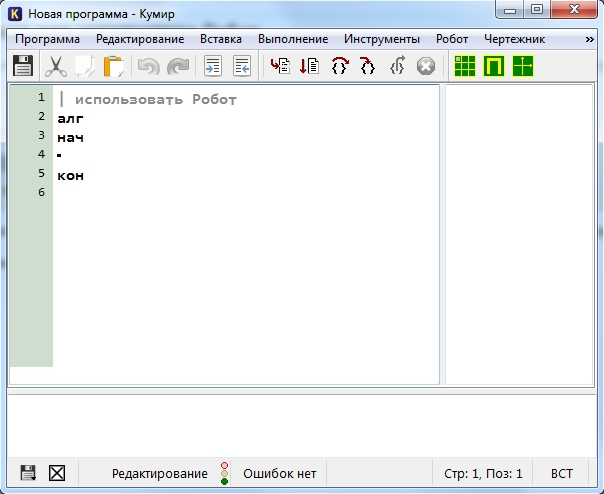
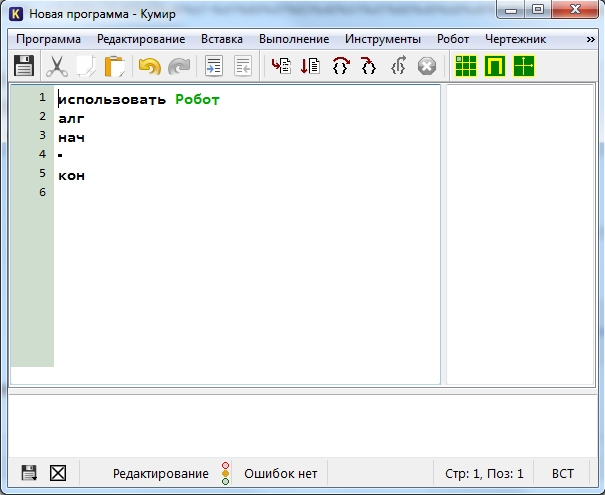
Запущенная программа Кумир выглядит так.



Первым делом мы должны раскомментировать первую строку нашей программы, убрав символ **|**



**Что такое стартовая обстановка**

Наш Робот находится в некой среде — это клетчатое поле, размер которого известен. Так же на этом поле могут находится стены и закрашенные клетки, а сам Робот может находится в любой клетке. Так вот — **стартовая обстановка** задает положение Робота на поле и расположение всех остальных элементов — стен, закрашенных клеток. И перед тем, как писать алгоритм для Робота необходимо задать стартовую обстановку. Насколько это важно давайте рассмотрим на примере. Пусть есть две стартовые обстановки:

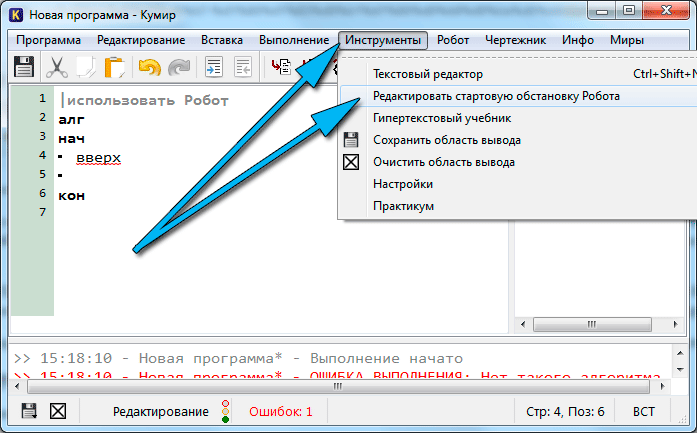
|  |  |
| --- | --- |
| Стартовая обстановка 1  Стартовая обстановка 1 | Стартовая обстановка 2  Стартовая обстановка 2 |

Отличаются они только тем, что в стартовой обстановке 2 справа от Робота находится стена.

