

**АКАДЕМИЯ НАУК СССР  
НАУЧНЫЙ ЦЕНТР БИОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ  
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ**

**Н. Л. ЛУНИНА**

**ФАКУЛЬТАТИВНЫЙ КУРС  
ПРОГРАММИРОВАНИЯ  
ОСНОВНЫЕ ПОНЯТИЯ**

**ПУЩИНО.1989**

**АКАДЕМИЯ НАУК СССР  
НАУЧНЫЙ ЦЕНТР БИОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ  
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ**

**Н. Л. ЛУНИНА**

**ФАКУЛЬТАТИВНЫЙ КУРС  
ПРОГРАММИРОВАНИЯ  
ОСНОВНЫЕ ПОНЯТИЯ**

**ПУЩИНО.1989**

УДК 512.519.68

В работе изложена первая часть факультативного курса программирования. Курс рассчитан на два года и ориентирован на учащихся 7-8 классов. Общая программа курса

Основные понятия

Бейсик

Паскаль

Фортран

Работа будет полезна старшеклассникам и преподавателям информатики и программирования.

Научный редактор - Т.А.Боровина.

© Научный центр биологических исследований  
АН СССР в Пущине, 1989 г.

## Предисловие

В настоящее время большое внимание уделяется обучению школьников основам информатики и программирования. Работа эта ведется в разных формах.

Настоящая работа является первой частью факультативного курса программирования. Курс рассчитан на два года и ориентирован на учащихся 7–8 классов. Общая программа курса

Основные понятия

Бейсик

Паскаль

Фортран

Каждая из частей рассчитана примерно на полгода.

При изложении основных понятий используется программное обеспечение, разработанное на механико-математическом факультете МГУ под руководством А.Р.Кушниренко.

В то же время данное пособие не дублирует учебник [1], созданный авторами программного обеспечения. Пособие отличается от учебника меньшей полнотой и объемом, стилем и последовательностью изложения, а также задачами и упражнениями. Учебник рассчитан на всеобщее обязательное обучение, как машинный, так и безмашинный вариант; данное пособие рекомендуется для факультативного обучения при наличии ЭВМ. При работе использовалась ЭВМ "Ямаха"; практически все остается верным для ЭВМ "Корвет".

Данное пособие успешно использовалось в различных условиях. Автору приходилось работать со школьниками обычной, не специализированной школы. Нагрузка составляла 4 часа в неделю (3 часа днем и 1 час вечером). Дневные занятия состояли из объяснения нового материала, письменной работы по новому материалу и выполнения упражнений на машине. В большинстве случаев одно дневное занятие соответствовало одному разделу пособия. Вечером ученики решали задачи либо самостоятельно выполняли другую работу на ЭВМ. Каждый школьник был обеспечен машиной и экземпляром пособия. По каждой из четырех частей курса ученики сдавали зачет.

Автор благодарит Т.А.Боровину за апробацию и редактирование данного пособия.

## 1. Основные конструкции

### 1.1. Исполнитель "Черепашка"

Черепашка бегаёт по песку с поднятым или опущенным хвостом. Когда хвост опущен, на песке остаются следы. Управляя черепашкой, можно получить на песке различные рисунки. Черепашка понимает команды:

```
черепашка
вперед ( )
назад ( )
направо ( )
налево ( )
поднять_хвост
опустить_хвост
```

При этом в командах ВПЕРЕД и НАЗАД указывается количество шагов; НАПРАВО и НАЛЕВО – угол в градусах. По своему характеру количество шагов должно быть целым. Если, однако, будет задано не целое число шагов, это не будет восприниматься как ошибка, а произойдет округление указанного числа до меньшего целого (предписание ВПЕРЕД(2.8) работает как ВПЕРЕД(2) ).

Размеры песочницы 255×85 черепашьих шагов. Первой командой в программе должна быть команда ЧЕРЕПАШКА. При выполнении ее черепашка встает в левый верхний угол песочницы; смотрит на юго-восток; хвост поднят. При выполнении программы

```
алг рисунок
данно !черепашка в левом верхнем углу
надо !нарисован квадрат 40×40; 20 шагов от края
нач
черепашка
вперед(20)
налево(45)
опустить_хвост
вперед(40)
направо(90)
вперед(40)
направо(90)
вперед(40)
направо(90)
вперед(40)
направо(90)
вперед(40)
направо(90)
```

кон

будет нарисован квадрат 40x40 со сторонами, параллельными стенкам песочницы, отстоящий на 20 шагов от края.

