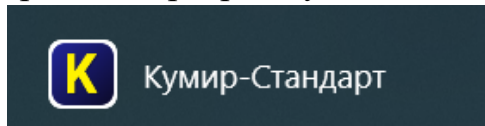
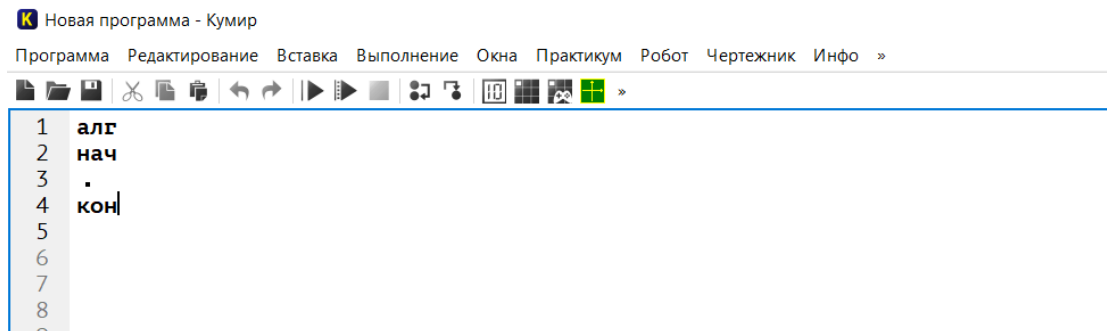


Уровень 0 (для тех, кого не было на прошлом уроке или для тех, кто не успел/не понял).

1. Открываем программу.

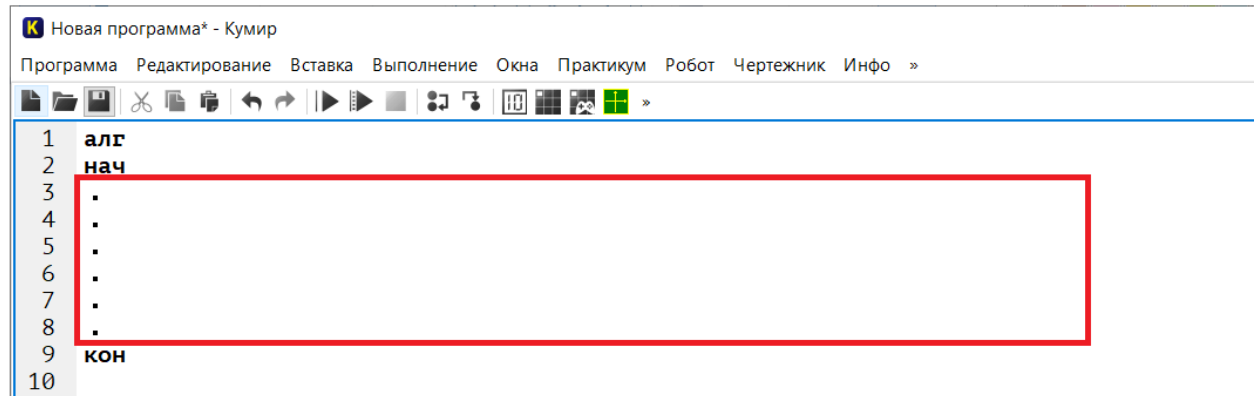


2. Получаем такое окно (какие-либо комментарии, выделенные серым шрифтом, можно стереть):

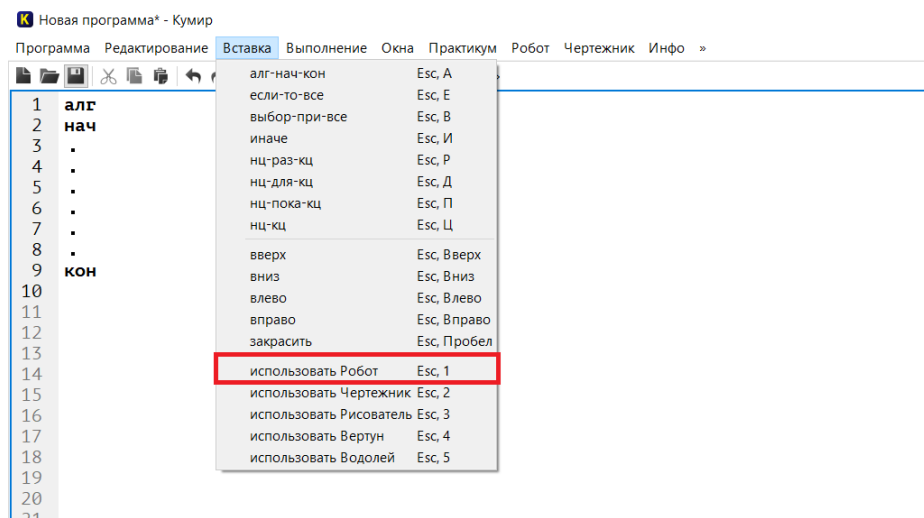


!!!ВНИМАНИЕ!!!

