



Запись алгоритмов. Исполнители

Способы записи алгоритмов

Словесные

Словесное
описание

Построчная
запись

Графические

Последователь-
ность рисунков

Структурограмма

Блок-схема

На алгоритмических
языках

Школьный
алгоритмический
язык

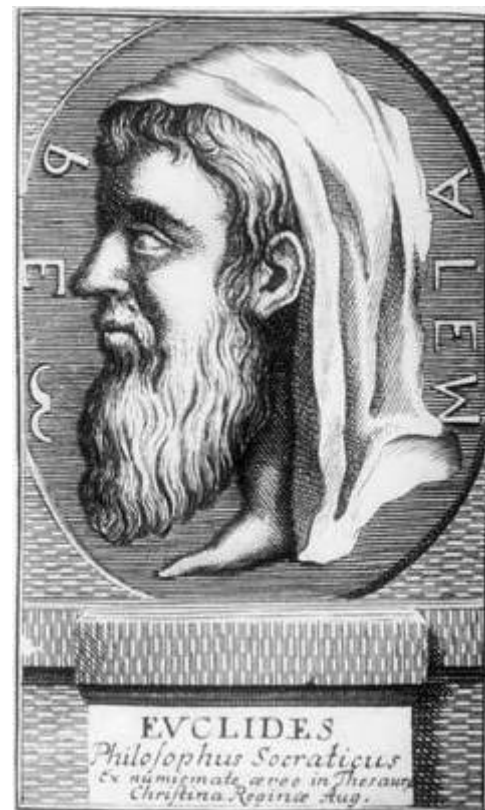
Язык
программирова-
ния

Словесное описание

Пример. Алгоритм нахождения НОДа пары целых чисел (алгоритм Евклида):

Чтобы найти НОД двух чисел, составьте таблицу из двух столбцов и назовите столбцы X и Y . Запишите первое из заданных чисел в столбец X , а второе - в столбец Y . Если данные числа не равны, замените большее из них на результат вычитания из большего числа меньшего.

Повторяйте такие замены до тех пор, пока числа не окажутся равными, после чего число из столбца X считайте искомым результатом.



Построчная запись

Правила построчной записи алгоритма

Каждое предписание записывается с новой строки

Предписание (шаги) алгоритма нумеруются

Исполнение алгоритма происходит в порядке возрастания номеров шагов, начиная с первого, если нет особых указаний

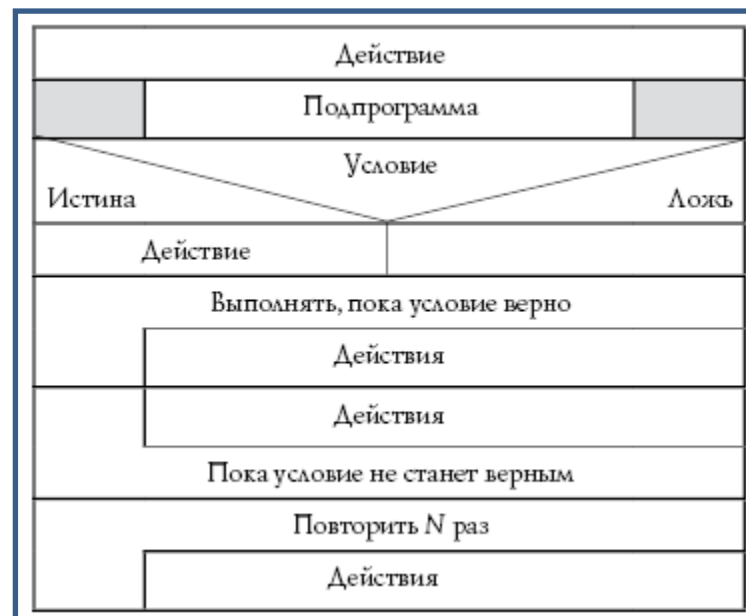
Кроме слов естественного языка предписания могут содержать математические выражения и формулы.