Слова АЛГ, ДАНО, НАДО, НАЧ, КОН являются обязательными элементами оформления программ; они появляются на экране независимо от программиста для каждой новой программы. Словами "алгоритм" и "программа" мы будем пользоваться как синонимами. Имя алгоритма (в нашем случае РИСУНОК) программист выбирает по своему желанию; на выполнение программы оно не оказывает влияния. Имя должно начинаться с буквы и не должно содержать "плохих" символов; вы сами со временем поймете, какие символы считаются "плохими". Буквы, цифры, подчеркивание являются "хорошими" символами.

Информация в строках ДАНО и НАДО не оказывает влияния на выполнение программы. Она облегчает чтение чужой программы (в любом случае) и написание своей (когда программа становится достаточно сложной).

Наша программа написана так, что в одной строке находится одно предписание. Для маленьких программ удобна именно такая форма записи. Однако можно записывать и несколько предписаний в одной строке, и тогда они должны разделяться знаком ";" (точка с запятой).

#### Как ввести и выполнить программу

Для выполнения программы на машине нужно ввести в машину ее текст. Когда вы включите машину, то, скорее всего, получите от учителя некоторую программу для демонстрации возможностей черепашки. Чтобы выполнить уже написанную программу, нужно нажать клавишу F9. Посмотрите внимательно на верхние черные клавиши, расположенные слева. Это функциональные клавиши. Их 5. На каждой из них две надписи, например F1 и чуть выше F6. Нижняя надпись на функциональной клавише соответствует тому случаю, когда клавиша нажимается сама по себе; верхняя - когда клавиша нажимается одновременно с клавишей SHIFT. На клавиатуре есть две клавиши SHIFT; они совершенно равноправны. Итак, для того, чтобы выполнить полученную от учителя программу, надо нажать две клавиши: клавишу SHIFT и одновременно с ней - функциональную клавишу с надписью F4 F9. Такую ситуацию мы будем называть нажатием клавиши F9.

Если нажать клавишу F4 без клавиши SHIFT, то программа тоже начнет выполняться. Однако это будет другой режим выполнения - пошаговый, когда для выполнения очередной команды необходимо оче-

редной раз нажать F4.

При выполнении программы часть экрана отводится под рисунок. Программа занимает нижнюю часть экрана. Справа на полях появляется надпись "Ж Выполнение Ж". Когда программа заканчивает работу, справа на полях появляется надпись "ЖЖЖЖ Конец ЖЖЖЖ".

После того, как демонстрационная программа выполнена, она нам больше не нужна. Ее текст нужно удалить с экрана. Для этого нам придется установить курсор в нужное нам положение. Посмотрите внимательно на часть экрана с текстом. В одном месте (на букве К слова КОН) вы увидите белый прямоугольник. Это курсор. Нажимая клавиши со стрелками, мы можем передвигать курсор по экрану. Нажмем стрелку "вверх". Курсор начнет передвигаться вверх по экрану, а затем – вверх по программе, пока не установится на самой первой строке программы. Проверьте – это должна быть строка со словом АЛГ. Дальнейшее нажатие стрелки не вызывает передвижение курсора; при этом мы слышим некоторый характерный звук, сообщающий о том, что наша команда не выполняется. В этот момент нужно нажать две клавиши – SHIFT и, не отжимая ее, – клавишу DEL. (Мы будем обозначать такое нажатие SHIFT/DEL). Текст алгоритма, строка за строкой, начнет исчезать с экрана. Если присланная к нам программа состояла из нескольких алгоритмов, то курсор установится на строке АЛГ следующего алгоритма. Тогда нажмем SHIFT/DEL еще раз.

Наконец, перед нами останется пустой экран, на котором будет видна надпись [Жконец текстаЖ]. Эту надпись нам уничтожить не удастся. На этом работа по уничтожению закончится; пора создавать новую программу. Прежде всего, для ввода новой программы нам понадобятся служебные слова АЛГ, НАЧ, КОН. Чтобы ввести эти слова (рамку для нового алгоритма), нужно нажать две клавиши – ESC а затем клавишу с буквой "А". Рамка выглядит так:

```
алг
  дано
  надо
  нач
  кон
```

Сами эти слова нельзя изменить; нельзя стереть одно из этих слов. Можно удалить лишь всю конструкцию целиком, установив курсор на строку АЛГ и нажав SHIFT/DEL.

Справа на полях в строке АЛГ мы видим сообщение "Нет имени". Сообщения такого типа будут появляться во время ввода текста программы и, как правило, сигнализировать об ошибках.

