

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ ВЛОЖЕННЫЕ ВЕТВЛЕНИЯ

На прошлых занятиях нами были рассмотрены подходы к программной реализации циклов, ветвлений и функций. Они являются ядром любого языка программирования и лежат в основе построения алгоритмов. Данные программные конструкции отличаются в различных языках только синтаксисом - правилами написания.

Умение реализовывать алгоритмы программно - одна из важнейших технологий, освоенных человечеством. Ведь с освоением этой технологии человек переложил труд по обработке информации с себя на вычислительную машину. Использование компьютера привело к фундаментальным изменениям в жизни человека и человечества.

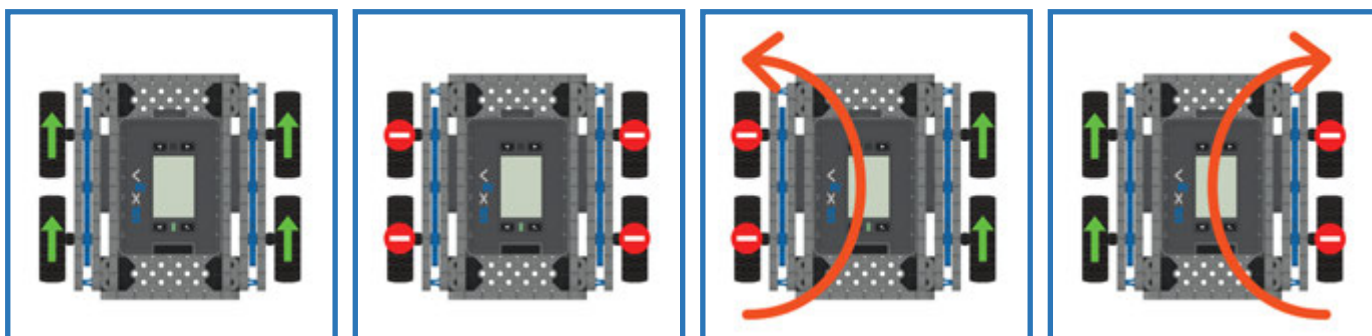
Мы уже умеем создавать программы, где используются ветвления. Но при помощи одного ветвления можно реализовать только два состояния. Напомним, что ветвление связано с вопросом, на который есть только два однозначных ответа: **true** и **false** (истина и ложь). С каждым из ответов можно связать какое-либо действие.

Очень важно научиться грамотно формулировать такие вопросы. Например, на вопрос «какая сегодня погода?» невозможно ответить «да» или «нет». Но в зависимости от ситуации этот вопрос можно переформулировать. Если необходимо принять решение, брать ли зонт, то вопрос следует переформулировать на «идет дождь?». На этот вопрос можно ответить либо да (**true**) либо нет (**false**) и связать с ответом какое-либо

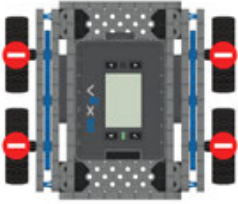
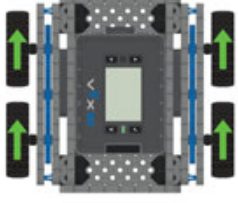
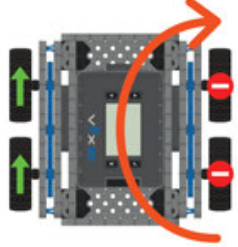
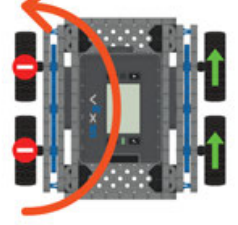
Но что делать, если вопрос и связанные с ним решения предполагают четыре состояния, а не два?

решение.

Например, мы хотим, чтобы наш робот умел откликаться на пульт таким образом, чтобы были осуществлены четыре состояния: движение вперед, стояние на месте, разворот вокруг своей оси против часовой стрелки и разворот вокруг своей оси по часовой стрелке.



Очевидно, что, для того чтобы описать работу одной пары колес, например левых, хватит одного вопроса: «Колеса должны вращаться?» Для правых колес справедлив тот же принцип. Получается, что для описания четырех состояний достаточно двух вопросов.