Построчная запись

Построчная запись алгоритма Евклида

1. Начало.
2. Обозначить первое из заданных чисел X , второе - Y .
3. Если $X = Y$ то перейти к п. 9.
4. Если $X > Y$, то перейти к п. 5, иначе перейти к п. 7.
5. Заменить X на $X - Y$.
6. Перейти к п. 3.
7. Заменить Y на $Y - X$.
8. Перейти к п. 3.
9. Считать X искомым результатом.
10. Конец.

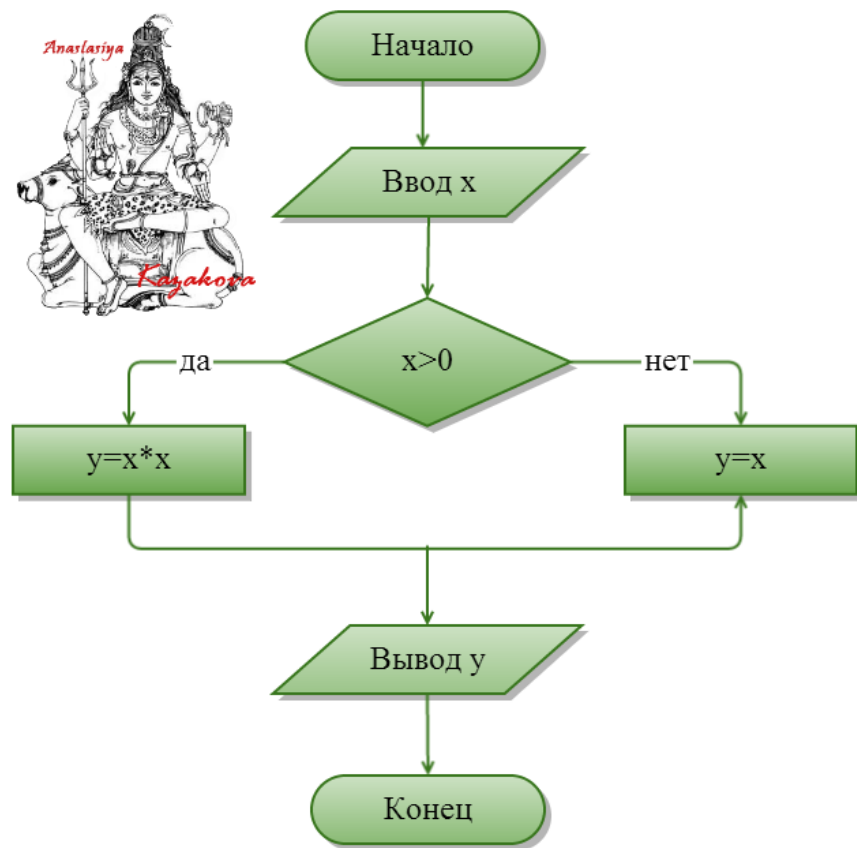
Графическая запись



Структурограмма

Последовательные картинки

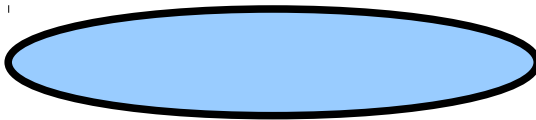
Блок-схема



Блок-схема – графическое изображение алгоритма, когда отдельные его действия изображаются графическими фигурами, а связи между ними с помощью стрелок.



