



# Исполнители алгоритмов

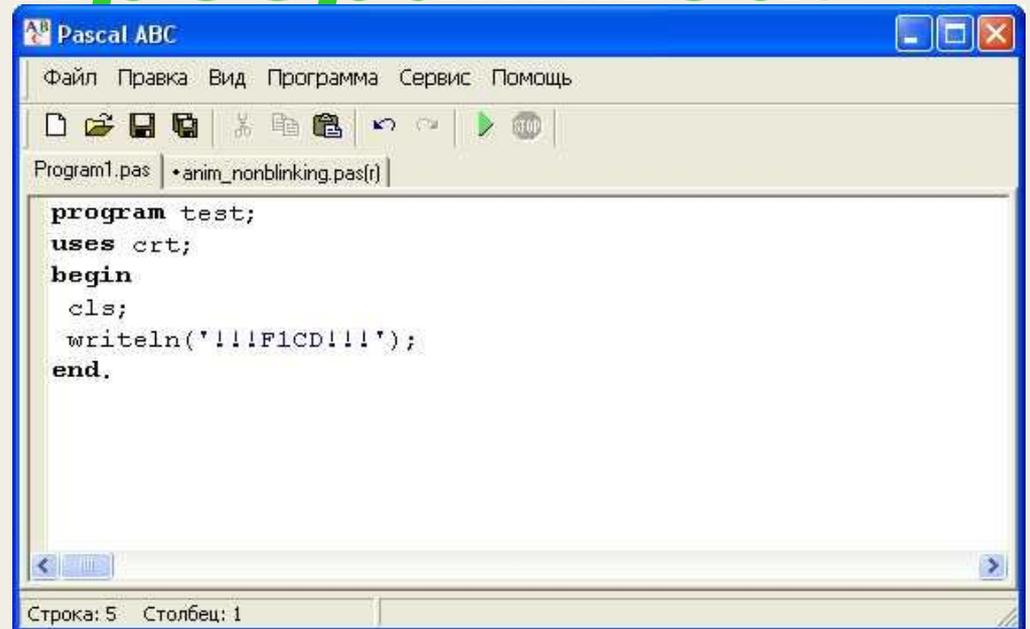
**Исполнитель алгоритма** - некоторая абстрактная или реальная система, способная выполнить действия, предписываемые алгоритмом.

Обычно исполнитель ничего не знает о цели алгоритма. Он выполняет все полученные данные

В информатике универсальным исполнителем алгоритмов является

***Компьютер.***

**Алгоритм, записанный на «понятном» компьютеру языке программирования, называется программой.**



The image shows a screenshot of a Pascal ABC IDE window. The window title is "Pascal ABC". The menu bar includes "Файл", "Правка", "Вид", "Программа", "Сервис", and "Помощь". The toolbar contains icons for file operations (new, open, save, print, copy, paste, undo, redo) and execution (run, stop). The main text area contains the following Pascal code:

```
program test;  
uses crt;  
begin  
  cls;  
  writeln('!!!F1CD!!!');  
end.
```

The status bar at the bottom indicates "Строка: 5" and "Столбец: 1".



**Язык Лого** был создан в **1967** году в  
Лаборатории Искусственного Интеллекта  
Массачусетского технологического  
института, профессором **Сеймуром**  
**Папертом**

*Исполнитель команд - черепашка*

Исполнителя характеризуют:

Сред  
Система  
Электронные  
ДЕЙСТВИЯ  
Ы

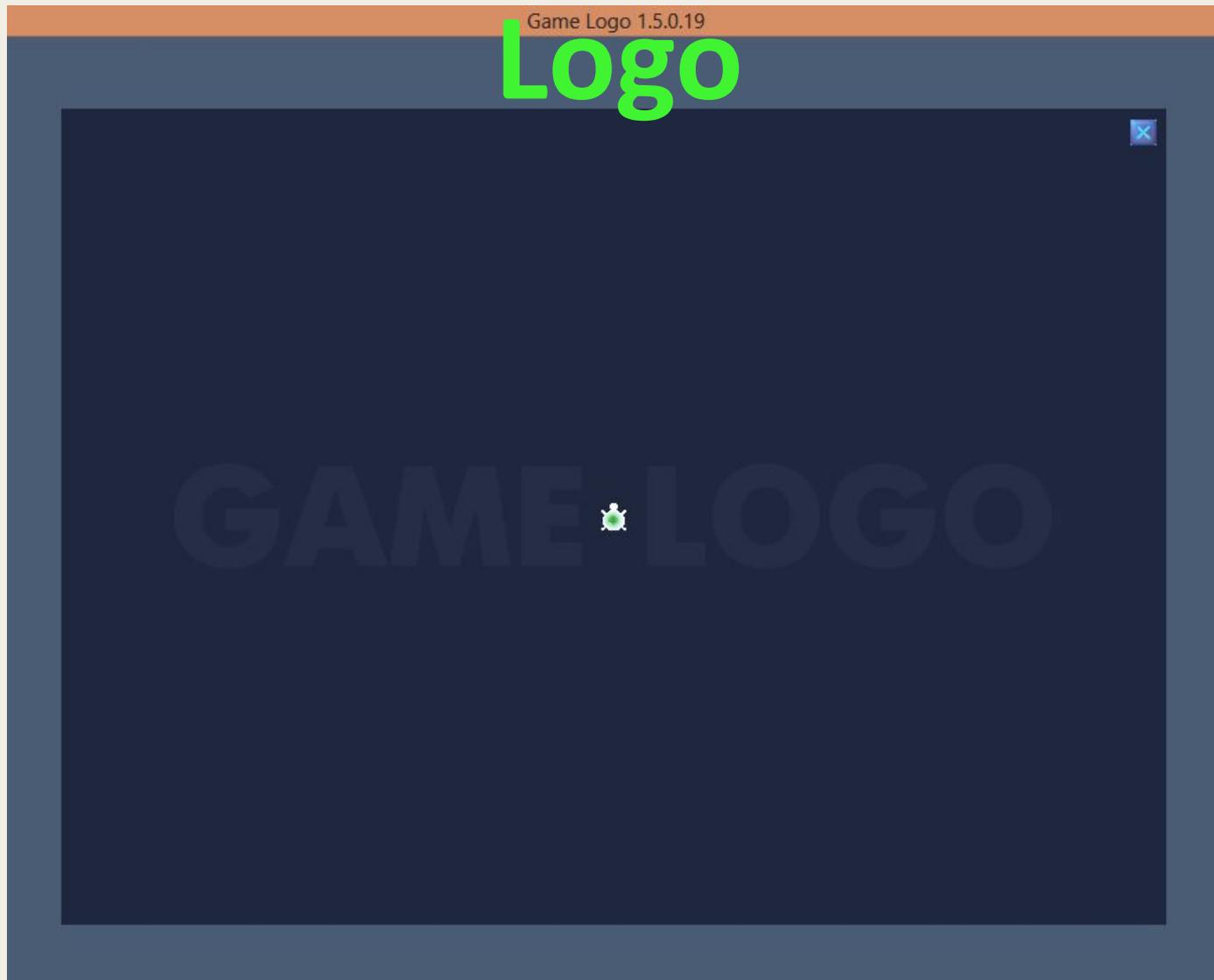
# Сред

**это "место обитания" исполнителя.**

Например, для исполнителя Logo, среда — это  
бесконечное клеточное поле.

Стены и покрашенные клетки тоже часть среды. А их  
расположение и положение самого робота задают  
конкретное состояние среды.

# Среда исполнителя

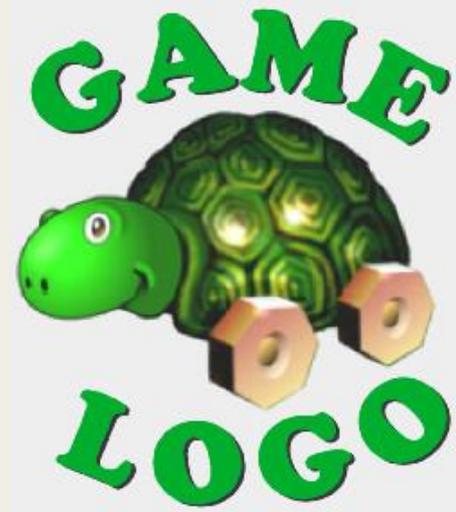


# Система

## команд

**Каждый исполнитель может выполнять команды только из некоторого строго заданного списка — системы команд исполнителя.**

Для каждой команды должны быть заданы условия применимости (в каких состояниях среды может быть выполнена команда) и описаны результаты выполнения команды.



