



ИНФОРМАТИКА

7

класс

КОНСТРУИРОВАНИЕ АЛГОРИТМОВ

ОСНОВЫ АЛГОРИТМИЗАЦИИ

КЛЮЧЕВЫЕ СЛОВА

- ◆ пошаговая детализация
- ◆ вспомогательный алгоритм
- ◆ рекурсивный алгоритм

МЕТОДЫ ПОСТРОЕНИЯ АЛГОРИТМОВ

При конструировании алгоритма решения сложной задачи её принято разбивать на более простые подзадачи, для этого используется один из следующих методов:

метод разработки «сверху вниз»

метод разработки «снизу вверх»

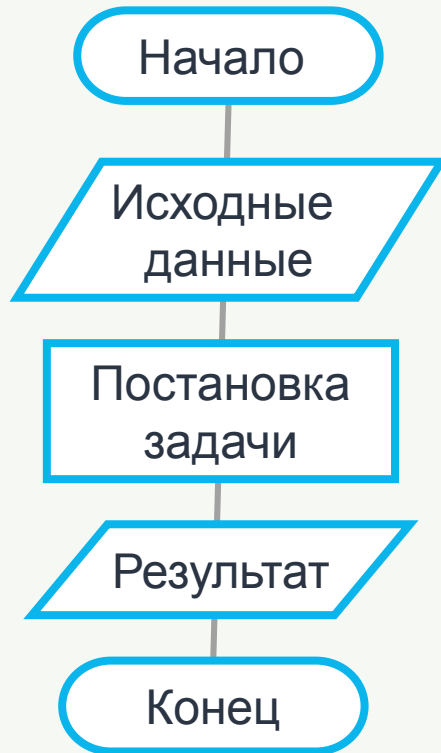
«СВЕРХУ ВНИЗ»
ПОШАГОВАЯ ДЕТАЛИЗАЦИЯ



«СНИЗУ ВВЕРХ»
РАЗРАБОТКА БИБЛИОТЕКИ



МЕТОД ПОШАГОВОЙ ДЕТАЛИЗАЦИИ



Первый шаг

Разбиваем задачу на более простые части

Решение каждой части задачи формулируем в отдельной команде (предписании)

Предписания, выходящие за пределы возможностей исполнителя, представляем в виде еще более простых команд

Последующие шаги

РАЗРАБОТКА АЛГОРИТМА МЕТОДОМ ПОШАГОВОЙ ДЕТАЛИЗАЦИИ ДЛЯ ИСПОЛНИТЕЛЯ РОБОТ

Система команд исполнителя Робот:

