

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЯЯ ОБРАЗОВАТЕЛЬНАЯ ШКОЛА №15 Г.НЕРЮНГРИ

**УЧЕБНОЕ-МЕТОДИЧЕСКОЕ ПОСОБИЕ
ОСНОВЫ РАБОТЫ В ПРОГРАММЕ «КУМИР»**

Харьковская Ольга Константиновна
учитель информатики и ИКТ

Нерюнгри, 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1. Лекция 1. Основы алгоритмизации.....	4
2. Лекция 2. Способы записи алгоритмов.....	5
3. Лекция 3. Разработка алгоритма с помощью «Исполнителя Робот».....	7
4. Практические работы.....	15
4.1. Линейный алгоритм.....	15
4.2. Циклический алгоритм. Цикл N раз.....	18
4.3. Циклический алгоритм с предусловием «ПОКА».....	22
4.4. Команды ветвления (если, выбор).....	28
СПИСОК ЛИТЕРАТУРЫ.....	35

ВВЕДЕНИЕ

Учебно-методическое пособие "Основы работы в программе Кумир" предназначено для учителей и обучающихся 8-9 классов общеобразовательных школ, изучающих раздел информатики: «Основы алгоритмизации».

Эти материалы могут быть полезны при подготовке к основному и единому государственному экзамену по информатике.

В пособие включен цикл лабораторных работ, основными целями для проведения которых являются:

- Углубление, обобщение и систематизация знаний по блок-схемам;
- Развитие алгоритмического мышления через составление блок-схем;
- Закрепление навыков составления всевозможных алгоритмов фиксированной длины на алгоритмическом языке для формального исполнителя с заданной системой команд;
- Закрепление навыков работы на персональном компьютере в среде программирования КУМИР;
- Развитие информационно-коммуникационной компетентности обучающихся

Лекция 1. Основы алгоритмизации

Известно множество областей применения компьютера: обработка текстов и графики, передача и получение информации, создание справочников, производство расчетов. Еще одно из важнейших направлений применения компьютеров – управление.

Управление – это целенаправленное воздействие одних объектов, которые являются управляющими, на другие объекты – управляемые. Все управляющие воздействия производятся с определенной целью с помощью команд. Таким образом, алгоритмом управления – это последовательность команд по управлению объектом, приводящую к достижению заранее поставленной цели.

Объект управления – исполнитель алгоритма, в данном случае исполнитель алгоритма – устройство. Все исполнители, которые относятся к такому типу, называются формальные. Формальный исполнитель не понимает смысл команд, в информатике рассматривают только формальных исполнителей.

Алгоритм – это точное описание последовательности шагов в решении задачи, приводящих от исходных данных к требуемому результату.

Алгоритм применительно к вычислительной технике должен обладать свойствами, которые обеспечивали бы его автоматическое выполнение:

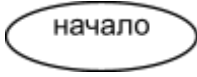


- 1) Процесс решения задачи должен быть разбит на последовательность отдельно выполняемых шагов – дискретность (прерывность).
- 2) Алгоритм для данного исполнителя содержит только те команды, которые входят в систему его команд – понятность.
- 3) Каждое правило алгоритма должно быть четким и однозначным – определенность (точность).
- 4) Исполнение алгоритма должно завершиться за определенное количество шагов конечность (результативность).
- 5) Алгоритм должен выполняться для любого набора исходных данных из некоторой области, удовлетворяющих условию задачи (массовость).

Лекция 2. Способы записи алгоритмов

Существуют различные способы записи алгоритмов. Основными среди них являются:

- Словесные
- Графические
- На алгоритмических языках.

1. Словесные способы записи алгоритма. Самой простой является запись алгоритма в виде набора высказываний на обычном разговорном языке. Но алгоритм в словесной форме может оказаться очень объемным и трудным для восприятия.
2. Блок-схема представляет собой графический документ, дающий представление о порядке работы алгоритма.

Название блока	Название блока	Название блока
Начало, остановка		Начало и конец блок
данные		Ввод-вывод данных
процесс		Вычислительное действие
Решение		Проверка условия
подготовка		Начало цикла
дисплей		Вывод результата на экран
документ		Вывод результата на печать

3. Алгоритмические языки – школьный алгоритмический язык,

Алгоритм применительно к вычислительной технике должен обладать свойствами, которые обеспечивали бы его автоматическое выполнение:

- 1) Процесс решения задачи должен быть разбит на последовательность отдельно выполняемых шагов –дискретность (прерывность).
- 2) Алгоритм для данного исполнителя содержит только те команды, которые входят в систему его команд
–понятность.
- 3) Каждое правило алгоритма должно быть четким и однозначным
–определенность (точность).
- 4) Исполнение алгоритма должно завершиться за определенное количество шагов конечность (результативность).
- 5) Алгоритм должен выполняться для любого набора исходных данных из некоторой области, удовлетворяющих условию задачи (массовость).

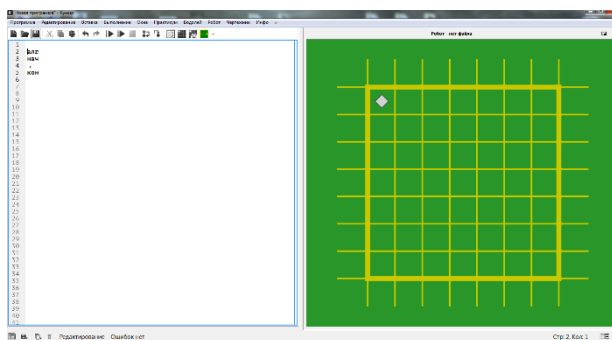
Лекция 3. Разработка алгоритма с помощью «Исполнителя Робот»

Прежде чем говорить о реализации алгоритмов для исполнителя, нужно понимать, что такое программа и чем она отличается от алгоритма.

Программа – алгоритм, записанный на языке программирования.

Как мы помним, алгоритмы могут иметь различные формы представления, а программа должна быть записана на языке исполнителя. Любому учебному исполнителю свойственна среда деятельности, система команд управления и режимы работы.

Среда Кумир версия 2.1.0 – Среда исполнителя Робот прямоугольное клетчатое поле, между клетками которого могут быть препятствия (стены) рисунок 1.



Система команд исполнителя Робот:

Простые команды: вверх, вниз, влево, вправо, закрасить.

Команды логические: (проверки условия)

сверху свободно, снизу свободно, слева свободно, справа свободно.

Логические связки: И, НЕ, ИЛИ (сложные условия)

Пример: (не слева свободно) или (не справа свободно)

команда ветвления:

если условие то

серия команд

всё

команда цикла:

нц пока условие

серия команд

кц

Важно помнить, что исполнитель Робот движется по клеточкам, получая простую команду, робот выполняет движение на одну клетку и ждет следующей команды.

И так, запустив программу Кумир, откройте меню Вставка, выберите исполнителя Робот. Далее, прежде чем начать писать программу для Робота, нужно настроить «поле» по которому Исполнитель Робот будет двигаться выполняя программу.

