

МОУ «Средняя общеобразовательная школа р.п. Пинеровка Балашовского района»  
Школьный центр математики и информатики

# Формальный исполнитель Роби

Исследовательская работа

Выполнил  
Рузиев Владислав,  
ученик 8 «Б» класса

Учитель-консультант  
Пичугин Виталий Владимирович

2016-2017 учебный год

## План

Введение

I. Исполнение алгоритмов

II. Построение задач и тестов проверки верного решения

Заключение

## Введение

Цель работы изучить систему команд формального исполнителя Роби, изучить возможность построения линейных алгоритмов, разветвляющихся и циклических алгоритмов, изучить правила построения и вызова вспомогательных алгоритмов (процедур).

Важно изучить правила и назначение тестирования алгоритмов.

## I. Исполнение алгоритмов

Каждый алгоритм строится в расчете на некоторого исполнителя. Для того, чтобы исполнитель мог решить задачу по заданной последовательности команд, необходимо, чтобы он был в состоянии выполнить каждое действие, предписываемое этой последовательностью команд.

Основными характеристиками исполнителя являются: среда, элементарные действия, система команд, отказы.

Исполнители бывают формальные (бездумные) и неформальные.

Формальный исполнитель имеет жестко заданный список выполняемых команд и при вызове каждой из них непременно совершает элементарное действие. Если исполнение невозможно, возникает «отказ». Примером формального исполнителя является Пылесосик Роби.

Рассмотрим несколько задач для исполнителя Роби.

Задача 1. Дано



Z1.pic

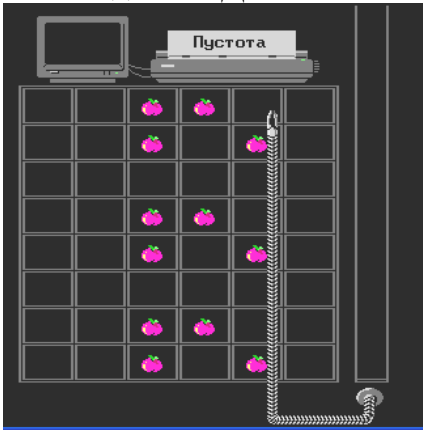
Надо



Z1.rob

Линейный алгоритм. Алгоритм работает для любого набора предметов в четырех отсеках.

### Задача 2. Дано

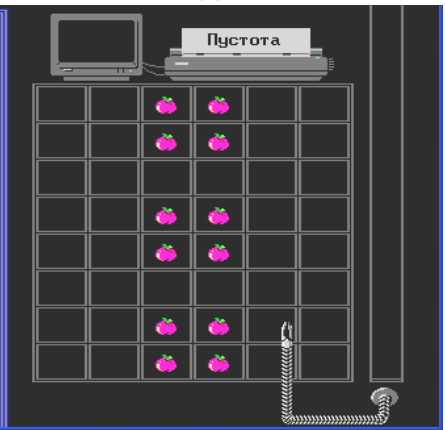


Z2.pic

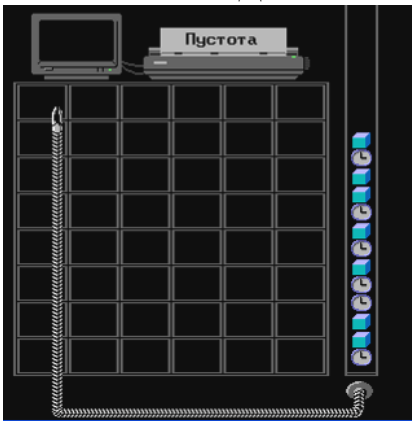
```
сделай namesto
вниз
вниз
сделай namesto
вниз
вниз
сделай namesto
вверх
процедура namesto
вниз
возьми
влево
положи
вправо
конец процедуры
```