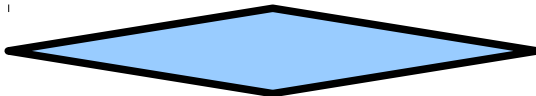
Элементы блок-схемы:



– начало и конец



– действие



– условие



– счётчик



– процедура



– ВВОД И ВЫВОД

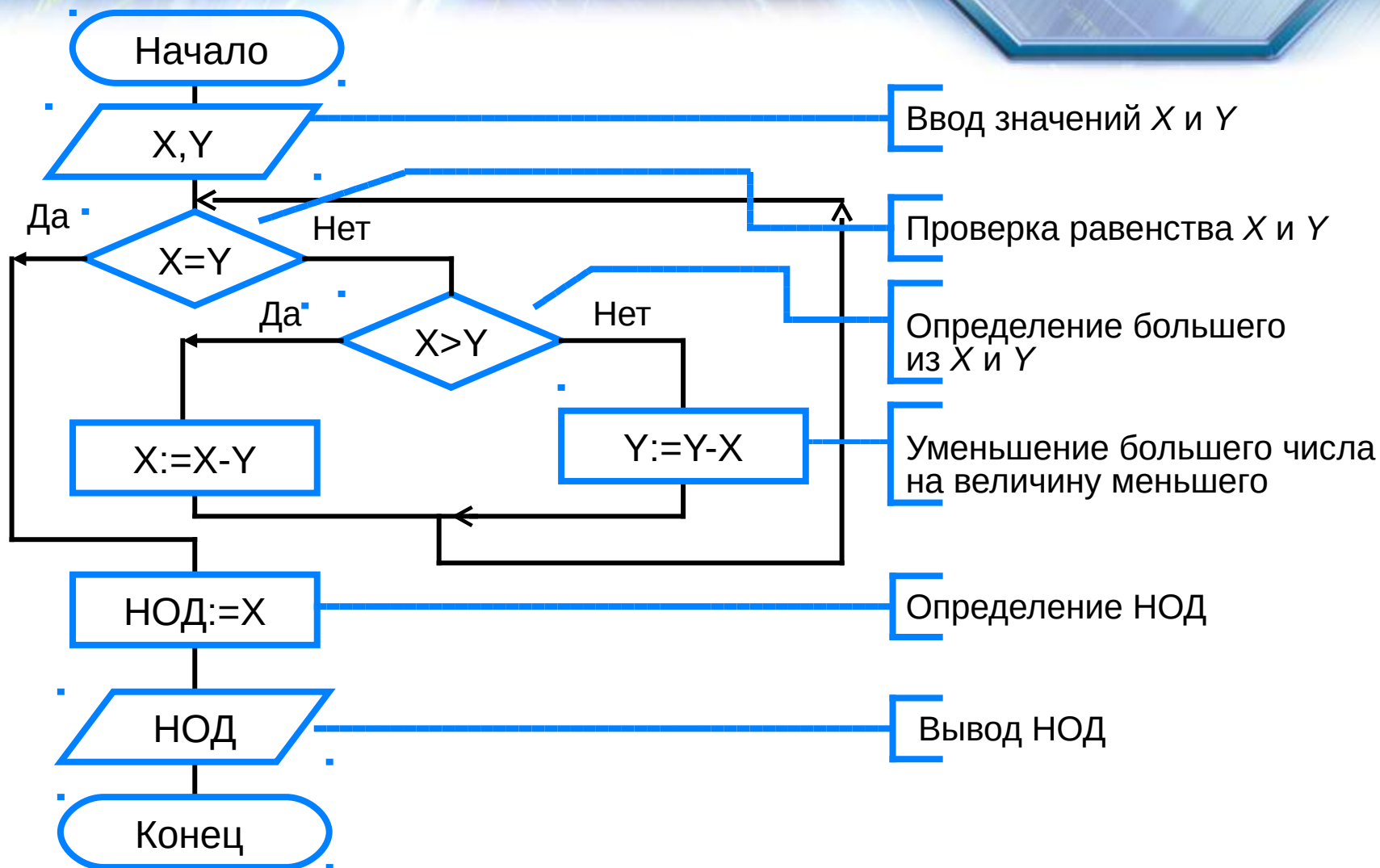


