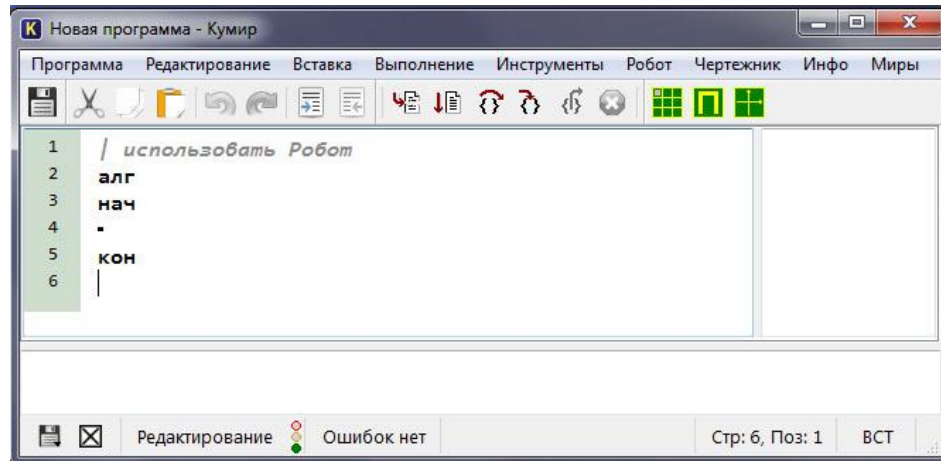


Практикум по КуМир

В данном практикуме будут рассмотрены *только* возможности исполнителя Робот, которые можно использовать для написания программы при выполнении задания ГИА.

После запуска среды КуМир на экране вы увидите следующее окно:



1. Обстановки исполнителя Робот

Исполнитель Робот существует в некоторой *обстановке* – прямоугольном поле, разбитом на клетки, между которыми могут стоять стены. Обстановка, в которой находится Робот, называется *текущей* обстановкой Робота. Определяется также *стартовая* обстановка, которая используется при управлении Роботом из программы.

Робот может передвигаться по полю, закрашивать клетки. Робот не может проходить сквозь стены, но может проверять, есть ли рядом с ним стена. Робот не может выйти за пределы заданного прямоугольного поля.

Обстановки Робота могут храниться в файлах специального формата (расширение .fil).

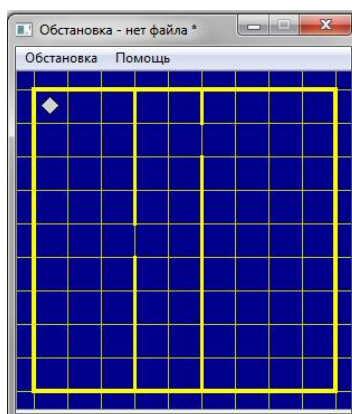
2. Как установить стартовую обстановку

Определить стартовую обстановку можно двумя способами:

1. Загрузить стартовую обстановку из ранее созданного файла.
2. Редактировать существующую стартовую обстановку с помощью специального редактора стартовых обстановок, а затем сохранить в файл или непосредственно использоваться в качестве стартовой обстановки.

3. Редактирование стартовой обстановки

Запуск редактирования производится с помощью команды *Редактировать стартовую обстановку* меню *Инструменты*. По этой команде появляется специальное окно редактирования стартовой обстановки (окно редактирования), фон которого синий.



В окне редактирования над рабочим полем есть стандартная полоса, содержащая главное меню окна и инструментальные кнопки (*Свернуть, Развернуть, Заккрыть*)

Выход из состояния редактирования стартовой обстановки производится с помощью соответствующей кнопки на окне редактирования или по команде *Выход* меню *Обстановка* на этом окне.

Примечание

Редактировать одновременно две стартовые обстановки нельзя.

4. Команды меню *Обстановка*

Новая обстановка

Вначале необходимо задать размер формы (количество строк от 1 до 10 и столбцов от 1 до 16), а затем открывается стандартная форма.

Открыть

Необходимо выбрать один из уже существующих файлов *.fil, который содержит ранее созданные обстановки.

Недавние обстановки

В качестве недавних обстановок запоминаются все обстановки, к которым применялись операции сохранения и записи.

Сохранить

Команда сохраняет внесенные изменения обстановки в текущий файл. Если обстановка была создана командой *Новая обстановка*, то запрашивает имя файла.

Сохранить как. . .

Работает так же, как *Сохранить*, но с обязательным запросом имени файла сохранения.

Сохранить как стартовую

Работает так же, как *Сохранить как*, но с изменением имени файла стартовой обстановки.

Печать в файл

Создание PDF-файла с изображением окна.

Выход

Завершает редактирование. Если последняя обстановка не сохранена, то выдает запрос о сохранении изменений.