Z2.rob

### Надо



### Задача 3. Дано

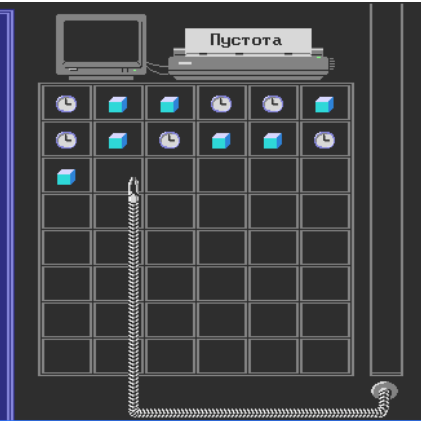


Z3.pic

```
пока карман не пуст, повторять
положи
если можно вправо, то
вправо
иначе
сделай назад
конец ветвления
конец цикла
процедура назад
вниз
пока можно влево, повторять
влево
конец цикла
если можно влево, то
влево
конец ветвления
конец процедуры
```

Z3.rob

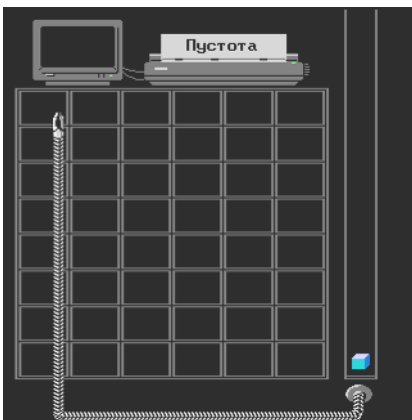
### Надо



Алгоритм включает цикл, ветвление, вспомогательные процедуры.

Работоспособность алгоритма проверяется набором самостоятельно подготовленных тестов.

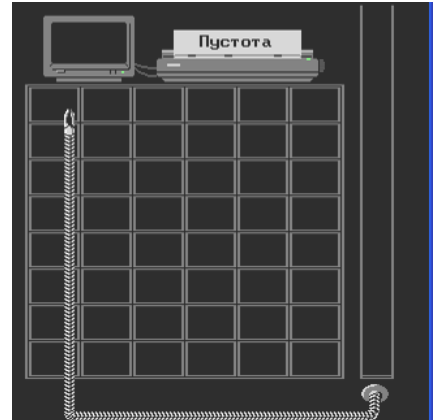
### Тесты



Z31.pic

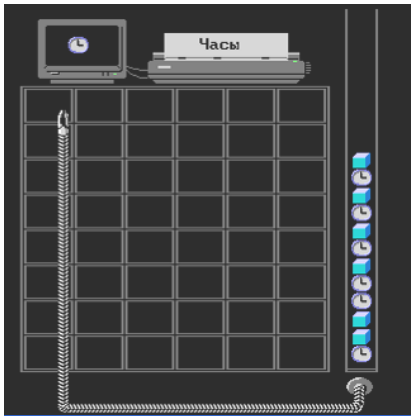


Z32.pic



Z33.pic

## Задача 4. Дано



Z4.pic

## Надо



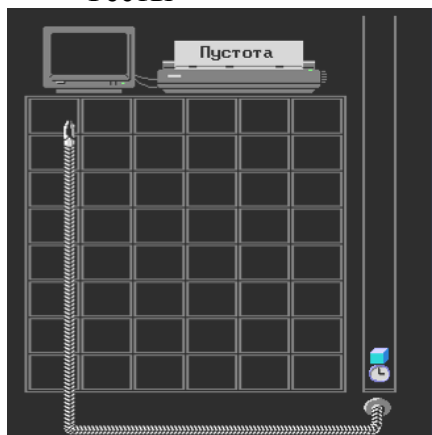
Z4.rob

Алгоритм включает цикл, ветвление, вспомогательные процедуры.

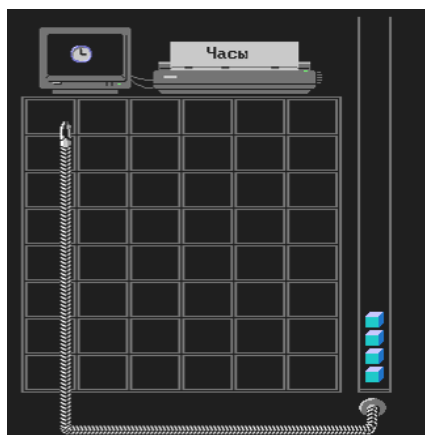
```
пока карман не пуст, повторять
положи
если предмет в памяти, то
сделай niz
конец ветвления
если можно вправо, то
вправо
конец ветвления
конец цикла
    процедура niz
    возьми
    пока можно вниз, повторять
    вниз
    конец цикла
    пока можно влево, повторять
    влево
    конец цикла
    пока отсек не пуст, повторять
    вправо
    конец цикла
    положи
    пока можно вверх, повторять
    вверх
    конец цикла
    пока можно влево, повторять
    влево
    конец цикла
    пока отсек не пуст, повторять
    вправо
    конец цикла
конец процедуры
```