– комментарий



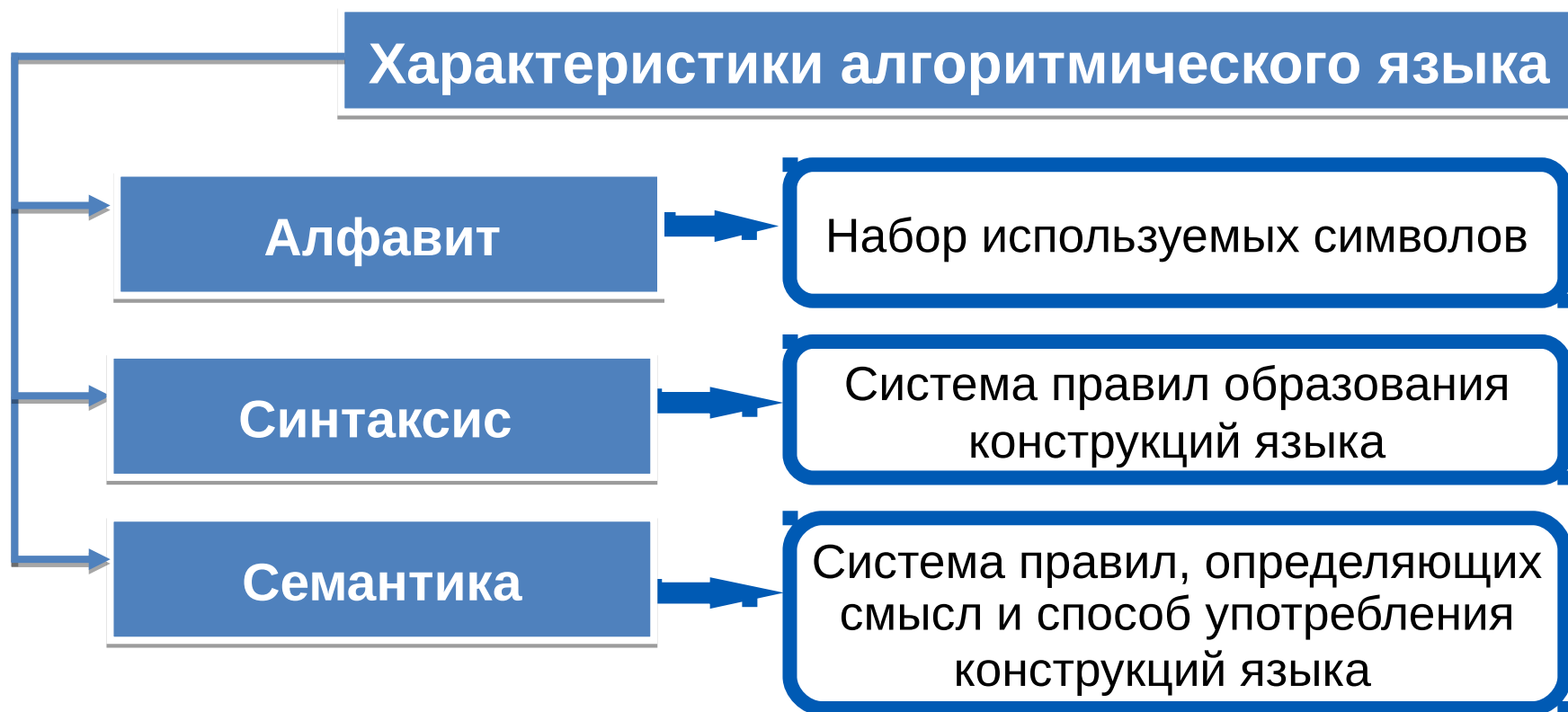
– соединитель

Алгоритм Евклида



Алгоритмические языки

Алгоритмические языки - формальные языки, предназначенные для записи алгоритмов.