5. Непосредственное редактирование обстановки

Все команды редактирования обстановки выполняются с помощью мыши:

- поставить/убрать стену – щелкнуть по границе между клетками,
- закрасить/очистить клетку – щелкнуть по клетке,
- переместить Робота – перетащить мышью в нужную клетку.

6. Команды меню *Робот*

Показать поле Робота

Делает видимым окно наблюдения за Роботом.

Напечатать обстановку

Создает файл в формате PDF, изображающий текущую обстановку в цветном или в черно-белом варианте.

Сохранить обстановку в файл

Создает текстовый файл с описанием обстановки во внутреннем формате .fil. Этот файл в дальнейшем может быть загружен в качестве стартовой обстановки (команда *Сменить стартовую обстановку*) или при редактировании стартовой обстановки (команда *Открыть* окна редактирования стартовой обстановки).

Сменить как стартовую обстановку

Устанавливает новое имя файла стартовой обстановки (с помощью стандартного диалога) и загружает новую стартовую обстановку.

Вернуться в стартовую обстановку

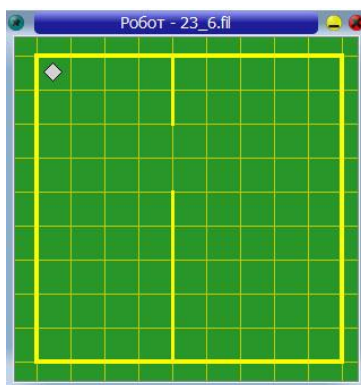
Делает стартовую обстановку текущей.

7. Изображение текущей обстановки в окне наблюдения

Изображение текущей обстановки всегда полностью помещается в рабочем поле окна наблюдения за Роботом.

Фон рабочего поля – зеленый. Закрашенные клетки – серые. Между клетками – тонкие черные линии. Стены изображаются толстыми желтыми линиями.

В клетке рабочего поля окна наблюдения Робот изображается ромбиком.



8. Команды исполнителя *Робот*

Команды перемещения:

вверх, вниз, влево, вправо

Робот перемещается на одну клетку вверх, вниз, влево, вправо.

Команда **закрасить** – закрашивает клетку, в которой стоит Робот.

Проверка истинности условия:

сверху свободно, снизу свободно, слева свободно, справа свободно

Робот проверяет истинность условия отсутствия стены у той клетки, где находится Робот.

Можно использовать запись составных условий, образованных логическими операциями **И, ИЛИ, НЕ**.

Пример

(справа свободно) И (не снизу свободно)

Проверка истинности условия:

закрашена, чистая

Робот проверяет истинность условия – закрашена или чистая клетка, где находится Робот.

9. Команды алгоритмического языка КуМир

Команда ветвления

Полная форма:

```
если <проверка условия>
    то    <последовательность команд-1>
    иначе <последовательность команд-2>
все
```

Команда ветвления выполняет *последовательность команд-1*, если *проверка условия* принимает значение истины, в противном случае выполняется *последовательность команд-2*.

Примечание

Каждая команда начинается с новой строки. Допускается написание нескольких команд в одной строке, отделяя команды точкой с запятой.

Сокращенная форма:

```
если <проверка условия>
    то    <последовательность команд-1>
все
```

Пример

```
если (справа свободно) И (не снизу свободно) то
    вправо
все
```

Команда повторения ПОКА

```

нц пока <проверка условия>
    <последовательность команд>
кц

```

Команда повторяет *последовательность команд*, пока *проверка условия* истина.

Пример

```

нц пока не справа свободно
    вниз
    закрасить
кц

```

Команда повторения заданное число раз

```

нц <число повторений> раз
    <последовательность команд>
кц

```

Команда повторяет *последовательность команд*, пока *проверка условия* истина.

Пример

```

нц 6 раз
    нц пока справа свободно
        вправо
    кц
    нц пока (не справа свободно) и (снизу свободно)
        вниз
    кц
    нц пока (не справа свободно) и (сверху свободно)
        вверх
    кц
кц

```

Цикл «для»

```

нц для К от <начальное значение> до <конечное значение>
    <последовательность команд>
кц

```

Команда повторяет *последовательность команд* для переменной К, которая изменяется от *<начальное значение>* до *<конечное значение>* с шагом 1.

Если *<начальное значение>* = *<конечное значение>*, то тело цикла выполнится один раз.

Если *<начальное значение>* > *<конечное значение>*, то тело цикла не выполнится ни разу.

