

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«МИРЭА – Российский технологический университет»  
РТУ МИРЭА  
Институт Информационных Технологий  
Кафедра Промышленной Информатики



## ПРОЦЕДУРНОЕ ПРОГРАММИРОВАНИЕ

Тема «ЗАДАЧА ПРО ШАРИКИ»

Лектор **Каширская Елизавета Натановна** (к.т.н., доцент, ФГБОУ ВО "МИРЭА - Российский технологический университет") e-mail: [liza.kashirskaya@gmail.com](mailto:liza.kashirskaya@gmail.com)

**Практическое занятие**



В качестве примера использования рекурсивного вызова функции давайте разберем задачу, которая у многих вызывает затруднения.

*Из урны с 10 пронумерованными шариками вынимают по одному шарiku. Подсчитать общее количество ситуаций, когда номер хотя бы одного вынутого шарика совпадает с порядковым номером действия "вынимания", например, когда шарик № 3 будет вынут 3-им по порядку.*

### **Один из возможных вариантов алгоритма решения задачи про шарики**

1. Задать количество шариков  $n$ .
2. Создать массив пронумерованных шариков от 1 до  $n$ .
3. Целочисленная переменная  $i$  - номер шарика (от 1 до  $n$ ), одновременно являющаяся счетчиком действий.
4. Создать функцию *perestanovka* от целочисленных  $m$  и  $n$ , которая генерирует перестановки, в зависимости от количества шариков ( $n$ ) и в которой фигурирует номер очередного переставляемого шарика ( $m$ ). В этой функции использовать условие: когда номер шага  $i$  равен номеру вынимаемого шарика  $m$ , печатать очередную перестановку. Во всех остальных случаях менять местами элементы с номерами  $i$  и  $m$ , после чего вызывать функцию *perestanovka* со следующим значением шага и опять же менять местами элементы с номерами  $i$  и  $m$ .
5. Основная программа: присваивание шарикам порядковых номеров, вызов функции *perestanovka* с параметрами 1 (первый шаг) и  $n$  (количество шариков).

## ЗАДАЧА «ШАРИКИ»



Чтобы понять, какие комбинации шариков надо учитывать, проведем эксперимент с небольшим количеством шариков, составив для них все возможные перестановки.

2 шарика: **1, 2** (подходит, так как шарик № 1 вынут первым; и № 2 тоже, но это не важно)

2, 1 (нет)

*Ответ: 1*

3 шарика: **1, 2, 3** (подходит, так как шарик № 1 вынут первым; и № 2 тоже, но это не важно)

**1, 3, 2** (подходит, так как шарик № 1 вынут первым)

2, 1, **3** (подходит, так как шарик № 3 вынут третьим)

2, 3, 1 (нет)

3, 1, 2 (нет)

3, **2**, 1 (подходит, так как шарик № 2 вынут вторым)

*Ответ: 4*

## Вывод рекуррентной формулы



п- количество шариков	P=n!-число перестановок	к-искомый результат	Расчет результата
1	1	1	1
2	2	1	1
3	6	4	(1+1)*2=4
4	24	15	(4+1)*3=15
5	120	76	(15+4)*4=76
6	720	455	(76+15)*5=455
7	5040	3186	(455+76)*6=3186
8	40320	25480	(3186+455)*7=25487
9	362880	229384	(25487+3186)*8=229384
10	3628800	2293839	(229384+25487)*9=2293839
11	39916800	25232230	(2293839+229384)*10=25232230
12	479001600	302786759	(25232230+2293839)*11=302786759
13	6227020800		

$$k_n = (k_{n-1} + k_{n-2})(n-1)$$

Мы с вами изучаем не комбинаторику, а программирование! Эта формула годится для проверки полученного программным способом результата.



```
void generate (int t) // Создает все перестановки шариков, число которых равно t
{
    if (t==n-1)
    {
        //Вывод очередной перестановки
        for (int i=0;i<n;++i)
            cout<<a[i]<< " ";
        cout<<endl;
    }
    else
    {
        for (int j=t;j<n;++j)
        {
            //Запускаем процесс обмена
            //a[t] со всеми последующими
            swap(a[t],a[j]);
            t++;
            generate(t); //Рекурсивный вызов
            t--;
            swap(a[t],a[j]);
        }
    }
}
```

**ЗАВЕРШИТЬ РАБОТУ ПРЕДЛАГАЮ ВАМ САМИМ!**



1. Материалы из открытого университета INTUIT.RU

Алгоритмизация. Введение в язык программирования С++ [Электронный ресурс].

Режим доступа: <https://www.intuit.ru/studies/courses/16740/1301/info>

**2. Уроки программирования на языке С++.** [Электронный ресурс]. Режим доступа:

<https://ravesli.com/uroki-cpp/#toc-0>