# Команды

**ВПЕРЕД** \_\_\_\_\_ (количество шагов)

**НАЗАД** \_\_\_\_\_ (количество шагов)

**НАЛЕВО** \_\_\_\_\_ (градус)

**НАПРАВО** \_\_\_\_\_ (градус)



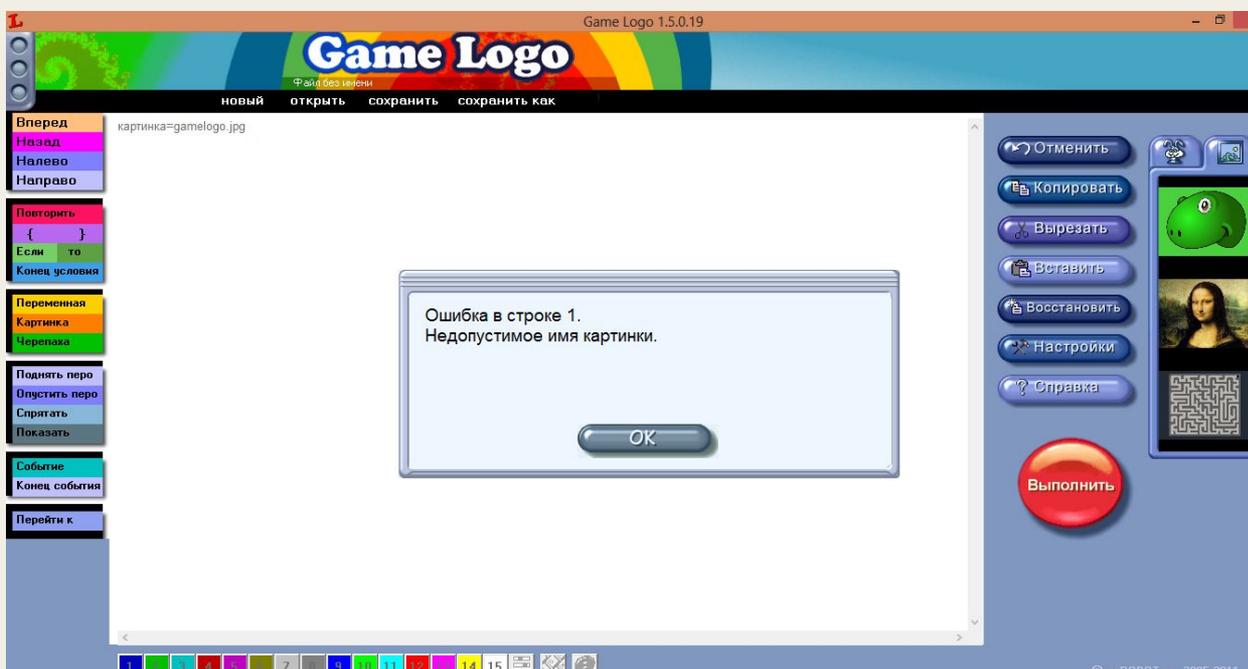
# Элементарные действия

после вызова команды исполнитель  
совершает элементарное действие

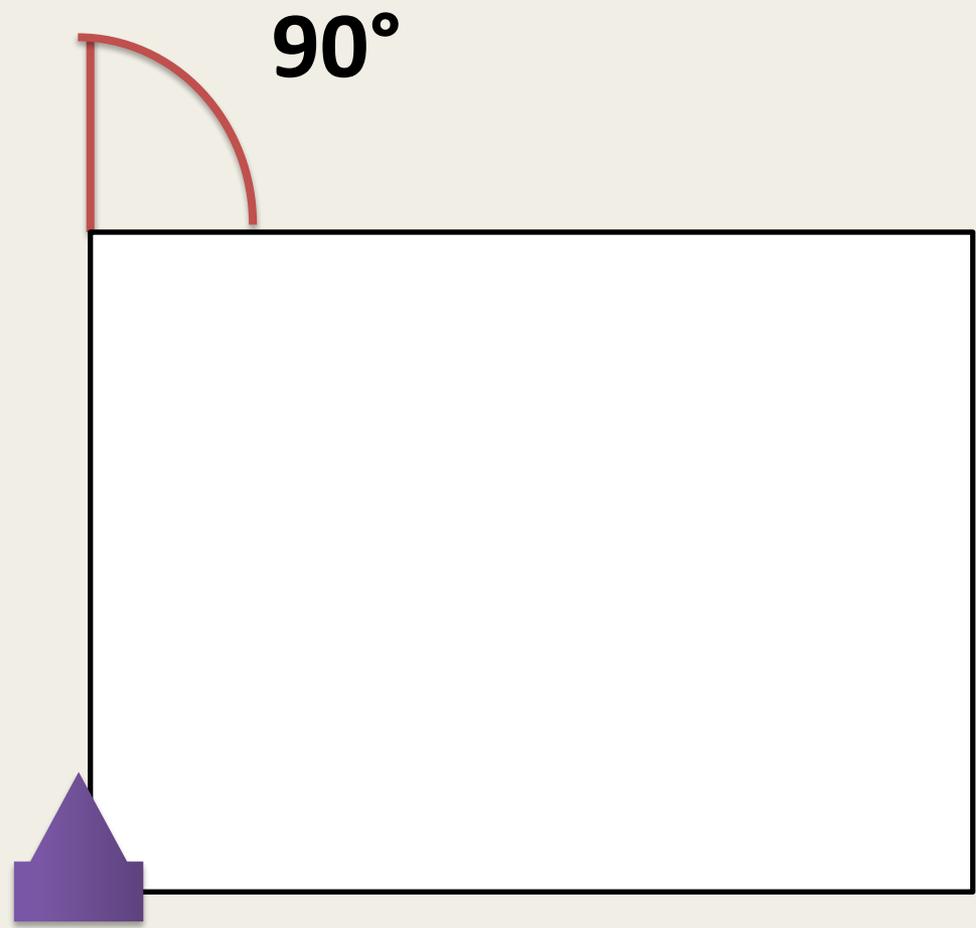


# Отказ

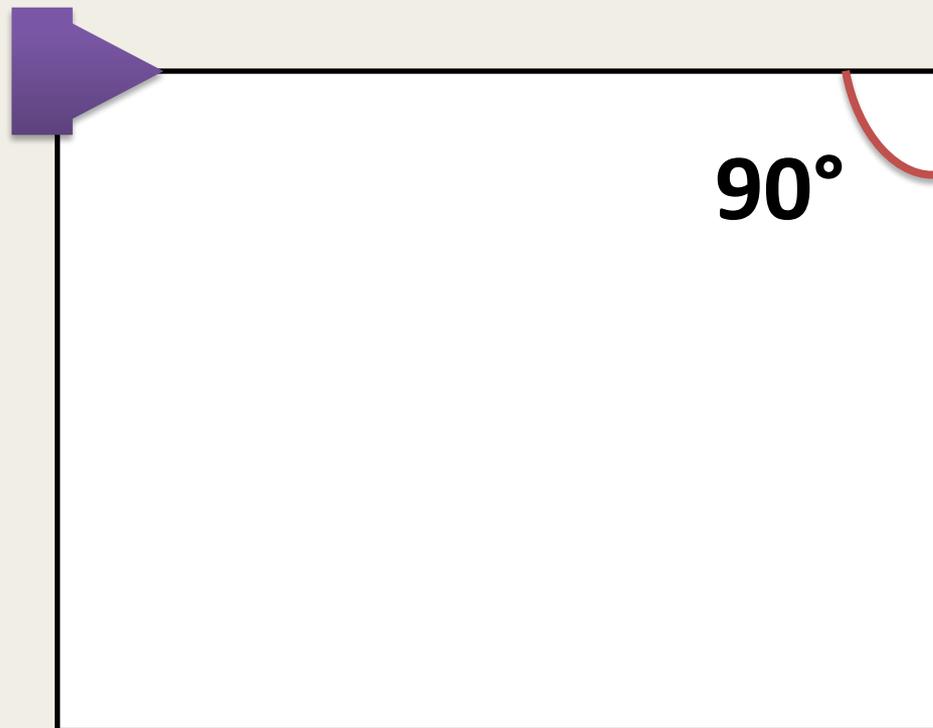
