

Работа с системой программирования КуМир (Комплект Учебных Миров)

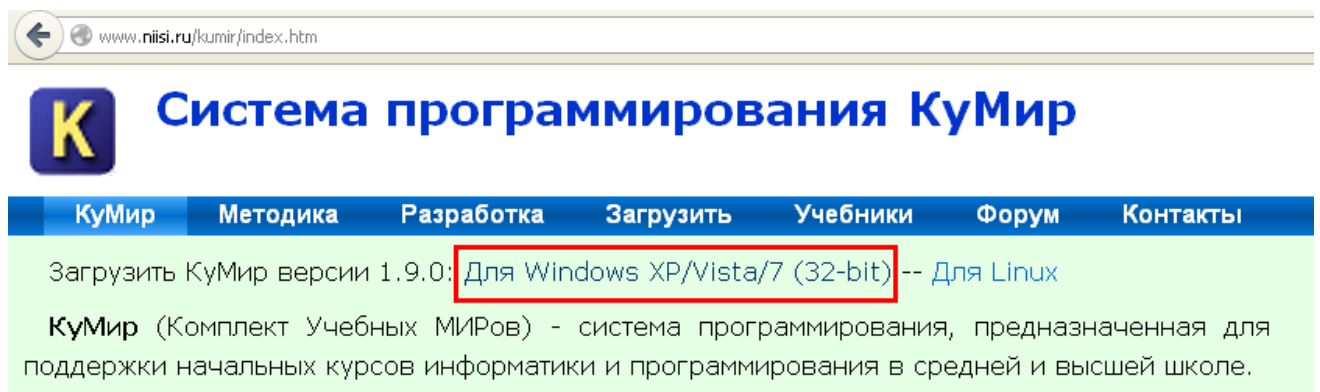
1. Установка КуМир
2. [Знакомство с исполнителем Робот](#)
3. [Основные команды исполнителя Робот](#)
4. [Циклы, условия, примеры решения задач.](#)
5. [Задачи для самостоятельной работы](#)

1. Установка КуМир

Чтобы установить систему программирования КуМир, необходимо сначала перейти на официальный сайт разработчиков этой системы и скачать установочный дистрибутив.

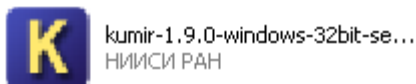
Сайт находится по адресу: <http://www.niisi.ru/kumir/>

И выглядит следующим образом:

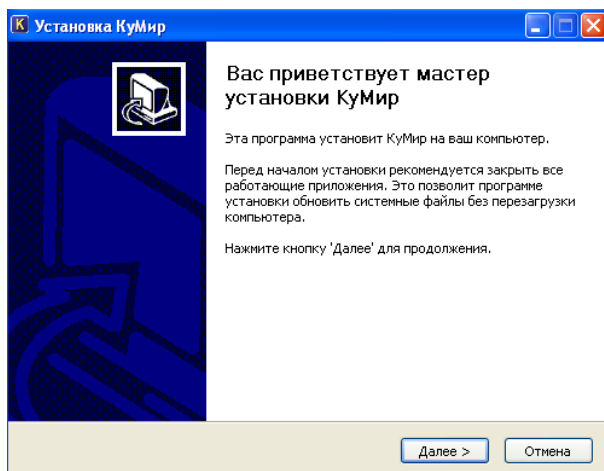


Если нажать на надпись, выделенную красным прямоугольником, то начнется загрузка установщика текущей версии программы.

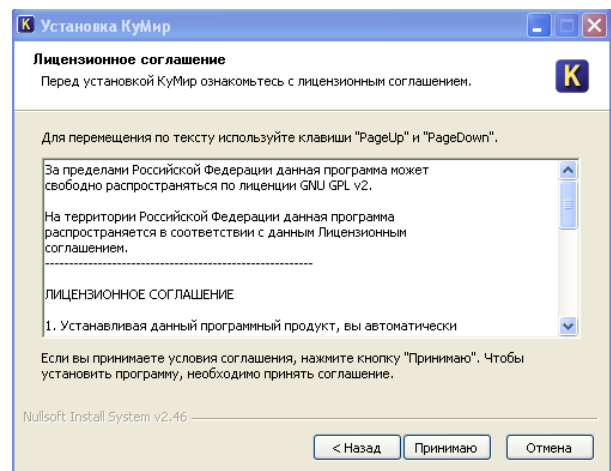
Так будет выглядеть скачанный файл:



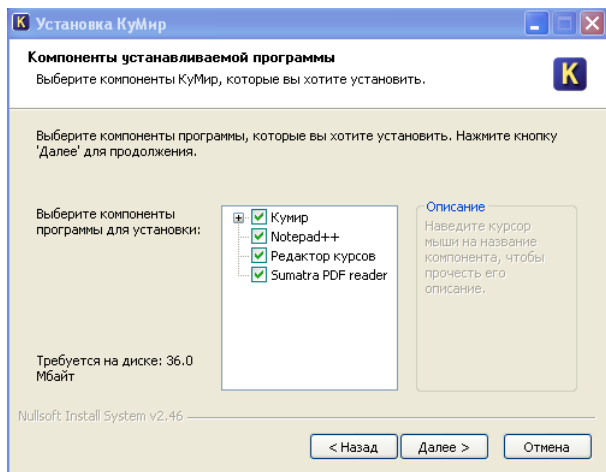
Запускаем установку:



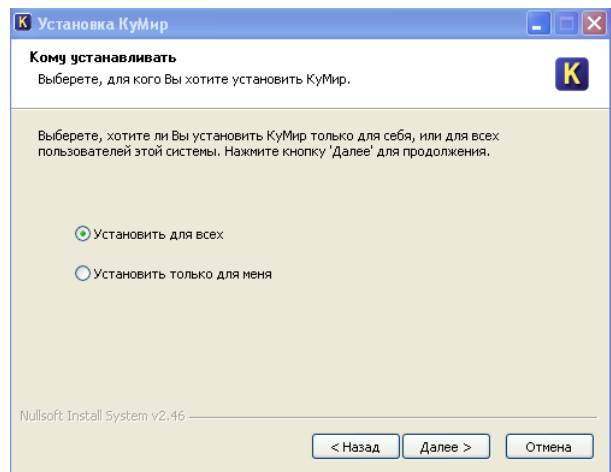
1. Нажимаем «Далее»



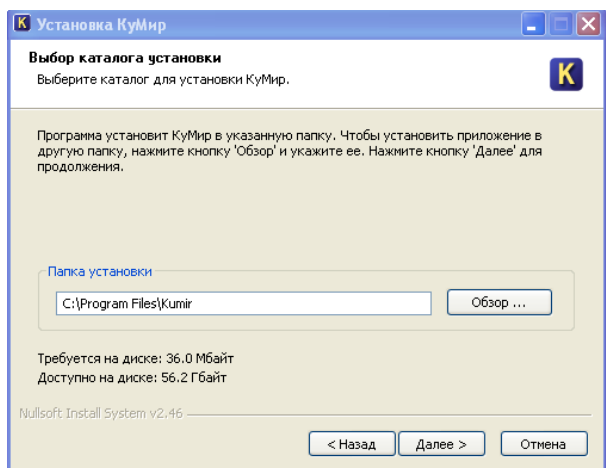
2. Нажимаем «Принимаю»



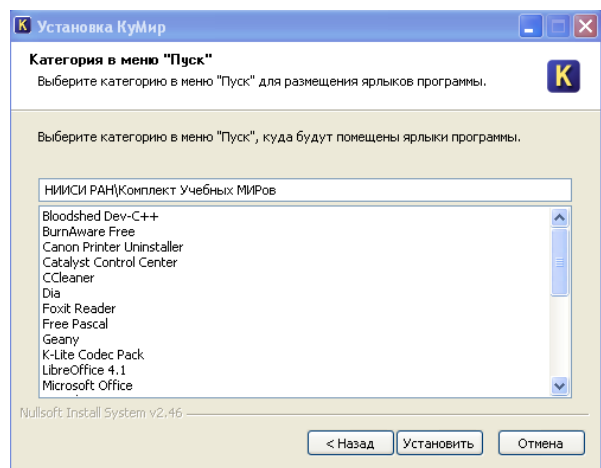
3. Нажимаем «Далее»



4. Нажимаем «Далее»

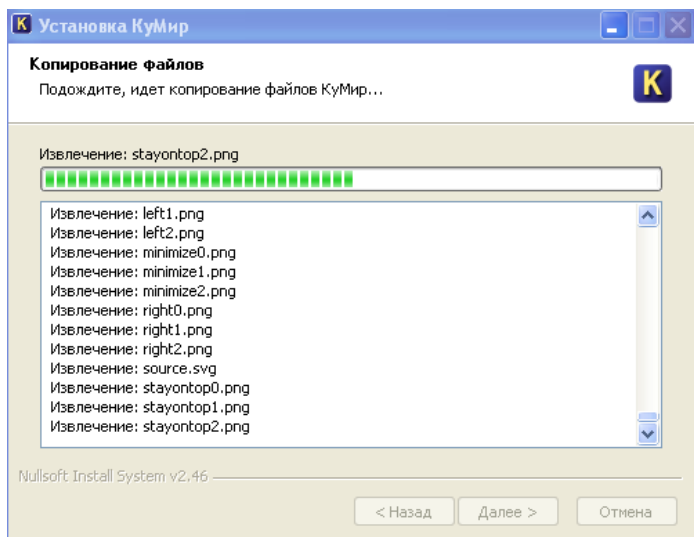


5. Нажимаем «Далее»