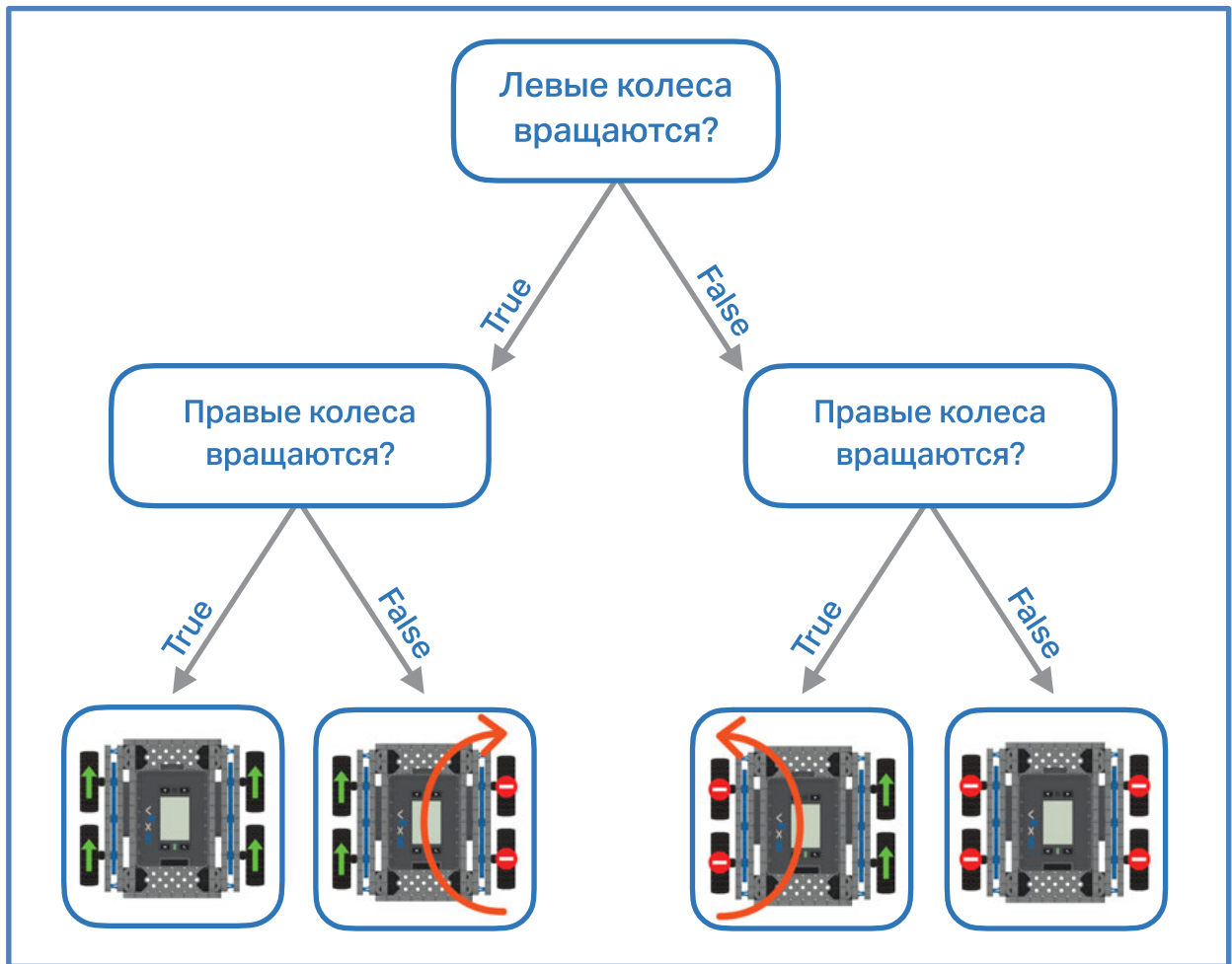
Левое колесо должно вращаться?	Правое колесо должно вращаться?	Рисунок
Нет	Нет	
Да	Да	
Да	Нет	
Нет	Да	

Обратите внимание, что в первом и втором столбцах таблицы перечислены все возможные комбинации ответов на вопросы:

- нет нет;
- да да;
- да нет;
- нет да;

Других комбинаций просто не может быть!

