикмахер просыпается
Парикмахер стрижет клиента
Парикмахер засыпает
Клиент 0x700000e9b000 проходит в зал ожидания.
Клиент 0x700000e9b000 проходит на стрижку.
Парикмахер просыпается
Парикмахер стрижет клиента
Парикмахер засыпает
Клиент 0x700000f1e000 проходит в зал ожидания.
Клиент 0x700000f1e000 проходит на стрижку.
Парикмахер просыпается
Парикмахер стрижет клиента
Парикмахер засыпает
Всего клиентов пришло за день: 3
```

### ■ Простой тест с заполнением комнаты ожидания

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Введеine количество мест в зале ожидания (целочисленное значение из диапазона (0; 10000]: 1  
 Введеine количество людей в городе (целочисленное значение из диапазона (0; 10000]: 4  
 Клиент 0x700004429000 проходит в зал ожидания.  
 Клиент 0x700004429000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Клиент 0x7000044ac000 проходит в зал ожидания.  
 Парикмахер засыпает  
 Клиент 0x7000044ac000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Клиент 0x70000452f000 проходит в зал ожидания.  
 Нет мест в зале ожидания. Поток 0x7000045b2000 уходит домой  
 Парикмахер засыпает  
 Клиент 0x70000452f000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Парикмахер засыпает  
 Всего клиентов пришло за день: 4

## Тест нормального размера

Парикмахер засыпает  
 Клиент 0x70000ae7b000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Клиент 0x70000ae87000 проходит в зал ожидания.  
 Клиент 0x70000af0a000 проходит в зал ожидания.  
 Клиент 0x70000af5d000 проходит в зал ожидания.  
 Парикмахер засыпает  
 Клиент 0x70000acfe000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Клиент 0x70000b010000 проходит в зал ожидания.  
 Парикмахер засыпает  
 Клиент 0x70000ad81000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Клиент 0x70000b070000 проходит в зал ожидания.  
 Клиент 0x70000b114000 проходит в зал ожидания.  
 Клиент 0x70000b199000 проходит в зал ожидания.  
 Парикмахер засыпает  
 Клиент 0x70000ae84000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Клиент 0x70000b21c000 проходит в зал ожидания.  
 Парикмахер засыпает  
 Клиент 0x70000ae87000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Клиент 0x70000b29f000 проходит в зал ожидания.  
 Клиент 0x70000b322000 проходит в зал ожидания.  
 Парикмахер засыпает  
 Клиент 0x70000af0a000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Клиент 0x70000b3a5000 проходит в зал ожидания.  
 Парикмахер засыпает  
 Клиент 0x70000af8d000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Парикмахер засыпает  
 Клиент 0x70000b010000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Парикмахер засыпает  
 Клиент 0x70000b073000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Парикмахер засыпает  
 Клиент 0x70000b116000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Парикмахер засыпает  
 Клиент 0x70000b199000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Парикмахер засыпает  
 Клиент 0x70000b21c000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Парикмахер засыпает  
 Клиент 0x70000b29f000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Парикмахер засыпает  
 Клиент 0x70000b322000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Парикмахер засыпает  
 Клиент 0x70000b3a5000 проходит на стрижку.  
 Парикмахер просыпается  
 Парикмахер стрижет клиента  
 Парикмахер засыпает  
 Всего клиентов пришло за день: 20

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

■ Тест нормального размера с заполнением комнаты ожидания

Введи количество мест в зале ожидания (целочисленное значение из диапазона (0; 10000]): 3  
Введи количество людей в городе (целочисленное значение из диапазона (0; 10000]): 20  
Клиент 0x7000078ee000 проходит в зал ожидания.  
Клиент 0x7000078ee000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x700007971000 проходит в зал ожидания.  
Клиент 0x7000079f4000 проходит в зал ожидания.  
Клиент 0x700007a77000 проходит в зал ожидания.  
Парикмахер засыпает  
Клиент 0x700007971000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x700007afa000 проходит в зал ожидания.  
Парикмахер засыпает  
Клиент 0x7000079f4000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x700007b7d000 проходит в зал ожидания.  