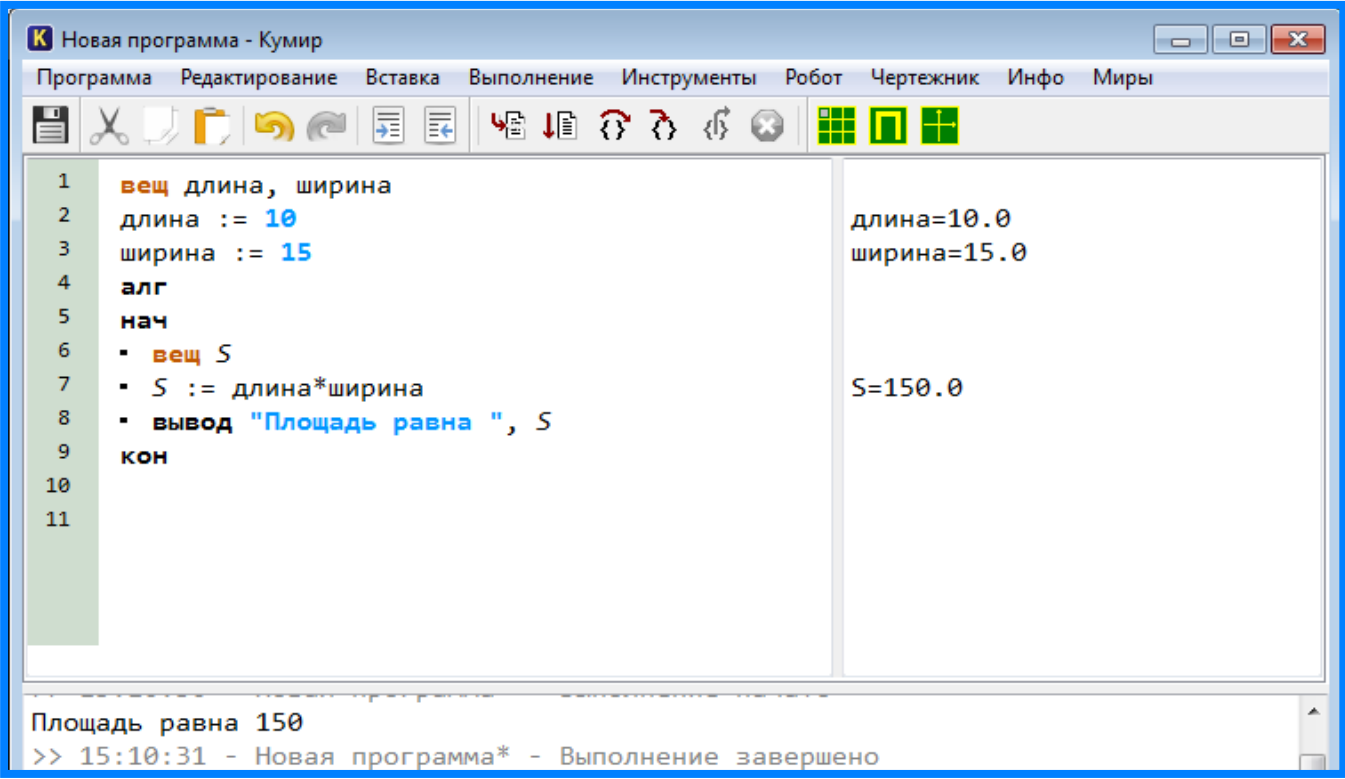
Школьный алгоритмический язык

алг <название алгоритма>

нач

<последовательность команд>

кон



The screenshot shows the 'Кумир' (KUMIR) programming environment. The window title is 'Новая программа - Кумир'. The menu bar includes 'Программа', 'Редактирование', 'Вставка', 'Выполнение', 'Инструменты', 'Робот', 'Чертежник', 'Инфо', and 'Мир'. The toolbar contains various icons for file operations, editing, and execution. The main text area contains the following code:

```
1  вещ длина, ширина
2  длина := 10
3  ширина := 15
4  алг
5  нач
6  ▪ вещ S
7  ▪ S := длина*ширина
8  ▪ вывод "Площадь равна ", S
9  кон
```