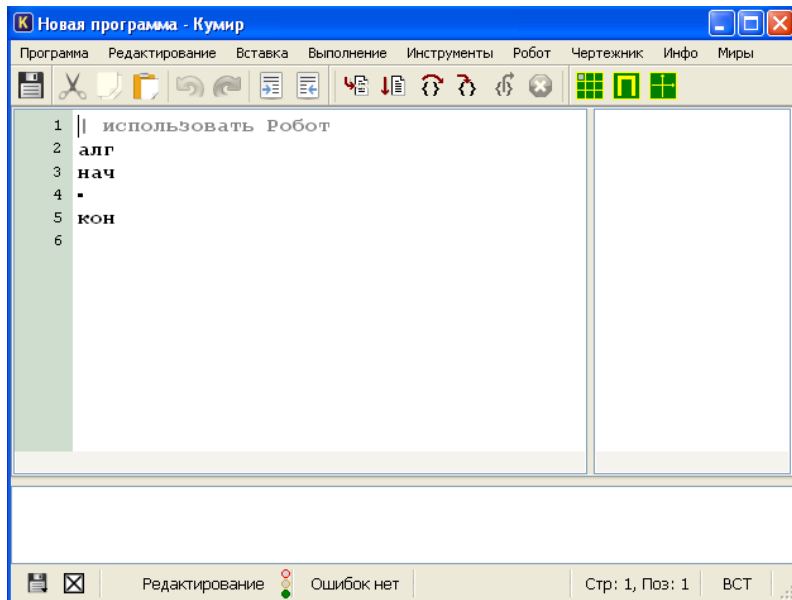
6. Нажимаем «Установить»

7. Должна начаться установка программы:



В конце установки нужно нажать кнопку «Готово».

Находим ярлык программы на рабочем столе или в меню «Пуск» и пробуем запустить среду КуМир:

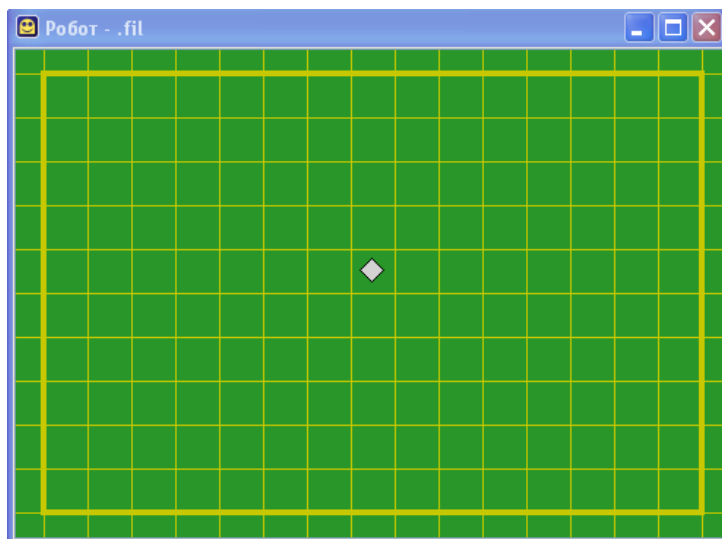


2. Знакомство с исполнителем Робот

Робот – один из исполнителей алгоритмов, доступный в системе программирования КуМир. Чтобы открыть окно робота, нужно нажать на кнопку на панели инструментов, указанную ниже на скриншоте:

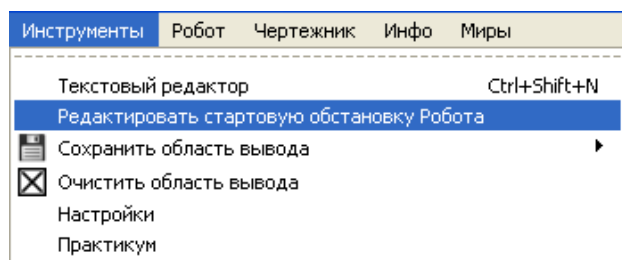


Должно открыться следующее окно:

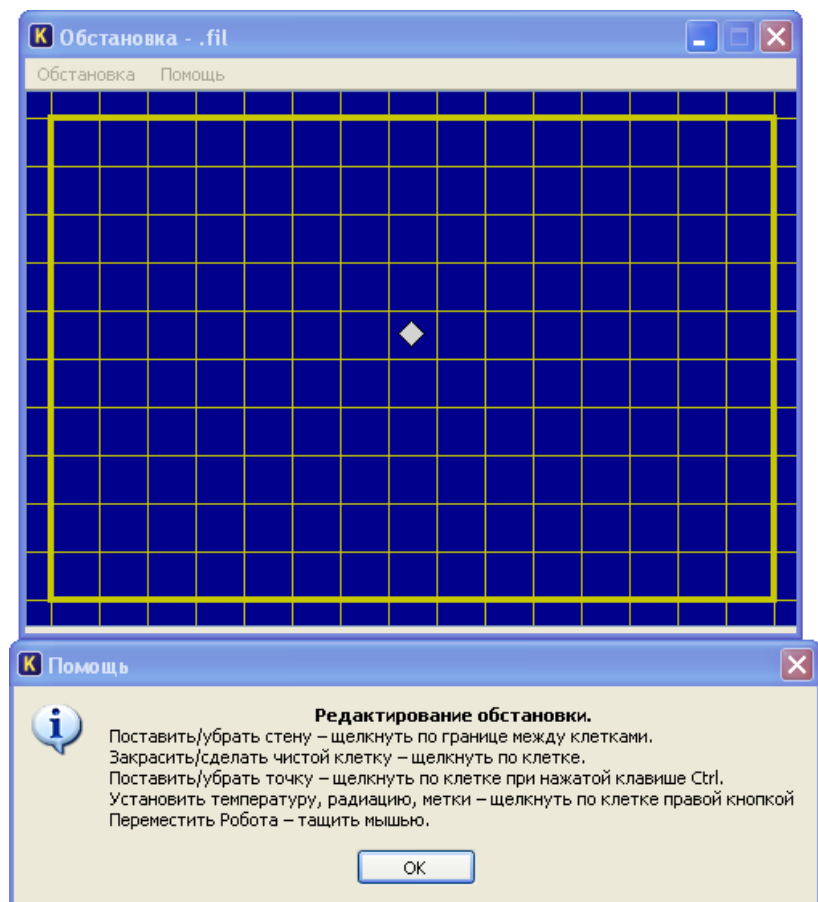


Как вы видите, робот «живет» на прямоугольном поле, разделённом на ячейки. Сам робот – это маленький ромбик в центре этого поля. Это поле по-другому называется **стартовой обстановкой** робота и сейчас мы научимся, как её менять.

Итак, заходим в меню «Инструменты» - «Редактировать стартовую обстановку робота».

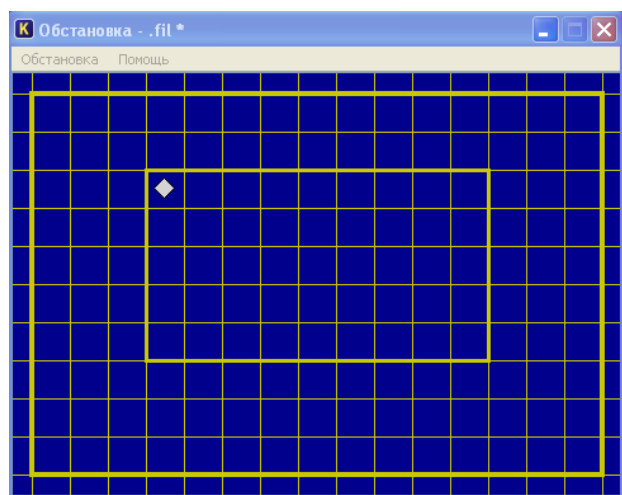


Должно открыться следующее окно:



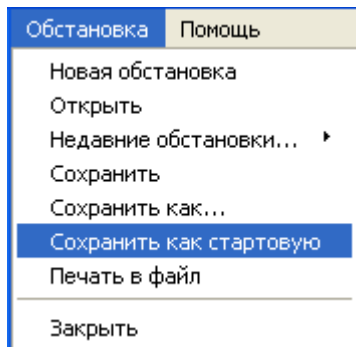
Как видно из раздела «помощь», чтобы добавить препятствия на поле робота, нужно щелкнуть по границе между ячейками, а чтобы переместить робота, необходимо щелкнуть на нём левой кнопкой мыши и перетащить в другое место.