Приступим к заполнению рамки. После слова АЛГ через пробел напишем имя алгоритма. Напомним, что имя может содержать русские

и латинские буквы, а также цифры и некоторые служебные символы. Первым символом имени должна быть буква. Заметим, что установлен русский регистр (горит клавиша РУС). Если нам понадобится написать что-то по-английски, мы можем поступить двумя путями:

1) набирать английские буквы при горячей клавише РУС, нажимая одновременно с нужной буквой клавишу CAPS. Как только мы прекратим пользоваться клавишей CAPS, снова будут набираться русские буквы;

2) нажать клавишу CTRL и, не отжимая, – клавишу РУС. Клавиша РУС погаснет. При нажатии на клавиши с буквами на экране будут появляться латинские буквы. Чтобы вернуться к русскому регистру, нужно будет снова нажать CTRL/РУС. Заметим, что когда установлен латинский регистр, можно использовать клавишу CAPS для ввода русских букв.

После вывода имени алгоритма сообщение "Нет имени" исчезнет.

Заполняя строки ДАНО и НАДО, не забывайте перед текстом ставить знак "!" (восклицательный знак). Он показывает, что вся информация справа от него является комментарием, т.е. предназначена человеку, а не машине. На клавиатуре этот знак расположен на той же клавише, где и цифра 1. В нашем режиме символ "!" (и другие расположенные под цифрами символы) будет изображаться при нажатии этой клавиши одновременно с клавишей SHIFT; цифры вводятся без нажатия клавиши SHIFT.

Текст самой программы располагается между словами НАЧ и КОН. Чтобы "раздвинуть" эти слова, нужно установить курсор на строку со словом КОН и нажать SHIFT/INS. На экране будут появляться строки, в которых в первой позиции стоит символ "." (точка). Свой текст мы можем писать через пробел от этой точки (левее не получится – набранные символы не появляются на экране).

Может получиться, что при вводе слова мы допустим ошибку. Если для исправления нужно заменить одну букву на другую, то необходимо установить курсор в нужное место и набрать правильный символ (ошибочный стирать не нужно!). Если мы пропустили букву, нужно подвести курсор к месту пропущенной буквы и нажать клавишу INS. Правая часть слова отодвинется на одну позицию. Если ввели лишнюю букву, то нужно установить курсор на лишний символ и нажать клавишу DEL. Символ сотрется, а правая часть слова "подъедет" влево на одну позицию.

При нажатии клавиш с буквами буквы появляются там, где стоит курсор. Проверьте, как можно передвигать курсор по экрану. Слева он доходит до границы экрана, справа – до полей из символов "!". Однако ввести символы в две крайние левые позиции и крайнюю пра-



вую позицию мы не можем. Проверьте, к чему приводит попытка ввести символы в эти позиции.

Введите столько пустых строк, чтобы слова НАЧ и КОН не были видны на экране одновременно. Проверьте, что происходит при движении курсора вверх и вниз. Убедитесь, что движение курсора вверх-вниз вызывает передвижение по программе "окна" - помещающегося на экране участка программы.

При вводе строки вида "вперед(10)" нам понадобятся скобки. Символы "(" и ")" расположены на тех же клавишах, где цифры 8 и 9. Чтобы ввести скобки, эти клавиши нужно нажимать одновременно с клавишей SHIFT.

Запомним команды, необходимые при вводе, исправлении и выполнении программ:

- |   |   |           |
|---|---|-----------|
| 1. Вставка рамки для нового алгоритма   | - | ESC/A     |
| 2. Вставка строки над курсором  | - | SHIFT/INS |
| 3. Удаление строки<br>(когда курсор стоит на строке АЛГ, SHIFT/DEL приводит к удалению всего алгоритма) | - | SHIFT/DEL |
| 4. Вставка одной позиции  | - | INS       |
| 5. Удаление символа   | - | DEL       |
| 6. Длительный переход от русского регистра к латинскому и обратно                                       | - | CTRL/ПУС  |
| 7. Временный переход от русского регистра к латинскому и обратно  | - | CAPS      |
| 8. Выполнение программы целиком   | - | F9        |
| 9. Выполнение программы по шагам  | - | F4        |

#### ЗАДАНИЯ.

1. Что будет изображено в результате выполнения программы:

```
алг что
  дано
  надо
нач
  черепашка
  опустить_хвост
  вперед(50)
  налево(90)
  вперед(50)
  налево(135)
  вперед(50*1.4)
кон
```

2. Нарисовать прямоугольник.
3. Нарисовать правильный треугольник.
4. Нарисовать правильный шестиугольник.
5. Нарисовать кораблик.