Первое – открываем меню **Робот**, выбираем команду **Редактировать обстановку** (рис.1),

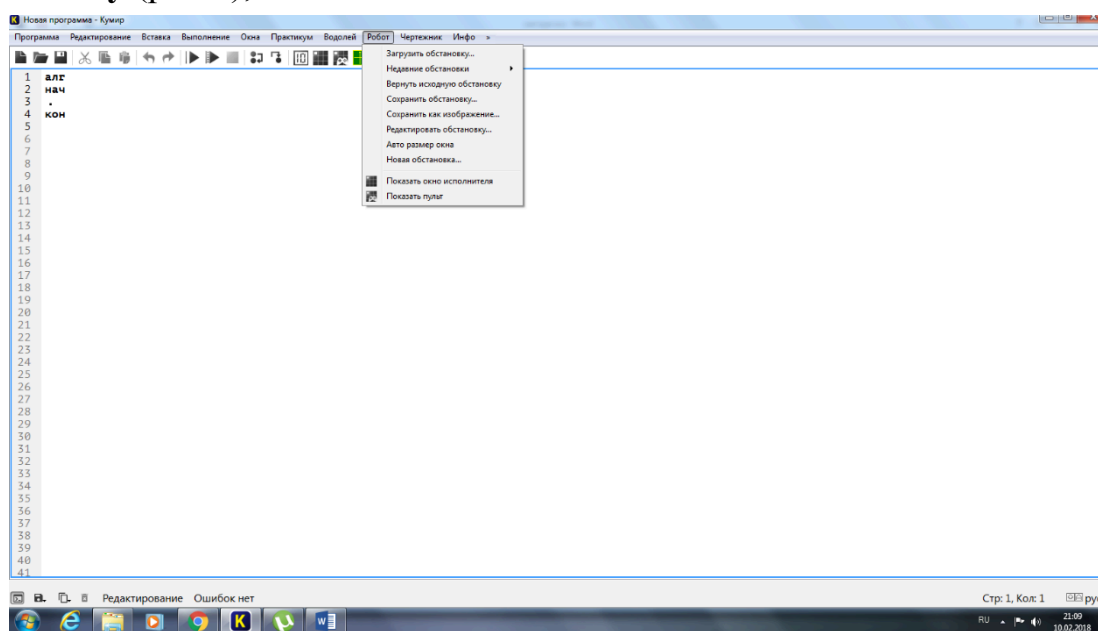


Рисунок 1 – выбор команды *Редактировать обстановку*

Далее, справа диалоговое окно обстановка Робот, становится синего цвета, и появляются кнопки управления (рис.2), которыми можно добавить или убрать

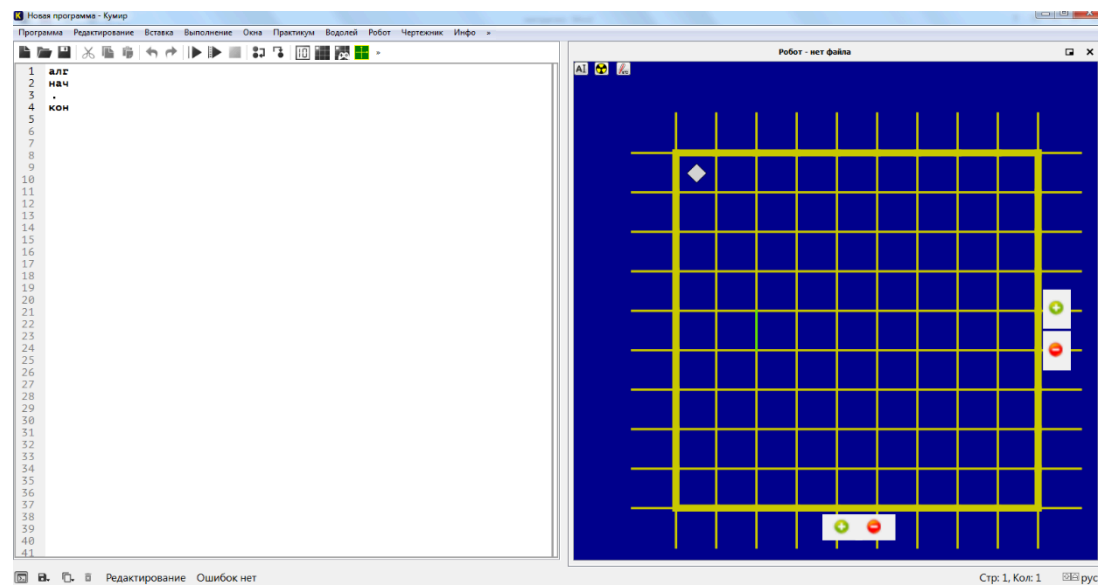


Рисунок 2 – Редактирование поля

клетки поля (при первом запуске поле имеет размер 7 x 7 клеток).

Затем, щелкая левой кнопкой мыши (рис.3) выставляем препятствия для Робота.

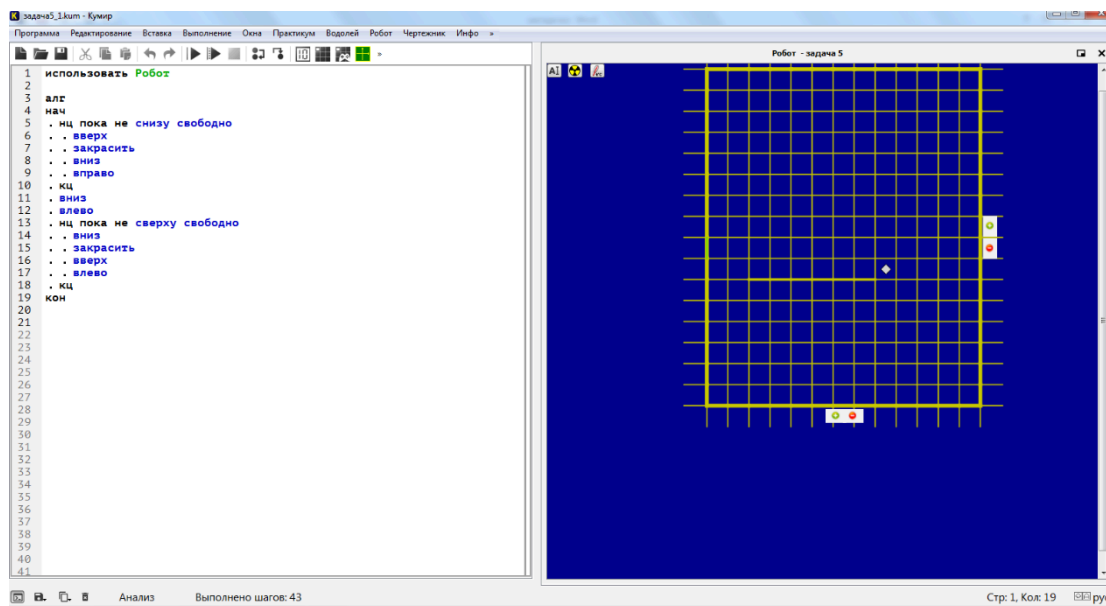
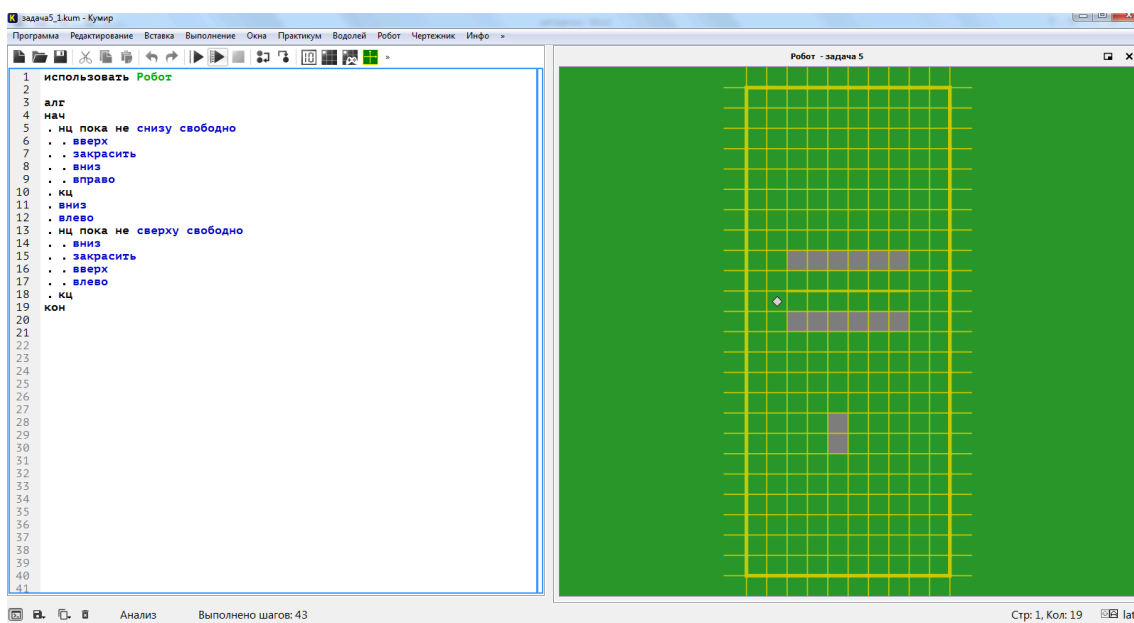