Нет мест в зале ожидания. Поток 0x700007c00000 уходит домой  
Парикмахер засыпает  
Клиент 0x700007a77000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x700007c33000 проходит в зал ожидания.  
Нет мест в зале ожидания. Поток 0x700007d06000 уходит домой  
Парикмахер засыпает  
Клиент 0x700007afa000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x700007d09000 проходит в зал ожидания.  
Нет мест в зале ожидания. Поток 0x700007e0c000 уходит домой  
Нет мест в зале ожидания. Поток 0x700007e0f000 уходит домой  
Парикмахер засыпает  
Клиент 0x700007b7d000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x700007f12000 проходит в зал ожидания.  
Парикмахер засыпает  
Клиент 0x700007c83000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x700007f95000 проходит в зал ожидания.  
Нет мест в зале ожидания. Поток 0x700000010000 уходит домой  
Нет мест в зале ожидания. Поток 0x700000090000 уходит домой  
Парикмахер засыпает  
Клиент 0x700007d09000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x7000001a0000 проходит в зал ожидания.  
Нет мест в зале ожидания. Поток 0x7000001a1000 уходит домой  
Парикмахер засыпает  
Клиент 0x700007f12000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x700000224000 проходит в зал ожидания.  
Нет мест в зале ожидания. Поток 0x7000002a7000 уходит домой  
Парикмахер засыпает  
Клиент 0x700007f95000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Клиент 0x7000001a0000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Всего клиентов пришло за день: 20

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

Большой тест

Парикмахер стрижет клиента  
Клиент 0x7000edbc000 проходит в зал ожидания.  
Нет мест в зале ожидания. Поток 0x7000ee3f000 уходит домой  
Парикмахер засыпает  
Клиент 0x7000e1f7000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x7000e0ec2000 проходит в зал ожидания.  
Парикмахер засыпает  
Клиент 0x7000e2fd000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x7000ef45000 проходит в зал ожидания.  
Нет мест в зале ожидания. Поток 0x7000efc8000 уходит домой  
Нет мест в зале ожидания. Поток 0x7000f04b000 уходит домой  
Парикмахер засыпает  
Клиент 0x7000e403000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x7000ef0ce000 проходит в зал ожидания.  
Парикмахер засыпает  
Клиент 0x7000e58c000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x7000f151000 проходит в зал ожидания.  
Нет мест в зале ожидания. Поток 0x7000f1d4000 уходит домой  
Парикмахер засыпает  
Клиент 0x7000e715000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Клиент 0x7000f257000 проходит в зал ожидания.  
Парикмахер засыпает  
Клиент 0x7000e81b000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Клиент 0x7000e921000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Клиент 0x7000e9aa000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Клиент 0x7000ec33000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Клиент 0x7000edbc000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Клиент 0x7000ec2000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Клиент 0x7000ef45000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Клиент 0x7000f0ce000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Клиент 0x7000f151000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Клиент 0x7000f257000 проходит на стрижку.  
Парикмахер просыпается  
Парикмахер стрижет клиента  
Парикмахер засыпает  
Всего клиентов пришло за день: 100

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата



**ИСТОЧНИКИ:**

1. Примеры реализаций семафоров на C++ [Электронный ресурс] – Режим доступа: <https://riptutorial.com/cplusplus/>, свободный. (дата обращения: 29.10.2020).
2. Документация по языку C++ [Электронный ресурс] – Режим доступа: <https://en.cppreference.com>, свободный. (дата обращения: 29.10.2020).
3. Stackoverflow Forum [Электронный ресурс] – Режим доступа: <https://stackoverflow.com>, свободный. (дата обращения: 29.10.2020).
4. Habr Forum [Электронный ресурс] – Режим доступа: <https://habr.com>, свободный. (дата обращения: 29.10.2020).

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

ТЕКСТ ПРОГРАММЫ