### 1.2. Понятие переменной. Оператор присваивания

Вспомним программу рисования прямоугольника 200x40:

```

алг рисунок1
  дано !черепашка в левом верхнем углу
  надо !нарисован прямоугольник 200x40; 20 шагов от края
  нач
    черепашка
    вперед(20)
    налево(45)
    опустить_хвост
    вперед(200)
    направо(90)
    вперед(40)
    направо(90)
    вперед(200)
    направо(90)
    вперед(40)
    направо(90)
  кон

```

Для того, чтобы нарисовать прямоугольник другого размера, нам нужно сделать 4 исправления в разных местах программы. Постараемся построить программу иначе, сделав ее более удобной, а именно такой, чтобы при изменении размера прямоугольника изменений в программе было поменьше. Нам понадобятся 2 переменные – а и б, которые будут соответствовать размерам по горизонтали и вертикали. Программа с использованием этих переменных будет выглядеть так:

```

алг рисунок2
  дано !черепашка в левом верхнем углу
  надо !нарисован прямоугольник аxб; 20 шагов от края
  нач цел а,б
    а:=200; б:=40

```

```

черепашка
вперед (20)
налево (45)
опустить_хвост
вперед (a)
направо (90)
вперед (b)
направо (90)
вперед (a)
направо (90)
вперед (b)
направо (90)
кон

```

Теперь для изменения размера прямоугольника нам придется изменить одну строку в начале программы. (Напомним, что если в одной строке программы встречаются два предписания, их нужно разделять точкой с запятой).

В строке НАЧ содержится описание переменных: указание их имени и типа. Тип переменной – это некоторая характеристика, указывающая

- какие значения может принимать переменная;
- какие операции можно с ней выполнять.

Например, значение 3,1416 может принимать переменная вещественного типа и не может принимать переменная целого типа. С переменными целого и вещественного типа можно выполнять сложение, умножение, вычитание, деление; с переменными логического типа можно выполнять операции И, ИЛИ, НЕ. Свои специфические операции существуют для переменных литерного типа – "склеивание" строк и выделение подстроки.

В следующей строке программы РИСУНОК содержатся два оператора присваивания. Переменной, записанной слева от знака "=", присваивается значение константы, переменной или выражения, указанных справа. Выражение может содержать константы и переменные, соединенные знаками операций. Употребляются следующие знаки операций

- +           - для сложения;
- - для вычитания;
- \*           - для умножения;
- /           - для деления;
- ^           - для возведения в степень;
- sqrt( )   - для извлечения корня.

#### ЗАДАНИЯ.

1. Что будет изображено при выполнении программы:

```
алг угадай
  дано
  надо
  нач цел а
    а:=60
  черепашка
  вперед(10)
  налево(45)
  опустить_хвост
  вперед(а)
  направо(90)
  а:=а-10
  вперед(а)
  направо(90)
  а:=а-10
  вперед(а)
  направо(90)
  а:=а-10
  вперед(а)
  направо(90)
кон
```

2. Нарисовать квадрат заданного размера так, чтобы размер легко менялся.
3. В квадрате заданного размера провести диагонали.
4. Нарисовать правильный треугольник заданного размера так, чтобы размер легко менялся.

#### 1.3. Цикл ДЛЯ

Вспомним программу рисования квадрата:

```
алг квадрат1
  дано !черепашка в левом верхнем углу
  надо !нарисован квадрат аХа; 20 шагов от края
  нач цел а
    а:=40
  черепашка
  вперед(20)
```

```
налево (45)
опустить_хвост
вперед (a)
направо (90)
вперед (a)
направо (90)
вперед (a)
направо (90)
вперед (a)
направо (90)
кон
```

Здесь группа команд

```
вперед (a)
направо (90)
```

повторяется 4 раза. Можно не писать их все, а указать этот факт:

```
алг квадрат2
дано !черепашка в левом верхнем углу
надо !нарисован квадрат aXa; 20 шагов от края
нач цел a, k
a := 40
черепашка
вперед (20)
налево (45)
опустить_хвост
нц для k от 1 до 4
. вперед (a)
. направо (90)
кц
кон
```

Для вставки конструкции НЦ ДЛЯ – КЦ нужно нажать ESC/Д. Для удаления конструкции нужно поставить курсор на строку НЦ и нажать SHIFT/DEL.

**ЗАДАНИЯ.**

1. Что будет изображено при выполнении программы

```
алг догадайся
```

```

дано
надо
нач цел а, к
а := 30
черепашка
вперед(40)
налево(45)
опустить_хвост
нц для к от 1 до 4
. вперед(а)
. налево(90)
. назад(а)
. вперед(а)
. направо(90)
кц
кон

```

2. Напишите программу рисования правильного треугольника с помощью конструкции цикла.
3. Напишите программу рисования правильного n-угольника.
4. Что будет изображено при выполнении программы

```

алг что_и_сколько
дано
надо
нач цел а, к, л
а := 20
черепашка
вперед(20)
налево(45)
опустить_хвост
нц для к от 1 до 5
. нц для л от 1 до 4
. . вперед(а)
. . направо(90)
. кц
. поднять_хвост
. вперед(а*2)
. опустить_хвост
кц
кон

```

5. Напишите программу рисования ряда из 4 треугольников.