Если наша программа начнется с команды, которая переместит Робота на одну клетку вправо (о простых командах Робота), то в первом случае (стартовая обстановка 1) Робот выполнит эту команду, а во втором программа завершится аварийно, так как Робот не может ходить сквозь стены. Получается, что одна и та же программа в первом случае работает, а во втором приводит к ошибке. Именно поэтому так важно задавать стартовую обстановку для Робота.

**Как задать стартовую обстановку**

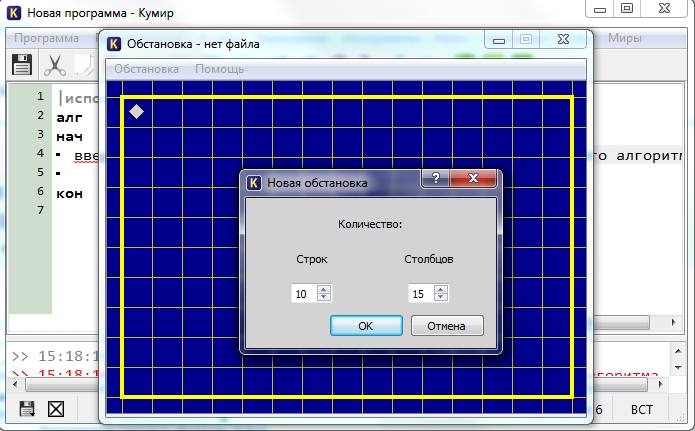
Запустив среду Кумир в меню ***Инструменты*** выбираем пункт ***Редактировать*** стартовую обстановку Робота



Редактировать стартовую обстановку Робота

Откроется окно с синим фоном. Это и есть **стартовая обстановка Робота**. И мы ее можем изменить.

По-умолчанию, размер окна 10 на 15 клеток. Если нам необходимо изменить количество строк и столбцов, то щелкаем ***Обстановка*** -> ***Новая обстановка*** и задаем необходимые значения

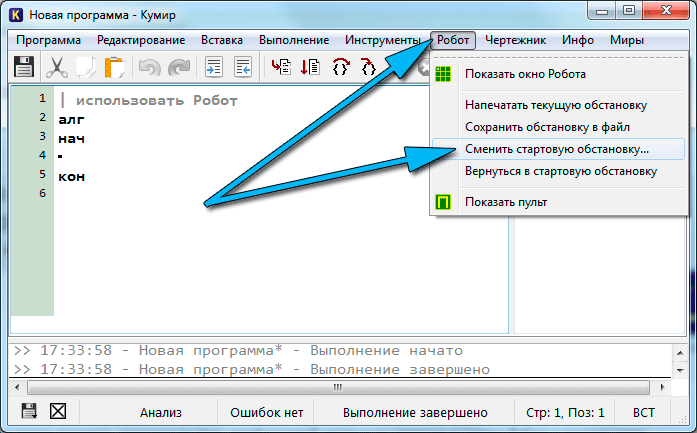


Задаем размер поля

Далее,

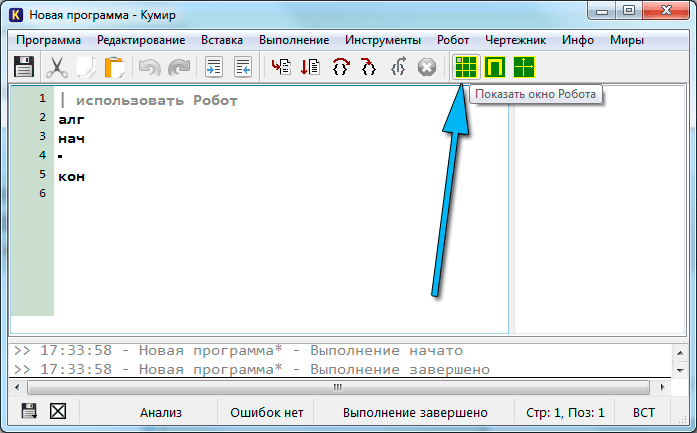
* чтобы переместить Робота в новую позицию, щелкаем по нему левой кнопкой мыши и не отпуская ее тащим Робота в нужное место.
* чтобы добавить/удалить стену, щелкаем левой кнопкой мыши по границе клетки.
* чтобы закрасить/очистить клетку, щелкаем по ней левой кнопкой мыши
* чтобы добавить или убрать точку в клетку щелкаем по клетке, удерживая клавишу ***Ctrl***