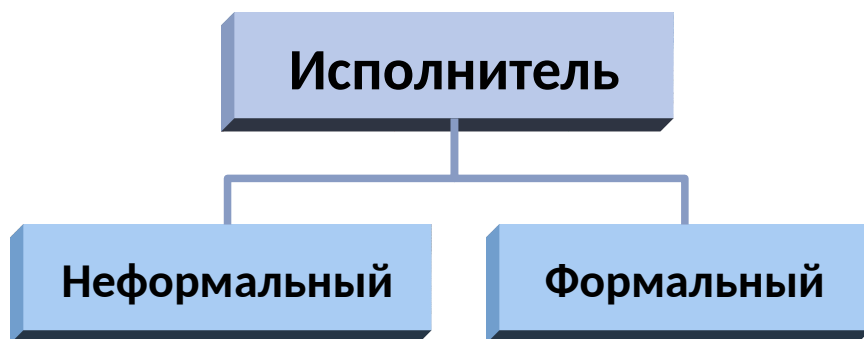
Line numbers 10 and 11 are visible but empty. To the right of the code, the current values of variables are displayed: 'длина=10.0', 'ширина=15.0', and 'S=150.0'. At the bottom of the window, the status bar shows 'Площадь равна 150' and '>> 15:10:31 - Новая программа* - Выполнение завершено'.

Исполнитель

Исполнитель — это некоторый объект (человек, животное, техническое устройство), способный выполнять определённый набор команд.



Формальный исполнитель ничего не знает о цели алгоритма. Он выполняет все полученные команды формально (не обдумывая). В информатике универсальным исполнителем алгоритмов является **компьютер**.



Характеристики формального исполнителя

Круг решаемых задач. Каждый исполнитель создаётся для решения некоторого круга задач.

Среда исполнителя. Область, обстановку, условия, в которых действует исполнитель, принято называть средой данного исполнителя.

Система команд исполнителя. Совокупность всех команд, которые могут быть выполнены некоторым исполнителем, образует систему команд данного исполнителя (СКИ).

Режимы работы исполнителя. Для большинства исполнителей предусмотрены режимы непосредственного управления и программного управления.

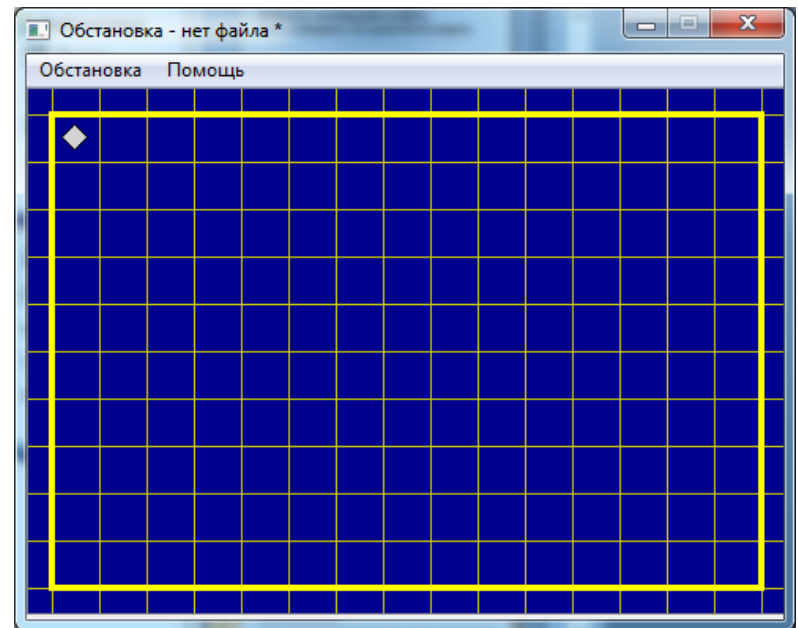
- В первом случае исполнитель ожидает команд от человека и каждую поступившую команду немедленно выполняет.
- Во втором случае исполнителю сначала задаётся полная последовательность команд (программа), а затем он выполняет все эти команды в автоматическом режиме.