Возникают при вызове команды в недопустимом для данной команды состоянии среды.



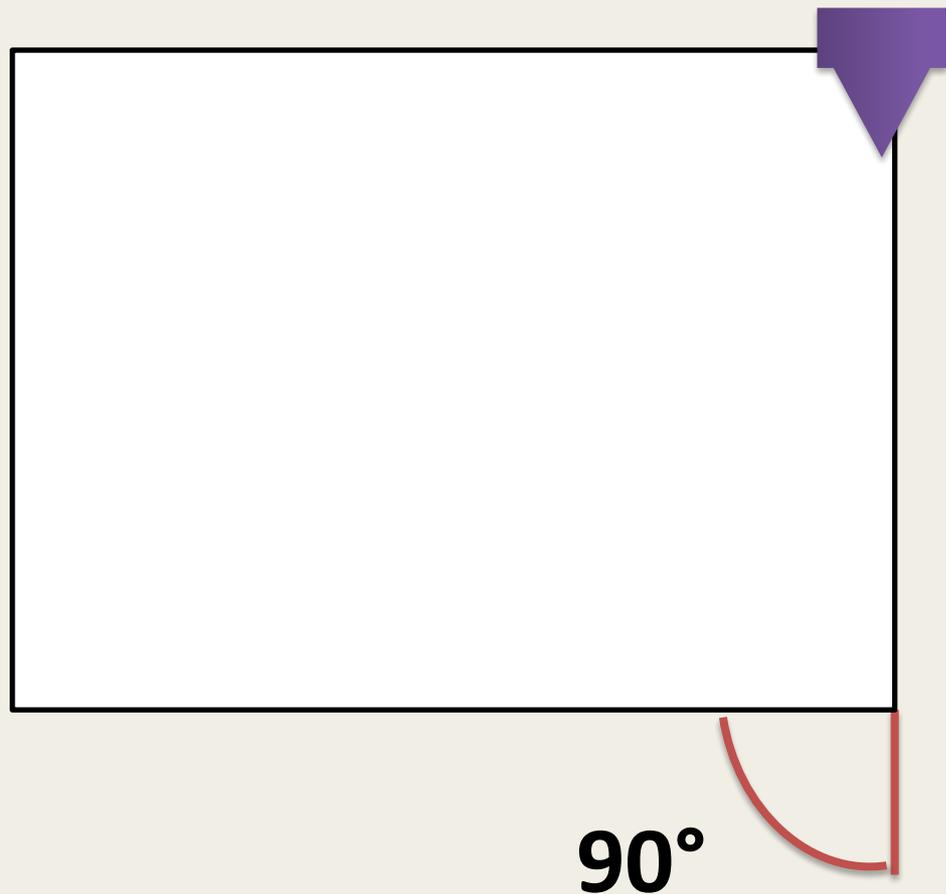
**вперед 100**  
**направо 90**



**вперед 100**  
**направо 90**  
**вперед**  
**направо**  
**90**



**вперед 100**  
**направо 90**  
**вперед**  
**100**  
**направо**  
**90**  
**вперед**  
**100**  
**направо**  
**90**