Рисунок 3 – Окончание редактирования

После того как обстановка для исполнителя Робот готова, нужно её сохранить. Заходим в меню **Робот**, выбираем команду **Сохранить обстановку**. Далее в открывшемся диалоговом окне, выбираем куда сохранить файл (например, Рабочий стол или личная папка учащегося), даем название файлу и сохраняем.

После того, как сохранили обстановку, заходи меню **Робот**, выбираем команду **Загрузить обстановку**, поле для Робота становится зеленого цвета (рис.4) и нет кнопок управления.



Второе – составляем по условию задачи алгоритм для Робота, для этого в левой части окна программы выполняем следующее:

1. меню Вставка команда Использовать Робот, слово робот закрашивается зеленым цветом;
2. составляем нужный алгоритм по условию задачи;
3. проверяем работает ли составленный алгоритм, для этого на панели инструментов нажимаем кнопку **Обычное** или клавишу **F9**.

Для каждой конкретной задачи используется свой алгоритм. Как помним, алгоритмы есть линейные, циклические, алгоритмы ветвления.

Рассмотрим примеры с исполнителем Робот, для каждого алгоритма по отдельности.

1. Линейные программы для исполнителя

Задача: Робот в произвольной точке поля. Передвинуть Робота на 5 клеток влево, закрасив их (рис.5). Напишем программу для Робота:

использовать Робот

алг лабиринт

нач

влево; закрасить

влево; закрасить

влево; закрасить

влево; закрасить

влево; закрасить

конец

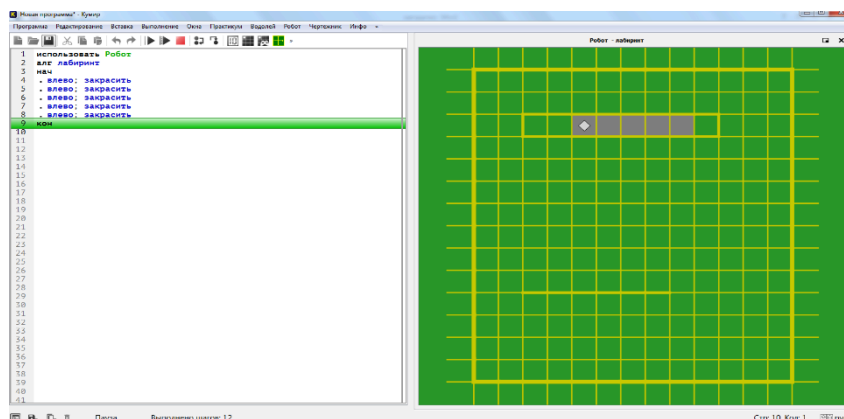


Рисунок 5 – Выполнение алгоритма «Лабиринт»

Как видно, клетки, первоначально находящиеся слева от Робота закрашены. В данном случае показан пример создания линейной программы для исполнителя.

2. Циклические алгоритмы для исполнителя

Цикл «пока»

Задача: закрасить все клетки справа от Робота, при условии, что справа на неизвестном расстоянии есть стена и количество клеток, которые нужно закрасить неизвестно.

Очевидно, что пока будет выполняться условие справа свободно, нужно выполнять команды: закрасить; вправо (рис.6)

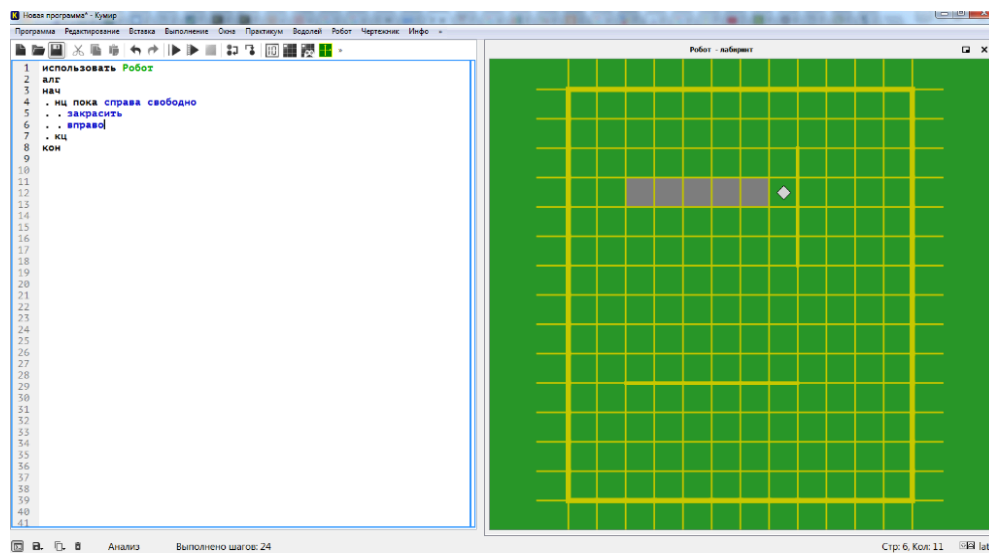


Рисунок 6 – Использование цикла «Пока»

Для оформления таких последовательностей действий используется специальная конструкция алгоритмического языка — цикл «пока».