Исполнитель Робот

Среда исполнителя: Клеточное поле, клетки могут отделяться друг от друга стенами.

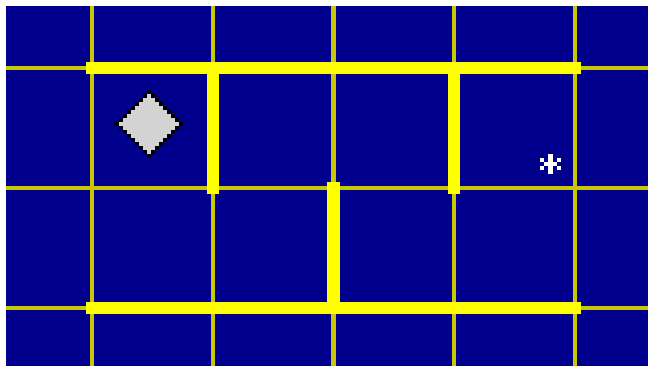
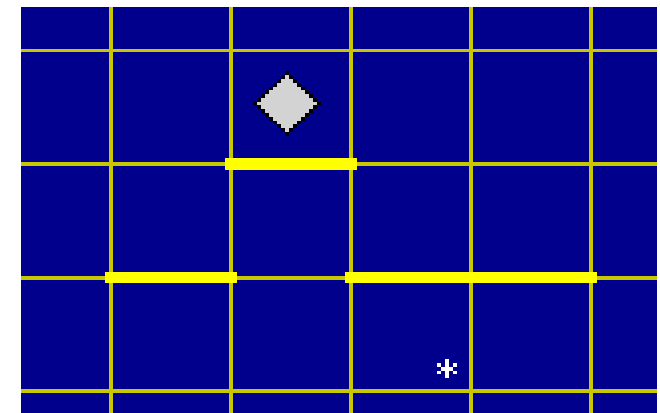
СКИ Робота:

вправо
влево
вверх
вниз
закрасить.



Исполнитель Робот

Задание. Переместить Робота в клетку, помеченную знаком «*», и закрасить данную клетку.



```
1  использовать Робот
2  алг
3  нач
4  ▪  вправо
5  ▪  вниз
6  ▪  влево
7  ▪  вниз
8  ▪  вправо
9  ▪  закрасить|
10 кон
11
```

Исполнитель Чертежник

Среда исполнителя: система

координат
СКИ:

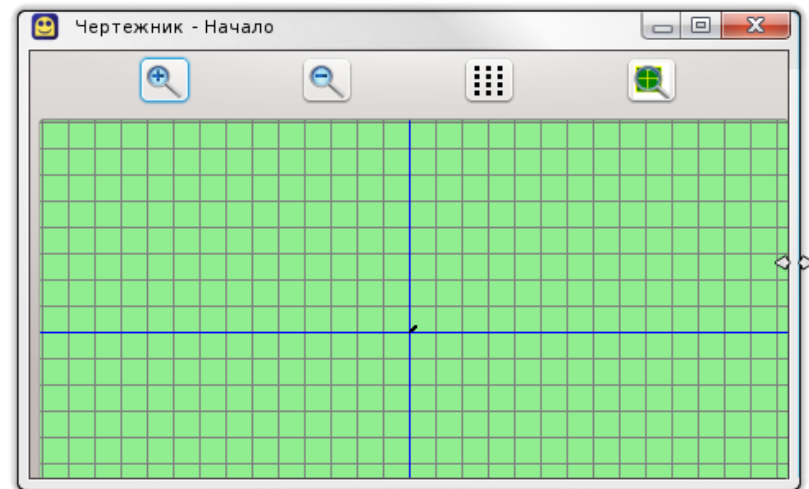
поднять перо - переводит чертежника в режим перемещения без рисования;