После того, как мы задали нужную стартовую обстановку, ее необходимо сохранить (***Обстановка*** -> ***Сохранить*** или ***Обстановка*** -> ***Сохранить как***). После этого закрываем окно **Обстановка** и в основном окне программы выбираем ***Робот*** -> ***Сменить стартовую обстановку***



Сменить стартовую обстановку Робота

Находим сохраненную ранее обстановку и загружаем ее. После этого убедимся, что загрузили правильную стартовую обстановку, щелкнув по кнопке ***Показать окно Робота***



Показать окно Робота

Если в окне с зеленым фоном (текущая обстановка Робота) вы увидите вашу обстановку, то можно переходить к написанию алгоритма, используя [простые команды Робота](http://easyinformatics.ru/ispolnitel-robot/ispolnitel-robot-prostye-komandy).

У любого исполнителя должна быть система команд (**СКИ** — **система команд исполнителя**). **Система команд исполнителя** — совокупность всех команд, которые может выполнить исполнитель. В качестве примера рассмотрим дрессированную собаку. Она умеет выполнять некоторые команды — «Сидеть», «Лежать», «Рядом» и т. п. Это и есть ее система команд.

**Простые команды Робота**

У нашего Робота тоже есть система команд. Сегодня мы рассмотрим **простые команды Робота**. Всего их 5:

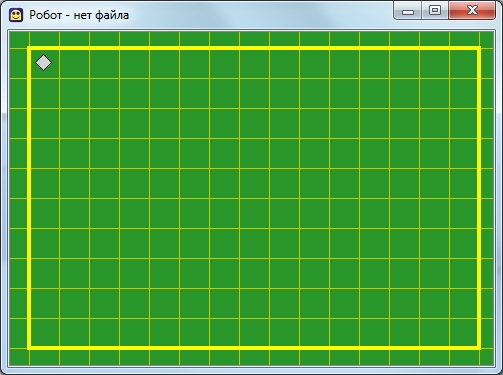
* вверх
* вниз
* влево
* вправо
* закрасить

Результат выполнения этих команд понятен из их названия:

1. **вверх** — переместить Робота на одну клетку вверх
2. **вниз** — переместить Робота на одну клетку вниз
3. **влево** — переместить Робота на одну клетку влево
4. **вправо** — переместить Робота на одну клетку вправо
5. **закрасить** — закрасить текущую клетку (клетку в которой находится Робот).

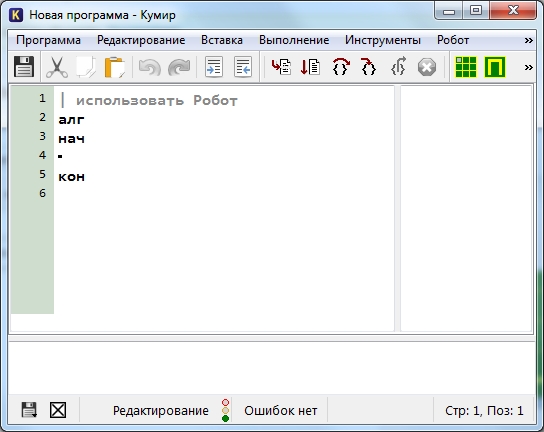
Эти команды можно писать с клавиатуры, а можно использовать горячие клавиши (нажав их команды будут вставляться автоматически):

