

# Исполнитель Перевозчик. Программа

Вот старинная задача под названием «Волк, коза и капуста»:

На левом берегу реки стоит крестьянин с лодкой, а рядом с ним — волк, коза и капуста. Крестьянин должен переправиться сам и перевезти волка, козу и капусту на правый берег. Однако в лодку, кроме крестьянина, помещается либо только волк, либо только коза, либо только капуста. Оставлять же волка с козой или козу с капустой без присмотра нельзя: волк может съесть козу, а коза — капусту. Как должен действовать крестьянин?

Давай подумаем, кого крестьянину перевезти первым. Ясно, что нельзя брать волка — тогда коза останется с капустой и съест её. По той же причине нельзя брать и капусту. Но можно забрать козу — волки не едят капусту. После того как крестьянин отвезёт козу на правый берег, он может переправиться только один в лодке обратно — везти обратно козу бессмысленно.

Чтобы описать действия крестьянина, придумаем специальный исполнитель *Перевозчик*. Пусть по команде переправь крестьянина *Перевозчик* переправляет на другой берег одного крестьянина в лодке (без крестьянина лодка не переправляется). По команде перевези волка *Перевозчик* переправляет на другой берег волка (и конечно, крестьянина). По команде



перевези козу *Перевозчик* переправляет на другой берег козу (и крестьянина). По команде перевези капусту *Перевозчик* переправляет на другой берег капусту (и крестьянина).

По условию задачи в лодке возможно переправить, помимо крестьянина, только одного «пассажира», значит, такие команды, как, например, перевези волка и капусту вместе, нам для решения задачи не понадобятся. В задаче нет других участников, кроме крестьянина, волка, козы и капусты. Поэтому такие команды, как, например, перевези собаку, нам тоже не пригодятся.

Мы придумали исполнителя *Перевозчик* с системой команд, изображённой справа:

Запишем решение нашей задачи в виде последовательности команд для исполнителя *Перевозчик*. Первые команды этой последовательности мы уже определили:

перевези козу  
переправь крестьянина

После выполнения этих двух команд *Перевозчик* окажется опять на левом берегу вместе с волком и капустой, а коза останется на правом берегу. Как *Перевозчик* должен поступить дальше? Какую команду мы ему должны дать? Если он перевозит волка и возвращается на левый берег, то волк и коза останутся на правом берегу, и коза будет съедена волком. А если он перевозит капусту и возвращается назад один, то коза съест капусту. Решение неожиданно: давайте дадим команду *Перевозчику* перевезти волка, а на обратном пути захватить с собой козу. Какие команды дать *Перевозчику* дальше, становится очевидным. Получается, что для решения задачи нужно дать *Перевозчику* такую последовательность команд:

**Система команд  
исполнителя  
*Перевозчик***

переправь крестьянина  
перевези волка  
перевези козу  
перевези капусту

перевези козу  
переправь крестьянина  
перевези волка  
перевези козу  
перевези капусту  
переправь крестьянина  
перевези козу



Договоримся называть последовательность команд программой.

**150**

Реши задачу «Волк, коза и капуста» по-другому: составь другую программу (другую последовательность команд) для исполнителя *Перевозчик*. Первые две команды этой программы будут такими же, как в первой программе. Какой может быть третья команда?

**151**

Реши задачу.

Два солдата подошли к реке и увидели двух мальчиков, которые катались на лодке и причалили к тому же берегу. Как солдатам переправиться на другой берег, если лодка вмещает либо только одного солдата, либо двух мальчиков, а солдата и мальчика уже не вмещает?

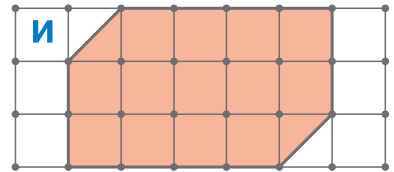
Сможет ли справиться с этой задачей исполнитель *Перевозчик*? Если да, запиши программу для исполнителя *Перевозчик* для решения этой задачи. Если нет, то придумай такой исполнитель *Другой Перевозчик*, который сможет справиться с этой задачей. Выпиши систему команд этого исполнителя, напиши программу для решения задачи.

**152**

Реши задачу 151, если солдат будет больше: а) 5 солдат; б) 7 солдат, в) 10 солдат. Для решения каждой задачи напиши программу для исполнителя *Другой Перевозчик*.

**153**

Найди площадь многоугольника И.

**154**

Из множества Щ выдели подмножество всех латинских букв. Запиши буквы этого подмножества в алфавитном порядке.

Й	Ж	W	Ю	Д	6	S	Ъ	G	Z	V	Щ
1	У	Г	Р	U	П	Л	З	Ј	Щ	Ф	
Ц	7	Ь	Ш	Ы	Ё	Б	F	L	Э	N	
Q	Я	4	У	І	5	D	И	2	8	9	

**155**

Придумай какого хочешь своего исполнителя, запиши его систему команд. Придумай для этого исполнителя две разные задачи и напиши программы для их решения.

**156**

Перепиши в тетрадь программу, которая у тебя получилась после решения задачи 5 из урока *Водолея* на компьютере.

Сравни свою программу с программами других учеников в классе. Скорее всего, программы будут разными: для решения одной и той же задачи можно составить разные цепочки команд, разные программы для *Водолея*. Рассмотрите все получившиеся программы в вашем классе. Найди длину каждой программы, выясни, сколько мер воды было использовано (налито в сосуды по команде *наполни...*) в процессе выполнения каждой программы. У кого получилась самая короткая программа? У кого в процессе выполнения программы было истрачено меньше всего воды?

## Исполнитель Удвоитель

Исполнитель *Удвоитель* имеет две кнопки и экран. На экране отображается число. В момент включения *Удвоителя* число на экране равно 0. На кнопках написано «умножь на 2» и «прибавь 1».

При нажатии на кнопку «умножь на 2» число на экране удваивается — вместо него появляется число, которое вдвое больше (например, если на экране было число 3, то после нажатия этой кнопки на экране будет число 6).

При нажатии на кнопку «прибавь 1» число на экране увеличивается на 1 — вместо него на экране появляется следующее число (например, если на экране было число 56, то после нажатия этой кнопки будет отображаться число 57).

Справа приведён пример программы для исполнителя *Удвоитель*. После исполнения этой программы на экране будет отображаться число 11.

**Система команд исполнителя Удвоитель**

умножь на 2  
прибавь 1

прибавь 1  
прибавь 1  
умножь на 2  
прибавь 1  
умножь на 2  
прибавь 1