Важно понимать, что в нашем случае для любого состояния задается всего два вопроса: сначала про левый двигатель, затем, какой бы ответ ни был получен, - про правый. Или сначала про правый, а затем про левый. Последовательность вопросов может быть любой, например:



Настало время реализовать программно управление роботом с пульта дистанционного управления для четырех вышеописанных состояний. Будем управлять роботом двумя верхними курками.

В нашей программе тело бесконечного цикла начинается в 13 и заканчивается в 36 строке.

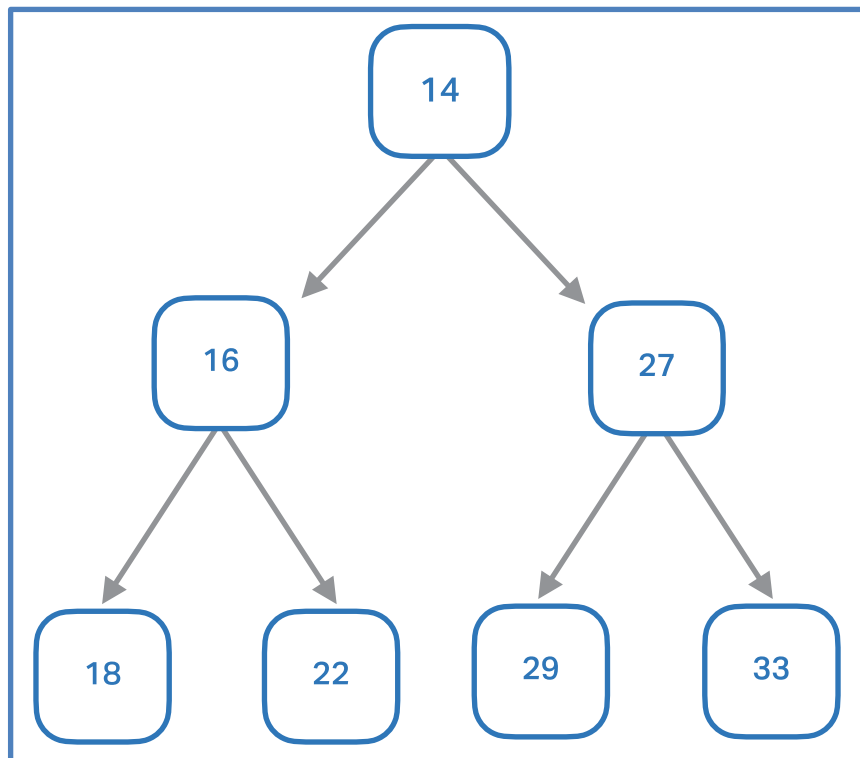
```

10 task main()
11 {
12     while (1)
13     {
14         if (getJoystickValue (BtnLUp))
15         {
16             if (getJoystickValue (BtnRUp))
17             {
18                 move (100,100,1);
19             }
20             else
21             {
22                 move (100,0,1);
23             }
24         }
25         else
26         {
27             if (getJoystickValue (BtnRUp))
28             {
29                 move (0,100,1);
30             }
31             else
32             {
33                 move (0,0,1);
34             }
35         }
36     }
37 }
  
```



В 14 строке находится ветвление: в зависимости от того, какое значение примет аргумент `if (getJoystickValue (BtnLUp))`, программа перейдет либо к 16, либо к 27 строке. В этих строках объявляется одно и то же вложенное ветвление. Важно понимать, что за одну итерацию выполнятся либо строки кода с 16 по 24, либо с 25 по 34. То есть ветвление, связанное с `BtnRUp`, будет задействовано только один раз.

Выполнение программы иллюстрируется деревом ниже. Числа в узлах дерева - это номер выполняемой строки кода:



Итак, любую систему можно рассматривать как находящуюся в каком-либо состоянии из множества возможных. Этим состояниям может быть в разы или миллионы раз больше, чем два состояния (рассмотренные нами четыре состояния - такой же частный случай). В случае, если эти состояния можно свести к серии вопросов с ответами да-нет (`true-false`), описание системы и влияние на ее свойства и поведение можно возложить на вычислительную машину, которая с каждым из ответов свяжет какое-либо действие.