Попробуем изменить обстановку следующим образом:

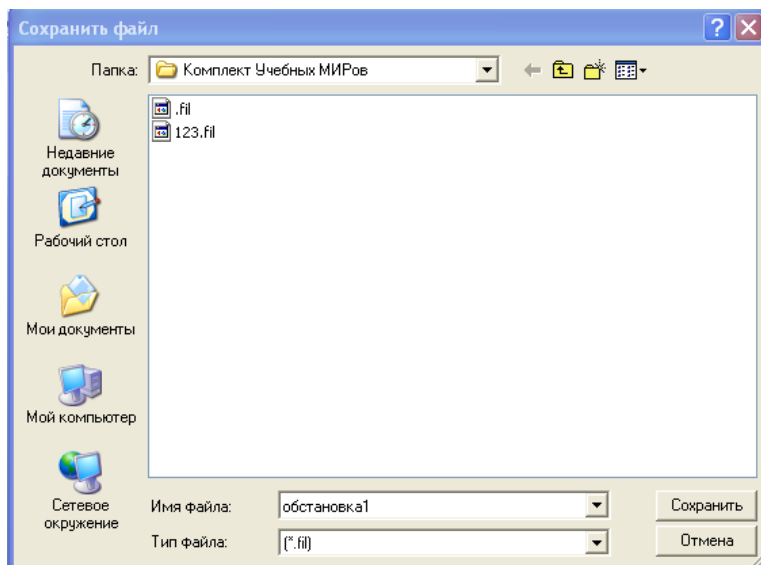


Теперь осталось сохранить эту обстановку.

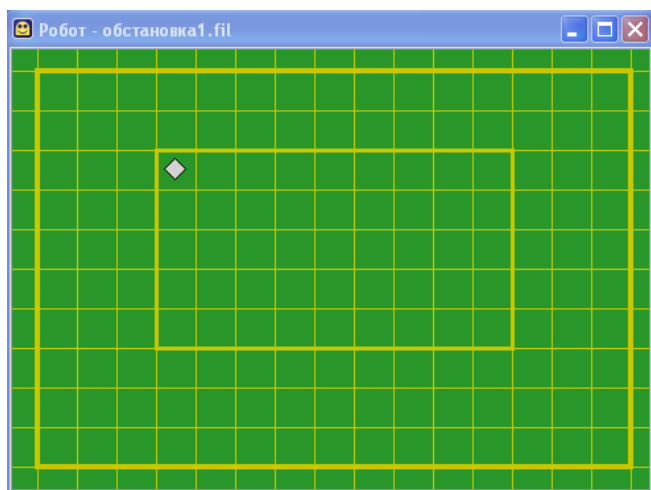
Выбираем пункт меню «Обстановка» - «Сохранить как стартовую»



И выбираем произвольную папку для сохранения и имя файла, жмём «сохранить»:



Теперь, если ещё раз открыть текущую обстановку робота, то мы увидим, что произошли изменения:



Обратите внимание, что если открыто синее поле, то вы изменяете стартовую обстановку робота. Если же поле зелёное, то это текущее поле для работы с роботом.

Не путайте их!

3. Основные команды исполнителя Робот

Примечание: чтобы научиться хорошо работать с роботом, набирайте и выполняйте все алгоритмы, приведённые в данном пособии.

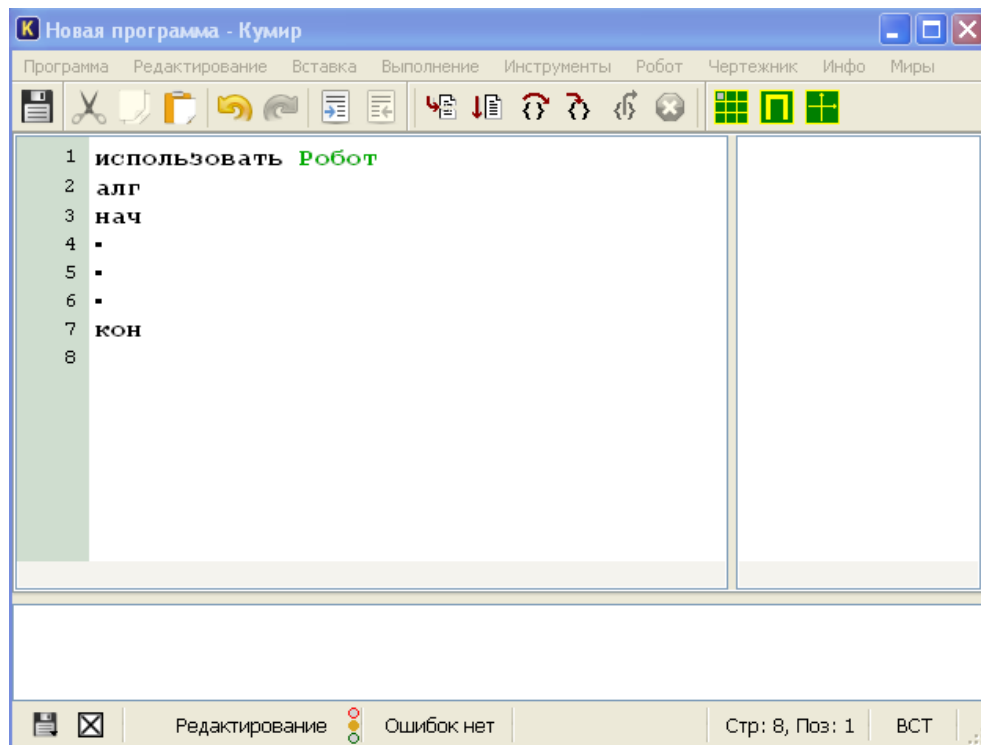
Как мы определились ранее, робот – это исполнитель алгоритмов. И как у каждого исполнителя, у него есть своя система команд. То есть набор команд, которые робот понимает.

Вот основные команды робота:

вверх	перемещает робота на одну клетку вверх
вниз	перемещает робота на одну клетку вниз
влево	перемещает робота на одну клетку влево
вправо	перемещает робота на одну клетку вправо
закрасить	закрашивает ячейку, в которой стоит робот

Как вы уже поняли, первые 4 команды нужны для перемещения робота по полю. Итак, попробуем эти команды в действии.

В КуМире напишем следующий текст (точки в строках 4, 5, 6 ставятся автоматически):



Команда «**Использовать Робот**» позволяет указать системе КуМир, что мы будем работать с исполнителем робот. Если мы не напишем эту команду, КуМир не сможет распознать команды, которые мы будем вводить.

Запись

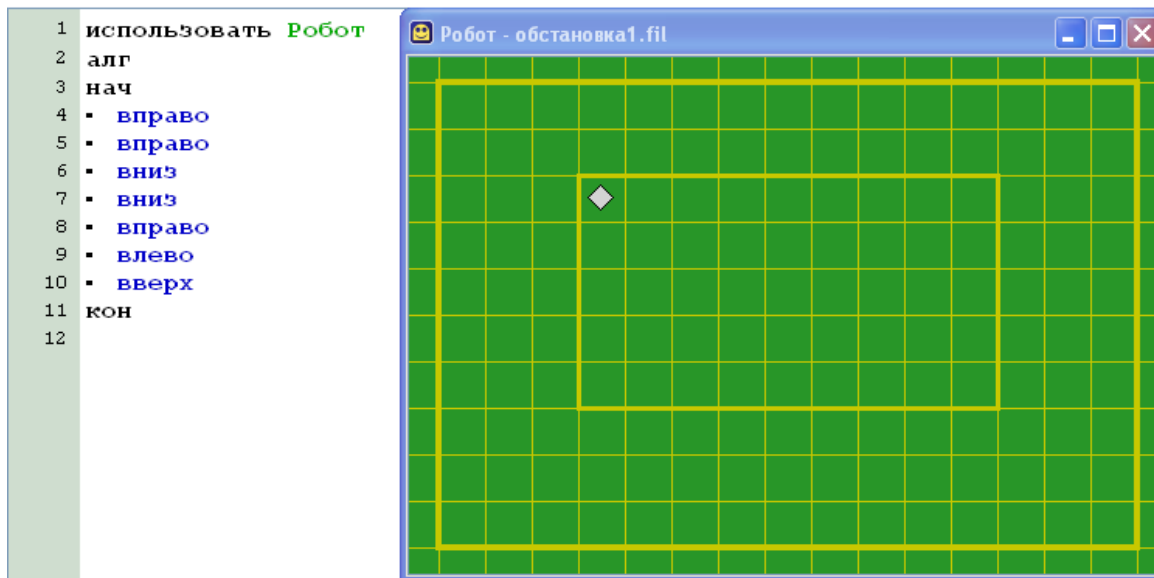
алг

нач

кон

также обязательна и обозначает: **алгоритм**, **начало**, **конец**. Все команды нашего алгоритма для робота мы будем писать между ключевыми словами **нач** и **кон**.