#### 1.4. Исполнитель РОБОТ

Исполнитель РОБОТ работает в прямоугольном лабиринте, разбитом на клетки, и понимает предписания

```
робот
шаг_на_север
шаг_на_юг
шаг_на_запад
шаг_на_восток
закрасить
```

Кроме того, у него есть средства контролировать свое положение в лабиринте. Он понимает вопросы

```
на_севере_свободно
на_юге_свободно
на_западе_свободно
на_востоке_свободно
```

и вырабатывает ответы ДА или НЕТ в зависимости от ситуации. Так, в левом верхнем углу на вопросы НА\_СЕВЕРЕ\_СВОБОДНО и НА\_ЗАПАДЕ\_СВОБОДНО будут выработаны ответы НЕТ, а на два других вопроса – ответы ДА.

Кроме того, робот умеет измерять температуру и радиацию в текущей клетке. Достигается это тем, что по предписаниям

```
температура
радиация
```

вырабатывается числовое значение, равное значению соответствующей величины в текущей клетке. При этом радиация принимает положительные вещественные значения от 0 до 7.0, температура – целые значения от -79 до +79. Пример использования:

```
r1:=радиация
шаг_на_восток
r2:=радиация
```

После этого переменные r1 и r2 получат значения, равные значению

радиации в соседних точках.

Задача. Вычислить среднюю температуру в трех верхних левых клетках лабиринта.

Возможное решение:

```
алг среднее
  дано !робот в левом верхнем углу
  надо !посчитано среднее значение температуры
нач вещь сум, ср_темп
  робот
  сум:=0
  сум:=сум+температура
  шаг_на_запад
  сум:=сум+температура
  шаг_на_запад
  сум:=сум+температура
  ср_темп:=сум/3
кон
```

У робота имеется режим редактирования лабиринта со следующими возможностями:

1. ESC/л – переход в режим редактирования лабиринта.
2. HOME – изменение состояния закрашенности текущей клетки.
3. CTRL/стрелка – установка стенки с указанной стороны от текущей клетки.
4. Число без знака – установление нового значения радиации в текущей клетке.
5. Число со знаком – установление нового значения температуры в текущей клетке.
6. INS – установка нового начального положения робота.
7. RETURN – выход из режима редактирования.

ЗАДАНИЯ.

1. Что будет изображено при работе программы:

```
алг вопрос
  дано
  надо
нач цел к
  робот
  нц для к от 1 до 3
```



```

. шаг_на_восток
. шаг_на_восток
. закрасить
кц
кон

```

2. Закрасить квадрат 2X2 в левом верхнем углу.
3. Закрасить 5 клеток, лежащих на биссектрисе левого верхнего угла.
4. В режиме редактирования лабиринта создать из стен прямоугольное препятствие 4X2. Поставить робота у его левого верхнего угла. Обвести вокруг препятствия и закрасить след.

#### 1.5. Цикл ПОКА

Перед началом работы робота мы не знаем размеры лабиринта. Чтобы довести его до восточной стены, можно пользоваться следующей конструкцией:

```

нц пока на_востоке_свободно
. шаг_на_восток
кц

```

Выполнение такой конструкции прекращается, когда условие, указанное после "пока", оказывается нарушенным. То есть, действуя по такой программе, робот остановится около восточной стены.

Задача. Закрасить биссектрису левого верхнего угла.  
Возможное решение:

```

алг закрасить_биссектрису
  дано
  надо !закрашена биссектриса левого верхнего угла
нач
  робот
  нц пока на_юге_свободно и на_востоке_свободно
  . шаг_на_восток
  . шаг_на_юг
  . закрасить
  кц
кон

```

Конструкцию **НЦ ПОКА – КЦ** вставляют в текст программы при помощи клавиш **ESC/П**. Чтобы уничтожить всю конструкцию, нужно установить курсор на строку **НЦ** и нажать **SHIFT/DEL**.