```
#include <iostream>

#include <thread>
#include <mutex>
#include <random>
#include <condition_variable>
#include <vector>
#include <queue>

using namespace std;

class semaphore {
private:
    mutex mtx;
    condition_variable cv;
    int count;
public:
    explicit semaphore(int count_ = 0) : count(count_) {}

    void notify() {
        unique_lock<mutex> lock(mtx);
        ++count;
        cv.notify_one();
    }

    void wait() {
        unique_lock<mutex> lock(mtx);
        while(count == 0) {
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

        cv.wait(lock);
    }

    --count;
}

};

vector<thread> v; // вектор потоков для сохранения потоков
(чтобы они не уничтожались после создания)

queue<thread::id> q; // очередь потоков (для соблюдения
порядка в зале ожидания)

semaphore waiting_customers(0); // семафор для посетителей,
ожидающих в очереди

semaphore barbers(0); // семафор для парикмахеров, в нашем
случае он всегда один, но можно сделать и больше

mutex console_mutex; // мутекс для записи в консоль, ибо иначе
вывод будет некрасивым

std::atomic_int waiting = 0, limit = 0, result = 0;

int chairs = 0;

void barber() {
    while (limit > 0 || waiting != 0) {
        barbers.notify(); // парикмахер готов принять клиента
        waiting_customers.wait(); // ждем пока семафор
посетителей не объявит о добавлении нового посетителя
    }
    lock_guard<mutex> lock_console(console_mutex);
    cout << "Парикмахер просыпается\n";
}

waiting--; /* уменьшаем кол-во ожидающих клиентов */

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

    {
        lock_guard<mutex> lock_console(console_mutex);
        cout << "Парикмахер стрижет клиента\n";
    }

    this_thread::sleep_for(1s); // каждая стрижка длится
1 секунду (для удобства)
    {
        lock_guard<mutex> lock_console(console_mutex);
        cout << "Парикмахер засыпает\n";
    }

    result++;
}
}

void customer() {
    if (waiting < chairs) /* есть место для ожидания */ {
        q.push(this_thread::get_id());
        waiting++; /* увеличиваем кол-во ожидающих клиентов
*/
        {
            lock_guard<mutex> lock_console(console_mutex);
            cout << "Клиент " << this_thread::get_id() << "
проходит в зал ожидания." << endl;
        }

        while (this_thread::get_id() != q.front()); //
проверка на очередь, чтобы к парикмахеру первым заходил не
самый быстрый поток)

        barbers.wait(); /* если парикмахер занят, переходим в
состояние ожидания, иначе – занимаем парикмахера*/

        waiting_customers.notify(); /* если парикмахер спит,
это его разбудит */
    }
}

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```

    {
        lock_guard<mutex> lock_console(console_mutex);

        cout << "Клиент " << this_thread::get_id() << "
проходит на стрижку.\n";

    }

    q.pop();

} else {

    lock_guard<mutex> lock_console(console_mutex);

    cout << "Нет мест в зале ожидания. Поток " <<
this_thread::get_id() << " уходит домой" << endl;

    result++;

}

}

```

```

/// Метод для проверки ввода
/// \param text текст запроса
/// \return корректное значение
int get_int_input(const string &text) {

    string str;

    int value = 0;

    while (value <= 0 || value > 10'000) { // проверка ввода

        cout << text;

        getline(cin, str);

        try {

            value = stoi(str);

        } catch (...) {

            cout << "Неверный ввод! Попробуйте еще раз.\n";

            value = 0;

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```
}

```

```
}

```

```
return value;

```

```


```

```
chrono::duration<double> rand_some_chrono_sec(double left,
double right) {

```

```
    random_device rd; // источник энтропии

```

```
    mt19937 generator(rd()); // генератор чисел

```

```
    uniform_real_distribution<> dist(left, right); //
    равномерное распределение от left до right

```

```
    double k = dist(generator);

```

```
    return chrono::duration<double>(k);

```

```


```

```
int main() {

```

```
    chairs = get_int_input("Введите количество мест в зале
ожидания "

```

```
                           "(целочисленное значение из
диапазона (0; 10000]: ");

```

```
    limit = get_int_input("Введите количество людей в городе "

```

```
                           "(целочисленное значение из
диапазона (0; 10000]: ");

```

```
    thread barb(barber);

```

```
    thread helper([] {

```

```
        while (limit-- != 0) {

```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

```
v. empl ace_back( cust omer );

t hi s_ t hread: : sl ee p_ f or ( r and_ some_ ch r on o_ sec ( 0. 1, 2. 5) ) ;

}

} );

bar b. j oi n ( ) ;

hel per. j oi n ( ) ; // джоиним все потоки с мейном, ибо они все
что-то выводят, а терять строки в логe не хочется

for ( auto & t hread : v )

t hread. j oi n ( ) ;

cout << " Всего клиентов пришло за день: " << res ul t ;

return 0;

}
```

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.15-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	Подп. и дата