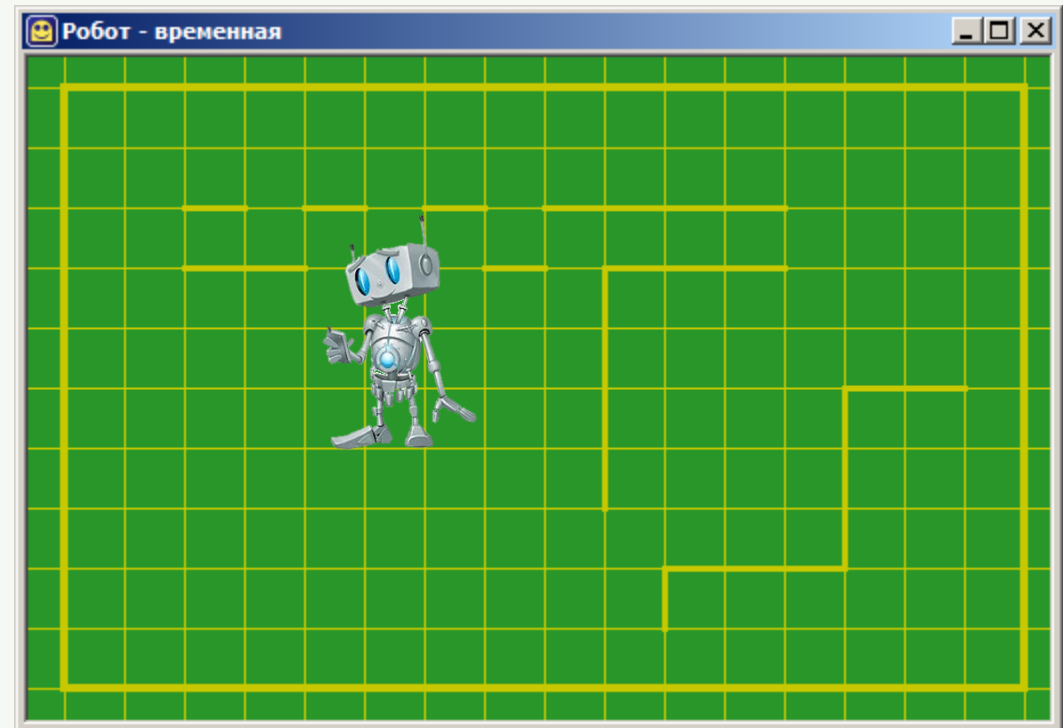
вправо

влево

вверх

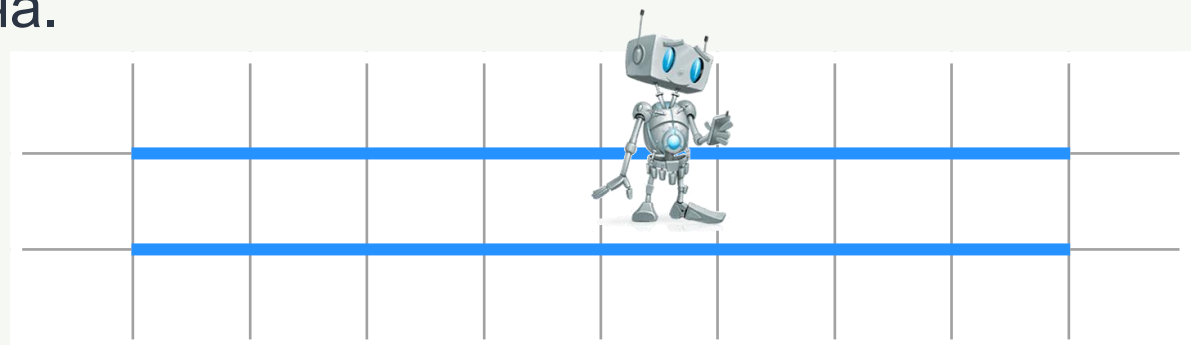
вниз

закрасить

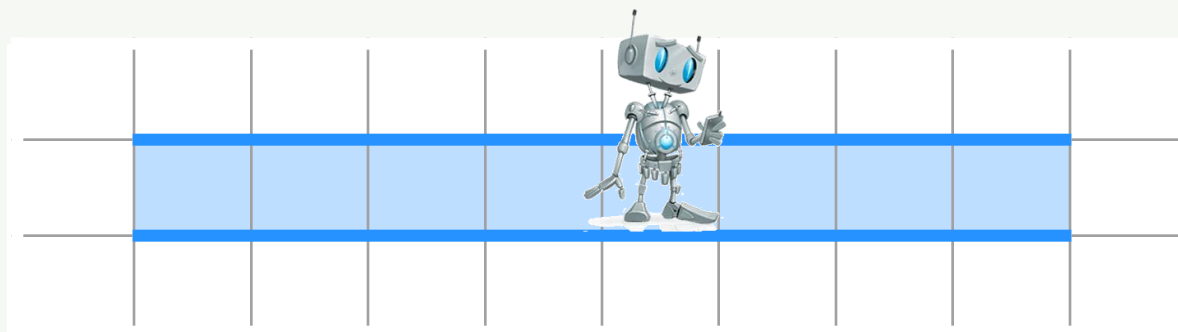


ПОСТАНОВКА ЗАДАЧИ

Робот находится в некоторой клетке горизонтального коридора. Ни одна из клеток коридора не закрашена.



Робот должен закрасить все клетки этого коридора и вернуться в исходное положение.



УКРУПНЁННЫЙ ПЛАН ДЕЙСТВИЙ РОБОТА

Начало

1. Закраска всех клеток коридора левее исходной

2. Возвращение в исходное положение

3. Закраска всех клеток коридора правее исходной

4. Возвращение в исходное положение

5. Закраска исходной клетки

Конец



ДЕТАЛИЗАЦИЯ ПЛАНА ДЕЙСТВИЙ РОБОТА

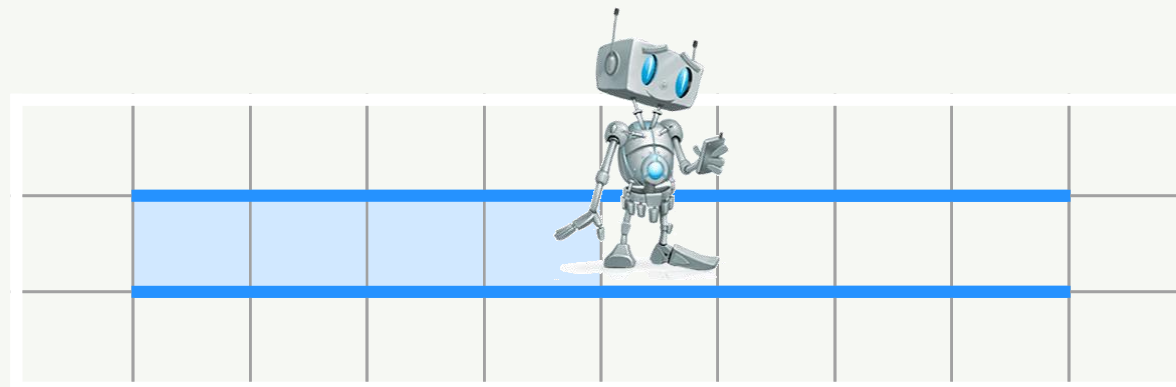
1. Закраска всех клеток коридора, находящихся левее Робота:

влево

нц пока сверху стена **и** снизу стена
закрасить; влево

кц

Положение Робота после выполнения этого алгоритма:



ДЕТАЛИЗАЦИЯ ПЛАНА ДЕЙСТВИЙ РОБОТА

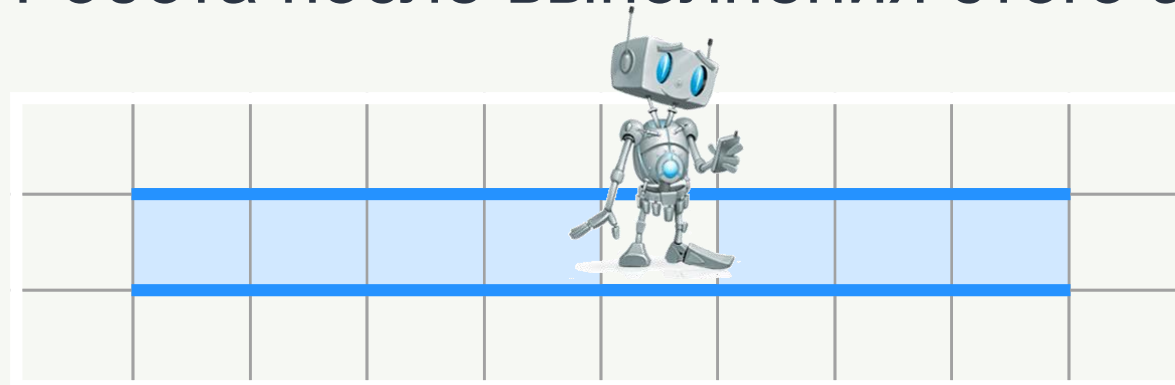
3. Закраска всех клеток коридора, находящихся правее Робота:

вправо

нц пока сверху стена **и** снизу стена
закрасить; вправо

кц

Положение Робота после выполнения этого алгоритма:



ДЕТАЛИЗАЦИЯ ПЛАНА ДЕЙСТВИЙ РОБОТА

4. Возвращение Робота в коридор в исходную точку:

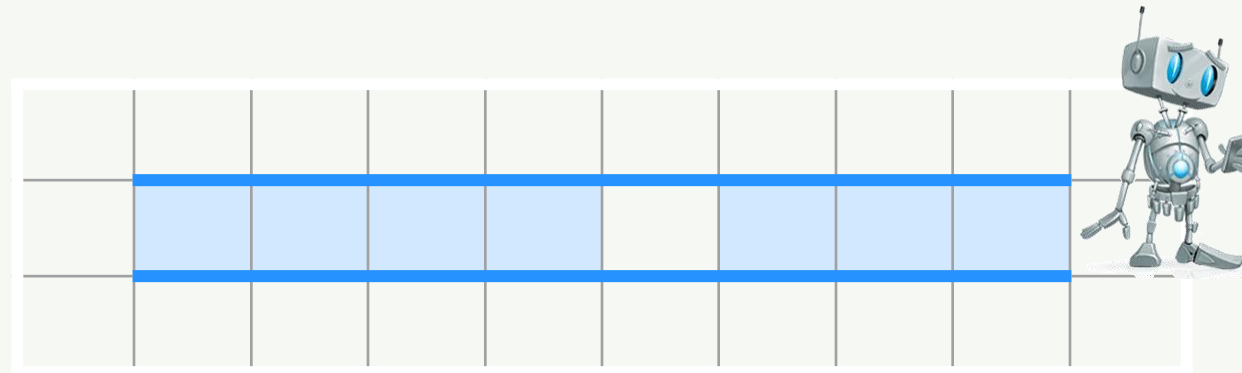
влево

нц пока клетка закрашена

влево

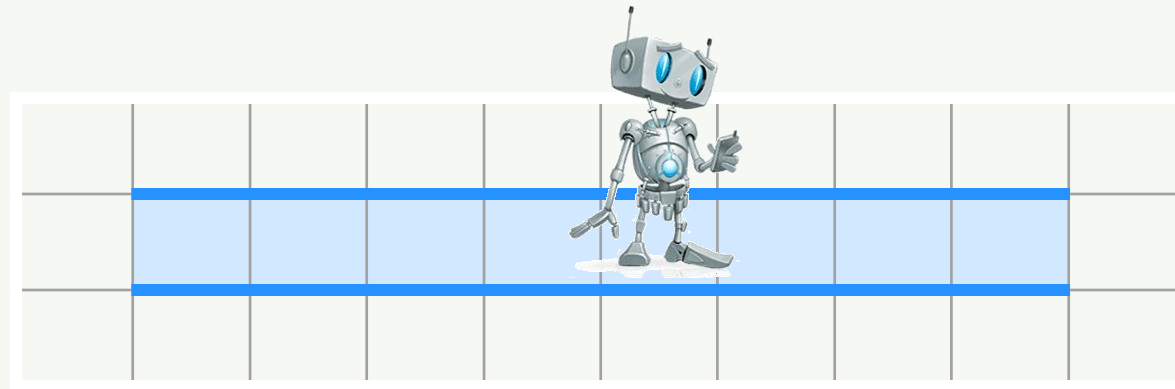
кц

Положение Робота после выполнения этого алгоритма:



ДЕТАЛИЗАЦИЯ ПЛАНА ДЕЙСТВИЙ РОБОТА

5. По команде закрасить Робот закрашивает исходную точку.