**вперед 100**

**направо 90**

**вперед**

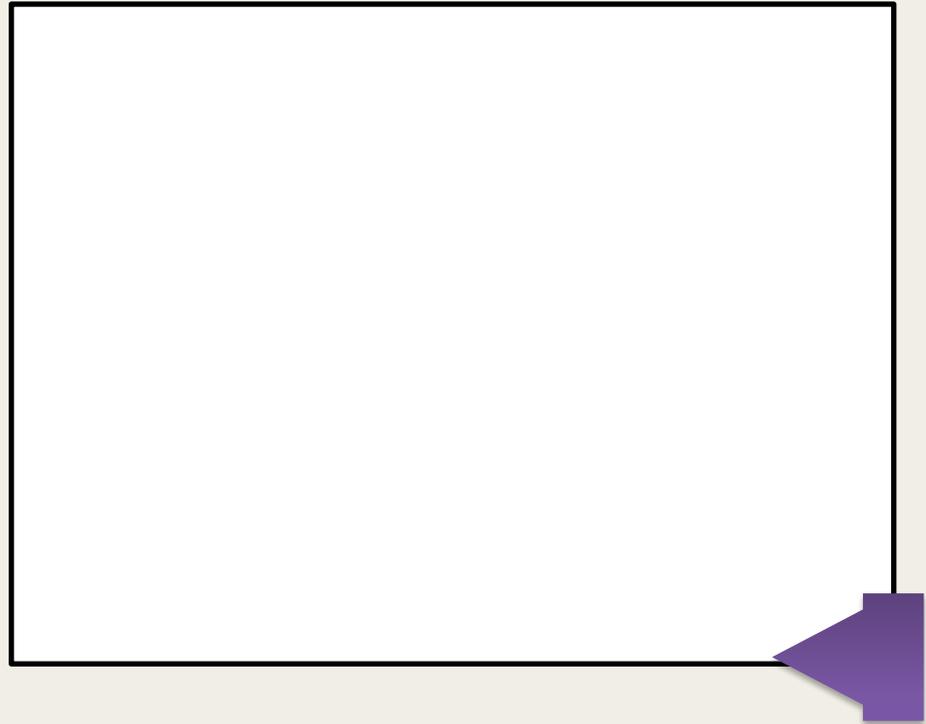
**100  
направо**

**90  
вперед**

**100  
направо**

**90  
вперед**

**100**



**вперед 100**

**направо 90**

**вперед**

**100  
направо**

**90  
вперед**

**100  
направо**

**90  
вперед**

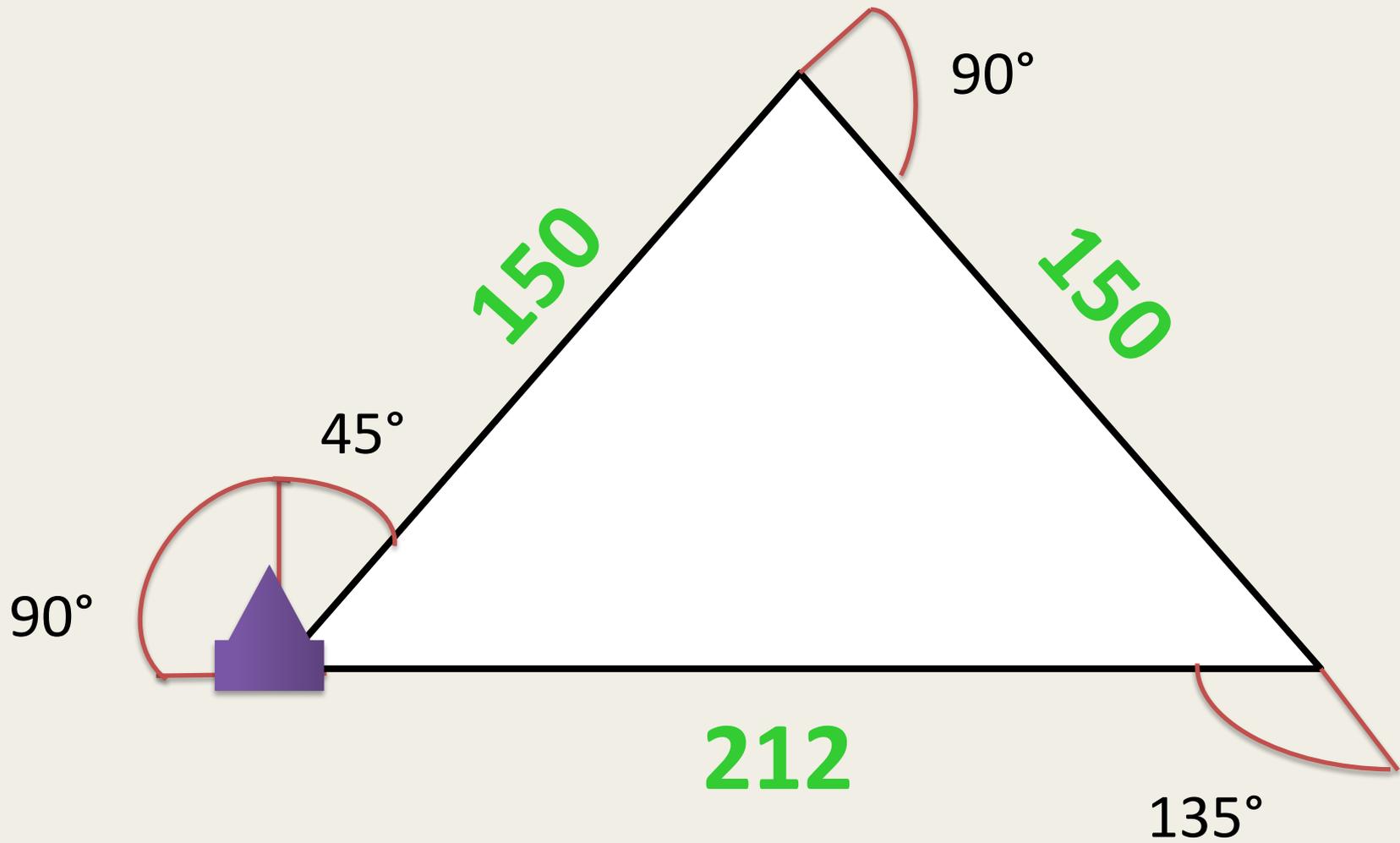
**100  
направо**

**90**



**повторить 4 {  
вперед 150  
направо 90  
}**





**Направо 45**

**Вперед 150**

**Направо 90**

**Вперед 150**

**Направо**

**135**

**Вперед 212**

**Направо 90**

$$360^\circ : 3 = 120^\circ$$

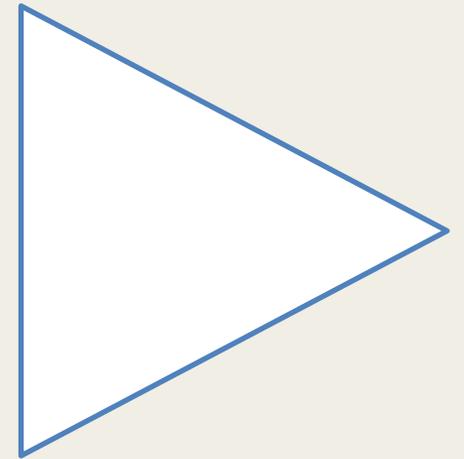
**повторить 3**

{

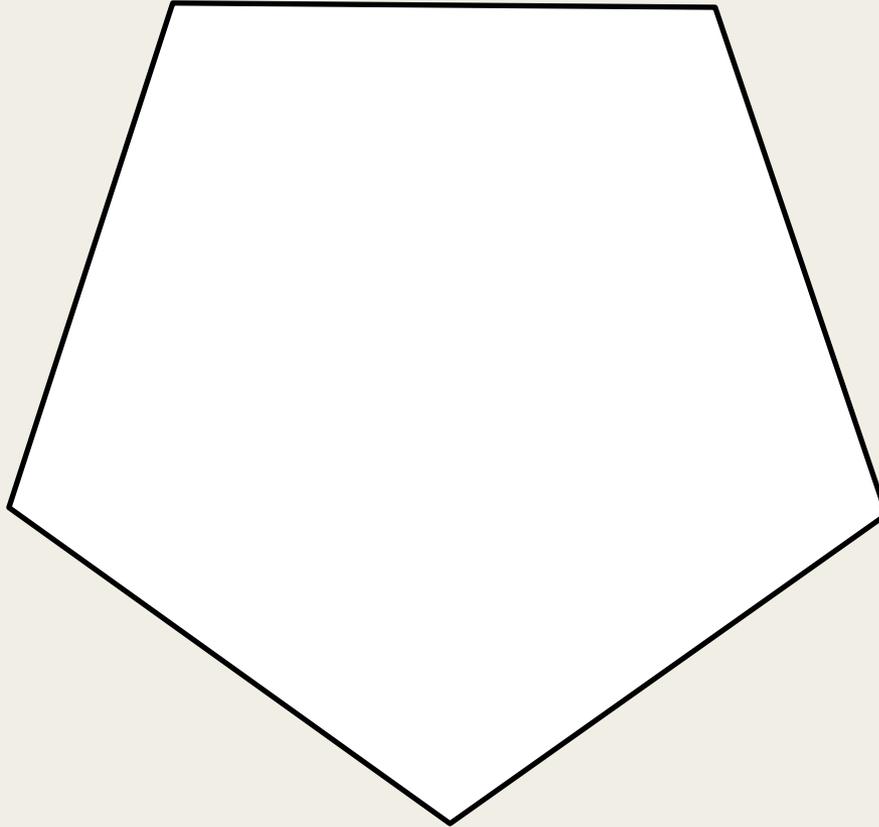
**вперед 100**

**налево 120**

}

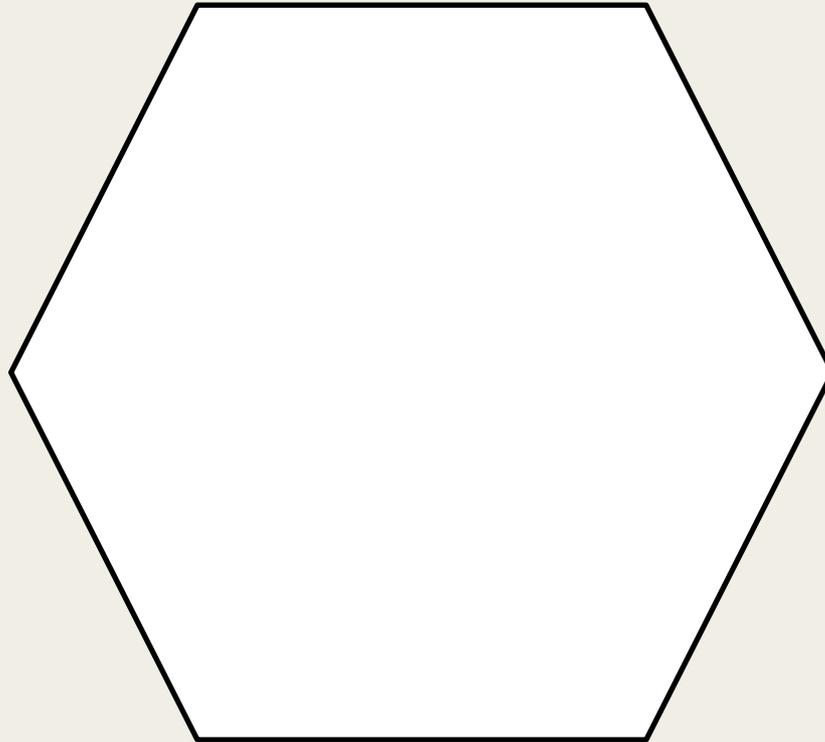