Мы видим, что условие может быть и сложным, состоящим из нескольких простых. При этом простые могут быть связаны словами **И**, **ИЛИ**, **НЕ**. В следующих табличках указано, в каких случаях является истинным сложное условие, если известна истинность или ложность входящих в них простых.

|            |            |            |
|------------|------------|------------|
| <b>И</b>   | <b>да</b>  | <b>нет</b> |
| <b>да</b>  | <b>да</b>  | <b>нет</b> |
| <b>нет</b> | <b>нет</b> | <b>нет</b> |

|            |           |            |
|------------|-----------|------------|
| <b>ИЛИ</b> | <b>да</b> | <b>нет</b> |
| <b>да</b>  | <b>да</b> | <b>да</b>  |
| <b>нет</b> | <b>да</b> | <b>нет</b> |

|           |            |            |
|-----------|------------|------------|
|           | <b>да</b>  | <b>нет</b> |
| <b>НЕ</b> | <b>нет</b> | <b>да</b>  |

#### ЗАДАНИЯ.

1. Что будет изображено при работе программы:

```

алг узор
  дано
  надо
  нач
    робот
    нц пока на_востоке_свободно
      . шаг_на_восток
      . шаг_на_юг
      . закрасить
      . шаг_на_восток
      . шаг_на_север
      . закрасить
    кц
  кон
  
```

2. Пройти левый ряд и закрасить его.
3. Пройти лабиринт по периметру и закрасить его.
4. Закрасить биссектрису правого верхнего угла.
5. Вычислить среднее значение радиации в верхнем ряду.

## 1.6. Ветвление

Задача. Пройти по верхнему ряду и закрасить все клетки, в которых температура выше  $10^{\circ}\text{C}$ . Для этого нам понадобится возможность проверки

температура  $> 10$

и возможность закрашивать либо не закрашивать клетку в зависимости от результата такой проверки.

Возможное решение

```
алг горячие_клетки
  дано !робот в левом верхнем углу лабиринта
  надо !ряд пройден; закрашены клетки, где температура  $> 10$ 
нач
  робот
  если температура  $> 10$ 
    . то закрасить
  все
  нц пока на_востоке_свободно
    . шаг_на_восток
    . если температура  $> 10$ 
      . . то закрасить
    . все
  кц
кон
```

Конструкция ЕСЛИ-ТО-ИНАЧЕ-ВСЕ вставляется путем нажатия ESC/E. При необходимости строку ИНАЧЕ можно удалить. Для удаления всей конструкции нужно установить курсор на строку ЕСЛИ и нажать SHIFT/DEL.

Условие, стоящее после ЕСЛИ, может быть и сложным. Правила определения истинности сложного условия на основе двух простых приведены в табличках прошлого раздела. Поработаем со сложными условиями.

Пусть робот работает в лабиринте, где температура и радиация заданы следующим образом:

|     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0   | 1   | 2   | 3   | 4   | 4   | 4   | 4   | 4   | 4   | 4   |
| +35 | +35 | +35 | +35 | +35 | +35 | +35 | +30 | +25 | +20 | +15 |
| 0   | 1   | 2   | 3   | 4   | 5   | 5   | 5   | 5   | 5   | 4   |
| +35 | +40 | +40 | +40 | +40 | +40 | +35 | +30 | +25 | +20 | +15 |
| 0   | 1   | 2   | 3   | 4   | 5   | 6   | 6   | 6   | 5   | 4   |
| +35 | +40 | +45 | +45 | +45 | +40 | +35 | +30 | +25 | +20 | +15 |
| 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 6   | 5   | 4   |
| +35 | +40 | +45 | +50 | +45 | +40 | +35 | +30 | +25 | +20 | +15 |
| 0   | 1   | 2   | 3   | 4   | 5   | 6   | 6   | 6   | 5   | 4   |
| +35 | +40 | +45 | +45 | +45 | +40 | +35 | +30 | +25 | +20 | +15 |
| 0   | 1   | 2   | 3   | 4   | 5   | 5   | 5   | 5   | 5   | 4   |
| +35 | +40 | +40 | +40 | +40 | +40 | +35 | +30 | +25 | +20 | +15 |
| 0   | 1   | 2   | 3   | 4   | 4   | 4   | 4   | 4   | 4   | 4   |
| +35 | +35 | +35 | +35 | +35 | +35 | +35 | +30 | +25 | +20 | +15 |

Какая часть лабиринта будет закрашена при выполнении следующей программы:

```

алг область
  дано
  надо
  нач цел к, л
  робот
  нц для к от 1 до 7
  . нц для л от 1 до 11
  . . если температура >= 40 !это условие будем менять
  . . .то закрасить
  . . все
  . . если на_востоке_свободно
  . . .то шаг_на_восток
  . . все
  . кц
  . нц пока на_западе_свободно
  . . шаг_на_запад

```

```

. кц
. если на_юге_свободно
. .то шаг_на_юг
. все
кц
кон

```

А теперь определите, какая часть лабиринта будет закрашена, если отмеченное условие поочередно заменять следующими:

