

# 1. Ход работы

## 1.1. Код программы

```
class ParkMiller:
    def __init__(self, seed):
        self.seed = seed
        self.multiplier = 16807
        self.modulus = 214748364

    def rand(self):
        self.seed = (self.multiplier * self.seed) % self.modulus
        return self.seed / self.modulus

class ShuffledParkMiller:
    def __init__(self, seed):
        self.generator = ParkMiller(seed)

    def rand(self):
        values = []
        for i in range(0, 16):
            values.append(self.generator.rand())

        # shuffle the values
        for i in range(15, 0, -1):
            j = int(self.generator.rand() * (i+1))
            values[i], values[j] = values[j], values[i]

        # return the first value
        return values[0]

def main():
    seed = 123456789
    park_miller = ParkMiller(seed)
    shuffled_park_miller = ShuffledParkMiller(seed)

    # generate 10 random numbers with Park-Miller generator
    print("Random numbers generated by Park-Miller:")
    for i in range(0, 10):
        print(park_miller.rand())

    # generate 10 random numbers with shuffled Park-Miller generator
```

```

print("\nRandom numbers generated by Shuffled Park-Miller:")
for i in range(0, 10):
    print(shuffled_park_miller.rand())

if __name__ == "__main__":
    main()

```

## 1.2. ТЕОРИЯ

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

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

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

$$a = 7^5 = 16807, m = 2^{31} - 1 = 2147483647 \quad (2)$$

Модуль разлагается в выражение:

$$m = a * q + r \quad (3)$$

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:

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

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

## 2. Пример вставки результата отладки программы изображения

Пример отладки программы представлен на рис. 1.

```
Random numbers generated by Park-Miller:
0.2184182969939049
0.9563175765594084
0.8295092339764858
0.5616954427965429
0.415307081497883
0.06611873491952137
0.257577792395641
0.10995679353827462
0.04382899778142059
0.6339657123358761

Random numbers generated by Shuffled Park-Miller:
0.10995679353827462
0.36702666774719334
0.01744227903775977
0.8322809398324605
0.04062438106193411
0.7808003140524031
0.4920846458953268
0.9083391390314042
0.2904765225436895
0.6181980756196185
```

Рис. 1. отладка