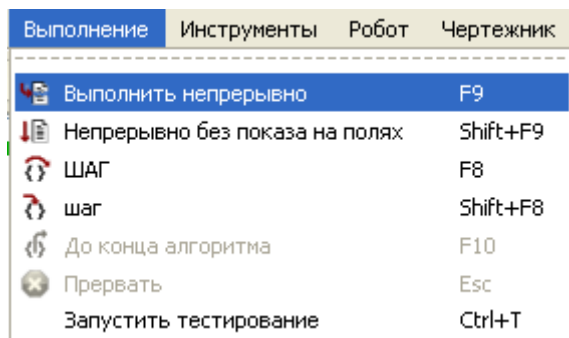
Итак, реализуем простейший алгоритм для робота.



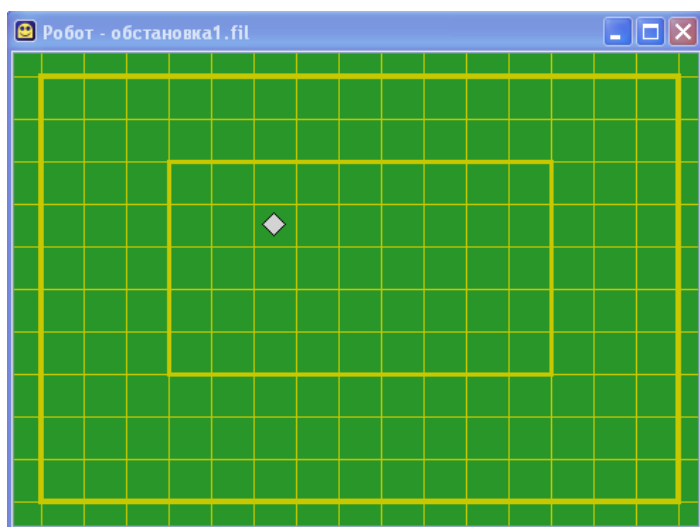
Обратите внимание, что если команды написаны верно, то они подсвечиваются синим цветом. (нельзя написать команды «в низ», «право» или «верх», будьте внимательны!).

Выполнить алгоритм вы можете двумя способами:

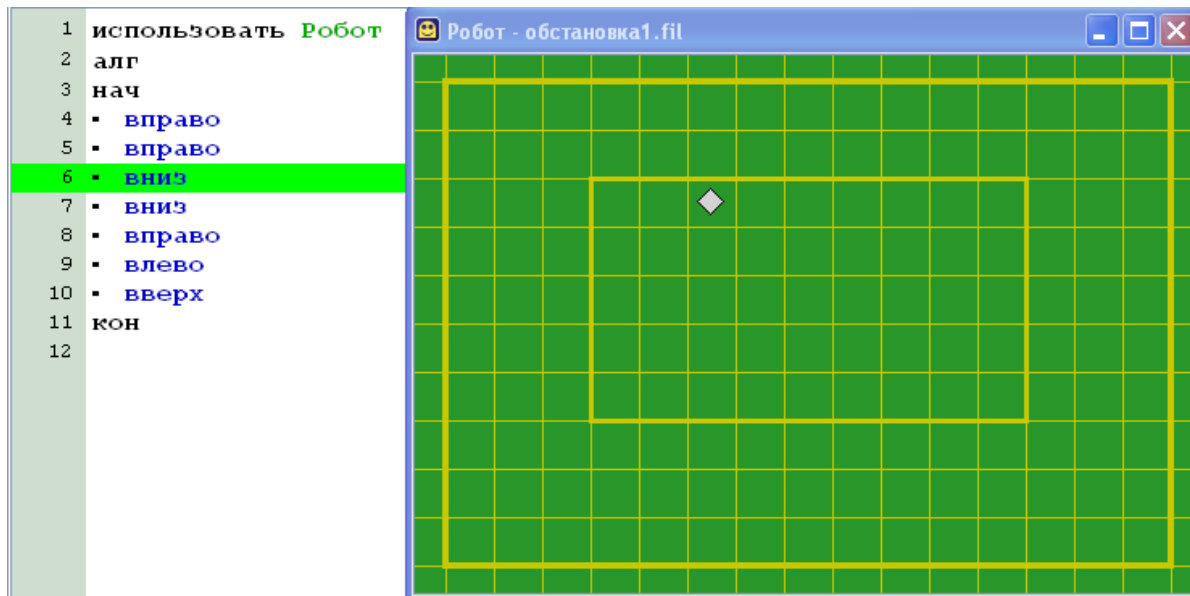
1. **По шагам** – выполнить одну команду алгоритма и остановиться
2. **Непрерывно** – выполнить все команды алгоритма сразу.



Попробуем запустить непрерывное выполнение алгоритма, робот перейдет в конечное положение:



Чтобы понять, как «ходит» робот, можно попробовать пошаговое выполнение: для этого нажимайте клавишу F8 или выберите пункт меню «Выполнение» - «ШАГ».

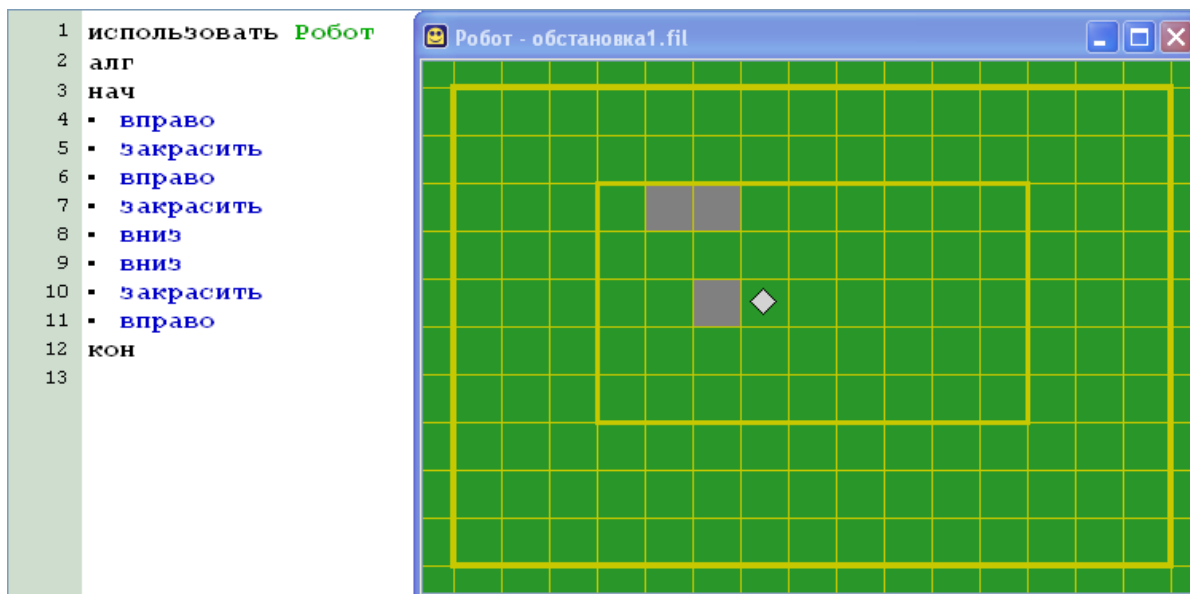


Робот сделал 2 команды «вправо» и остановился на команде «вниз». Если ещё раз нажать F8 (один шаг алгоритма), то робот шагнёт вниз и остановится на следующей команде.

Таким образом, много раз нажимая F8 мы можем выполнить все шаги алгоритма.

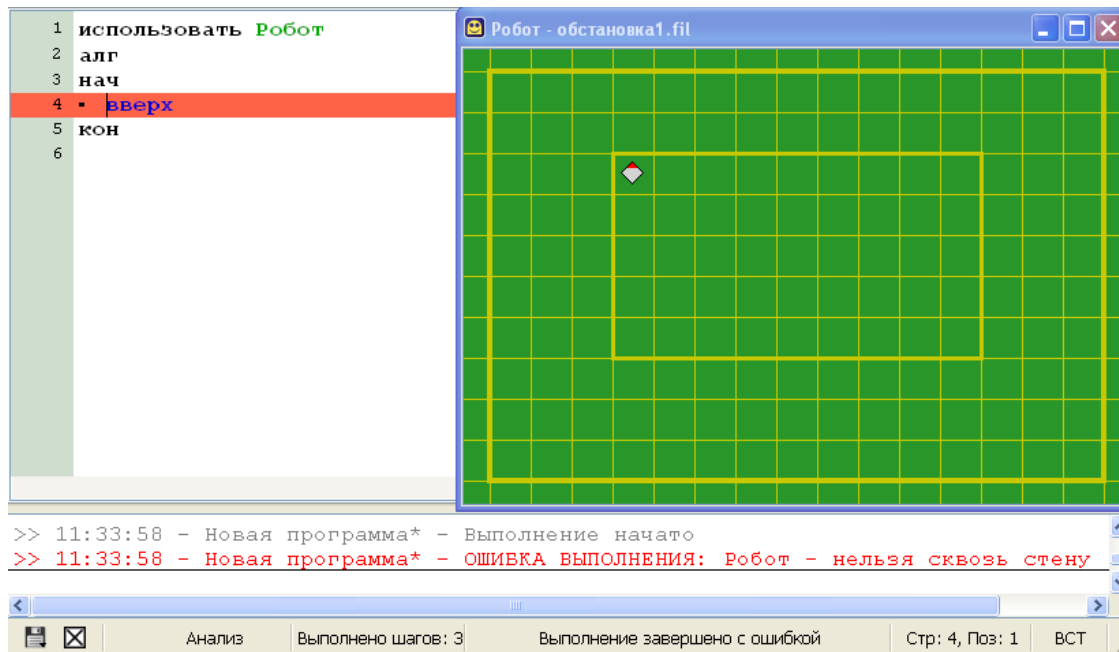
Обратите внимание, что выполнение алгоритма всегда начинается из той точки, в которой изначально стоял робот.