Цикл «раз»

Задача: нарисовать букву «П» размер высотой 6 клеток, шириной 4 клетки с помощью исполнителя Робот. Исходное положение Робота показано на рисунке 7. В этом алгоритме известно количество шагов исполнителя, поэтому используем цикл «раз». Его конструкция в общем виде выглядит так:

НЦ количество РАЗ

КЦ

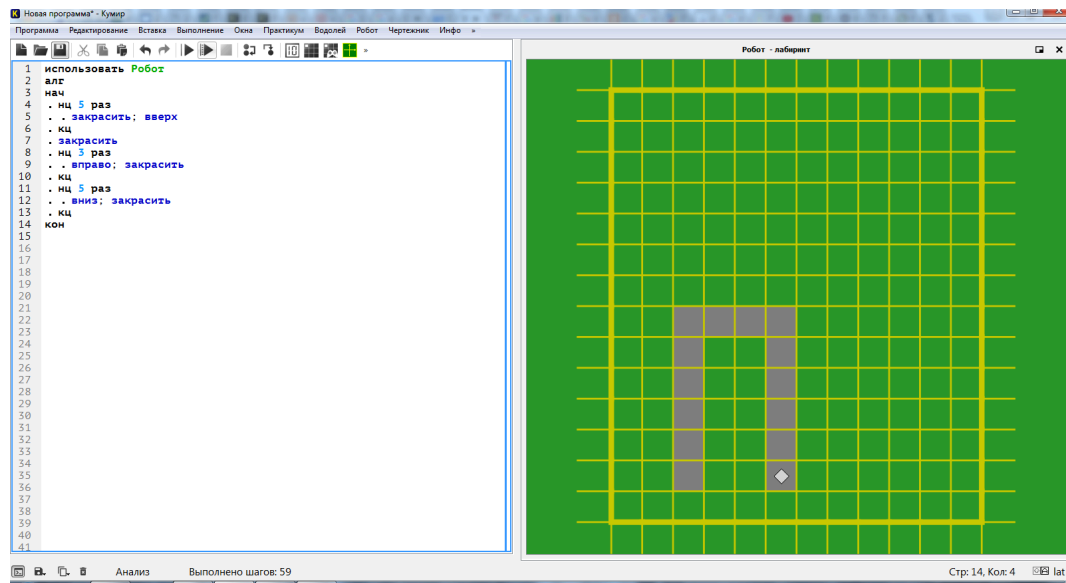


Рисунок 7 – «Буква»

По аналогии с предыдущим примером Робот должен закрашивать очередную клетку, а затем переходить на следующую – сначала вверх, затем вправо и вниз.

Программа для Робота:

использовать Робот

алг буква

нач

нц 5 раз

закрасить; вверх

кц

закрасить

нц 3 раз

вправо; закрасить

кц

нц 5 раз

вниз; закрасить

кц

кон

3. Алгоритм ветвление для исполнителя Робот

Вспомним, что форма организации действий, при которой в зависимости от выполнения или невыполнения некоторого условия совершается либо одна,

либо другая последовательность действий, называется ветвлением.

Для организации ветвлений исполнителя Робота предусмотрена специальная команда ЕСЛИ. Ее общий вид:

ЕСЛИ <условие> ТО <серия действий 1>

ИНАЧЕ <серия действий 2>

КОНЕЦ

Задача: Робот находится в горизонтальном коридоре, нижняя граница которого сплошная, а в нижней имеются выходы (рис. 8). Требуется провести Робота через весь коридор и закрасить клетки коридора, не имеющие границ сверху.

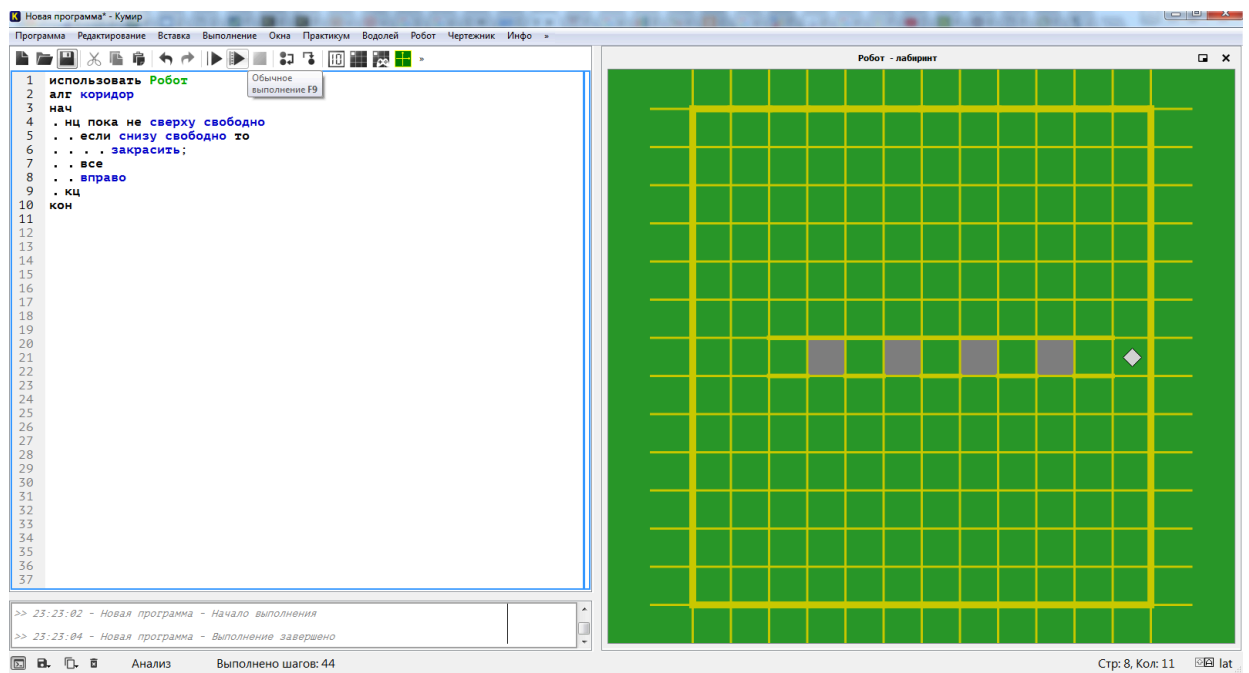


Рисунок 8 – «Коридор»

Единственным признаком коридора является наличие границы сверху, значит, условие НЕ сверху свободно должно выполняться. Если при этом выполняется условие снизу свободно, то клетку нужно закрасить, иначе —

закрашивать не надо. Известно, что слева и справа от горизонтального коридора есть клетки.

Таким образом, рассмотрев примеры написания трех основных типов алгоритмов и ознакомившись с пошаговым примером работы с исполнителем Робот, можно приступить к выполнению практической части.

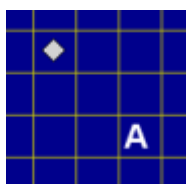
Практические работы для самостоятельного выполнения

1. Линейные алгоритмы

Задания рассчитаны на группу в 10 человек. Каждая карточка содержит 3 задания разной степени сложности.

Вариант 1

1. Составьте программу закрашивания прямоугольника 3×4 , считая, что Робот находится где-то в центре поля.
2. Необходимо перевести Робота из начального положения (\diamond) в точку А за минимальное число шагов любым из возможных способов.

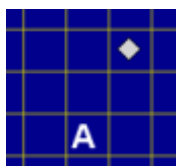


2. Необходимо перевести Робота по лабиринту из начального положения (\diamond) в точку А.



Вариант 2

1. Составьте программу закрашивания прямоугольника 4×2 , считая, что Робот находится где-то в центре поля.
2. Необходимо перевести Робота из начального положения (\diamond) в точку А за минимальное число шагов любым из возможных способов.



3. Необходимо перевести Робота по лабиринту из начального положения (\diamond) в точку А.



Вариант 3

1. Составьте программу закрашивания периметра квадрата 3×3 , считая, что Робот находится где-то в центре поля.
2. Необходимо перевести Робота из начального положения (\diamond) в точку А за минимальное число шагов любым из возможных способов.



3. Необходимо перевести Робота по лабиринту из начального положения (◇) в точку А.