ПРОГРАММА ДЛЯ РОБОТА

алг
нач

влево

нц пока сверху стена **и** снизу стена
закрасить; влево

кц

вправо

нц пока клетка закрашена
вправо

кц

вправо

нц пока сверху стена **и** снизу стена
закрасить; вправо

кц

влево

нц пока клетка закрашена
влево

кц

закрасить

кон

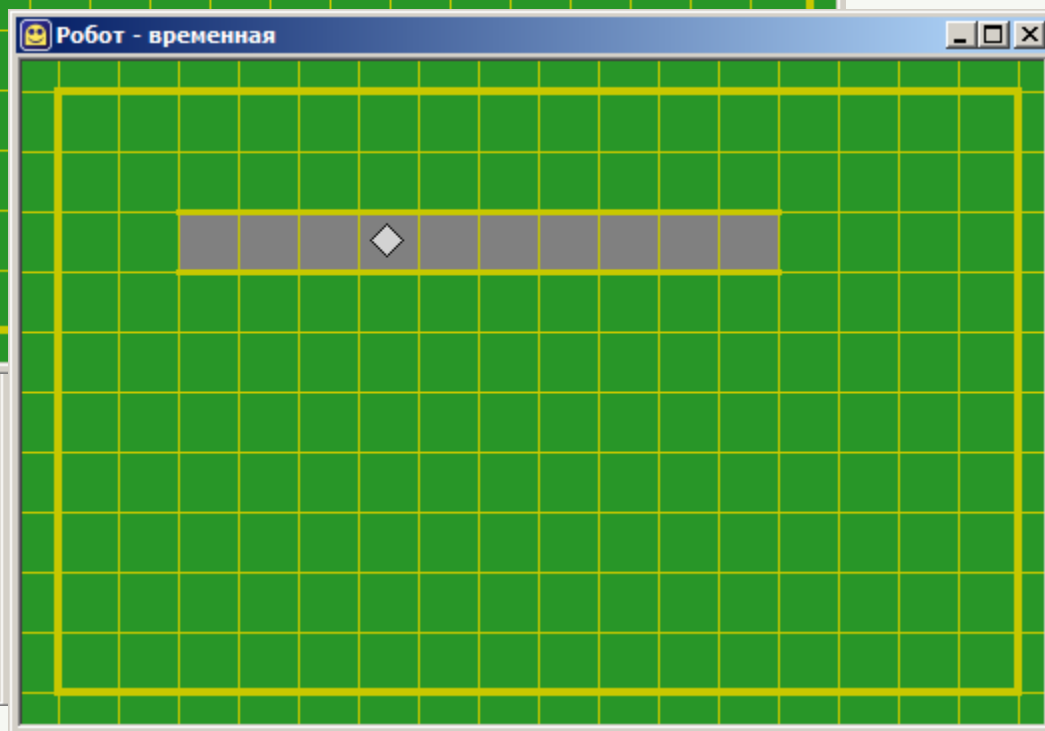
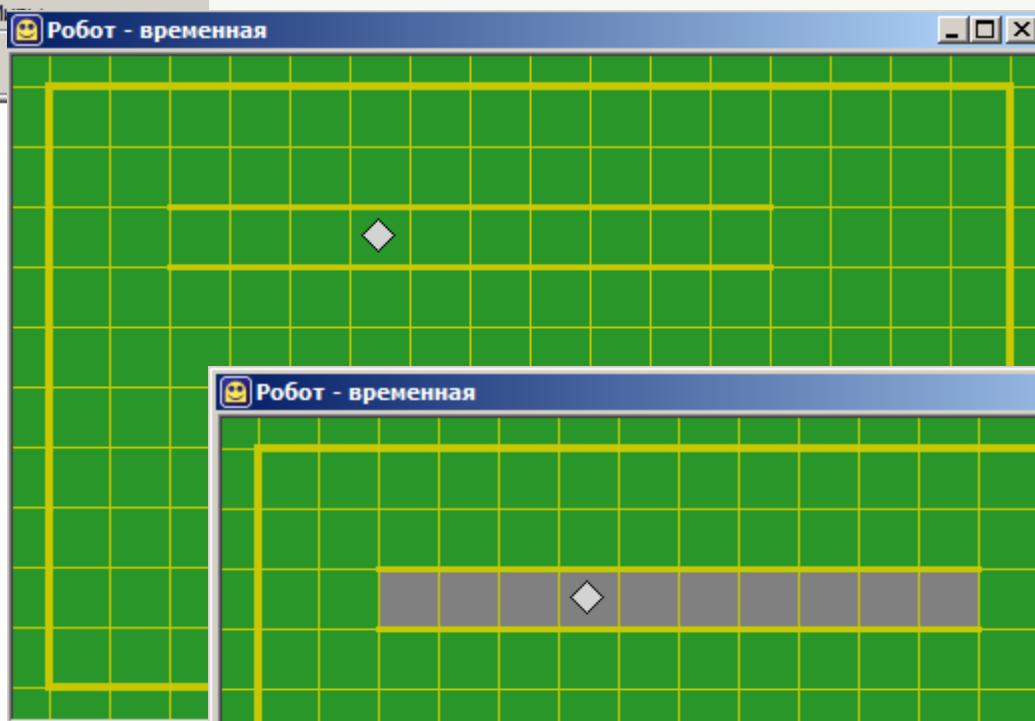


К Новая программа - Кумир

Программа Редактирование Вставка Выполнение Инструменты Робот Чертежник Инфо М



```
1 использовать Робот
2 алг
3 нач
4   влево
5   нц пока сверху стена и снизу стена
6     закрасить; влево
7   кц
8   вправо
9   нц пока клетка закрашена
10    вправо
11  кц
12  вправо
13  нц пока сверху стена и снизу стена
14    закрасить; вправо
15  кц
16  влево
17  нц пока клетка закрашена
18    влево
19  кц
20  закрасить
21 кон
22
23
```



ВСПОМОГАТЕЛЬНЫЙ АЛГОРИТМ

Вспомогательный алгоритм - алгоритм, целиком используемый в составе другого алгоритма для решения некоторой подзадачи основной задачи.