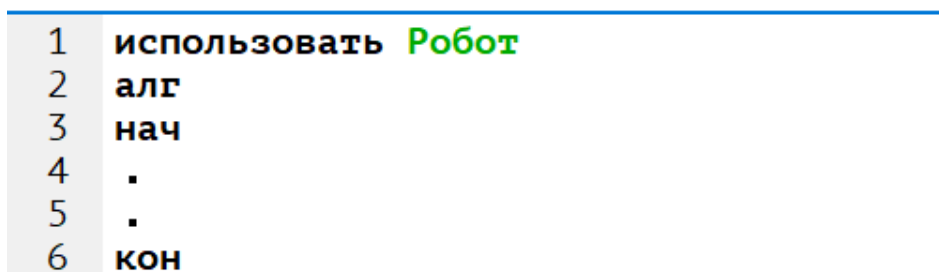
Программу для Робота мы пишем МЕЖДУ слов «нач» и «кон».



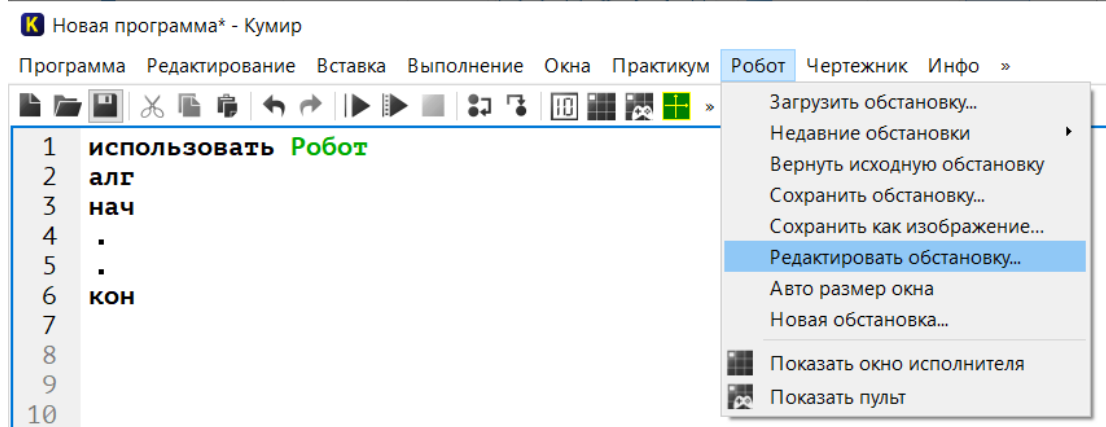
3. Чтобы писать алгоритм для Робота, надо зайти в «Вставка» => «Использовать Робот»:



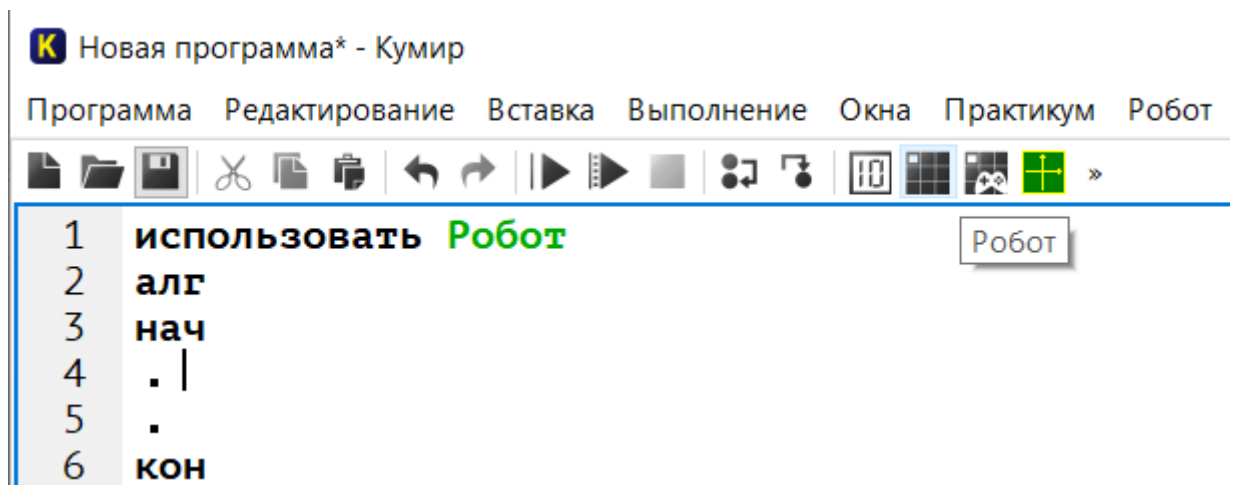
Теперь наше окно выглядит вот так:



4. Заходим в «Робот», выбираем пункт «Редактировать обстановку», выбираем эту опцию.