Попробуем выполнить следующий алгоритм:



Робот стоял в левом верхнем углу прямоугольника. После непрерывного выполнения алгоритма он закрасит 3 клетки и окажется в клетке, указанной на скриншоте.

Что же произойдет, если мы попытаемся шагнуть «вверх» из начального положения, то есть пойти «в стенку»?



Как видно на скриншоте выше, робот разбился, потому что он не умеет проходить сквозь стенки.

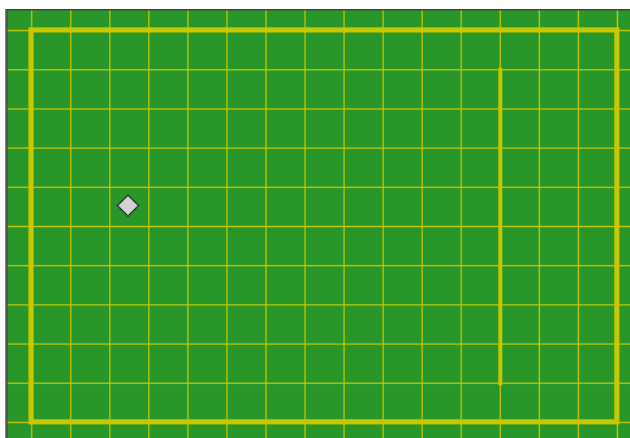
Алгоритмы, в которых робот разбивается (неважно о какую стену), считаются ошибочными.

4. Циклы, условия, примеры решения задач

В среде КуМир есть возможность менять размеры стартовой обстановки. Мы можем уменьшать количество клеток на поле робота, а также увеличивать их.

Также можно представить бесконечное поле для робота и решать на нём задачи.

Допустим, имеется стартовая обстановка, как показано ниже на скриншоте:

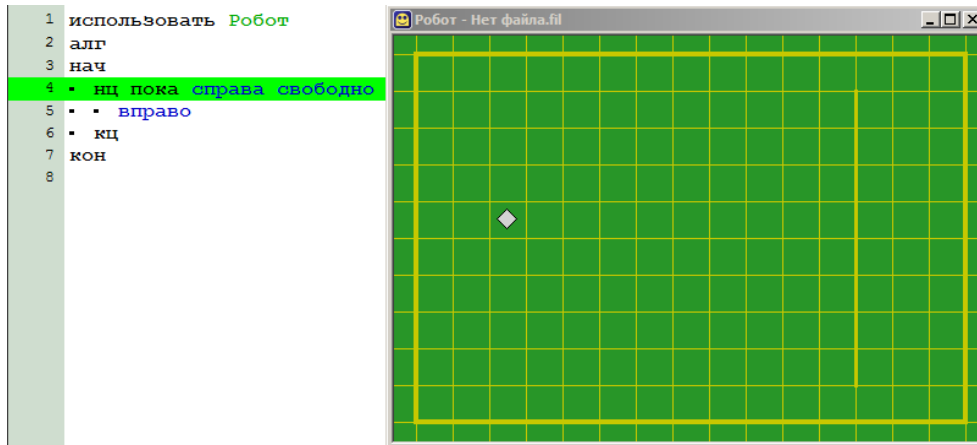


Задача 1:

Робот расположен на поле напротив вертикальной стены, слева от неё, размер поля и расстояние до стены неизвестны. Необходимо дойти роботом до стены.

Очевидно, что невозможно решить эту задачу просто написав последовательно много команд «вправо», потому что робот либо разобьется о стену, либо не дойдет до неё. Угадать это невозможно, так как расстояние до стены неизвестно.

Но из условия задачи можно понять, что нам нужно идти роботом вправо до тех пор, пока не встретится стена. То есть нам нужно много раз выполнить команду **вправо**, но остановиться тогда, когда на пути встретится стена. В этом нам поможет цикл типа «пока».



Обратите внимание, что у нас в алгоритме появилась новая алгоритмическая структура – цикл. Разберем, как она работает.

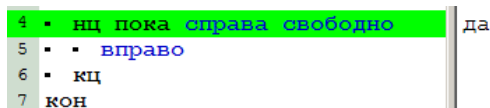
нц пока – начало цикла «пока»

справа свободно – условие, проверяющее, свободна ли клетка справа. Если стены между текущей клеткой и соседней справа нет, то условие верно (возвращает значение «да»), иначе - условие неверно (возвращает «нет»).

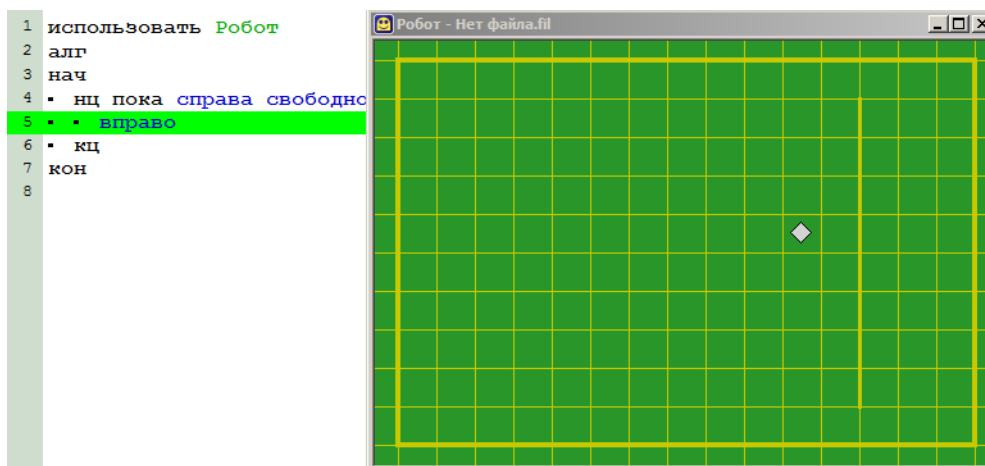
кц – конец цикла

Цикл выполняется следующим образом:

1. Проверяется условие «справа свободно» (т.е. есть справа стена или её нет)
2. Если условие верно, делаем команды, написанные между **нц** и **кц**. Если нет – переходим на команду, записанную после **кц**
3. Возвращаемся на проверку условия



В начальном положении у робота свободна клетка справа, поэтому условие верно (надпись «да») и робот выполняет команду «вправо». Очевидно, что робот остановится в клетке, которая прилегает к стене (условие станет неверным), то есть дойдет до неё.



Аналогично, можно решить задачи движения до стены сверху, стены слева и стены снизу на бесконечном поле. Для этого стоит лишь поменять команду условия.

Роботу доступны следующие проверки:

слева свободно
справа свободно
снизу свободно
сверху свободно

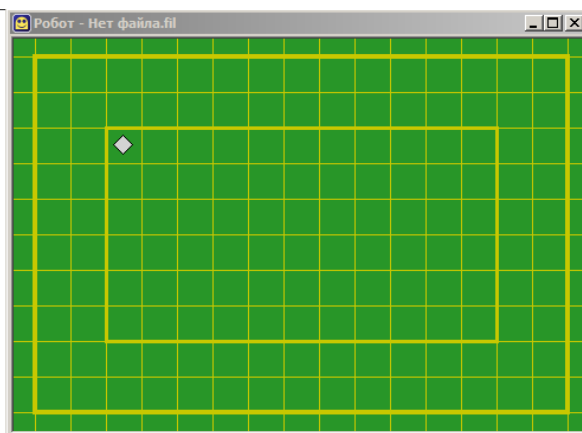
Эти команды также, как и команды движения, относятся к системе команд робота.

Обратите внимание, что команды «**вверху** свободно», «**низ** свободно» или любые другие, отличающиеся от приведённых выше, будут неверными.

Как и с командами движения, правильность написания условий можно отследить по выделению шрифта синим цветом.