* вверх — Escape, Up (стрелка вверх)
* вниз — Escape, Down (стрелка вниз)
* влево — Escape, Left (стрелка влево)
* вправо — Escape, Right (стрелка вправо)
* закрасить — Escape, Space (пробел)
* Обратите внимание, что набирать нужную комбинацию горячих клавиш нужно не привычным нам способом! Мы привыкли нажимать клавиши одновременно, а здесь их нужно **нажимать последовательно**. К примеру, чтобы ввести команду вверх, нужно нажать Escape, отпустить ее и после этого нажать стрелку вверх. Это нужно помнить.
* Теперь мы готовы написать первый алгоритм для Робота. Предлагаю начать с простого — нарисуем квадрат со стороной 3 клетки. Поехали!
* Запускаем Кумир, [настраиваем](http://easyinformatics.ru/ispolnitel-robot/nastrojka-sredy-kumir-dlya-ispolnitelya-robot) его. Можно начинать писать программу? Конечно нет! Мы же не [задали стартовую обстановку](http://easyinformatics.ru/ispolnitel-robot/startovaya-obstanovka-robota)! Делаем это. Предлагаю использовать вот такую:



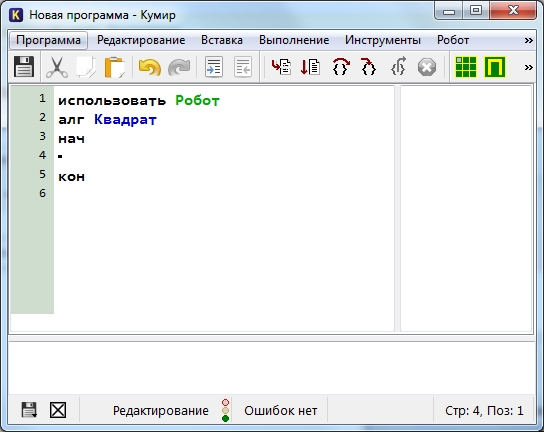
Стартовая обстановка Робота

Вот теперь все готово. Начинаем писать программу. Пока она выглядит так



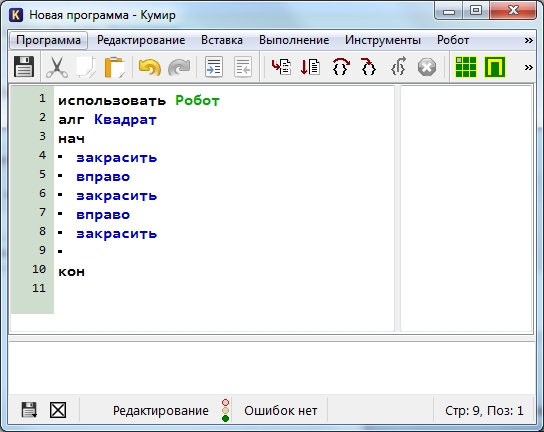
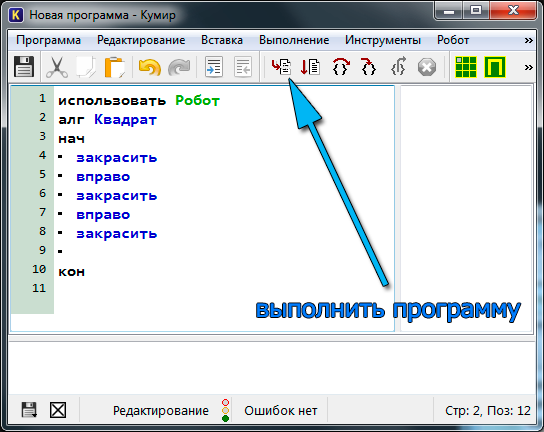
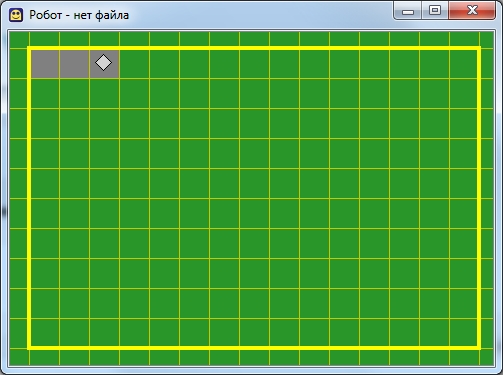
**Первая программа для Робота**

Удаляем символ «|» и называем наш алгоритм «Квадрат»