ВСПОМОГАТЕЛЬНЫЙ АЛГОРИТМ ДЛЯ РОБОТА

ИСПОЛЬЗОВАТЬ Робот

алг узор

нач

фигура

вправо; вниз

фигура

вправо; вверх;

фигура

кон

алг фигура

нач

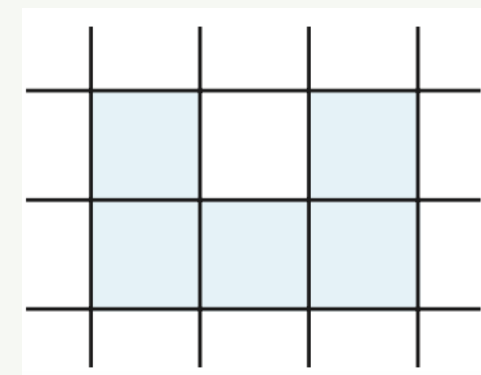
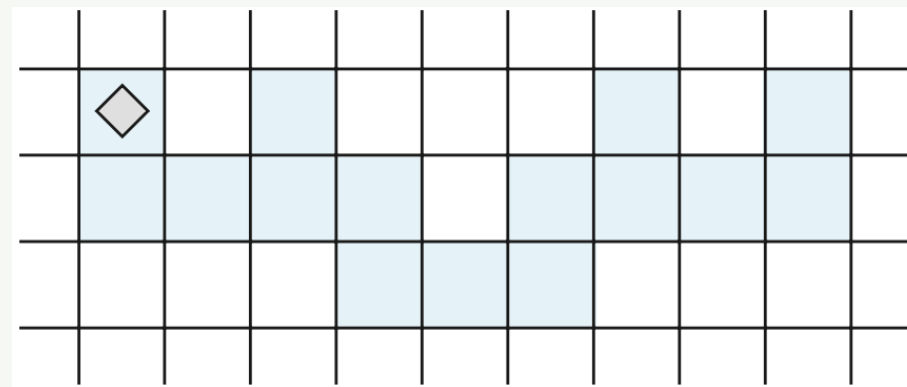
закрасить; вниз

закрасить; вправо; закрасить;

вправо; закрасить; вверх;

закрасить

кон

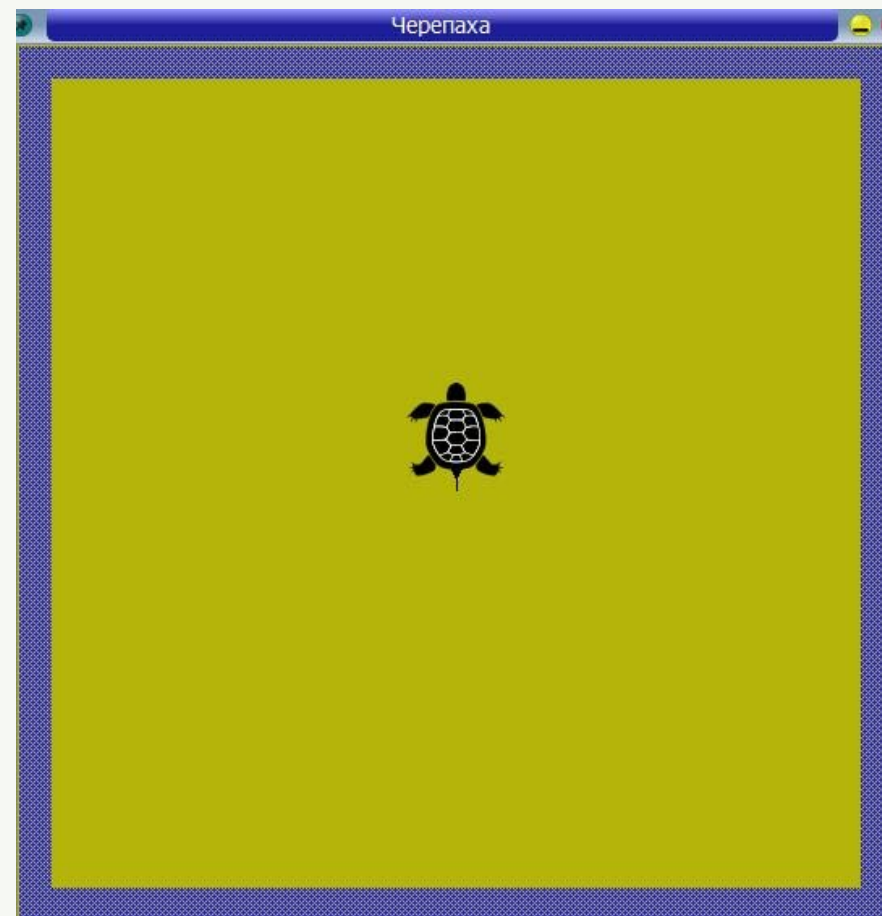


СИСТЕМА КОМАНД ИСПОЛНИТЕЛЯ ЧЕРЕПАХА

ПОДНЯТЬ ХВОСТ
ОПУСТИТЬ ХВОСТ
ВПЕРЕД (n)
НАЗАД (n)
ВПРАВО (m)
ВЛЕВО (m)