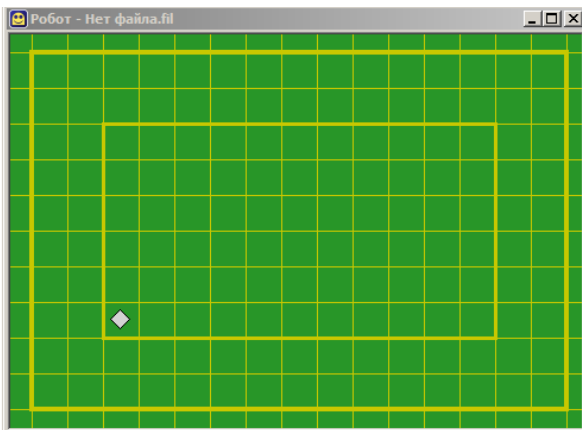
Задача 2: Обойти роботом прямоугольник изнутри и закрасить все клетки, прилегающие к стенкам (начальное положение робота: верхний левый угол).

```
1 использовать Робот
2 алг
3 нач
4   |
5   кон
6
```



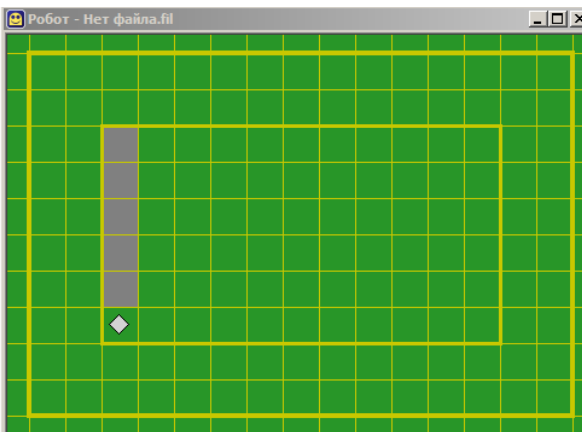
Начальное положение

```
1 использовать Робот
2 алг
3 нач
4   нц пока снизу свободно
5     вниз
6   кц
7   кон
8
```



Алгоритм «дойти до нижней стены»

```
1 использовать Робот
2 алг
3 нач
4   нц пока снизу свободно
5     закрасить
6     вниз
7   кц
8   кон
9
```



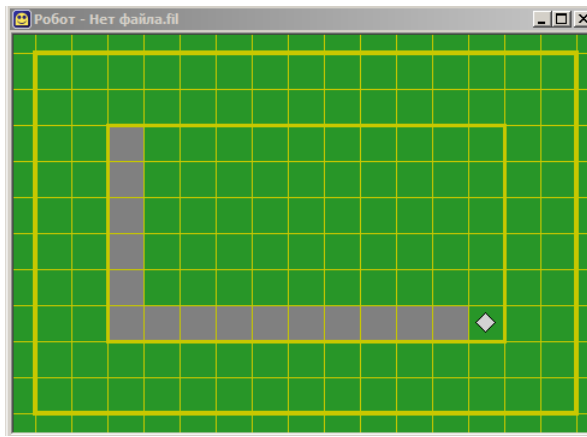
Алгоритм «дойти до нижней стены и закрасить»

Подумайте, почему не закрасилась последняя клетка?

```

1 использовать Робот
2 алг
3 нач
4   нц пока снизу свободно
5     закрасить
6     вниз
7   кц
8   нц пока справа свободно
9     закрасить
10    вправо
11  кц
12 кон
13

```

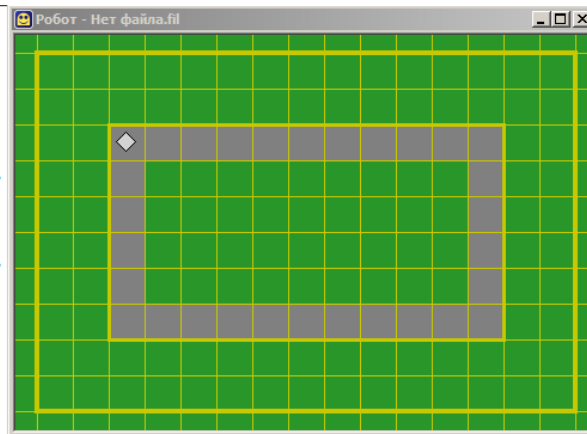


Алгоритм «дойти до нижней стены и закрасить, дойти до правой стены и закрасить»

```

1 использовать Робот
2 алг
3 нач
4   нц пока снизу свободно
5     закрасить
6     вниз
7   кц
8   нц пока справа свободно
9     закрасить
10    вправо
11  кц
12   нц пока сверху свободно
13     закрасить
14     вверх
15   кц
16   нц пока слева свободно
17     закрасить
18     влево
19   кц
20 кон
21

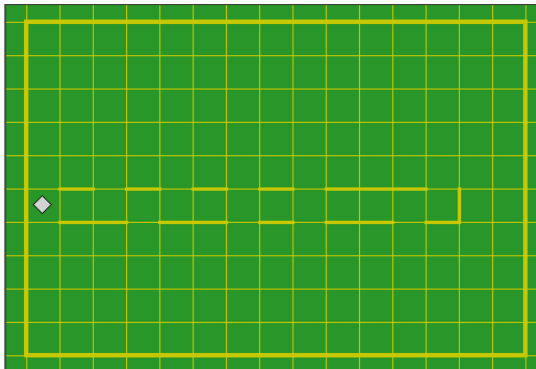
```



Аналогично добавим циклы для двух других стенок.

Задача решена.

Задача 3: Дана стартовая обстановка, представленная на скриншоте ниже. Пройти роботом по коридору, закрашивая только те клетки, у которых присутствует нижняя стенка.

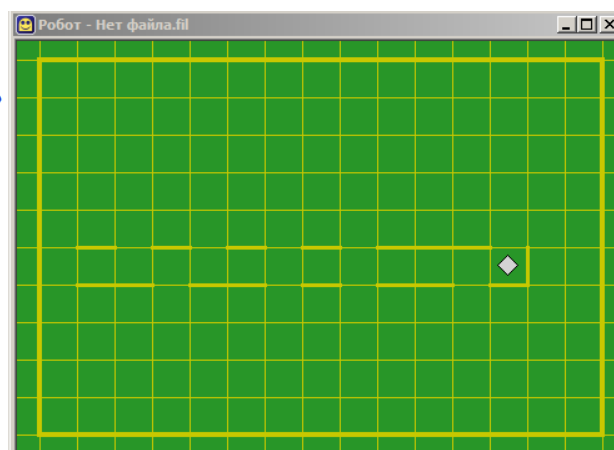


Мы знаем, как реализовать алгоритм движения до правой стены. Напишем и выполним его:

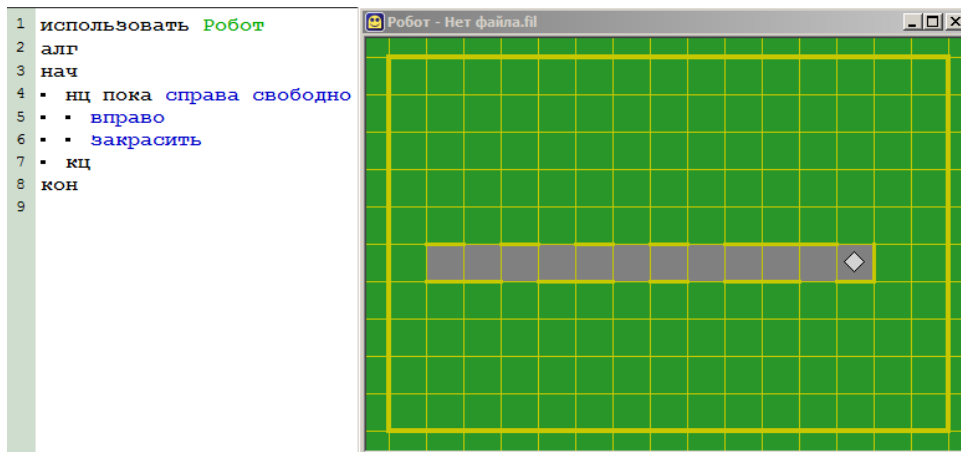
```

1 использовать Робот
2 алг
3 нач
4   нц пока справа свободно
5     вправо
6   кц
7 кон
8

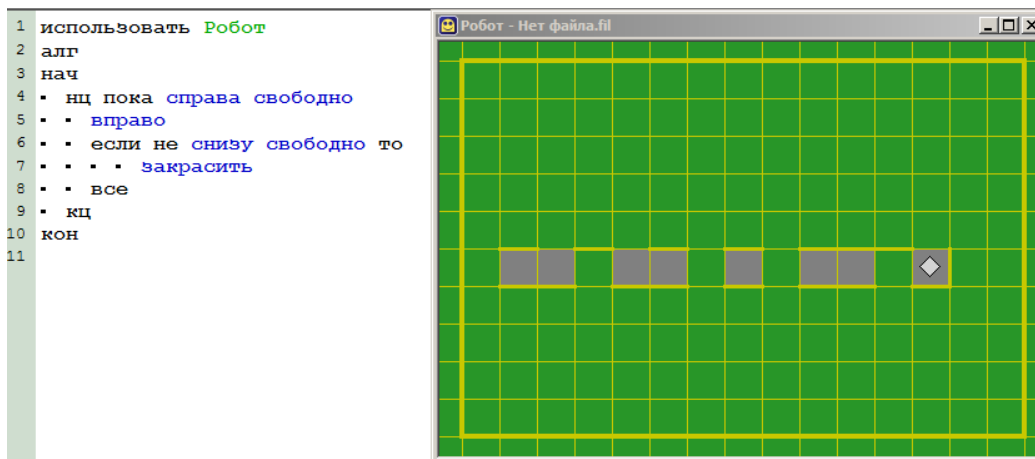
```



Теперь подумаем, как закрасить клетки, у которых есть нижняя стенка. Если мы просто добавим в цикл команду «закрасить», то робот закрасит нам все клетки:



А нам нужно закрашивать только с тем условием, что снизу есть стена. Посмотрим, как добавить такое условие в алгоритм:



Команда **если-то-все** называется **условной** (условным оператором) и позволяет по условию выполнять либо одно действие, либо другое.