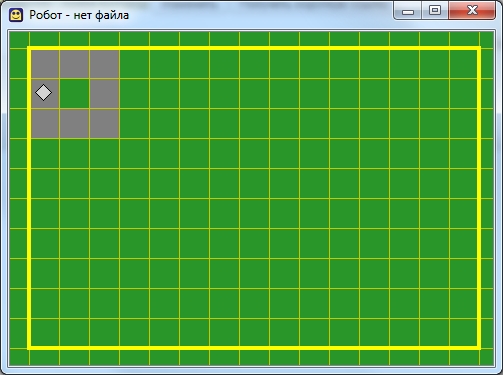


Алгоритм квадрат

Предлагаю рисовать квадрат, двигаясь по часовой стрелке. Для начала закрасим текущую клетку, дав команду **закрасить**. Потом делаем шаг вправо и опять закрашиваем клетку. И еще раз шаг вправо и закрасить.

* 
* Первые шаги
* Попробуем запустить программу и посмотреть что же получилось. Для запуска нажимаем **F9** или же кнопку на панели инструментов
* 
* выполнить программу
* В результате мы должны увидеть вот такую картину
* 
* Первый результат
* Если такое окно Робота у вас не появилось, то на панели инструментов щелкните «**Показать окно Робота**» или в меню Робот выберите пункт «**Показать окно Робота**«. Продолжаем дальше.
* Теперь мы будем двигаться вниз и закрашивать правую сторону квадрата:
* *вниз*
* *закрасить*
* *вниз*
* *закрасить*
* Потом пойдем влево, закрашивая нижнюю границу квадрата
* *влево*
* *закрасить*
* *влево*
* *закрасить*
* У нас осталась одна незакрашенная  клетка. Закрасим ее
* *вверх*
* *закрасить*
* Все готово! В итоге наша программа выглядит так:
* **использовать Робот**
* **алг Квадрат**
* **нач**
* **закрасить**
* **вправо**
* **закрасить**
* **вправо**
* **закрасить**
* **вниз**
* **закрасить**
* **вниз**
* **закрасить**
* **влево**
* **закрасить**
* **влево**
* **закрасить**
* **вверх**
* **закрасить**
* **кон**

А результат ее работы вот так



Итак, сегодня мы с вами написали программу, используя **простые команды Робота**. Рекомендую попрактиковаться самостоятельно — придумать себе задание и написать программу. Это могут быть самые различные фигуры, узоры, буквы. К примеру, попробуйте написать программу, рисующую букву П, Р, Ш, Щ, М. А если получится и захотите поделиться — комментируйте и прикрепляйте результат к комментарию.

Продолжаем [изучение исполнителя Робот](http://easyinformatics.ru/ispolnitel-robot) и сегодня поговорим о циклах. Давайте разберемся, что же такое цикл и как научить выполнять циклические алгоритмы нашего Робота.

Итак, **что такое цикл**? Представьте, что мы находимся на уроке физической культуры и перед нами стоит задача **сделать 7 приседаний**. Это задание можно оформить в виде линейного алгоритма и тогда оно будет выглядеть примерно так:

сделай приседание

сделай приседание

сделай приседание

сделай приседание

сделай приседание

сделай приседание

сделай приседание

Т. е мы повторили команду сделай приседание 7 раз. А есть ли смысл писать 7 одинаковых команд? Может проще дать команду **сделай 7 приседаний**? Конечно проще и правильнее. **Это и есть цикл**. Вы можете сами вспомнить примеры циклов из жизни — их довольно много.

Таким образом **линейный алгоритм**, где повторяются одни и те же команды мы можем оформить **в виде циклического алгоритма** — примерно так:

повторяй 7 раз

сделай приседание

конец цикла

Вот так, на придуманном нами языке мы оформили цикл. У исполнителя Робот тоже есть возможность записывать циклы. Причем, **циклы бывают разные**. Тот вариант, который мы только что рассмотрели называется **цикл со счетчиком** или **цикл с параметром**.

