Программирование в среде RobotC

Занятие 10: Структурное программирование

Компьютерная программа

- На следующем занятии мы перейдем к следующему этапу программирования – освоению структур. Сегодня поговорим о некоторых необходимых для дальнейшего понятиях и познакомимся с еще одной функцией языка Си.
- Что представляет собой компьютерная программа? Предположим, нам надо пройти по лабиринту. Как мы будем поступать? Например, мы можем использовать правило правой руки, то есть мы делаем шаг прямо, смотрим направо, если справа стенка, идём дальше на шаг. Если справа проём, поворачиваем направо и идём на шаг прямо. И так далее.
- Описание таких действий – это алгоритм решения задачи. Мы являемся исполнителем, а, если всё это мы запишем на языке программирования, получим компьютерную программу.

Компьютерная программа

- *Компьютерная программа* – алгоритм решения какой-либо задачи, записанный на языке программирования.
- *Алгоритм* – точное описание порядка действий, которые должен выполнить исполнитель для того, чтобы решить задачу.
- *Исполнитель* – человек или некоторое устройство, которое может понимать и выполнять определённый набор команд.
- *Система команд исполнителя* – набор команд, которые понимает и умеет выполнять исполнитель.

Парадигмы программирования

- Рассмотрим еще один пример. Предположим, я хочу из точки А попасть в точку В. Какие есть для этого варианты? Например:

  – дойти пешком;

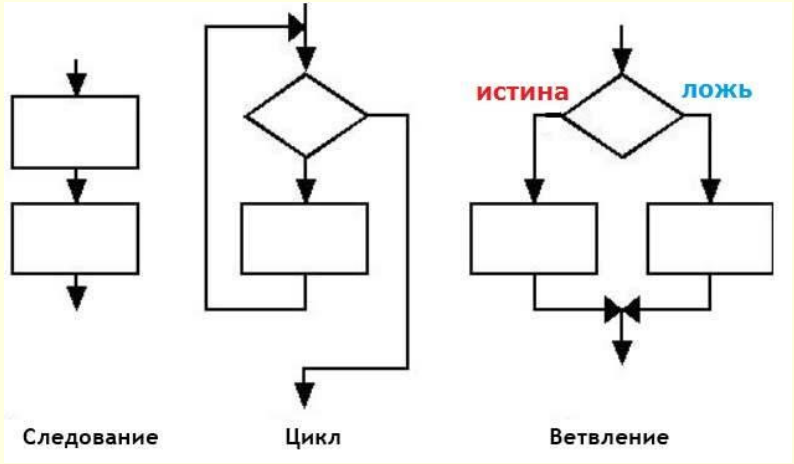
  – доскакать на лошади;

  – домчаться на мотоцикле;

  – докатиться на электрическом самокате.
- Задача одна, а вариантов достижения много. Между этими вариантами есть существенные различия. Они заключаются в устройстве процесса передвижения. В первом случае, мы просто используем силу своих мышц, во втором – силу мышц более сильного животного, в третьем – энергию двигателя внутреннего сгорания, в четвёртом – электрическую энергию. Минимум четыре различных подхода (парадигмы).
- В программировании тоже есть различные подходы к устройству программ. Эти подходы называются парадигмами программирования. Мы начнём с рассмотрения одного из таких подходов – структурного программирования.

Структурное программирование

- Идея в следующем. Любой алгоритм можно представить в виде всего трёх возможных структур:
  - последовательность;
  - ветвление;
  - цикл.
- И оказывается, этого будет достаточно, чтобы написать программу любой сложности.



Структурное программирование

- **Следование.**

  С этой конструкцией вы уже хорошо знакомы. Это самая простая структура. Все программы, которые вы писали ранее, использовали эту структуру. В ней команды выполняются друг за другом. Следующая команда выполняется после того, как завершится предыдущая.

  Команда 1;

  Команда 2;

  Команда 3;

Структурное программирование

Цикл.

- Вспомните задачу о лабиринте. Что мы будем делать, если до выхода из лабиринта надо сделать 1000 ходов? Писать 1000 раз команду вперёд или направо? Вот для подобных случаев и используется структура Цикл. Данная конструкция нужна для того, чтобы выполнять команды несколько раз.
- В самом простом случаем мы заранее задаём количество раз, которое должны выполнится команды.

  ПОВТОРЯЙ 1000 РАЗ Команда 1;
- Но есть и более продвинутый вид цикла. Которые выполняется пока не выполнится некоторое условие.