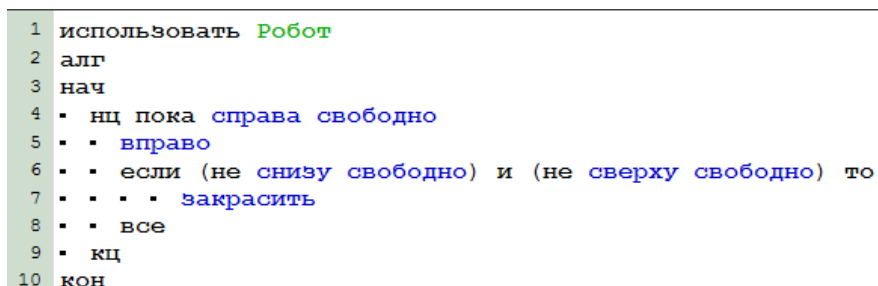
Посмотрим внимательно на условие «не снизу свободно»? Что же оно означает? Команда «снизу свободно» проверяет свободно ли снизу и если свободно возвращает «да», иначе «нет».

НЕ – это отрицание этой команды. То есть «не снизу свободно» проверяет есть ли снизу стена. И если стена есть (снизу несвободно), то возвращает «да», иначе «нет».

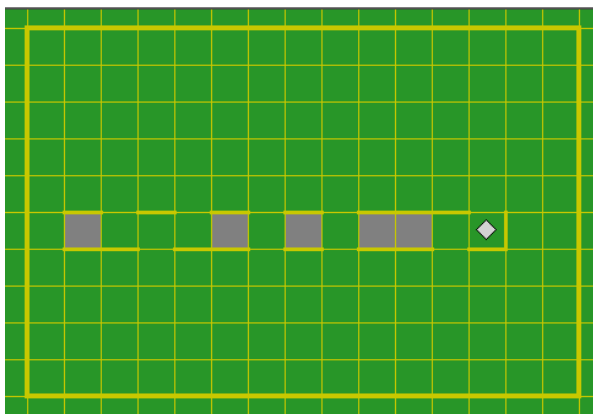
В итоге, команда «закрасить» выполняется только тогда, когда снизу стена. Это нам и было нужно.

Задача 3.1: Задание то же, что и в предыдущей задаче, но нужно закрасить только те клетки, у которых есть **и нижняя и верхняя** стенки.

Для этого напишем сложное условие (из двух простых, соединённых союзом **и**).



Выполнение алгоритма:

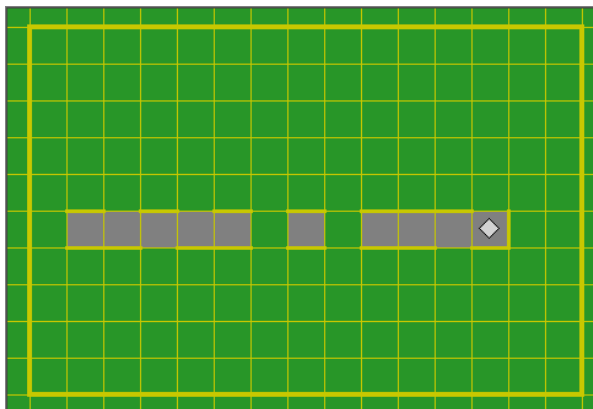


Робот закрасил только те клетки, где выполняются оба условия «не снизу свободно» и «не сверху свободно», то есть клетки, у которых есть стена сверху и снизу.

Задача 3.2: Стартовая обстановка та же, но необходимо закрасить клетки, у которых есть хотя бы одна стена сверху или снизу (или обе вместе).

Для этого между условиями используем команду «или»:

```
1 использовать Робот
2 алг
3 нач
4   нц пока справа свободно
5     вправо
6     если (не снизу свободно) или (не сверху свободно) то
7       закрасить
8     все
9   кц
10 кон
```



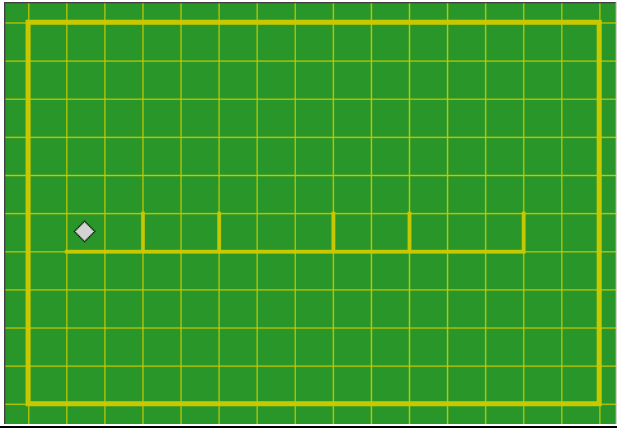
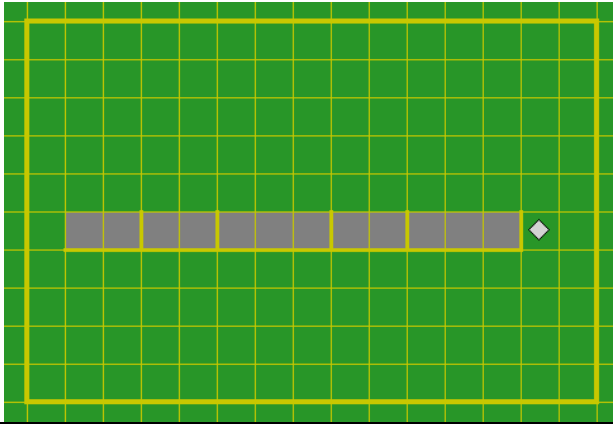
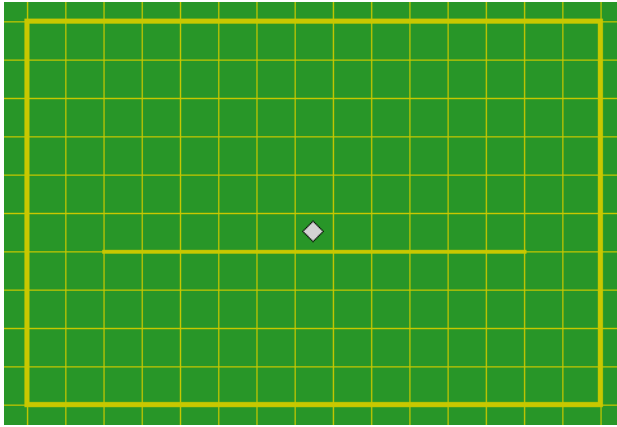
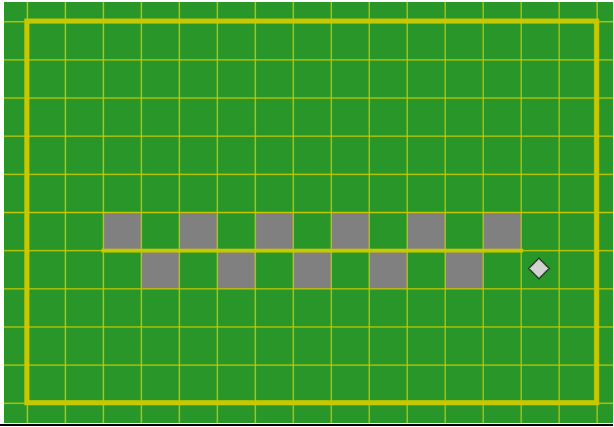
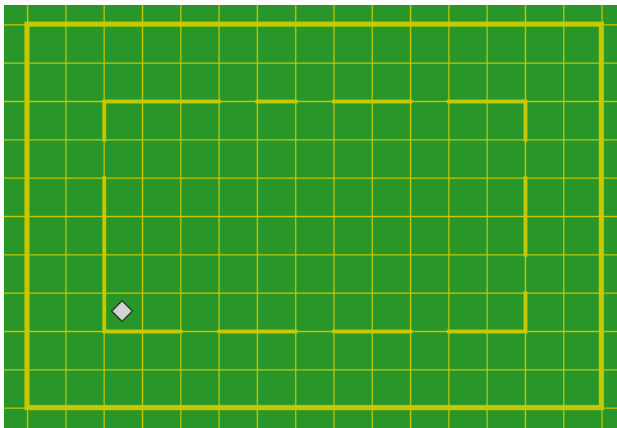
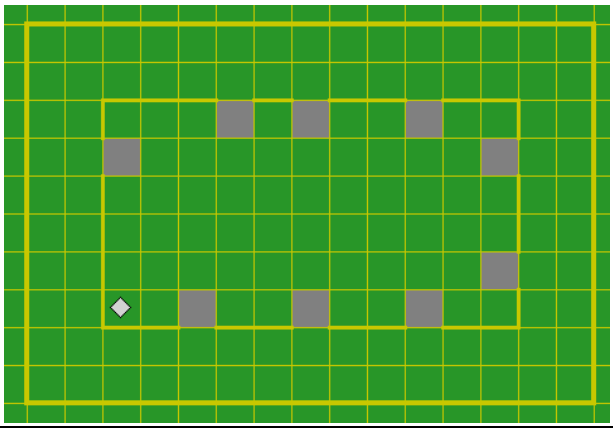
Обратите внимание, что робот **не закрасил** только те клетки у которых нет ни нижней, ни верхней стены.