опустить перо - переводит чертежника в режим перемещения с рисованием;

сместиться на вектор (dX, dY) - перемещает перо на dX вправо и dY вверх;

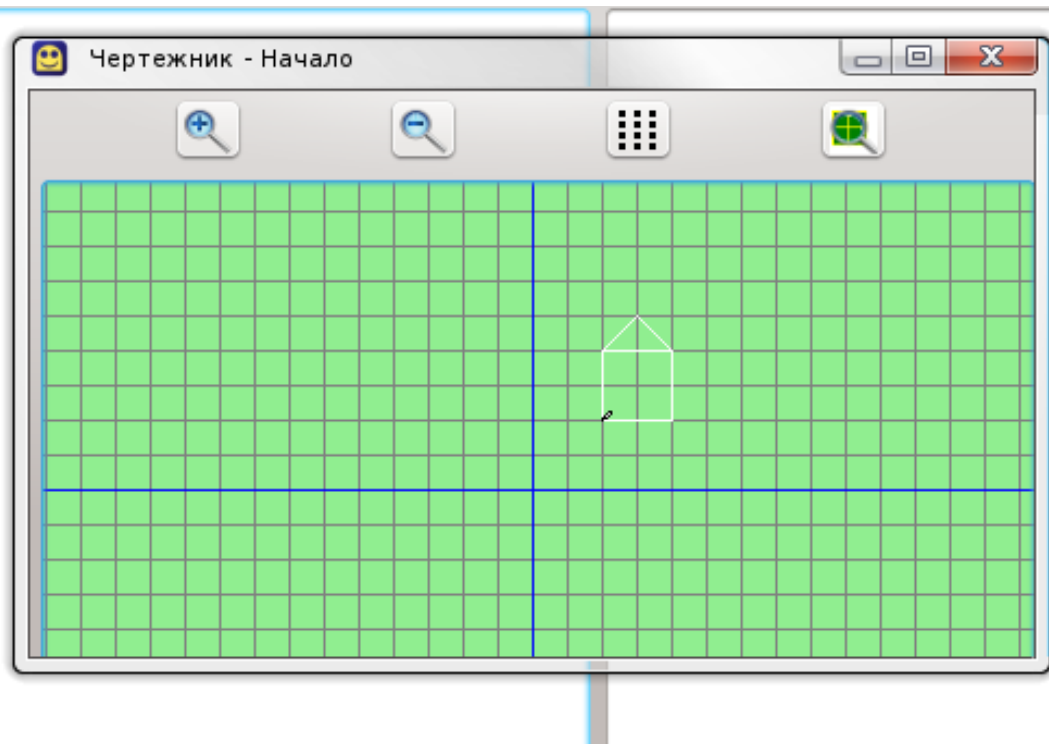
сместиться в точку (x, y) - перемещает перо в точку с координатами (x, y) ;

установить цвет - устанавливает цвет чернил.



Исполнитель Чертежник

```
1 использовать Чертежник
2 алг
3 нач
4   ▪ установить цвет ("белый")
5   ▪ сместиться в точку(2,2)
6   ▪ опустить перо
7   ▪ сместиться на вектор (0,2)
8   ▪ сместиться на вектор (1,1)
9   ▪ сместиться на вектор (1,-1)
10  ▪ сместиться на вектор (-2,0)
11  ▪ сместиться на вектор (2,0)
12  ▪ сместиться на вектор (0,-2)
13  ▪ сместиться на вектор (-2,0)
14  ▪ поднять перо
15 кон
16
```



Задание на дом

§ 2.1.1 и 2.1.3

Задание:

Чертежник:

Робот:

Закрасить