Пример

```

нач
    цел i, j, b
    b:=5

```

```

нц для j от 1 до b
нц для i от 1 до j
закрасить
вниз
кц
вправо
нц для i от 1 до j
вверх
кц
кц

```

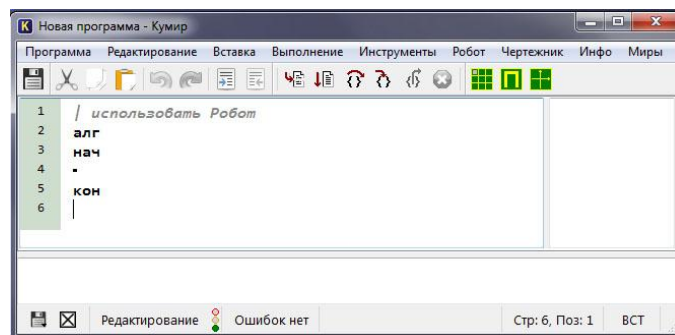
10. Начало работы

Последовательность создания программы для исполнителя Робот в среде КуМир:

1. Создать стартовую обстановку (*Редактировать стартовую обстановку* меню *Инструменты*). В меню *Обстановка*:
 - а. **Новая обстановка.** Выбор размера рабочего поля.
 - б. Редактирование обстановки – расстановка стенок, определение начальной позиции робота.
 - с. **Сохранить как стартовую.** Сохранить стартовую обстановку в файл .fil.
2. Написание программы в поле алгоритма.
3. Запуск программы на выполнение (*Выполнить непрерывно* меню *Выполнение* или клавиша F9).

11. Пример написания программы

После запуска среды КуМир на экране вы увидите поле алгоритма, в котором создается программа для исполнителя.



Первая строка содержит знак комментария «|», т.е. эта строка не воспринимается как код программы.

Внимание! Для работы с исполнителем Робот нам потребуется удалить знак «|».

Следующие строки содержат служебные слова **алг**, **нач**, **кон**. Они служат для оформления программы. Часть алгоритма до служебного слова **нач** называют заголовком, а часть между **нач** и **кон** – телом программы. Можно вспомнить, что в языке Pascal имеется похожая структура:

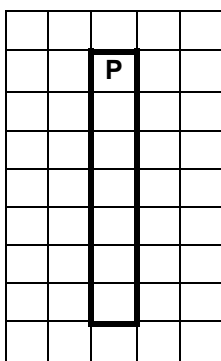
```

program
...
  begin
    ...
  end.

```

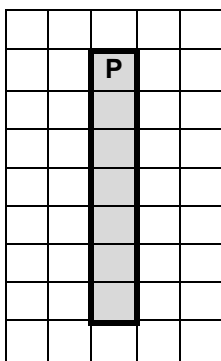
Задача

Робот находится в верхней клетке узкого вертикального коридора. Ширина коридора – одна клетка, длина коридора может быть произвольной. Возможный вариант начального расположения Робота приведён на рисунке (Робот обозначен буквой «Р»):



Напишите для Робота алгоритм, закрашивающий все клетки внутри коридора и возвращающий Робота в исходную позицию. Например, для приведённого выше рисунка

Робот должен закрасить следующие клетки:



Алгоритм должен решать задачу для произвольного конечного размера коридора. При исполнении алгоритма Робот не должен разрушиться.

Решение:

```

использовать Робот
алг
нач
  | Закрасить клетку и спуститься вниз
нц пока снизу свободно
  закрасить
вниз
кц

```

```
| Закрасить последнюю клетку
закрасить
| Подняться вверх в исходное положение
нц пока сверху свободно
  вверх
кц
кон
```

Для упрощения работы и исключения ошибок при наборе текста рекомендуется использовать команду меню *Вставка*. С помощью этой команды можно вставить готовые структуры команд.

алг-нач-кон	(ESC, A)	Esc, F
если-то-все	(ESC, E)	Esc, T
выбор-при-все	(ESC, B)	Esc, D
иначе	(ESC, И)	Esc, B
нц-раз-кц	(ESC, P)	Esc, H
нц-для-кц	(ESC, Д)	Esc, L
нц-пока-кц	(ESC, П)	Esc, G
нц-кц	(ESC, Ц)	Esc, W
исп-кон_исп	(ESC, С)	Esc, C
вверх		Esc, Up
вправо		Esc, Right
вниз		Esc, Down
влево		Esc, Left
закрасить		Esc, Space
использовать Робот		Esc, 1
использовать Чертежник		Esc, 2
использовать Файлы		Esc, 3
использовать Кузнечик		Esc, 4
использовать Черепаха		Esc, 5
использовать Водолей		Esc, 6

12. Сохранение текста программы

Сохранить программу, созданную в среде КуМир, можно с помощью команды *Сохранить программу* меню *Программа*. Сохраненный файл с текстом программы будет иметь расширение *.kum*.

Внимание! Не забудьте правильно установить папку, в которой вы будете сохранять свои программы.