Вариант 4

1. Составьте программу закрашивания квадрата 3×3, считая, что Робот находится где-то в центре поля.

2. Необходимо перевести Робота из начального положения (◇) в точку А за минимальное число шагов любым из возможных способов.



3. Необходимо перевести Робота по лабиринту из начального положения (◇) в точку А.



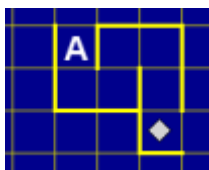
Вариант 5

1. Составьте программу закрашивания буквы "Г", состоящей из трех вертикальных и двух горизонтальных клеток. Начальное положение Робота - где-то в центре поля.

2. Необходимо перевести Робота из начального положения (◇) в точку А за минимальное число шагов любым из возможных способов.



3. Необходимо перевести Робота по лабиринту из начального положения (◇) в точку А.



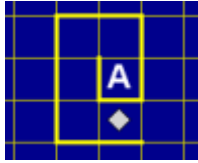
Вариант 6

1. Составьте программу закрашивания буквы "П", размерами три клетки по вертикали и две по горизонтали. Начальное положение Робота - где-то в центре поля.

2. Необходимо перевести Робота из начального положения (◇) в точку А за минимальное число шагов любым из возможных способов.



3. Необходимо перевести Робота по лабиринту из начального положения (◇) в точку А.



Вариант 7

1. Составьте программу закрашивания буквы "Т", размерами четыре клетки по вертикали и три по горизонтали. Начальное положение Робота - где-то в центре поля.
2. Необходимо перевести Робота из начального положения (◇) в точку А за минимальное число шагов любым из возможных способов.



3. Необходимо перевести Робота по лабиринту из начального положения (◇) в точку А.



Вариант 8

1. Составьте программу закрашивания клеток, отмеченных звездочкой. Начальное положение Робота - где-то в центре поля.



2. Необходимо перевести Робота из начального положения (◇) в точку А за минимальное число шагов любым из возможных способов.



3. Необходимо перевести Робота по лабиринту из начального положения (◇) в точку А.



Вариант 9

1. Составьте программу закрашивания клеток, отмеченных звездочкой. Начальное положение Робота - где-то в центре поля.



2. Необходимо перевести Робота из начального положения (◇) в точку A за минимальное число шагов любым из возможных способов.



3. Необходимо перевести Робота по лабиринту из начального положения (◇) в точку A.



Вариант 10

1. Составьте программу закрашивания клеток, отмеченных звездочкой. Начальное положение Робота - где-то в центре поля.



2. Необходимо перевести Робота из начального положения (◇) в точку A за минимальное число шагов любым из возможных способов.



3. Необходимо перевести Робота по лабиринту из начального положения (◇) в точку A.



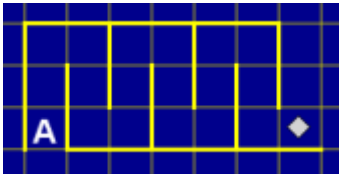
1. Циклические алгоритмы. Цикл N раз.

Задания рассчитаны на группу в 10 человек. Каждая карточка содержит 3 задания разной степени сложности.

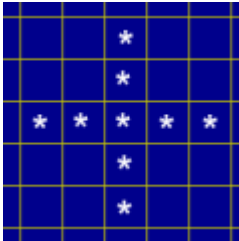
Вариант 1

1. Составьте программу закрашивания 6-ти клеток вправо от Робота, считая, что изначально Робот находится у левого края поля.

2. Необходимо провести Робота по лабиринту из начального положения (◇) в точку A.



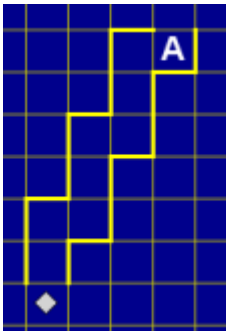
3. Составьте программу закрашивания клеток поля, отмеченных звездочкой. Начальное положение Робота где-то в центре поля.



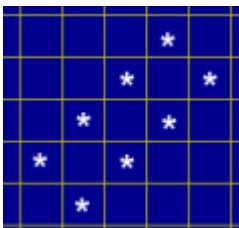
Вариант 4

1. Составьте программу закрашивания 7-ти клеток по диагонали вправо вверх от Робота, считая, что изначально Робот находится в левом нижнем углу поля.

2. Необходимо провести Робота по лабиринту из начального положения (◇) в точку A.



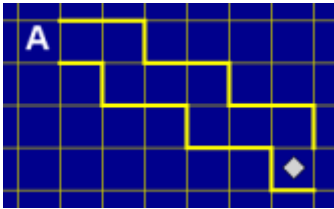
3. Составьте программу закрашивания клеток поля, отмеченных звездочкой. Начальное положение Робота где-то в центре поля.



Вариант 5

1. Составьте программу закрашивания 4-х клеток по диагонали влево вверх от Робота, считая, что изначально Робот находится в правом нижнем углу поля.

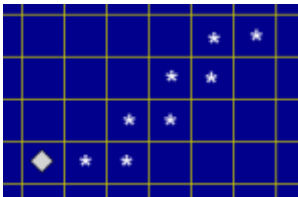
2. Необходимо провести Робота по лабиринту из начального положения (◇) в точку A.



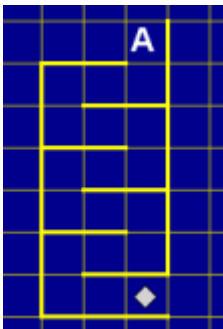
3. Считая, что Робот находится в левом нижнем углу поля, составьте программу закрашивания клеток поля в виде буквы "Г" высотой в 6 клеток.

Вариант 6

1. Составьте программу закрашивания клеток поля, отмеченных *. Начальное положение Робота обозначено значком \diamond .



2. Необходимо провести Робота по лабиринту из начального положения (\diamond) в точку А.



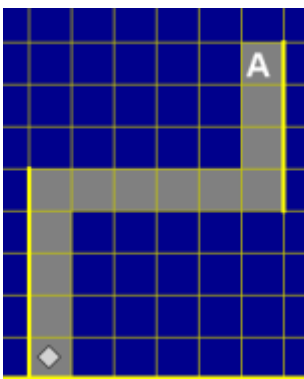
3. Считая, что Робот находится в левом нижнем углу поля, составьте программу закрашивания клеток поля в виде буквы "О" высотой в 6 клеток.

Вариант 7

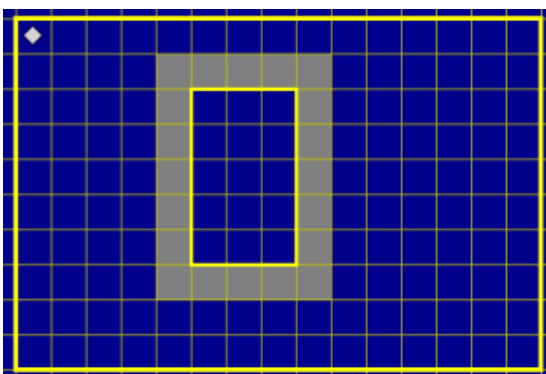
1. Составьте программу закрашивания клеток поля, отмеченных *. Начальное положение Робота обозначено значком \diamond .



2. Необходимо провести Робота вверх по лестнице из начального положения (\diamond) в точку А.