Работоспособность алгоритма проверяется набором самостоятельно подготовленных тестов.

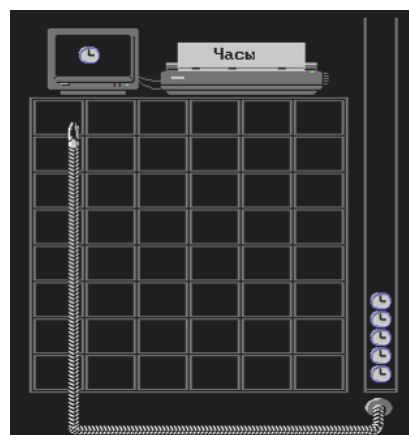
## Тесты



Z41.pic



Z42.pic

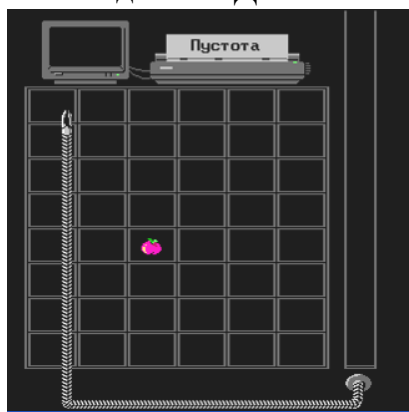


Z43.pic

## II. Построение задач и тестов проверки верного решения

На этом этапе пришлось самостоятельно составить задачу, решить её, продумать комплект проверочных тестов (учитывая возможные «критические» и граничные условия).

### Задача 5. Дано



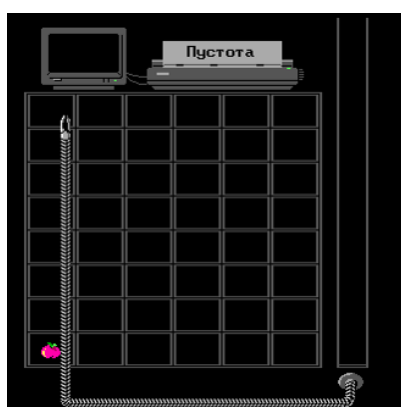
Z5.pic

### Надо

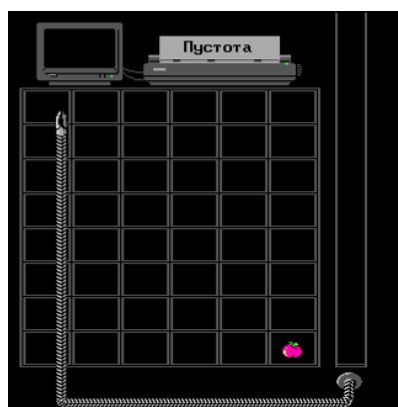


Z5.rob

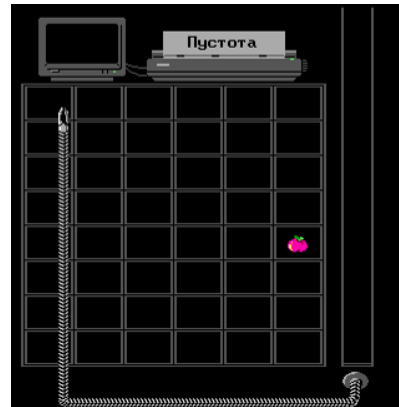
### Тесты



Z51.pic



Z52.pic



Z53.pic

## Заключение

Я работал над проектом 4 месяца. За это время я многому научился в программировании, осознал роль и назначение проверочных тестов. Мне очень понравилась исследовательская работа.