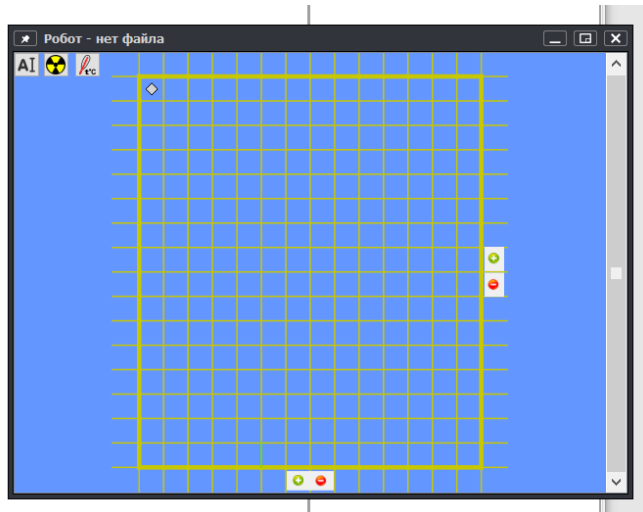


5. Выбираем сверху тот квадратик, который предназначен для работы с Роботом (навести мышкой на квадратик, высвечивается «Робот»):



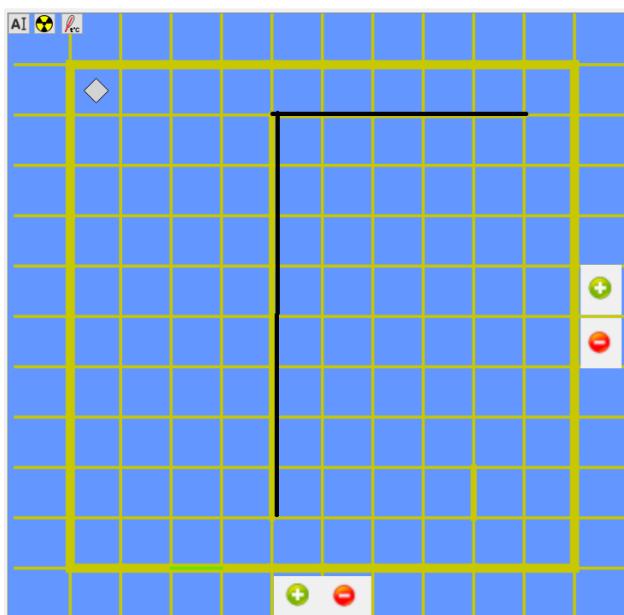
Выбираем этот «квадратик»

6. Появляется вот такое окошечко.

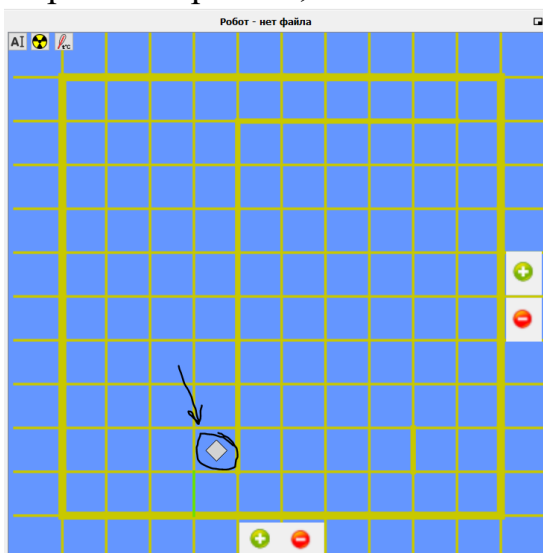


С помощью «+» и «-» можно добавлять или убирать строки и столбцы. Формируем свою обстановку для Робота.

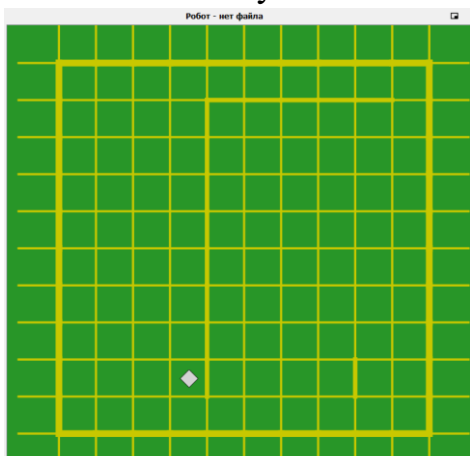
7. Наживаем на ребро клеточки, оно подсвечивается жирным. Формируем путь в виде буквы «Г» (для наглядности выделено черным цветом):



8. Переносим робота, зажав кнопкой мыши, в начало пути.