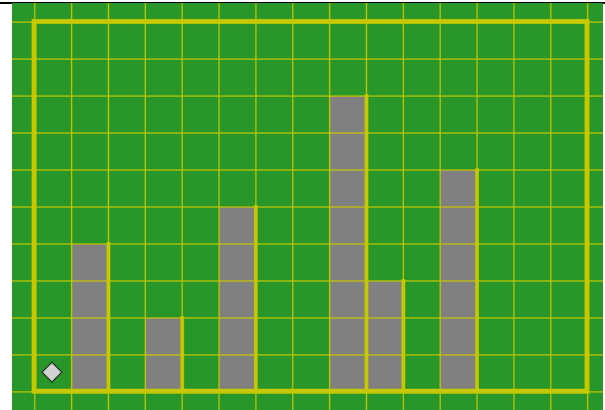
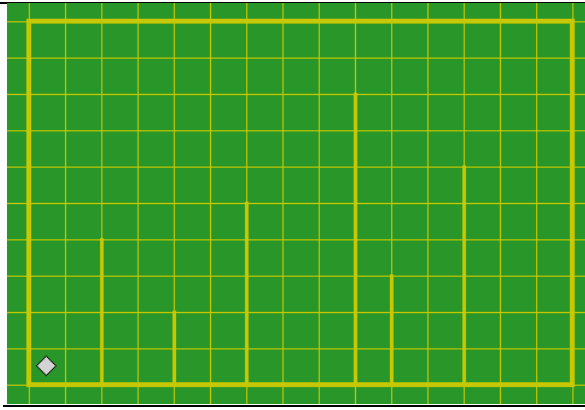
Операции **не**, **и**, **или** называются **логическими операциями** и позволяют из простых условий составлять сложные. Эти сложные условия вы можете записать как в цикл «**нц пока**», так и в условный оператор «**если-то-все**».

5. Задачи для самостоятельной работы:

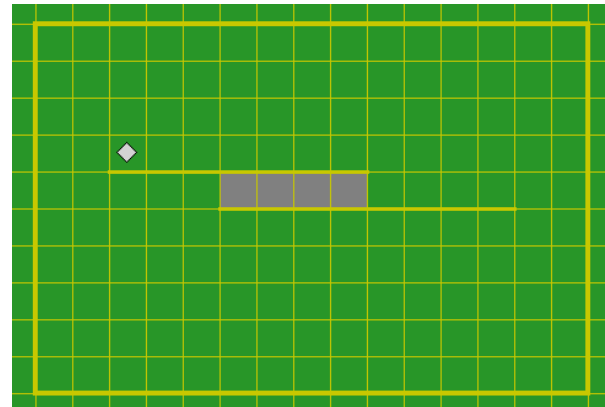
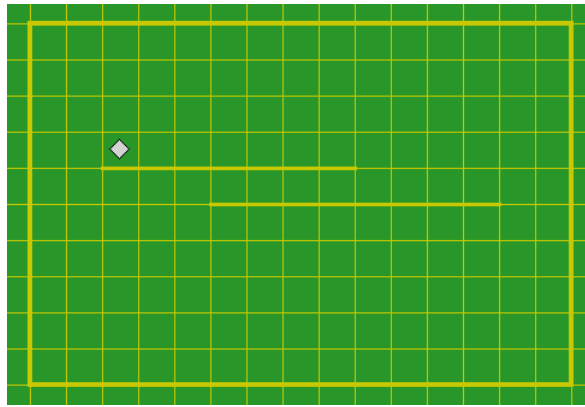
Примечание 1: Во всех задачах нужно закрасить клетки поля так, как указано в столбце «задание». Начальное положение робота имеет значение и указано на скриншотах со стартовой обстановкой.

Примечание 2: После того, как вы написали алгоритм, решающий задачу, попробуйте немного изменить стартовую обстановку, не меняя сути задачи (например, уменьшить или увеличить коридор для примера-задачи 3) и снова попробовать выполнить ваш алгоритм. Если и в этом случае все работает верно, то задача решена. Пробуйте ради интереса находить такие стартовые обстановки, в которых ваш алгоритм не будет работать (иначе за вас их найдет учитель ;)

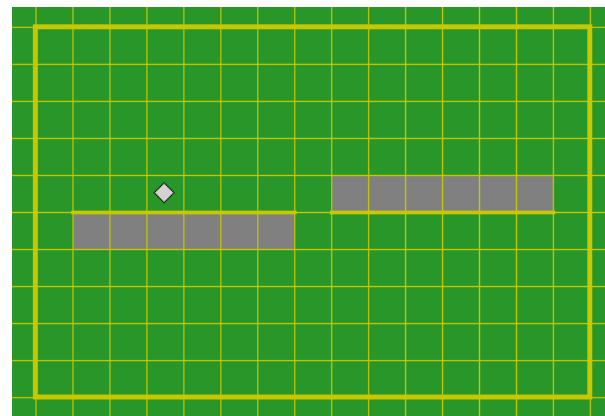
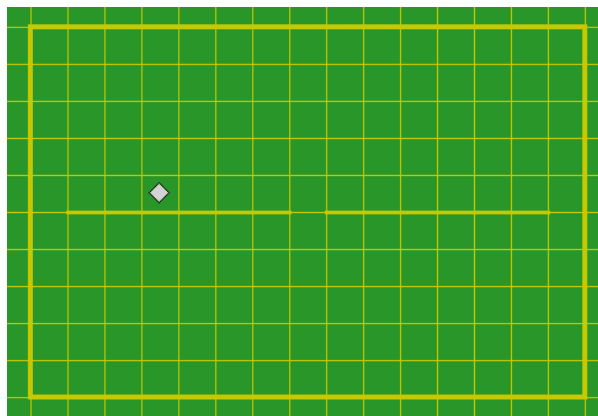
Стартовая обстановка	Задание
 <p>Расстояние между вертикальными стенками не менее 2 клеток.</p>	 <p>Конечное положение робота неважно</p>
 <p>Робот находится в середине горизонтальной стены.</p>	 <p>Конечное положение робота неважно</p>
 <p></p>	 <p>Конечное положение робота неважно</p>



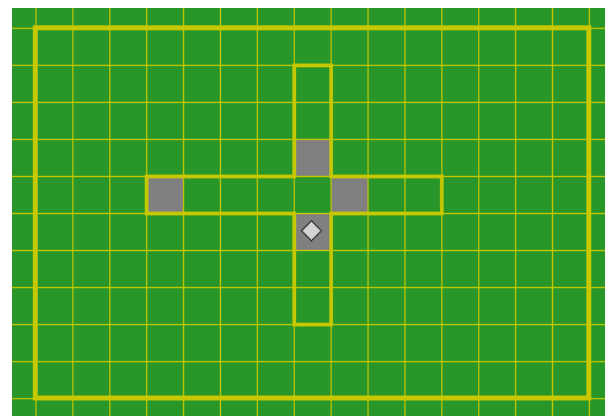
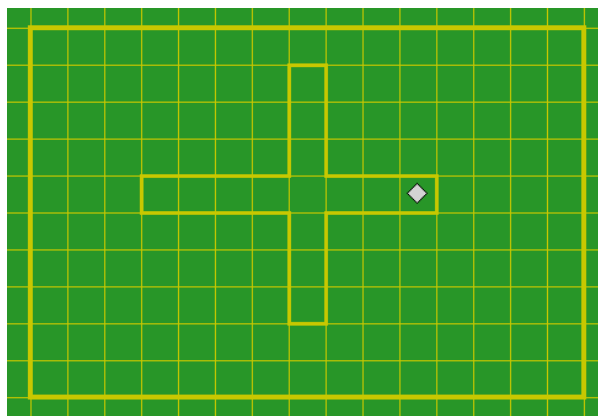
Конечное положение робота неважно



Конечное положение робота неважно



Конечное положение робота неважно



Закрасить указанные ячейки и перевести робота в конечное положение.

Другие стартовые обстановки можно взять из демо-версий ГИА по информатике (или придумать самим)