$$360^\circ : 5 = 72^\circ$$



**повторить 5{  
вперед 150  
налево 72  
}**

# Задание для самостоятельной работы



# Задание для самостоятельной работы

**повторить 18 {  
вперед 200  
налево 100  
}**

# Задание для самостоятельной работы

Нарисуйте любую  
произвольную фигуру

# ЦВЕТ



**цвет 12**

**вперед 50**

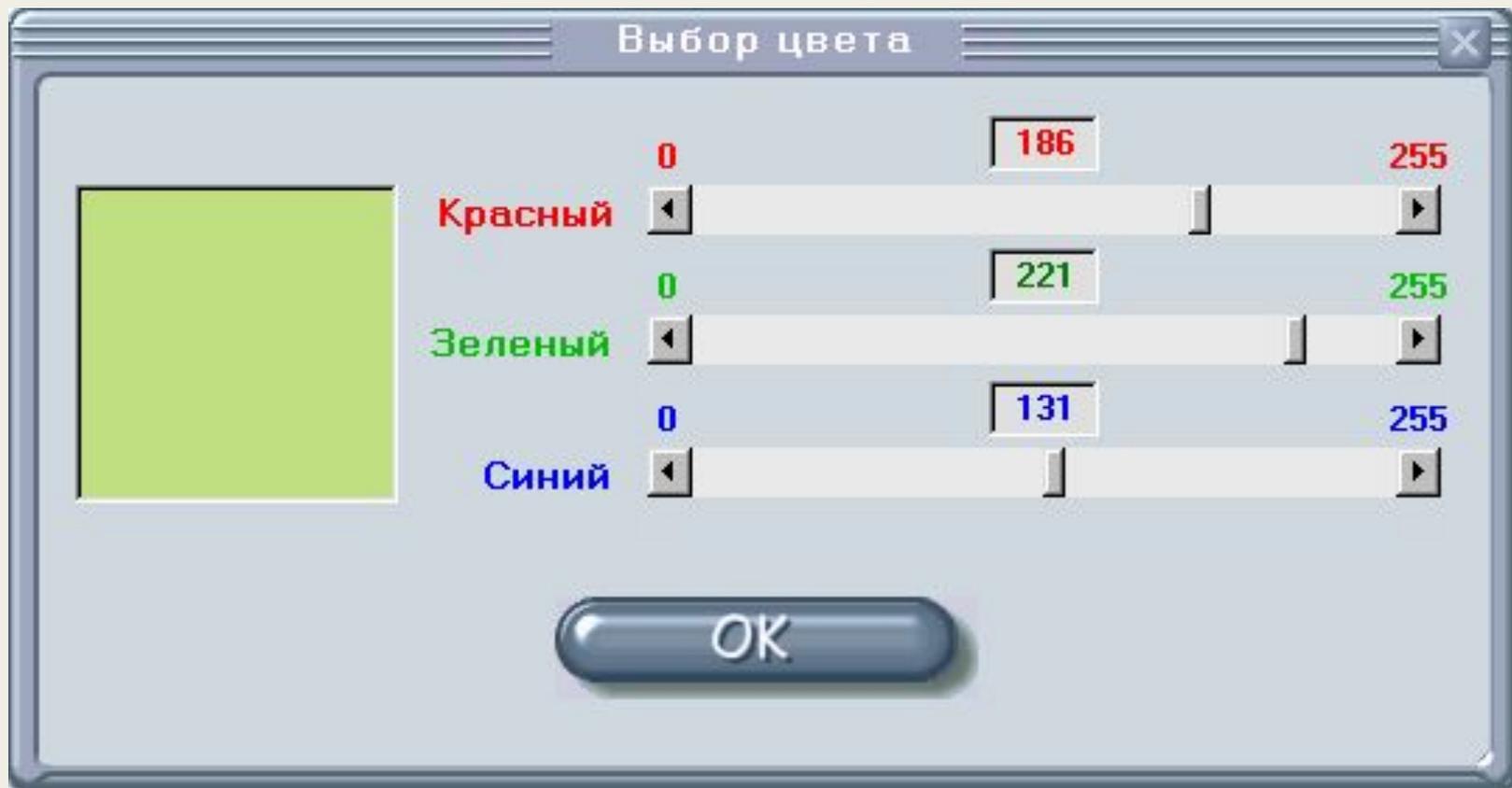
**цвет 14**

**вперед 50**

**цвет 10**

**вперед 50**

# RGB



Для управления толщиной пера служит команда

# ПЕРО.

Попробуем:

**перо 5**

**цвет RGB (186, 221, 131)**

**повторить 18 {**

**вперед 250**

**налево 140**

**}**

***ПОДНЯТЬ ПЕРО*** - поднимает перо у черепашки, после чего черепашка перестает оставлять след при движении.