- а) радиация  $\geq 5$
- б) радиация  $\geq 5$  и температура  $\geq 40$
- в) радиация  $\geq 5$  или температура  $\geq 40$
- г) радиация  $\geq 5$  и не температура  $\geq 40$
- д) не радиация  $\geq 5$  и температура  $\geq 40$

Вспомним задачу о вычислении средней температуры в трех верхних левых клетках лабиринта:

```

алг среднее
дано !робот в левом верхнем углу
надо !посчитано среднее значение температуры
нач вещь сум, ср_темп
робот
сум:=0
сум:=сум+температура
шаг_на_восток
сум:=сум+температура
шаг_на_восток
сум:=сум+температура
ср_темп:=сум/3
кон

```

Рассмотрим теперь другую задачу: определить максимальную температуру в трех верхних левых клетках.

Как нужно поступить в этом случае? Сначала запомнить значение температуры в самом левом углу. Затем шагнуть в соседнюю клетку и сравнить текущее значение температуры с запомненным. Если во второй клетке значение температуры выше запомненного, то запомнить именно его. Еще раз шагнуть. Если в третьей клетке значение температуры выше запомненного, то запомнить именно его. Запоминать будет в переменной ТМАКС. Программа будет выглядеть так:

```

алг максимум_температуры
  дано !робот в левом верхнем углу
  надо !запомнено максимальное значение температуры
  нач вещь тмакс
    робот
    тмакс:=температура
    шаг_на_восток
    если температура>тмакс
      . то тмакс:=температура
    все
    шаг_на_восток
    .если температура>тмакс
      . то тмакс:=температура
    все
  кон

```

#### ЗАДАНИЯ.

1. Что будет изображено при работе программы

```

алг нечто
  дано
  надо
  нач
    робот
    нц пока на_востоке_свободно
      . шаг_на_восток
      . если на_юге_свободно
        . . то шаг_на_юг
      . все
      . закрасить
    кц
  кон

```

2. Вычислить максимальную температуру в верхнем ряду.
3. Закрасить клетку верхнего ряда, где радиация максимальна.

#### 1.7. Исполнитель ГРАФИКА

Исполнитель ГРАФИКА выполняет следующие предписания:

```

графика
  поз(x,y)

```

точка (x, y)  
линия (x, y)  
контур (x, y)  
блок (x, y)  
круг (p)  
надпись (' текст')

Рисунки выполняются в прямоугольнике 256x86; начало координат – в левом верхнем углу; X возрастает вправо, Y – вниз. Координаты меняются от 0 до 255 по X и от 0 до 85 по Y. При рисовании используется понятие ТЕКУЩЕЙ ТОЧКИ.

Предписание ПОЗ переносит делает текущей точку с указанными координатами.

Предписание ТОЧКА ставит точку в указанном месте.

Предписание ЛИНИЯ рисует линию от текущей точки до указанной. Указанная точка становится текущей.

Предписание КОНТУР рисует прямоугольник со сторонами, параллельными осям, один угол которого находится в текущей точке, второй – в указанной. Указанная точка становится текущей.

Предписание БЛОК работает аналогично предписанию КОНТУР; прямоугольник при этом получается закрашенным. Указанная точка становится текущей.

Предписание КРУГ рисует круг указанного радиуса с центром в текущей точке.

Предписание НАДПИСЬ выводит указанный в скобках текст, начиная от текущей точки. Текущей становится точка – начало следующей буквы.

Задача. Нарисовать треугольник и подписать его.

Одно из возможных решений:

```
алг треугольник
  дано
  надо !нарисовать треугольник
нач
  графика
  поз(120, 10)
  линия(240, 70)
  линия(0, 70)
  линия(120, 10)
  поз(10, 74)
  надпись ('треугольник')
```

кон

#### ЗАДАНИЯ.

1. Что будет изображено при работе программы

```
алг объект
  дано
  надо
нач
  графика
  поз(0,0)
  блок(25,25)
  блок(50,50)
  блок(75,75)
кон
```

2. Нарисовать детскую пирамидку.
3. Провести 5 горизонтальных отрезков одинаковой длины.
4. Из прямоугольников и окружностей нарисуйте грузовой автомобиль.
5. Изобразите движение точки по экрану слева направо; сверху вниз.

#### 1.В. Литерные переменные

Познакомимся с переменными литерного типа. Пример описания литерных переменных –

лит а, б, ц

Пример операторов присваивания для литерных переменных –

```
а:= 'компьютер'
б:= 'активизация'
```

Существуют три операции для работы с литерными переменными. Первая – операция "вырезки", когда из строки выделяется подстрока с указанными позициями. Так, в результате выполнения операции

```
ц:=а[1:3]
```

переменная ц получит значение 'ком'. Эта операция допускает и обратное использование: в результате выполнения

```
а[5:6]:='ос'
```

переменная а получит значение 'компостер'. Если мы хотим вырезать из слова только одну букву, то в скобках должны указать номер этой буквы два раза. Так, в результате выполнения операции

```
в:=а[3:3]
```



переменная *v* получит значение 'м'.