НЦ k **РАЗ**
<последовательность
команд>

КЦ



АНАЛИЗ АЛГОРИТМА

нач

нц 8 раз

вперед (50)

нц 3 раз

вперед (50)

вправо (120)

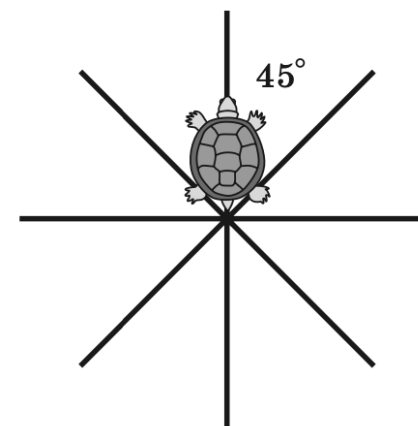
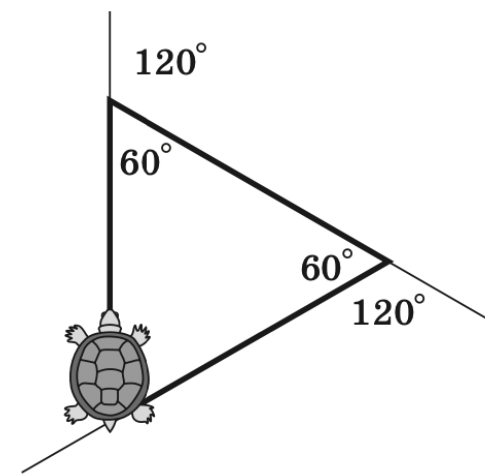
кц

назад (50)

вправо (45)

кц

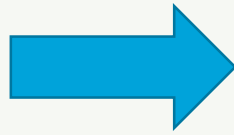
кон



Оформите внутренний цикл в виде вспомогательного алгоритма без параметров треугольник.

Модифицируем вспомогательный алгоритм `треугольник` в **вспомогательный алгоритм с параметрами** `многоугольник (k, n)`,
где k – количество углов, а n – длина стороны правильного многоугольника:

```
алг треугольник  
нач  
    нц 3 раз  
        вперед (50)  
        вправо (120)  
    кц  
кон
```



```
алг многоугольник (арг цел  $k, n$ )  
нач  
    нц  $k$  раз  
        вперед ( $n$ )  
        вправо ( $360/k$ )  
    кц  
кон
```



ФОРМАЛЬНЫЕ И ФАКТИЧЕСКИЕ ПАРАМЕТРЫ

Формальные параметры используются при описании алгоритма.

Фактические параметры - те величины, для которых будет исполнен вспомогательный алгоритм.

Типы, количество и порядок следования формальных и фактических параметров должны совпадать.



ФОРМАЛЬНЫЕ ПАРАМЕТРЫ

алг многоугольник (**арг** цел k, n)

нач

нц k **раз**

вперед (n)

вправо ($360/k$)

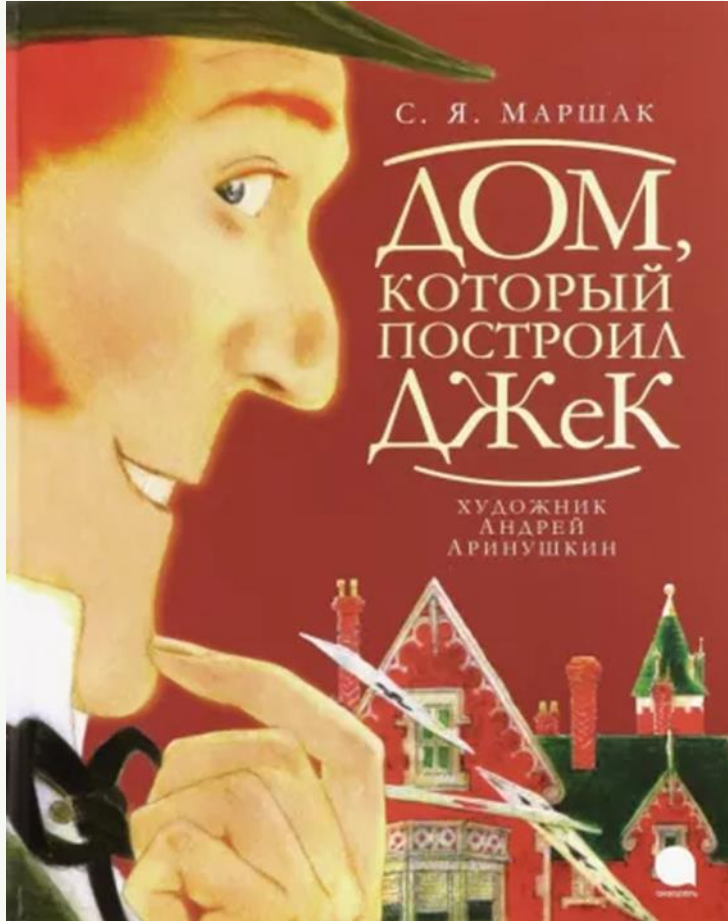
кц

кон

Здесь k, n — **формальные параметры**, они используются при описании вспомогательного алгоритма. При конкретном обращении к вспомогательному алгоритму формальные параметры заменяются **фактическими параметрами**.



РЕКУРСИВНЫЙ АЛГОРИТМ



Алгоритм, в котором прямо или косвенно содержится ссылка на него же как на вспомогательный алгоритм, называют **рекурсивным**.