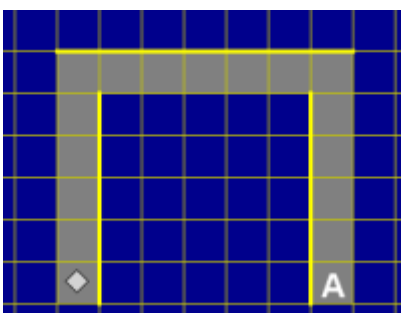


2. Где-то в поле Робота находится прямоугольник, размеры которой неизвестны. Робот из верхнего левого угла поля должен прийти до прямоугольника и закрасить все клетки по периметру вокруг него. Пример результата работы программы показан на рисунке.

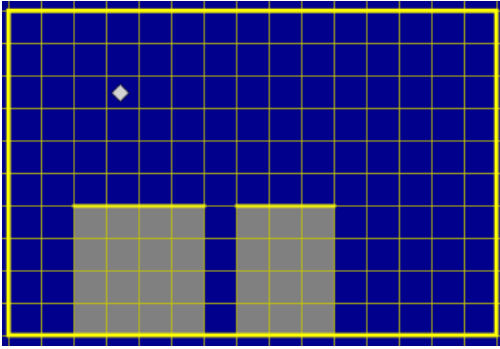


Вариант 3

1. Необходимо перевести Робота из начального положения (\diamond) в точку A, закрашивая при этом указанные клетки поля. Размеры стен и расстояние между ними могут быть произвольны.

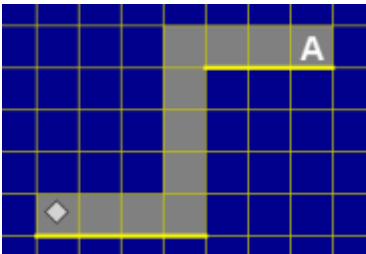


2. Где-то в поле Робота находится горизонтальная стена с отверстием в одну клетку, размеры которой неизвестны. Робот из произвольной клетки над стеной, но обязательно напротив нее, должен прийти до стены и закрасить все клетки под ней, как показано на рисунке.

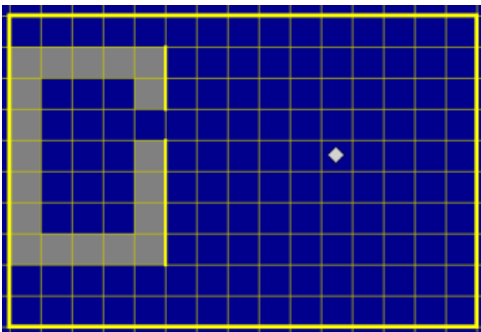


Вариант 4

1. Необходимо перевести Робота из начального положения (\diamond) в точку A, закрашивая при этом указанные клетки поля. Размеры стен и расстояние между ними могут быть произвольны.

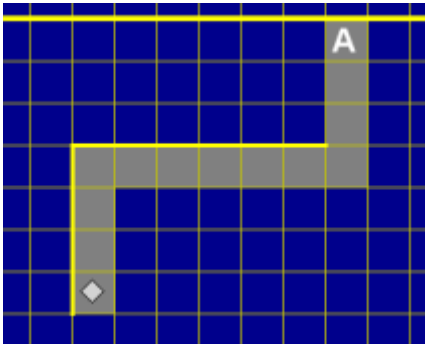


2. Где-то в поле Робота находится вертикальная стена с отверстием в одну клетку, размеры которой неизвестны. Робот из произвольной клетки справа от стены, но обязательно напротив нее, должен прийти до стены и закрасить клетки за стеной, как показано на рисунке.

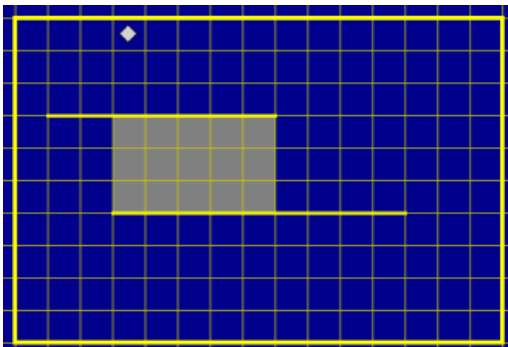


Вариант 5

1. Необходимо перевести Робота из начального положения (\diamond) в точку A, закрашивая при этом указанные клетки поля. Размеры стен и расстояние между ними могут быть произвольны.

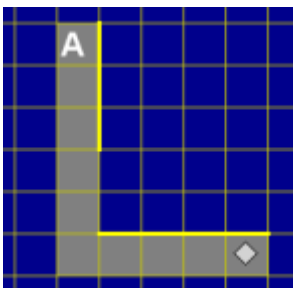


2. Где-то в поле Робота находятся две горизонтальные стены смещенные друг относительно друга, размеры которых неизвестны. Робот из произвольной клетки над верхней стеной, но обязательно напротив нее, должен дойти до стены и закрасить все клетки между стенами, как показано на рисунке.

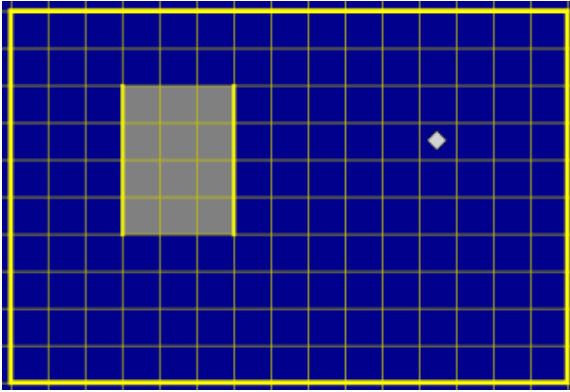


Вариант 6

1. Необходимо перевести Робота из начального положения (\diamond) в точку A, закрашивая при этом указанные клетки поля. Размеры стен и расстояние между ними могут быть произвольны.

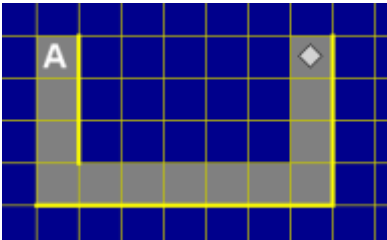


2. Где-то в поле Робота находятся две вертикальный стены равной, но неизвестной длины, расположенные одна за другой. Робот из произвольной клетки справа от правой стены, но обязательно напротив нее, должен дойти до стены и закрасить клетки между стенами, как показано на рисунке.

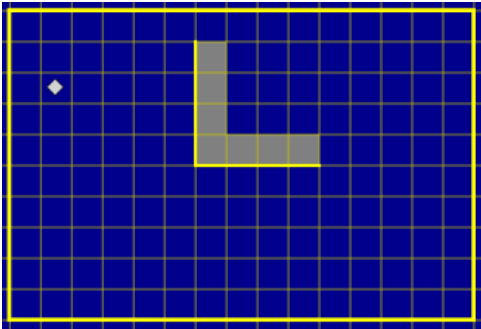


Вариант 7

1. Необходимо перевести Робота из начального положения (\diamond) в точку A, закрашивая при этом указанные клетки поля. Размеры стен и расстояние между ними могут быть произвольны.

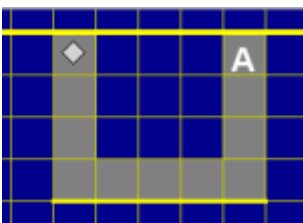


2. Где-то в поле Робота находится стена в виде уголка, размеры которой неизвестны. Робот из произвольной клетки слева от стены, но обязательно напротив нее, должен прийти до стены и закрасить все клетки вдоль стены за ней, как показано на рисунке.