9. Заходим в «Робот», находим «Редактировать обстановку», тыкаем, снимаем галочку. Поле становится зелёным.

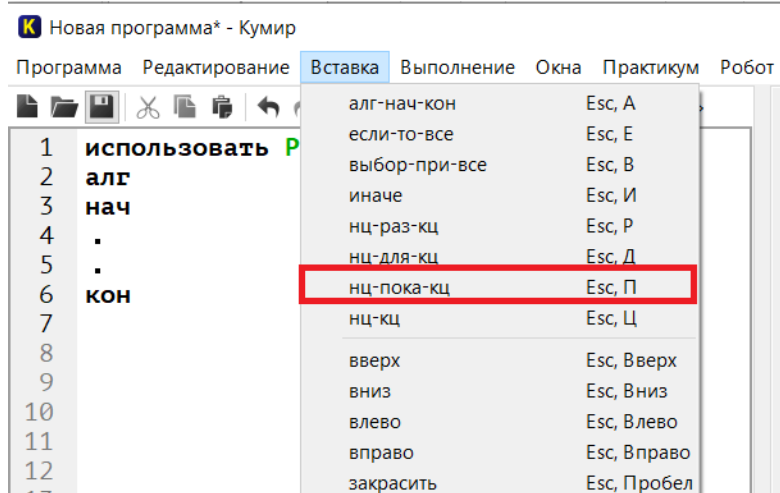


10. Первая часть нашего алгоритма - до тех пор пока справа стена, Робот будет закрашивать клеточку, на которой он находится, и сдвигаться вверх (понятие ЦИКЛ).

Ставим курсор в первую строчку после служебного слова «нач» (помечено красным крестиком):

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

11. Заходим в «Вставка», выбираем пункт «нц-пока-кц». Выбираем его.



нц – начало цикла

кц – конец цикла

Эти слова обязательны для выполнения цикла!!!!

После слова «пока» мы пишем условие для его выполнения.

Теперь наш алгоритм выглядит вот таким образом:

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

12. Заполняем его.

```

1  использовать Робот
2  алг
3  нач
4    . нц пока справа стена
5    .   . закрасить
6    .   . вверх
7    . кц
8    .
9  кон

```

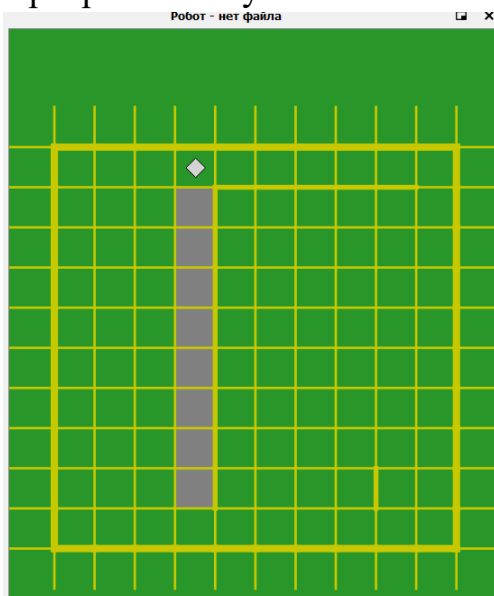
Пока справа стена (условие выполнения цикла), Робот красит клеточку и идет вверх.

13. Запустим нашу программу.

Нажимаем вот этот треугольничек.



14. Программа запускается.



15. Попадаем в верхнюю клетку. Она ни в какой цикл входить не будет. Поэтому Робот её просто закрасит и повернет направо.

```
1  использовать Робот
2  алг
3  нач
4  . нц пока справа стена
5  . . закрасить
6  . . вверх
7  . кц
8  . закрасить
9  . вправо
10 .
11 кон
```

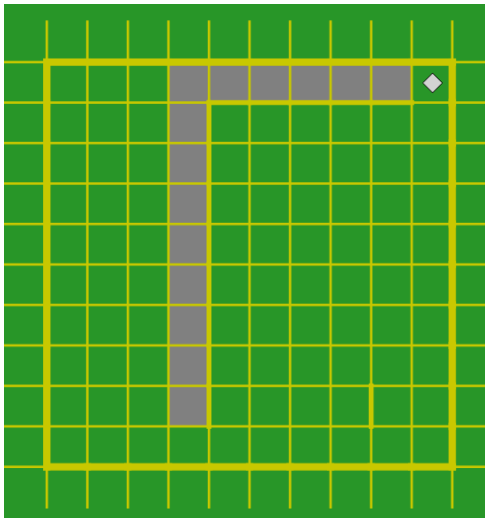
16. Теперь Робот будет идти вправо до тех пор, пока СНИЗУ будет стена.
Возникает снова цикл. «Вставка» = > «нц – пока – кц»:

```
1  использовать Робот
2  алг
3  нач
4  . нц пока справа стена
5  . . закрасить
6  . . вверх
7  . кц
8  . закрасить
9  . вправо
10 . нц пока
11 . .
12 . кц
13 кон
```