***ОПУСТИТЬ ПЕРО*** - опускает перо у черепашки, после чего черепашка оставляет след при движении.

**МЕСТО** - передвигает черепашку в место с указанными координатами.

Размеры экранного поля составляют 800 точек по горизонтали и 600 точек по вертикали.

Точка с координатами  $(0, 0)$  находится в верхнем левом углу.

Точка с координатами  $(800, 600)$  - в нижнем правом углу.

Центр экрана -  $(400, 300)$ .

**КРУГ** - рисует круг заданного радиуса с центром в месте, в котором стоит черепаха.

Пример:

**круг 250**

**ЗАКРАСЬ** - закрашивает замкнутую область, в которую входит точка с заданными координатами.

Пример:

**круг 200**

**закрась 400, 300**

**повторить 4 {**

**вперед 100**

**налево 90 }**

**закрась черепаха.x - 10 , черепаха.y**

**- 10**

# ФОН=

Название  
картинки



# ОЧИСТИТЬ ФОН

# ПЕРЕМЕННЫ

## Е

Для хранения различных значений в языках программирования используются переменные.

**Переменной** называется область памяти, имеющая имя.

**Объявление**  
**переменной**  $x$

**Присваивание**

$$x = 5 + 3$$

**Арифметические**  
**операции**

**+** (плюс)

**-** (минус)

**\*** (умножение)

**/** (деление)

**фон= ultrasea.gif**

**цвет 15**

**перо 4**

**переменная x**

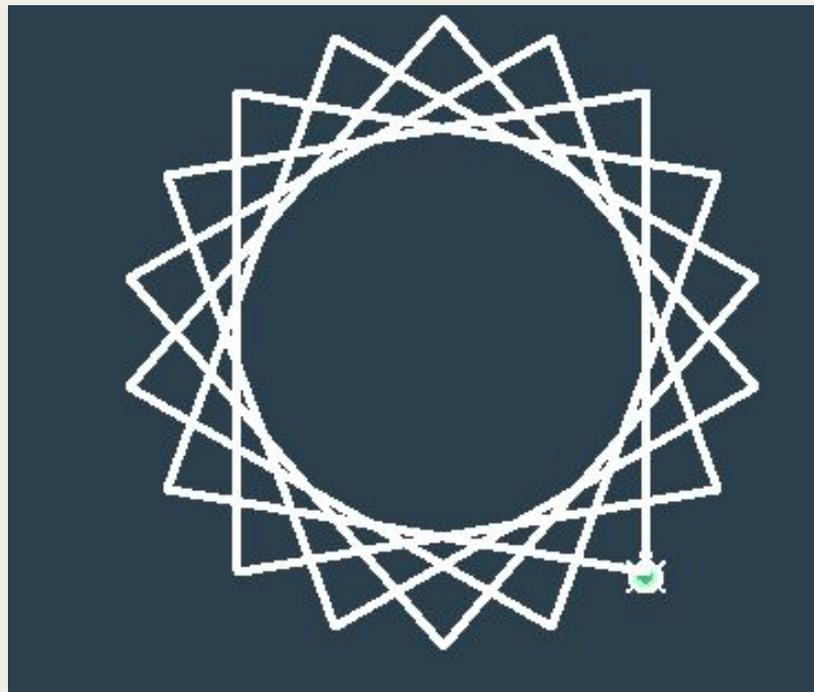
**X=20**

**повторить 18 {**

**вперед 200+x**

**налево 100**

**}**



# СЛУЧАЙНЫЕ

## ЧИСЛА

повторить 900 {  
перо случайный \* 10  
цвет случайный \* 15  
вперед случайный \*  
100  
налево случайный \*  
360  
}

