

ЗАДАНИЕ 2

Спроектируйте и реализуйте иерархию классов для генерации случайных чисел с дискретным вероятностным распределением, а также фабрику для создания экземпляра класса генератора с заданными параметрами.

```
class TRandomNumberGenerator {
public:
    virtual ~TRandomNumberGenerator();
    virtual double Generate() const = 0;
};
```

Необходимо поддерживать четыре типа распределения: распределение Пуассона, распределение Бернулли, геометрическое распределение и распределение конечной дискретной случайной величины. Тип распределения задаётся строковым параметром `type`, который принимает значения из множества `poisson`, `bernoulli`, `geometric`, `finite`. При создании генератора передаются также параметры распределений:

- в случае распределения Бернулли и геометрического распределения задается вероятность p выпадения единицы (или «успеха»);
- в случае конечного дискретного распределения задается два вектора параметров (x_1, \dots, x_k) — значения случайной величины и (p_1, \dots, p_k) — их вероятности.
- в случае распределения Пуассона задается параметр λ .

Предполагается, что $p \in [0, 1]$, $\lambda > 0$, $\sum_{i=1}^k p_i = 1$, $p_i \in [0, 1]$. Если эти ограничения не выполняются — нужно не создавать объект (возвращать `nullptr`).

Напишите функцию, которая реализует подсчет эмпирического среднего значения путем многократного вызова метода `Generate`. Протестируйте с несколькими параметрами каждое распределение (достаточно 4 методов, каждый из которых создаст `std::unique_ptr<TRandomNumberGenerator>` с некоторыми параметрами и проводит сравнение теоретического и эмпирического средних значений).

Разрешается использовать встроенные генераторы псевдослучайных чисел в C++.

Решение задачи нужно прислать (предпочтительнее ссылка на github, и т.п.) на vkonovodov@gmail.com. Решение должно быть оформлено в виде компилирующегося кода с комментариями про исправление ошибок отдельным текстом или непосредственно в коде.

Срок сдачи задания — 25 ноября 2018 г. 23:59.