

Златопольский Дмитрий Михайлович

*Кандидат технических наук, доцент кафедры
информатики и прикладной математики
Московского городского педагогического университета.*



Задача «Рыбаки и рыбки»

В статье рассматриваются различные варианты программы решения задачи, которую придумал известный английский физик Поль Дирак. В дополнении к статье задача решается средствами электронной таблицы.

Задача такая: «Три рыбака легли спать, не поделив улова. Проснувшийся ночью первый рыбак решил уйти, взяв свою долю. Но число рыб не делилось на три. Тогда он выбросил одну рыбу, а из числа оставшихся забрал треть. Второй и третий рыбаки поступили аналогично (выбросили по одной рыбе и взяли треть из оставшихся). Спрашивается, какое наименьшее количество рыб может удовлетворить условию задачи?»

Поль Дирак был мастер давать различным существительным приставку «анти» – античастица, например. И в этой задаче он, по-видимому, не изменил своей привычке, оригинально решив её – ответом является значение «минус две рыбы». Выбрасываем одну – получаем минус три, забираем треть – останется минус две и т. д.

Попробуем решить задачу, не допуская возможности ловли «анти-

рыб», то есть найдём наименьшее *положительное* число, удовлетворяющее условию приведённой задачи. Правда, при этом сделаем небольшое уточнение – число рыб, доставшееся каждому рыбаку, не обязательно должно быть одинаковым, как в решении Дирака.

Сначала рассмотрим такую, более простую, задачу: «Имеется некоторое количество рыб. Определить, возможен ли делёж рыб между тремя рыбаками в соответствии с условием задачи Дирака» (очевидно, что не при любом числе возможен такой делёж – убедитесь в этом!).

Идея решения такая. Нужно проверять, кратно ли трём число рыб, остающееся после выбрасывания одной рыбы каждым из трёх рыбаков. Если для какого-то рыбака ответ отрицательный, то дальнейшие действия, естественно, надо прекратить

(используем оператор цикла с условием, в котором учтём, что всего рыбаков 3).

В программе решения задачи на школьном алгоритмическом языке используем следующие величины:

Коб – общее количество имеющихся (пойманных) рыб;

Кос – количество рыб, оставшееся тому или иному рыбаку;

взял – количество рыб, которые взял тот или иной рыбак;

i – номер рыбака;

можно_делить – величина логического типа, определяющая возможность дележа (взятия каждым рыбаком трети оставшегося количества рыб).

Соответствующая программа имеет вид:

```
алг Проверка
нач цел Коб, Кос, взял, i, лог можно_делить
  вывод нс, "Введите количество рыб"
  ввод Коб
  Кос := Коб          | Сначала так
  i := 1
  нц                  | Цикл "действий" каждого рыбака
    Кос := Кос - 1    | Осталось после выбрасывания
                      | одной рыбы
    если mod(Кос, 3) = 0
      то
        | i-й рыбак может взять треть
        | оставшихся рыб
        можно_делить := да
        взял := div(Кос, 3) | Берет i-й рыбак
        Кос := Кос - взял   | Оставшееся количество рыб
      иначе
        можно_делить := нет | При Коб рыбах
                             | дележ невозможен

  все
    i := i + 1        | Проснулся следующий рыбак (☺)
  кц при i > 3 или не можно_делить
  | В зависимости от значения величины можно_делить
  | выводим ответ
  вывод нс, "При таком количестве рыб дележ "
  если можно_делить
    то
      вывод "возможен"
    иначе
      вывод "невозможен"

  все
кон
```

Здесь *div* – функция, возвращающая результат целочисленного деления своего первого аргумента на второй, а *mod* – функция, рассчитывающая остаток (в других языках программирования для этого используются

не функции, а специальные операции).

После этого программа нахождения минимального количества рыб, удовлетворяющего условию задачи Дирака, может быть оформлена так:

```
алг Задача_Рыбаки_и_рыбки
нач цел Коб, Кос, взял, i, лог можно_делить
  Коб := 7 | Начальное значение диапазона поиска
  | Проверяем, можно ли разделить Коб рыб
  | (см. выше)
нц
  Кос := Коб
  i := 1
нц | Цикл "действий" каждого рыбака
  Кос := Кос - 1 | Осталось после выбрасывания
  ...
  i := i + 1 | Проснулся следующий рыбак (☺)
кц при i > 3 или не можно_делить
  Коб := Коб + 1 | Очередное значение
кц_при можно_делить
вывод нс, "Наименьшее количество рыб,"
вывод "удовлетворяющее условию задачи:", Коб
кон
```