## Виды циклов.

### Цикл со счетчиком.

**Цикл со счетчиком** применяется когда заранее известно сколько повторений необходимо сделать. В примере выше с приседаниями именно такой случай.

Для того, чтобы написать цикл со счетчиком для исполнителя необходимо знать его синтаксис. А он такой:

нц <количество повторений> раз

<команда 1>

<команда 2>

…

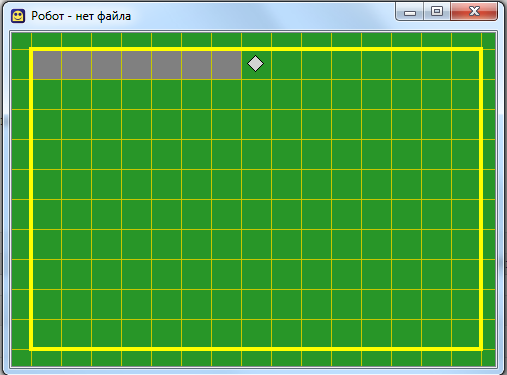
<команда n>

кц

Здесь мы должны указать количество повторений (число) и команды, которые будут повторяться.  Команды, которые повторяются в цикле называют **телом цикла**.

Давайте рассмотрим это на примере.

Закрасим 7 клеток, как на рисунке. Рекомендую почитать про [стартовую обстановку Робота](http://easyinformatics.ru/ispolnitel-robot/startovaya-obstanovka-robota) и про его [простые команды](http://easyinformatics.ru/ispolnitel-robot/ispolnitel-robot-prostye-komandy).

[](http://easyinformatics.ru/wp-content/uploads/2014/03/zadanie_1.png)

Задание на цикл со счетчиком

Изначально Робот находился в левой верхней клетке.

Давайте для начала решим задачу линейно. В этом случае мы будет закрашивать текущую клетку и перемещаться на 1 клетку вправо и программа будет выглядеть так:  
использовать Робот  
алг  
нач

закрасить

вправо

закрасить

вправо

закрасить

вправо

закрасить

вправо

закрасить

вправо

закрасить

вправо

закрасить

вправо

кон

Как видим, команды закрасить и вправо повторяются 7 раз. Давайте теперь перепишем программу с использованием цикла. Кстати, чтобы вставить цикл в свою программу можно в меню **Вставка** выбрать пункт **нц-раз-кц** или нажать одну из комбинаций клавиш **Esc, Р** (русская буква Р) или **Esc, H** (латинская буква H). Причем клавиши надо нажимать последовательно — сначала Esc, отпустить ее и только потом Р или H.

Так вот, наша **программа с циклом** будет выглядеть так:

использовать Робот

алг

нач

нц 7 раз

закрасить

вправо

кц

кон

Если мы ее запустим, то увидим, что в результате получится тоже самое — 7 закрашенных клеток. Однако программа стала короче и значительно грамотней с алгоритмической точки зрения!

В качестве разминки и закрепления предлагаю самостоятельно написать программу для Робота, которая нарисует квадрат со стороной 7 клеток. Естественно, используя цикл. Жду решения в комментариях.

### Цикл с условием.

При решении задачи 19 ГИА по информатике с Роботом использовать цикл со счетчиком не получится. Так как там поле, как правило, бесконечное и стены не имеют конкретной длины. Поэтому мы не сможем определить количество повторений для цикла со счетчиком. Но не беда — нам поможет **цикл с условием**.

Вернемся к физкультуре и изменим задачу. Ведь кто-то может и не сделать 7 приседаний, а другой способен сделать 27. Можно ли учесть это при создании цикла? Конечно. Только теперь мы будем использовать не счетчик (количество повторений), а условие. К примеру, пока не устал, делай приседания. В этом случае человек будет делать не конкретное число приседаний, а приседать до тех пор, пока не устанет. И наш цикл на абстрактном языке примет такой вид:

пока ***не устал***

сделай приседание

конец цикла

Слова не устал в нашем случае — это условие. Когда оно истинно, цикл выполняется. Если же оно ложно (устал) тело цикла не будет выполнено. У исполнителя Робот есть несколько условий

**сверху свободно**

**снизу свободно**

**слева свободно**

**справа свободно**

**сверху стена**

**снизу стена**

**слева стена**

**справа стена**

, но в условии задачи 19 ГИА указаны только первые 4. Так что будем пользоваться только ими.