ХАНОЙСКАЯ БАШНЯ

Даны три стержня, на один из которых нанизаны кольца, причём кольца лежат меньшее на большем. Задача состоит в том, чтобы перенести пирамиду с левого стержня на правый.

Требования:

1. За один раз разрешается переносить только одно кольцо.
2. Меньшее кольцо всегда должно лежать на большем.

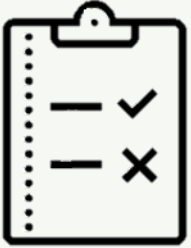


Вы сможете решить задачу для 4 колец, если сумеете решить чуть более простую задачу: перенести 3 верхних кольца на средний стержень.

Один из основных методов конструирования алгоритмов решения сложных задач — **метод пошаговой детализации**, когда исходная задача разбивается на несколько частей, каждая из которых проще всей задачи, и решение каждой части формулируется в отдельном предписании; если получаются предписания, выходящие за пределы возможностей исполнителя, то они представляются в виде совокупности ещё более простых предписаний. Процесс продолжается до тех пор, пока все предписания не будут понятны исполнителю.

Вспомогательный алгоритм — алгоритм, целиком используемый в составе другого алгоритма для решения некоторой подзадачи основной задачи.

Алгоритм, в котором прямо или косвенно содержится ссылка на него же как на вспомогательный алгоритм, называют **рекурсивным**.

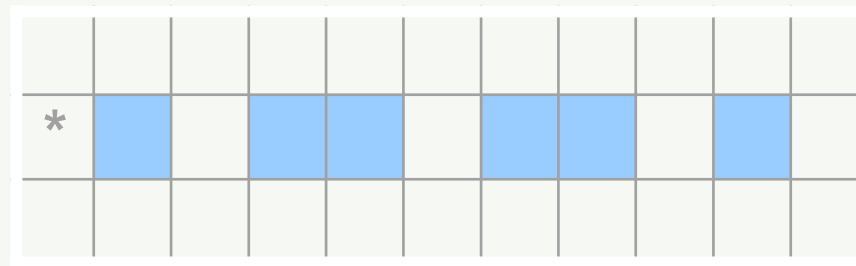
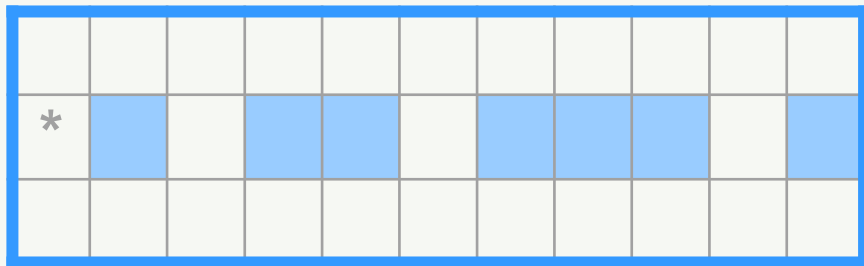


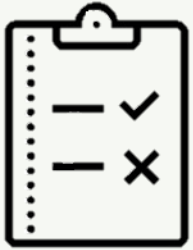
ВОПРОСЫ И ЗАДАНИЯ

В ряду из десяти клеток правее Робота некоторые клетки закрашены. Последняя закрашенная клетка может примыкать к стене.

Составьте алгоритм, который закрашивает клетки выше и ниже каждой закрашенной клетки.

Проверьте работу алгоритма в следующих случаях:





ВОПРОСЫ И ЗАДАНИЯ

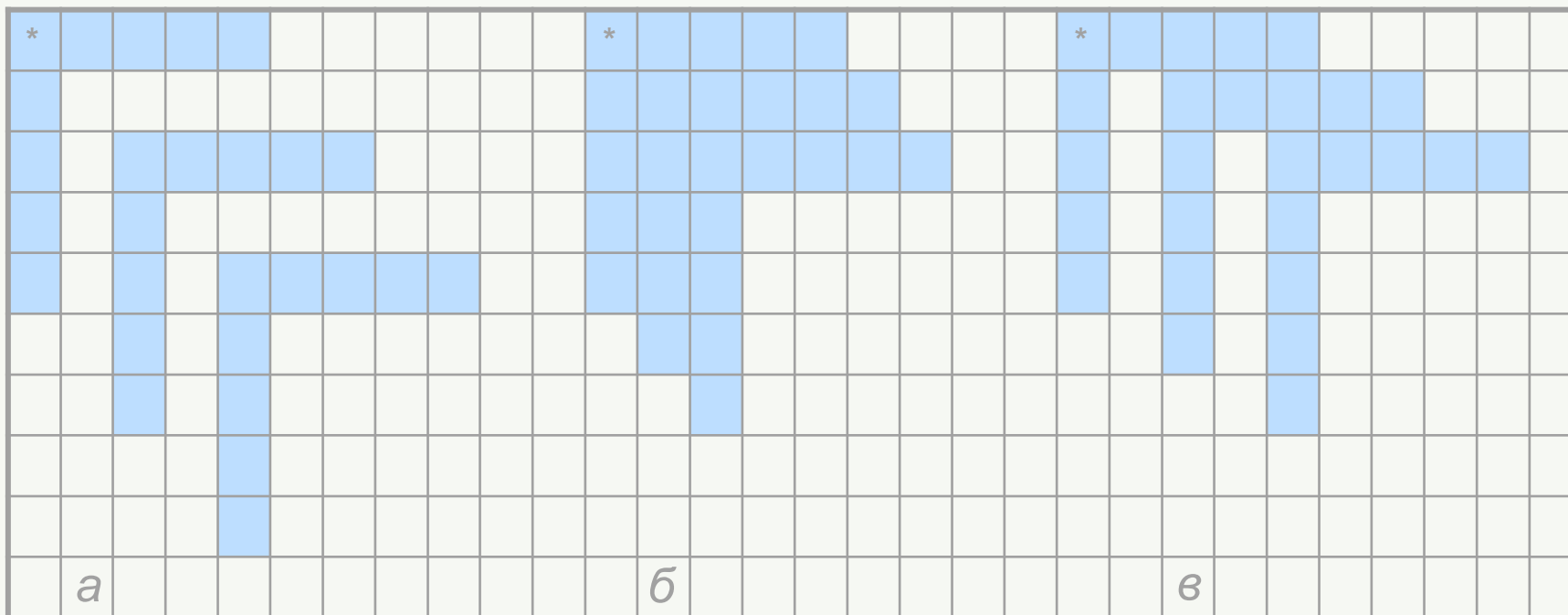
Для чего нужны вспомогательные алгоритмы?

