

Черепаха

Программное решение

Алгоритм аналитического решения №6:

1. Определить начальное положение Черепахи (в какой точке располагается, куда направлена голова, опущен ли хвост)
2. Проанализировать алгоритм исполнителя и понять, какую он описывает фигуру (если изначально этого не определить, то можно пошагово повторить команды и самим построить фигуру)
3. Перейти к математическому анализу для определения количества точек с целочисленными координатами внутри фигуры

Основные команды Черепахи в Кумир и Python:

Название операции	Кумир	Python	
Вперед n	вперед(n)	forward(n)	fd(n)
Назад n	назад(n)	backward(n)	bk(n)
Направо k	направо(k)	right(k)	rt(k)
Налево k	налево(k)	left(k)	lf(k)
Повтори n	нц n раз кц	for i in range(n)	
Поднять хвост	поднять хвост	penup()	pu()
Опустить хвост	опустить хвост	pendown()	pd()

Дополнительные функции в Python

tracer(0)	не показывать перемещения черепахи по полю, сразу после запуска программы выдает готовый рисунок
screensize(N, M)	задать размеры поля для черепахи размерами N*M пикселей
goto(x, y)	
setpos(x, y)	черепаха идет в точку с координатами (x, y) по кратчайшему пути
setposition(x, y)	
dot(d, 'color')	рисует круг диаметром d и цветом color

Что повторить по Python

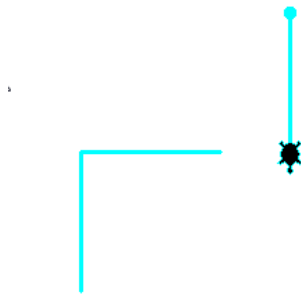
- **Циклы:** `for i in range(...)`, вложенные циклы, работа с диапазонами.
- **Условные операторы:** `if / else` — фильтрация точек по условиям.
- **Функции:** определение функций, возврат результата, использование `return`.
- **Импорт модулей:** `import turtle as t`, использование псевдонимов.
- **Типы данных:** числа, строки, списки.
- Библиотека [Turtle в Python](#)
- **Работа с циклами Повтори:** понимание, как запись Повтори k [Команда1 Команда2] превращается в цикл `for i in range(k):.`
- **Функции ускорения отрисовки:** `speed(0)`, `tracer(0)`, `update()`.

Шаблоны для решения через Python

1)

```
import turtle as t

t.shape("turtle") # сменить вид на черепаху
t.pensize(3)      # толщина линии
t.pencolor("cyan") # цвет линии
t.left(90)        # повернуть налево на 90°
t.forward(100)    # движение вперед на 100 шагов
t.right(90)       # повернуть направо на 90°
t.forward(100)    # движение вперед на 100 шагов
t.backward(50)    # движение назад
t.penup()         # поднять перо
t.goto(150, 200)  # переместиться в точку (x, y)
t.pendown()       # опустить перо
t.dot(10)         # нарисовать точку радиусом 10
t.right(90)       # повернуть направо на 90°
t.forward(100)    # движение вперед на 100 шагов
```



2)

Задача 1

Алгоритм: Повтори 7 [Вперед 10 Направо 120]

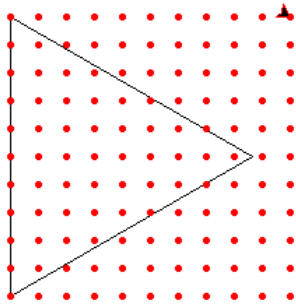
Что делает: рисует звездообразный многоугольник (семикратное повторение треугольного шага).

Нужно определить количество целочисленных точек внутри фигуры (без точек на границе).

```

import turtle as t
k = 20 # масштаб
t.left(90)
for i in range(7):
    t.forward(10*k)
    t.right(120)
t.up()
t.pencolor('red')
for x in range(0, 11):
    for y in range(0, 11):
        t.goto(x*k, y*k)
        t.dot(6)
# ответ по рисунку → 38

```



3)

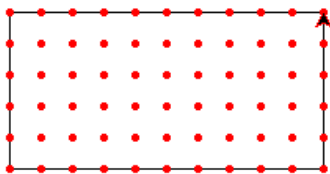
Задача 2

Алгоритм: Повтори 4 [Вперёд 5 Направо 90 Вперёд 10 Направо 90]
 Рисуется прямоугольная фигура, составленная из четырёх трапеций.
 Нужно посчитать точки внутри (без границы).

```

import turtle as t
k = 20 # масштаб
t.left(90)
for i in range(4):
    t.forward(5*k)
    t.right(90)
    t.forward(10*k)
    t.right(90)
t.up()
t.pencolor('red')
for x in range(0, 11):
    for y in range(0, 6):
        t.goto(x*k, y*k)
        t.dot(6)
# ответ по рисунку → 36

```



4)

Задача 3

Алгоритм: Повтори 4 [Вперёд 10 Направо 60 Вперёд 10 Направо 120]

Рисуется «ромбовидная звезда». Считаем количество целочисленных точек внутри без учёта границы.

```
import turtle as t
```

```
k = 15 # масштаб
```

```
t.left(90)
```

```
for i in range(4):
```

```
    t.forward(10*k)
```

```
    t.right(60)
```

```
    t.forward(10*k)
```

```
    t.right(120)
```

```
t.up()
```

```
t.pencolor('red')
```

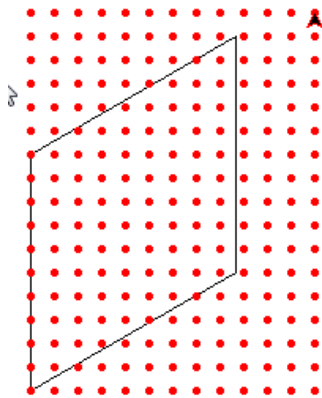
```
for x in range(0, 13):
```

```
    for y in range(0, 17):
```

```
        t.goto(x*k, y*k)
```

```
        t.dot(6)
```

```
# ответ → 80
```



4)

Задача 4

Алгоритм: Повтори 14 [Направо 60 Вперёд 2 Направо 60 Вперёд 2 Направо 270]

Здесь точки на границе учитывать **нужно**.

```
import turtle as t
```

```
k = 15 # масштаб
```

```
# t.left(90)
```

```
for i in range(14):
```

```
    t.right(60)
```

```
    t.forward(2*k)
```

```
    t.right(60)
```

```
    t.forward(2*k)
```

```
    t.right(270)
```

```
t.up()
```

```
t.pencolor('red')
```

```
for x in range(-14, 4):
```

```
    for y in range(-10, 7):
```

```
        t.goto(x*k, y*k)
```

```
        t.dot(6)
```

```
# ответ → 148
```