17. Заполняем тело цикла:

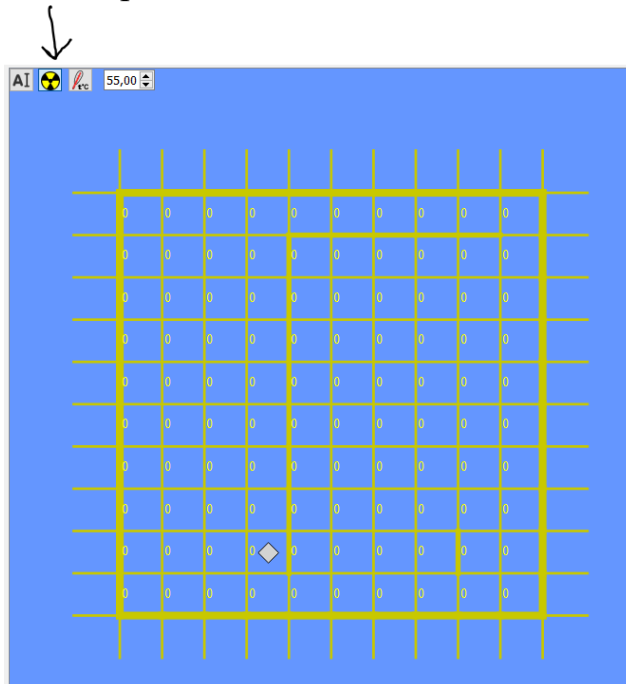
```
1  использовать Робот
2  алг
3  нач
4  . нц пока справа стена
5  . . закрасить
6  . . вверх
7  . кц
8  . закрасить
9  . вправо
10 . нц пока снизу стена
11 . . закрасить
12 . . вправо
13 . кц
14 кон
```

18. Готово, теперь наш Робот красить клетки вокруг лабиринта в виде буквы «Г»:



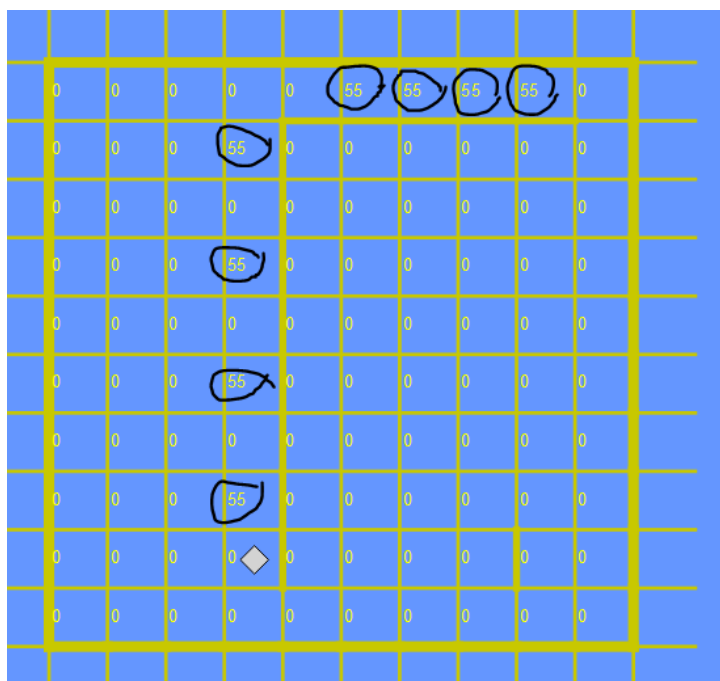
Теперь задачу усложним.

1. Заходим снова в «Робот» => «Редактировать обстановку». Тыкаем на значок радиоактивной опасности.



2. Будем заражать радиацией некоторые из клеток на пути Робота (маршрут в виде буквы «Г» останется таким же). Для этого надо тыкнуть на соответствующую клеточку, которую хотим заразить.

Пример:



3. Заходим в «Робот», «Редактировать обстановку», снимаем галочку. Поле станет зелёным.
4. Теперь мы будем красить только те клетки, которые заражены радиацией. Вспоминаем про алгоритм с ВЕТВЛЕНИЕМ.

