

# Содержание

1. Введение	2
2. Алгоритм решения	3
3. Программа	4
4. Список используемой литературы	5

# 1. Введение

Для генерирования случайных чисел Парка-Миллера с перетасовкой и без была использована среда разработки Visual Studio 2022. Для оформления и написания отчёта использовался онлайн-компилятор LaTeX Overleaf

## 2. Алгоритм решения

Самая простая последовательность, которую можно предложить для реализации генератора равномерного распределения:

$$I(j+1) = a * I(j) \pmod{m}$$

при соответствующем выборе констант. Константы были предложены Park и Miller:

$$a = 75 = 16807, \quad m = 2^{31} - 1 = 2147483647.$$

Модуль разлагается в выражение:

$$m = a * q + r$$

Если  $r < q$  и  $0 < z < m - 1$ , то при этом величины  $a * (z \bmod q)$  и  $r * [z/q]$  всегда лежат в интервале  $0, \dots, m - 1$ . Для умножения  $(a * z) \pmod{m}$  при этом используется алгоритм:

- $t = a(z \bmod q) - r[z/q]$
- если  $t < 0$ , то  $t += m$ .
- $(a * z) \pmod{m} = t$ .

В случае констант Парка-Миллера можно использовать  $q = 12773$  и  $r = 2836$ .