  ПОВТОРЯЙ Команда 1;

  ПОКА Условие1

Структурное программирование

- **Ветвление.**
- Мы сделали шаг по лабиринту и дальше нам надо принять решение в зависимости от того, есть ли справа стенка. Здесь нам и придёт на помощь структура «ветвление».

  ЕСЛИ условие ТО Команда 1;

  ИНАЧЕ Команда 2;
- Если условие выполнено, то выполнится Команда1, иначе (если условие не выполнено) выполнится Команда2.
- Как вы наверное уже поняли, в следующих уроках вам предстоит разобраться с тем, как данные структуры управления реализованы в языке Си.

Генерация случайных чисел в языке Си

- Сейчас мы познакомимся с еще одной функцией языка Си – командой генерирования случайных чисел.
- Начнем с примера. Нам надо составить распределить 16 футбольных команд по четырём группам. Если сделать это вручную, то могут обвинить в предвзятости. Поэтому мы занумеруем все команды и напишем программу, которая выдает случайное число от 1 до 16. Запустив четыре раза программу, мы получим первую группу и т.д.
- Как получить случайное число. Для этого нам понадобиться команда

  random(n);

  Данная функция возвращает случайное целое число в диапазоне от нуля до числа n включительно. При этом n должно быть в диапозоне от 0 до 32 767.

Генерация случайных чисел в языке Си

- Давайте посмотрим на эту функцию в действии. Запустим следующий код:

```
task main(){
  int a;
 a=random(90);
  nxtDisplayTextLine(1,”%d”,a);
}
```

На экране NXT появится какое-то число от 0 до 90.

Генерация случайных чисел в языке Си

- **Ограничение числа снизу.**
- Функция random возвращает случайные числа из отрезка [0, n]. А что если нам нужны только числа большие числа m. Как быть? Просто прибавим к тому, что вернула функция random, наше значение m. Тогда если функция вернёт 0, итоговый ответ будет m, если 2439, то итоговый ответ будет m + 2439. Этим действием мы как бы сдвигаем все числа на m единиц вперёд.

```
task main(){
 int m=50, a;
 a=random(100);
 nxtDisplayTextLine(1,”%d”,m+a);
}
```

Генерация случайных чисел в языке Си

- **Ограничение числа сверху и снизу.**
- Если мы запустим предыдущую программу, то будем получать числа от 50 до 150. Подумайте, почему.
- Если нам нужны числа от 50 до 80? Тогда нам надо получить случайное число из промежутка от 50 до 80 и прибавить к 50.
- В общем случае если нам нужно получить числа из отрезка [a;b], то необходимо воспользоваться следующей конструкцией:
  a + random(b-a).
- Программа будет выглядеть так:

```
task main(){
int a=50, b=80;
nxtDisplayTextLine(1,”%d”,a+random(b-a));
}
```

Задача 1.

- Введите максимальное число, которое может быть сгенерировано следующей конструкцией:
- int a = 55 + random(32767)%501;

Задача 2.

- Введите минимальное число, которое может быть сгенерировано следующей конструкцией:
- int a = 10 + random(32767);

Задача 3.

- Вычислите нижнюю и верхнюю границу диапазона, сгенерированного следующей конструкцией:
- int a = 100 + random(32767)%100;

Задача 4.

- Вычислите нижнюю и верхнюю границу диапазона, сгенерированного следующей конструкцией:
- int a = -50 + random(32767)%101;

Задача 5.

- Поле для игры в рулетку состоит из ячеек от 0 до N. Число N получается случайным выбором из промежутка [1, 333].
- Написать программу, которая выдаёт случайное число от нуля до N.

Задача 6.

- Вначале программы задаются два случайных числа: число p от 100 до 200 и число q от 200 до 300. Написать программу, которая выводит одно случайное число из промежутка [p,q].

Ответы к задачам

1. 555
2. 10
3. 100, 199
4. -50, 50
5. 324
6.
```
task main(){
  int n, a;
  n=1+random(333);
  nxtDisplayTextLine(1,”%d”,random(n));
 }
```
7.
```
task main(){
  int p, q;
  p=100+random(100);
  q=200+random(100);
  nxtDisplayTextLine(1,”%d”,p+random(q-p));
 }
```

Завершение занятия

Теперь мы готовы перейти к написанию программ более высокого уровня сложности.

На этом занятие завершено.