Теперь новый алгоритм будет выглядеть вот так:

использовать Робот

алг

нач

```

. нц пока справа стена
. . если радиация > 0
. . . то закрасить
. . все
. . вверх
. кц
. закрасить
. вправо
. нц пока снизу стена
. . закрасить
. . вправо
. кц

```

кон

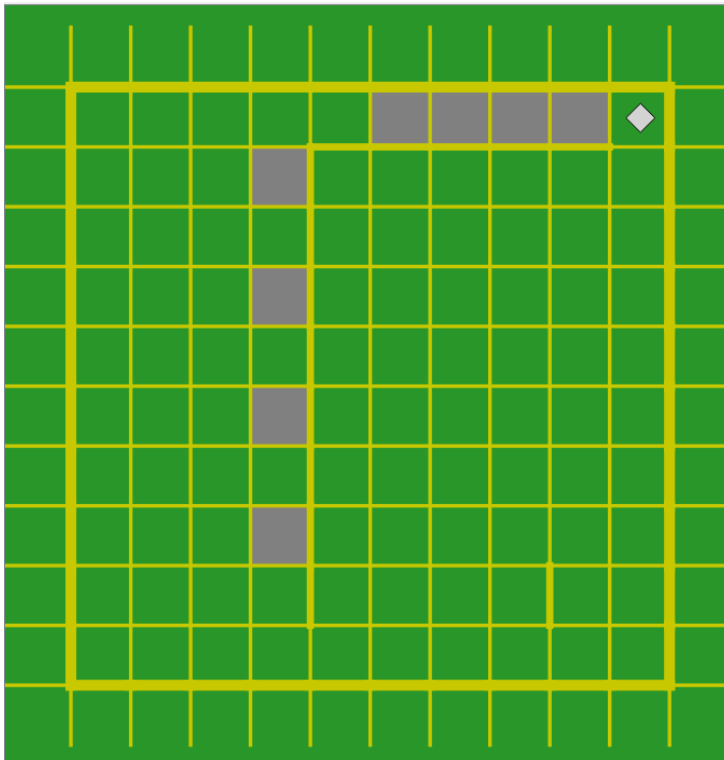
!!!ВНИМАНИЕ!!!

Обратите внимание на структуру алгоритма. Есть служебные слова «если», «то» и «все». Без них алгоритм работать не будет.

5. Дополним остальные инструкции нашего алгоритма. (красным выделены те кусочки алгоритма, где мы добавили алгоритм с ветвлением):

```
1  использовать Робот
2  алг
3  нач
4  . нц пока справа стена
5  . . если радиация > 0
6  . . . то закрасить
7  . . все
8  . . вверх
9  . кц
10 . если радиация > 0
11 . . то закрасить
12 . все
13 . вправо
14 . нц пока снизу стена
15 . . если радиация > 0
16 . . . то закрасить
17 . . все
18 . . вправо
19 . кц
20 кон
```


6. Теперь Робот красит только те клетки, которые заражены радиацией. Например, вот так:



Еще усложняем задачу.

1. Мы будем ещё и считать количество зараженных клеток. Для этого введём ЦЕЛОЧИСЛЕННУЮ переменную:

```
1  использовать Робот
2  алг
3  нач
4  . цел  $n$ 
5  . нц пока справа стена
6  . . если радиация > 0
7  . . . то закрасить
8  . . все
```



2. С ее помощью мы будем считать количество зараженных клеток. В самом начале своего пути Робот еще не нашел НИ ОДНУ клетку. Поэтому наша переменная-счётчик равна нулю:

```
нач
. цел  $n$ 
.  $n := 0$ 
```

Обратите внимание, как именно происходит присваивание значения. Ставим двоеточие «:», затем знак равно «=».

3. Теперь если Робот найдёт зараженную клетку, то значение нашей переменной-счётчика мы будем увеличивать на единицу:

```
. если радиация > 0
. . то закрасить
. . .  $n := n + 1$ 
. все
```

4. Теперь наш алгоритм выглядит вот так:

```

1  использовать Робот
2  алг
3  нач
4  . цел  $n$ 
5  .  $n := 0$ 
6  . нц пока справа стена
7  . . если радиация > 0
8  . . . то закрасить
9  . . . .  $n := n + 1$ 
10 . . все
11 . . вверх
12 . кц
13 . если радиация > 0
14 . . то закрасить
15 . . .  $n := n + 1$ 
16 . все
17 . вправо
18 . нц пока снизу стена
19 . . если радиация > 0
20 . . . то закрасить
21 . . . .  $n := n + 1$ 
22 . . все
23 . . вправо
24 . кц
25 кон