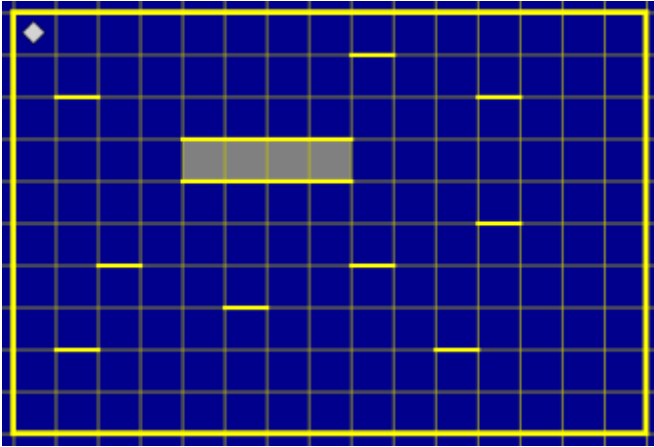


Вариант 8

1. Необходимо перевести Робота из начального положения (\diamond) в точку A, закрашивая при этом указанные клетки поля. Размеры стен и расстояние между ними могут быть произвольны.

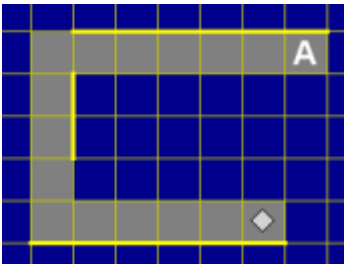


2. Где-то в поле Робота находится горизонтальный коридор шириной в одну клетку неизвестной длины. Робот из верхнего левого угла поля должен прийти до коридора и закрасить клетки внутри него, как показано на рисунке. По полю Робота в произвольном порядке располагаются стены, но расстояние между ними больше одной клетки.

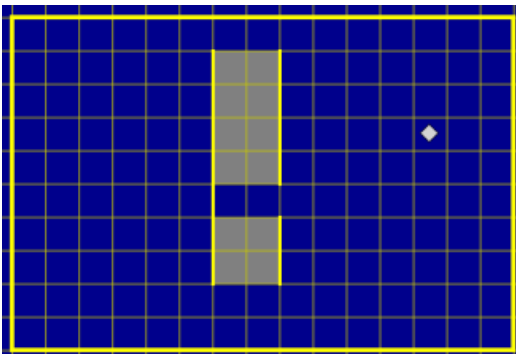


Вариант 9

1. Необходимо перевести Робота из начального положения (\diamond) в точку A, закрашивая при этом указанные клетки поля. Размеры стен и расстояние между ними могут быть произвольны.

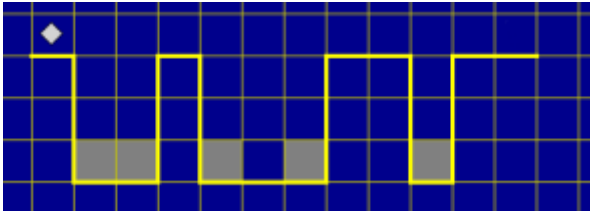


2. Где-то в поле Робота находятся две стены равной, но неизвестной длины. В правой стене имеется отверстие шириной в одну клетку. Робот из произвольной клетки справа от правой стены, но обязательно напротив нее, должен прийти до стены и закрасить все клетки между двумя стенами, как показано на рисунке.




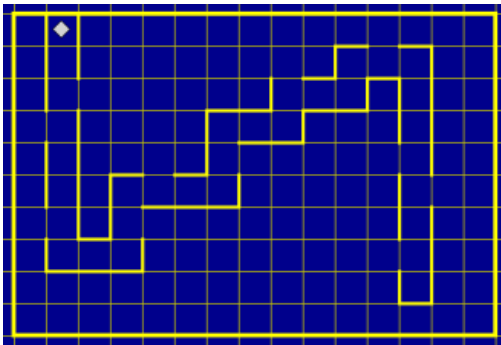
Вариант 10

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.



Вариант 2


1. Необходимо провести Робота по коридору шириной в одну клетку из начального положения (◇) до конца коридора, закрасивая 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 . Пример коридора показан на рисунке.

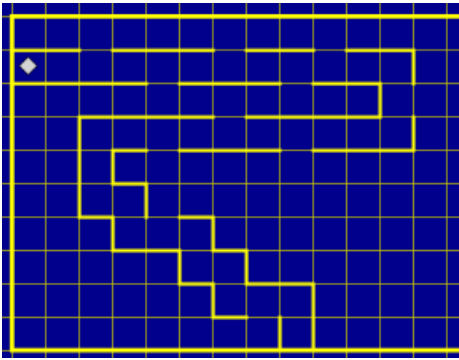


2. Робот движется вдоль стены, профиль которой показан на рисунке, от начального положения (◇) до конца стены. Необходимо закрасить все внутренние углы стены, как показано на примере. Размеры стены могут быть произвольны.

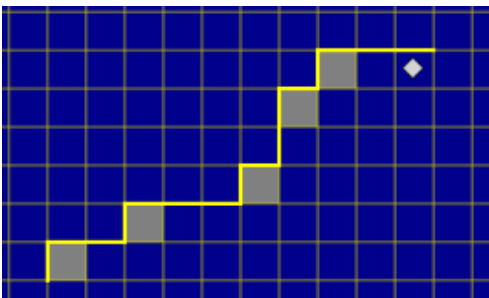


Вариант 3

1. Необходимо провести Робота по коридору шириной в одну клетку из начального положения (◇) до конца коридора, закрасивая 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 . Пример коридора показан на рисунке.

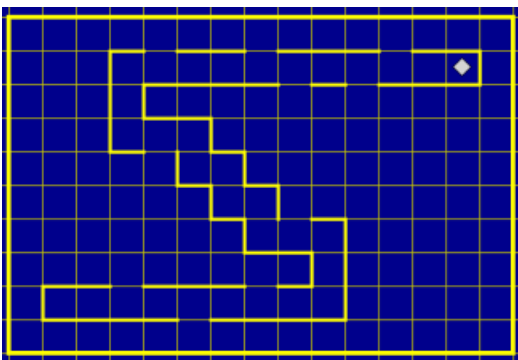


2. Робот движется вдоль стены от начального положения (\diamond) до конца стены. Стена располагается по диагонали от правого верхнего к левому нижнему углу поля. Необходимо закрасить все внутренние углы стены, как показано на примере. Размеры стены могут быть произвольны.