Читаемость и понятность всей программы может быть существенно улучшена, если создать вспомогательную функцию *Можно_делить(Коб)* логи-

ческого типа, определяющую возможность дележа *Коб* рыб в соответствии с условием задачи. Её можно составить на основе программы, приведённой первой:

```
алг лог Можно_делить (арг цел Коб)
нач цел Кос, взял, i, лог можно_делить
  Кос := Коб
  i := 1
нц
  Кос := Кос - 1
  если mod(Кос, 3) = 0
  то
    можно_делить := да
```

```
        взял := div(Кос, 3)
        Кос := Кос - взял
    иначе
        можно_делить := нет
    все
    i := i + 1
кц при i > 3 или не можно_делить
|Значение функции:
знач := можно_делить
кон
```

Примечание.

В школьном алгоритмическом языке регистр букв в именах учитывается, поэтому в функции *Можно_делить* можно использо-

вать переменную величину логического типа *можно_делить*.

С использованием этой функции наша «главная» программа оформляется очень кратко:

```
алг Задача_Рыбаки_и_рыбки_Вариант_2
нач цел Коб
    Коб := 7 |Начальное значение диапазона поиска
    нц пока не Можно_делить(Коб)
        Коб := Коб + 1 |Очередное значение
    кц
    вывод нс, "Наименьшее количество рыб,"
    вывод "удовлетворяющее условию задачи:", Коб
кон
```

Есть и другой способ решения задачи. Можно идти не от общего количества пойманных рыб, а от числа рыб, доставшихся третьему рыбаку (то есть искать, так сказать, «с конца»). Если эту величину обозначить *взял3*, а количество рыб, оставшееся тому или иному ры-

баку, – соответственно *К3*, *К2* и *К1*, то, перебирая значения *взял3*, равные 1, 2, 3, ..., можно найти такое минимальное число, при котором значения величин *К2* и *К1* есть целые числа. Соответствующий вариант программы разработайте самостоятельно.

Другие задания для самостоятельной работы

1. Разработав программы (на языке программирования, который вы изучаете), найдите значение, о ко-

тором идёт речь в начале статьи, а также следующее большее число.

2. Разработайте аналогичные про-

граммы для случая, когда рыбаков 4, и также найдите соответствующие значения.

Дополнение

Опишем методику решения за-

дачи «Рыбаки и рыбки» средствами электронной таблицы на примере программы Microsoft Excel.

Оформим верхнюю часть листа следующим образом:

А	В	С	Д	Е	Ф	Г	Н	І	Ј
Число рыб	–1	Взял 1-й	Осталось	–1	Взял 2-й	Осталось	–1	Взял 3-й	Про- верка
7	6	2	4	3	1	2	1	0,333 333	

В столбце А запишем возможные значения общего числа пойманных рыб (допустим, что оно не превышает 300). Эти значения можно получить, используя, например, автозаполнение.

Формулы в ячейках В2:І2 достаточно очевидны. А как проверить тот или иной вариант (столбец Ј)? Можно рассуждать так. Данный вариант исходного количества рыб подходит, если в столбце І будет записано целое

число. Это условие можно формализовать следующим образом:

$I2=ЦЕЛОЕ(I2)$

Если оно соблюдается, то выведем в ячейке Ј2 соответствующее количество рыб (из ячейки А2), в противном случае – «пустое» значение («»).

Вся формула в ячейке Ј2 имеет вид:
 $=ЕСЛИ(I2=ЦЕЛОЕ(I2);A2;"")$

Все формулы диапазона В2:Ј2 можно распространить (скопировать) на другие строки:

	А	В	С	Д	Е	Ф	Г	Н	І	Ј
1	Число рыб	–1	Взял 1-й	Оста- лось	–1	Взял 2-й	Оста- лось	–1	Взял 3-й	Проверка
2	7	6	2	4	3	1	2	1	0,333333	
...										
295	300	299	99,67	199,33	198,33	66,11	132,22	131,22	43,74	
296	От- вет:									

После этого искомое количество в ячейке В296 может быть получено как минимальное числовое значение в диа-

пазоне Ј2:Ј295 (с использованием функции МИН):

$=МИН(Ј2:Ј295)$

Задания для самостоятельной работы

1. Подготовив лист электронной таблицы, найдите значение, о котором идёт речь в начале статьи, а также следующее большее число.

2. Подготовьте лист для случая, когда рыбаков 4, и также найдите соответствующие значения.