```

5. Выведем сообщение. Если у нас есть зараженные клетки, мы будем выводить на экран их количество и выводить сообщение: «Заражено!».

```

5  .  $n := 0$ 
6  . нц пока справа стена
7  . . если радиация > 0
8  . . . то закрасить
9  . . . .  $n := n + 1$ 
10 . . все
11 . . вверх
12 . кц
13 . если радиация > 0
14 . . то закрасить
15 . . .  $n := n + 1$ 
16 . все
17 . вправо
18 . нц пока снизу стена
19 . . если радиация > 0
20 . . . то закрасить
21 . . . .  $n := n + 1$ 
22 . . все
23 . . вправо
24 . кц
25 . если  $n > 0$ 
26 . . то вывод "Заражено: |",  $n$ 
27 . . иначе вывод "Коридор чист"
28 . все
29 кон

```

6. Запускаем программу.
Вывод сообщения:

```
>> 0:32:11 - Новая программа - Начало выполнения
Заражено: 8
>> 0:32:14 - Новая программа - Выполнение завершено
```

Уровень 1.

Иногда при построении лабиринта может возникнуть такая ситуация, что Робот разобьется.

Здесь Робот идет влево ДО ТЕХ ПОР, пока снизу стена. Он идет влево и разбивается о стену:

1	использовать Робот	
2	алг	
3	нач	
4	нц пока снизу стена	да
5	закрасить	
6	влево	
7	кц	
8		
9	кон	

0:36:14 - Новая программа - Начало выполнения
ОШИБКА ВЫПОЛНЕНИЯ: Робот разбился: слева стена!
0:36:16 - Новая программа - Выполнение завершено

Как избежать таких ситуаций?

Изменим формулировку наших инструкций.

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

Теперь Робот будет красить клетку и идти влево только в том случае, если слева будет свободно. Ура, теперь он не разобьётся!

Пример.

Робот красит клетки и идет влево, доходит до стены, начинает идти
наверх:

<pre>1 использовать Робот 2 алг 3 нач 4 . нц пока слева свободно 5 . . закрасить 6 . . влево 7 . кц 8 . нц пока сверху свободно 9 . . закрасить 10 . . вверх 11 . кц 12 кон</pre>	да; да;	
--	------------	--

>> 0:40:23 - Новая программа - Начало выполнения
>> 0:40:25 - Новая программа - Выполнение завершено

Еще пример:

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

>> 0:42:49 - Новая программа - Начало выполнения
>> 0:42:54 - Новая программа - Выполнение завершено

Задание 1.



Простые и составные алгоритмы.

Прочитать здесь: <http://www.klyaksa.net/htm/konspektsch/kumir/13.htm>

Задание 2. Имеется коридор П-образной формы (как в прошлом задании).

Вам необходимо не только заразить некоторые из клеток радиацией, но и задать им определенную температуру (это можно сделать в «Редакторе обстановки»).

Задание: закрасить только те клетки, которые заражены радиацией и у которых температура больше нуля.

Использовать составное условие.

Условие для проверки температуры:

температура > 0

Задание 3.

Прочитать и выполнить:

<http://www.klyaksa.net/htm/konspektsch/kumir/22.htm>

Задание 4.

Построить такой же вытянутый (как в примере по ссылке в Задании 3) лабиринт. Закрашивать только те клетки, у которых ТЕМПЕРАТУРА является минимальной (*).

(*) В редакторе обстановки задаваемую температуру можно менять:



Это параметр можно уменьшить:

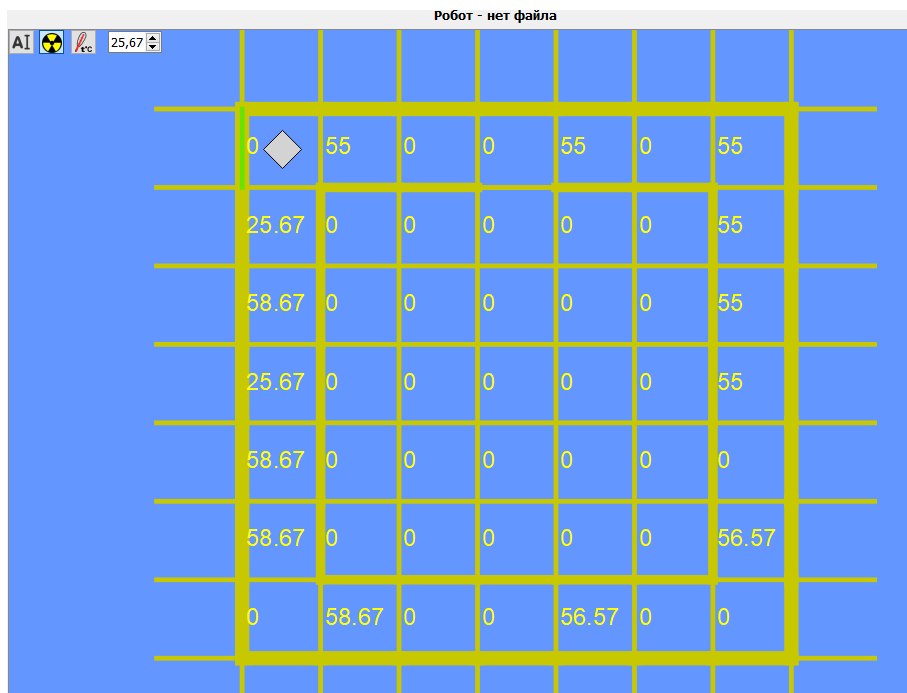


Можно увеличить:

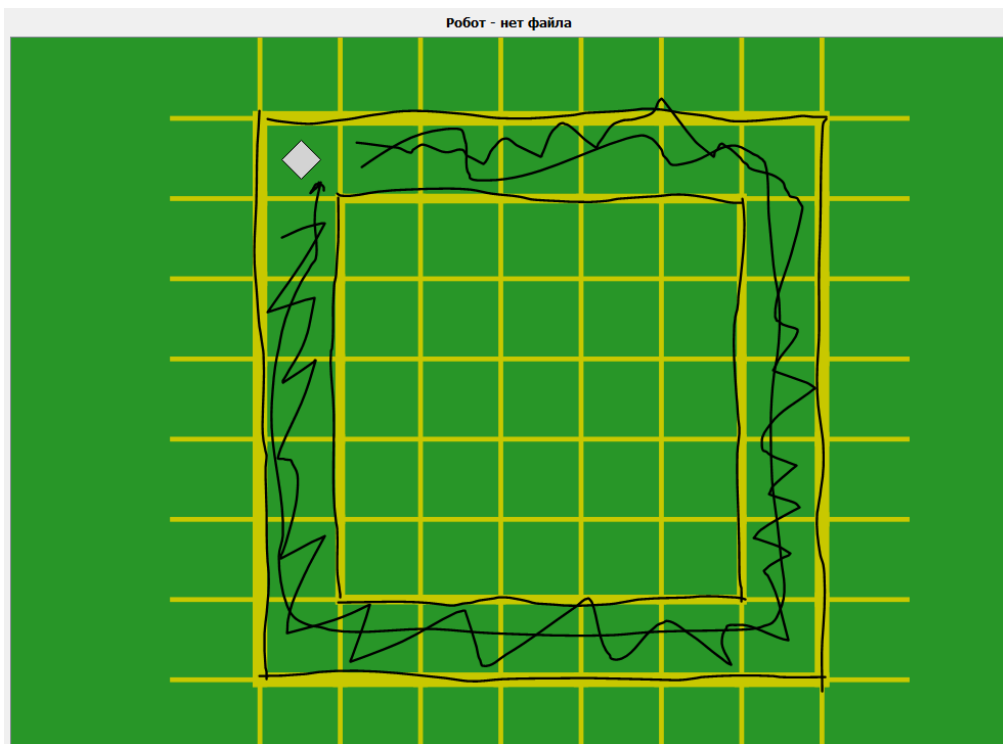


Задание 5.

Создать путь для Робота вдоль стены и произвольно заразим радиацией клетки. Обратите внимание, радиация может выражаться не только целым числом, а также рациональным:



Будем проходить вот этот коридор.



Теперь вам надо пройти по этому коридору и посчитать суммарное радиоактивное излучение от всех клеток, которому был подвержен Робот.

На каждом шаге проверять: есть ли радиация. Если есть, излучение от клетки прибавить к специальной переменной.

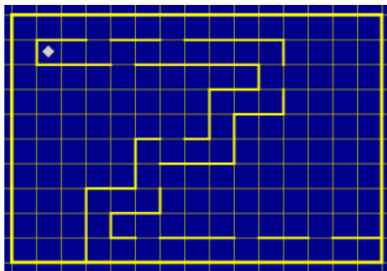
Так как в этом примере числа у нас уже как целые, так и дробные, то необходимо завести переменную ВЕЩЕСТВЕННОГО типа.

- **вещ** *n*;
- *n* := 0

Дополнительные задания.

Вариант 1

1. Необходимо провести Робота по коридору шириной в одну клетку из начального положения (◇) до конца коридора, закрашивая при этом все клетки коридора, которые имеют выход. Выходы размером в одну клетку располагаются произвольно по всей длине коридора. Коридор заканчивается тупиком. Коридор имеет два горизонтальных и диагональный участки в форме Σ . Пример коридора показан на рисунке.



2. Робот движется вдоль стены, профиль которой показан на рисунке, от начального положения (◇) до конца стены. Необходимо закрасить все внутренние углы стены, как показано на примере. Размеры стены могут быть произвольны.