Теперь давайте решим следующую задачу для Робота — нарисовать вертикальную линию от левой до правой границы поля использую цикл с условием. Изначально Робот находится в левом верхнем углу.

Давайте сначала сформулируем словесный алгоритм — т. е. опишем словами что нужно делать Роботу. Этот алгоритм будет звучать примерно так:

«***Пока справа свободно делай шаг вправо и закрашивай клетку***»

В результате Робот пробежит по всем клеткам вправо и будет их закрашивать до тех пор, пока справа не окажется стена.

Исходный код нашей программы для Робота будет примерно такой:

использовать Робот

алг

нач

нц пока справа свободно

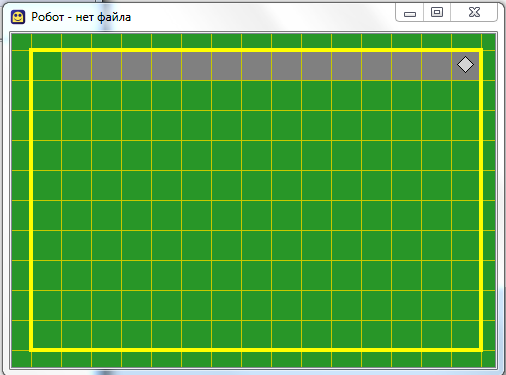
вправо

закрасить

кц

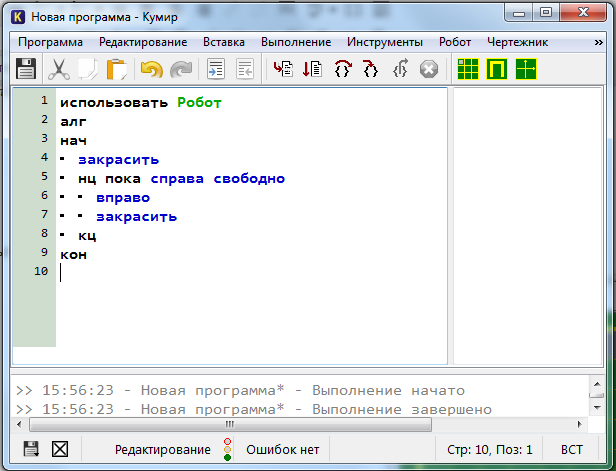
кон

В результате выполнения этой программы мы увидим вот такую картину:

[](http://easyinformatics.ru/wp-content/uploads/2014/03/zadanie_2.png)

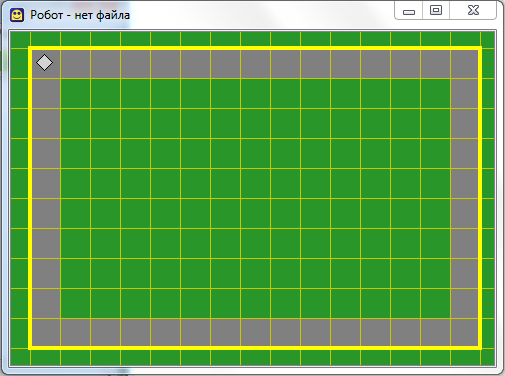
Задание для Робота цикл с условием

Как видим, не хватает только закрашенной первой клетки. Для этого перед циклом необходимо выполнить команду **закрасить**.

[](http://easyinformatics.ru/wp-content/uploads/2014/03/reshenie.png)

Решение задачи с циклом

Для закрепления прошу написать программу, которая будет делать рамку вокруг рабочего поля Робота независимо от его размера. Конечно же с использованием циклов с условием. В итоге должно получиться так:

[](http://easyinformatics.ru/wp-content/uploads/2014/03/samostoyatelno.png)

Задача для самостоятельного решения.