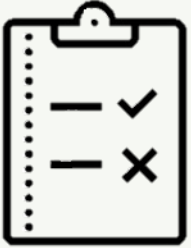




ВОПРОСЫ И ЗАДАНИЯ

Составьте алгоритмы, под управлением которых Робот закрасит указанные клетки.





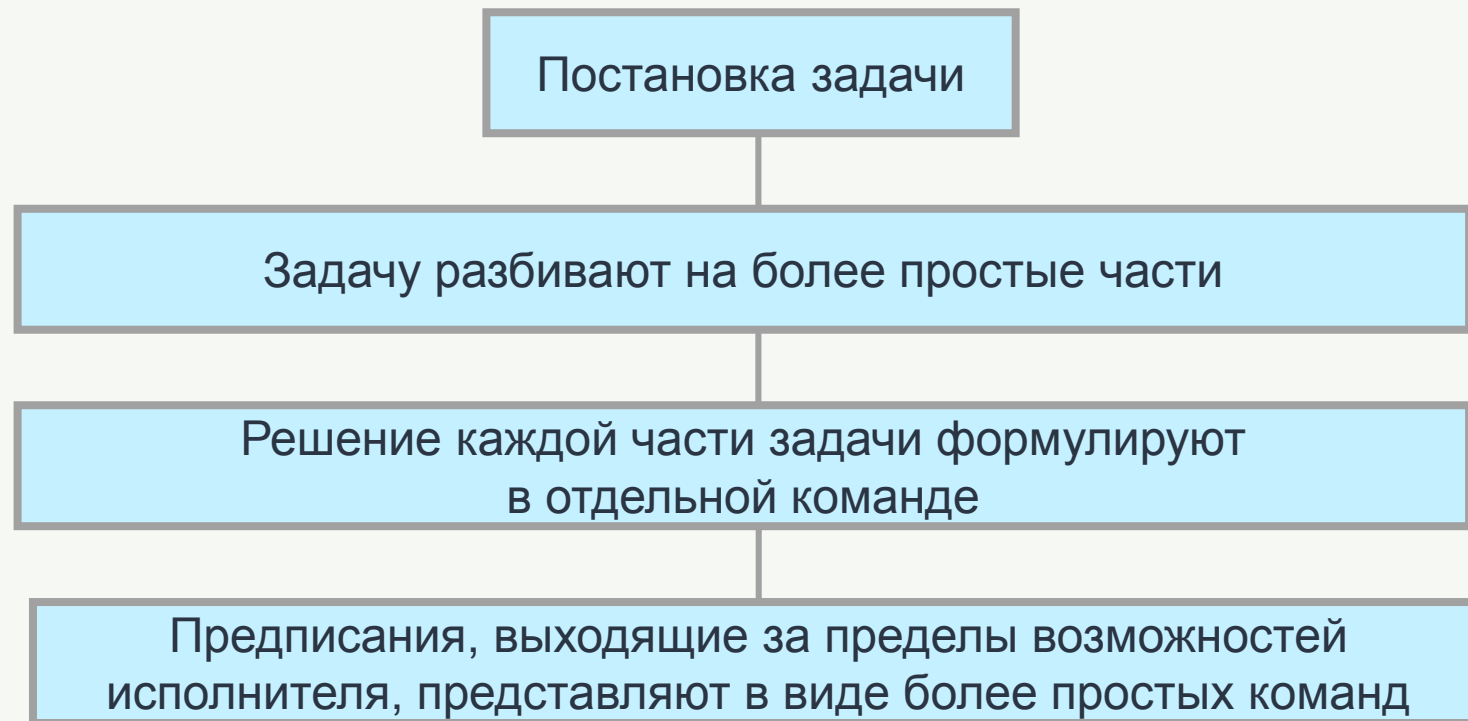
ВОПРОСЫ И ЗАДАНИЯ

Какие алгоритмы называют рекурсивными?

Приведите пример рекурсии из жизни.



Метод последовательного построения алгоритма - один из основных методов конструирования алгоритмов.



Вспомогательный алгоритм - алгоритм, целиком используемый в составе другого алгоритма.