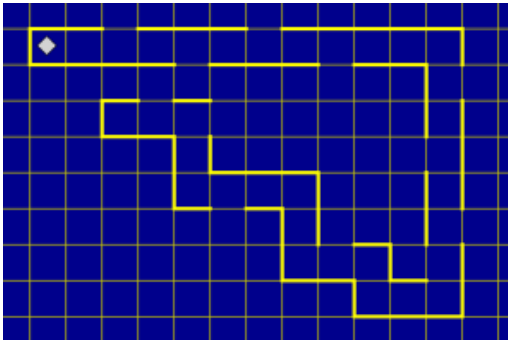


Вариант 4

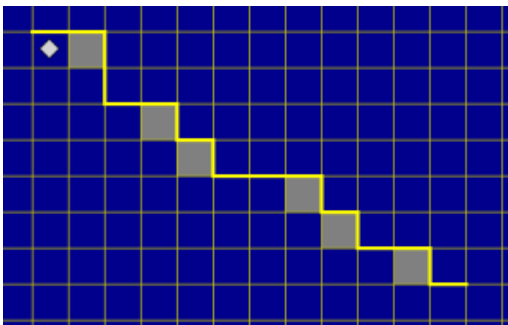
1. Необходимо провести Робота по коридору шириной в одну клетку из начального положения (\diamond) до конца коридора, закрасив при этом все клетки коридора, которые имеют выход. Выходы размером в одну клетку располагаются произвольно по всей длине коридора. Коридор заканчивается тупиком. Коридор имеет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




2. Робот движется вдоль стены от начального положения (\diamond) до конца стены. Стена располагается по диагонали от правого нижнего к левому верхнему углу поля. Необходимо закрасить все внутренние углы стены, как показано на примере. Размеры стены могут быть произвольны.

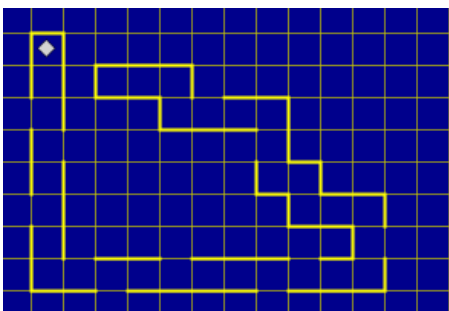


2. Робот движется вдоль стены от начального положения (◇) до конца стены. Стена располагается по диагонали от правого нижнего к левому верхнему углу поля. Необходимо закрасить все внутренние углы стены, как показано на примере. Размеры стены могут быть произвольны.

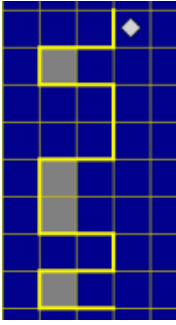


Вариант 7


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

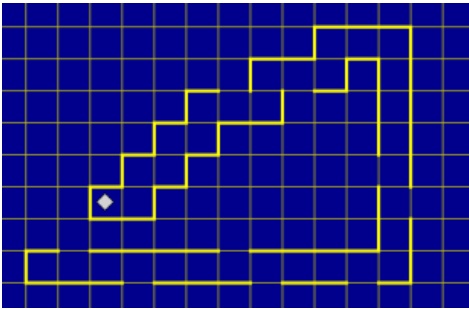


2. Робот движется вдоль стены, профиль которой показан на рисунке, от начального положения (◇) до конца стены. Необходимо закрасить все внутренние углы стены, как показано на примере. Размеры стены могут быть произвольны.

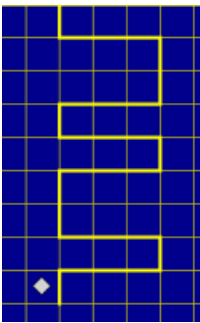


Вариант 8


1. Необходимо провести Робота по коридору шириной в одну клетку из начального положения (\diamond) 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 горизонтальный, вертикальный и диагональный участки в форме . Пример коридора показан на рисунке.

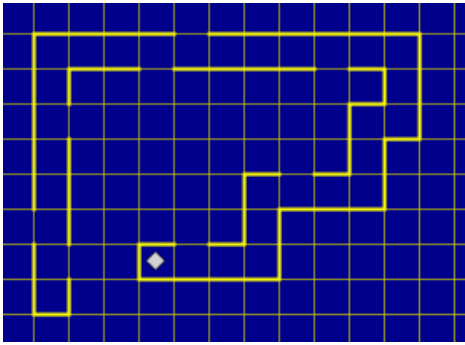


2. Робот движется вдоль стены, профиль которой показан на рисунке, от начального положения (\diamond) до конца стены. Необходимо закрасить все внутренние углы стены, как показано на примере. Размеры стены могут быть произвольны.

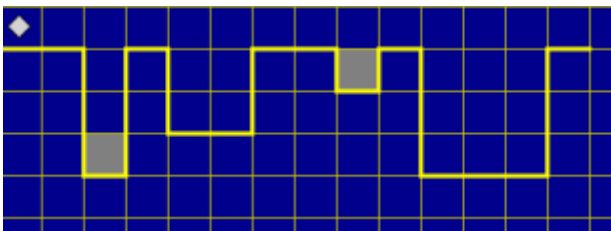


Вариант 9

1. Необходимо провести Робота по коридору шириной в одну клетку из начального положения (\diamond) 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 горизонтальный, вертикальный и диагональный участки в форме . Пример коридора показан на рисунке.

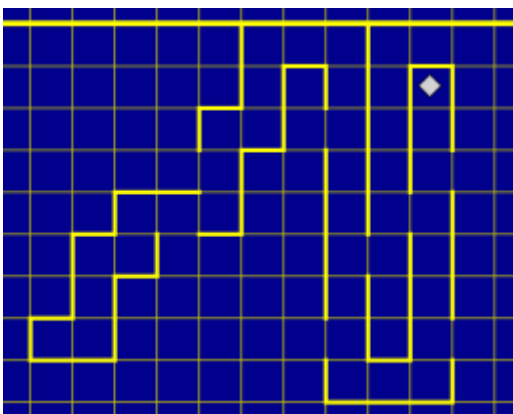


2. Робот движется вдоль стены, профиль которой показан на рисунке, от начального положения (\diamond) до конца стены. Необходимо закрасить все клетки стены, которые огорожены с трех сторон, как показано на примере. Размеры стены могут быть произвольны.

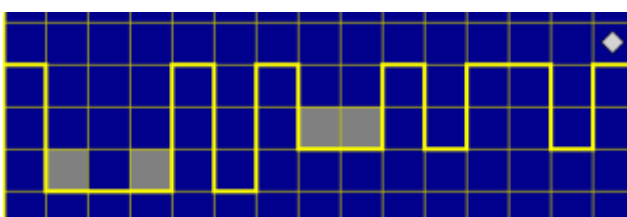


Вариант 10

1. Необходимо провести Робота по коридору шириной в одну клетку из начального положения (\diamond) 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 вертикальных и диагональный участки в форме $\begin{matrix} \diagup \\ | \end{matrix}$. Пример коридора показан на рисунке.



2. Робот движется вдоль стены, профиль которой показан на рисунке, от начального положения (\diamond) до конца стены. Необходимо закрасить все углы стены, которые огорожены только с двух сторон, как показано на примере. Размеры стены могут быть произвольны.



СПИСОК ЛИТЕРАТУРЫ

1. Босова Л.Л. Информатика: уч. для 9 кл. –2-е изд., испр. – М.: БИНОМ. Лаборатория знаний, 2014
2. Камалова Н.А.<http://festival.1september.ru/articles/594645/>
3. Нурмухамедов Г.М. Информатика. Теоретические основы: учеб. Пос. для подготовки к ЕГЭ. – СПб.: БХВ-Петербург, 2012
4. Поляков К.Ю., Еремин Е.А. Информатика. Углубленный уровень: уч. для 10кл.: в 2 ч. Ч.1. – М.: БИНОМ. Лаборатория знаний, 2013
6. Семакин И.Г., Залогова Л.А., Русаков С.В. Информатика и ИКТ: уч. для 9 кл. – М.: БИНОМ. Лаборатория знаний, 2012
7. Сухих Н.А.Поурочные разработки по информатике: 9 класс. –М.: ВАКО, 2012.