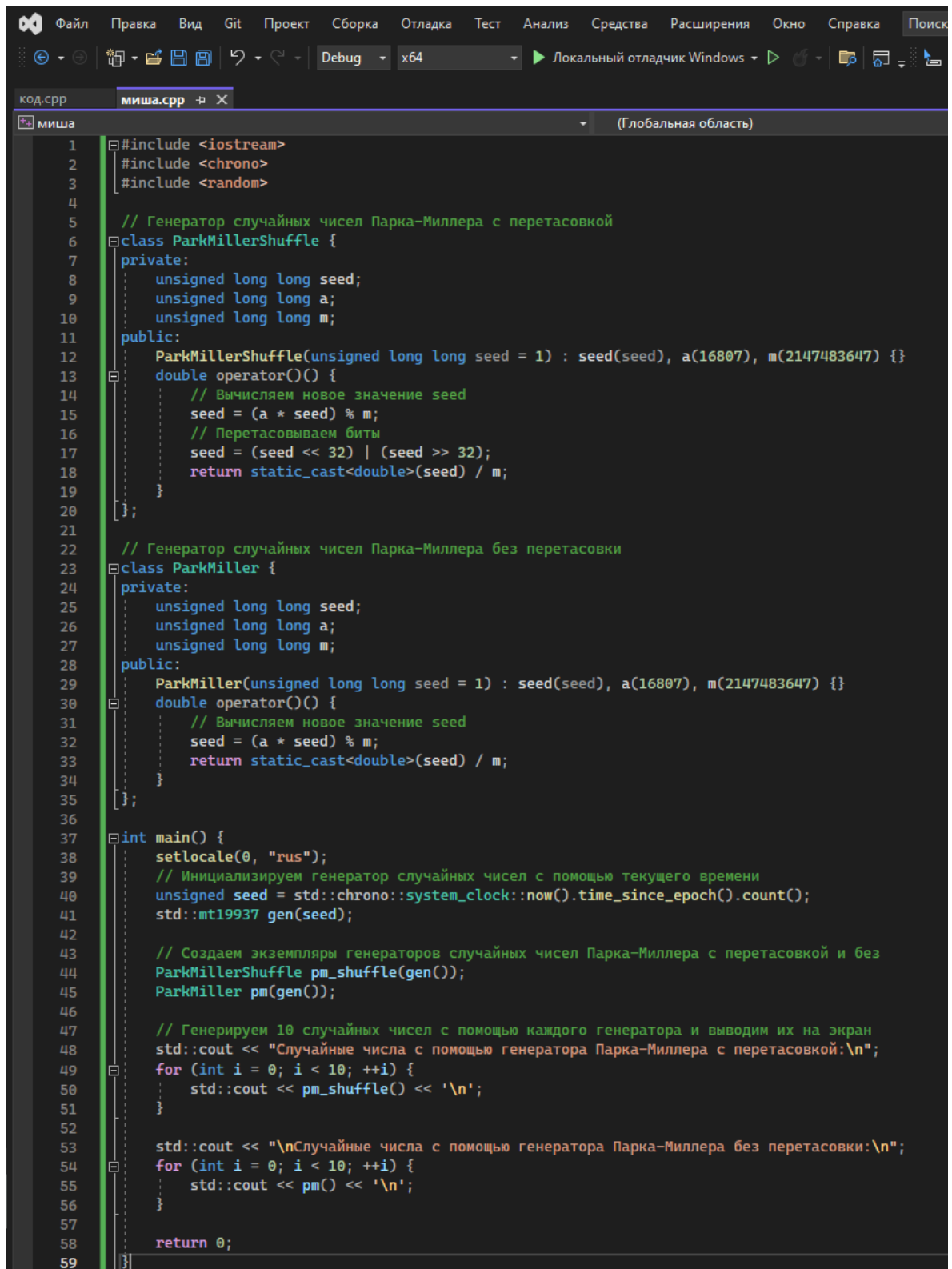
### 3. Программа

Этот код на C++ демонстрирует два класса генераторов случайных чисел, основанных на алгоритме Парка-Миллера: `ParkMillerShuffle` и `ParkMiller`. Оба класса имеют одинаковый набор членов: `seed`, `a` и `m`. `Seed` - это начальное значение для генератора случайных чисел, `a` и `m` - это параметры алгоритма Парка-Миллера.

Класс `ParkMillerShuffle` использует операцию перетасовки битов для улучшения качества генерируемых случайных чисел. Класс `ParkMiller` не использует операцию перетасовки битов.

В функции `main()` создаются экземпляры обоих классов и используется генератор случайных чисел `std::mt19937` для инициализации начального значения `seed`. Затем генерируются и выводятся на экран 10 случайных чисел с помощью каждого генератора.

Код может быть использован для демонстрации различных подходов к генерации случайных чисел и их применения в приложениях.



```
1 #include <iostream>
2 #include <chrono>
3 #include <random>
4
5 // Генератор случайных чисел Парка-Миллера с перетасовкой
6 class ParkMillerShuffle {
7 private:
8     unsigned long long seed;
9     unsigned long long a;
10    unsigned long long m;
11 public:
12    ParkMillerShuffle(unsigned long long seed = 1) : seed(seed), a(16807), m(2147483647) {}
13    double operator()() {
14        // Вычисляем новое значение seed
15        seed = (a * seed) % m;
16        // Перетасовываем биты
17        seed = (seed << 32) | (seed >> 32);
18        return static_cast<double>(seed) / m;
19    }
20 };
21
22 // Генератор случайных чисел Парка-Миллера без перетасовки
23 class ParkMiller {
24 private:
25     unsigned long long seed;
26     unsigned long long a;
27     unsigned long long m;
28 public:
29    ParkMiller(unsigned long long seed = 1) : seed(seed), a(16807), m(2147483647) {}
30    double operator()() {
31        // Вычисляем новое значение seed
32        seed = (a * seed) % m;
33        return static_cast<double>(seed) / m;
34    }
35 };
36
37 int main() {
38    setlocale(0, "rus");
39    // Инициализируем генератор случайных чисел с помощью текущего времени
40    unsigned seed = std::chrono::system_clock::now().time_since_epoch().count();
41    std::mt19937 gen(seed);
42
43    // Создаем экземпляры генераторов случайных чисел Парка-Миллера с перетасовкой и без
44    ParkMillerShuffle pm_shuffle(gen());
45    ParkMiller pm(gen());
46
47    // Генерируем 10 случайных чисел с помощью каждого генератора и выводим их на экран
48    std::cout << "Случайные числа с помощью генератора Парка-Миллера с перетасовкой:\n";
49    for (int i = 0; i < 10; ++i) {
50        std::cout << pm_shuffle() << '\n';
51    }
52
53    std::cout << "\nСлучайные числа с помощью генератора Парка-Миллера без перетасовки:\n";
54    for (int i = 0; i < 10; ++i) {
55        std::cout << pm() << '\n';
56    }
57
58    return 0;
59 }
```

Рис. 1. Результат выполнения программы

## 4. Список используемой литературы

- 1) "Язык программирования C++. Базовый курс" Бьерн Страуструп - Издательство: «Питер», 2006, 1104с.

- 2) "Роберт Лафоре. Объектно-ориентированное программирование в C++
- 3) "<https://www.youtube.com/watch?v=kRcbYLK3OnQlist=PLQOaTSbfxUtCrKs0nicOg2npJQYSPGO9r>