Вторая – операция "склеивания" строк (в литературе вы можете встретить термин "конкатенация"). В результате выполнения операции

```
ц:=a+b[6:11]
```

переменная *ц* получит значение 'компьютеризация'.

И, наконец, операция

```
длин(a)
```

для литерной переменной *a* выдает длину строки. Так, `длин('ЗВМ')` равно 3, `длин('Э В М')` равно 5.

В некоторых задачах удобно пользоваться пустой строкой. Это строка, в которой нет ни одного символа. Она записывается так:

```
a:''
```

Длина такой строки равна 0.

Задача. Подсчитать количество букв 'о' в указанном слове.

Возможное решение:

```
алг количество_о
  дано !слово
  надо !подсчитать количество букв 'о'
нач лит а, б, цел к
  а:='головоломка'
  к:=0 !это счетчик букв 'о'
  нц для к от 1 до длин(а)
  . б:=a[к:к] !это очередная буква слова
  . если б='о'
  . . то к:=к+1
  . все
кц
кон
```

ЗАДАНИЯ.

1. Что будет результатом программы

```
алг что_это
  дано
  надо
нач лит а, б, в, г
  а:='человекообразные'
  б:=a[1:7]
  в:=a[3:4]+a[7:8]+a[14:14]
  г:=a[13:13]+a[16:16]+a[10:12]
кон
```

2. Получите слова 'корфца', 'ситро', 'скала' из слова 'первоклассница'.

3. Что будет результатом программы

```
алг кое_что
  дано
  надо
  нач лит а,б,в,г
  а:='дискковод'
  б:=''
  нц для к от 1 до длин(а)
  . в:=а[к:к]+б
  кц
кон
```

4. Напишите программу, которая из данного слова получает другое, в котором вычеркнуты все буквы 'и', а остальные сохранены. Проверьте работу программы на слове 'пилигрим'.

5. Напишите программу, которая из данного слова получает другое, в котором все буквы 'о' заменены на буквы 'а'. Проверьте работу программы на слове 'водоворот'.

## 2. Дополнительные возможности

### 2.1. Массивы

Задача. Изобразить на плоскости 6 точек – координаты вершин правильного шестиугольника.

Воспользуемся тем, что вершины правильного шестиугольника располагаются на окружности. Радиусы, идущие к ним, отстоят друг от друга на  $60^\circ$ . Возможное решение:

```
алг шестиугольник1
  дано
  надо !отметить вершины правильного шестиугольника
  нач вещ x1,x2,x3,x4,x5,x6,y1,y2,y3,y4,y5,y6,r,x0,y0,пи
  x0:=100;y0:=40;r:=30 !центр и радиус
  пи:=3.1416
  x1:=x0+r*cos(пи/3);y1:=y0+r*sin(пи/3)
  x2:=x0+r*cos(2*пи/3);y2:=y0+r*sin(2*пи/3)
  x3:=x0+r*cos(3*пи/3);y3:=y0+r*sin(3*пи/3)
  x4:=x0+r*cos(4*пи/3);y4:=y0+r*sin(4*пи/3)
  x5:=x0+r*cos(5*пи/3);y5:=y0+r*sin(5*пи/3)
  x6:=x0+r*cos(6*пи/3);y6:=y0+r*sin(6*пи/3)
```

```

графика
точка(x1,y1)
точка(x2,y2)
точка(x3,y3)
точка(x4,y4)
точка(x5,y5)
точка(x6,y6)
кон

```

Мы видим, что здесь 6 раз повторяются очень похожие действия. Но непонятно, как воспользоваться циклом, поскольку надо вычислить и запомнить 6 разных пар переменных.

Для таких случаев, когда у нас имеется группа похожих переменных и к ним необходим доступ по номеру, используется специальный способ организации данных – массивы, или, другими словами, таблицы. Можно объявить, что у нас есть две таблицы – x и y, в каждой по шесть элементов. Затем в цикле вычислить каждый элемент таблицы x и таблицы y. Затем в цикле 6 раз поставить точку, используя каждый раз очередной элемент таблиц x и y.

```

алг шестиугольник2
дано
надо !отметить вершины правильного шестиугольника
нач вещтаб x[1:6],y[1:6], цел p,x0,y0,k,l, вещ пи
x0:=100;y0:=40;p:=30 !центр и радиус
пи:=3.1416
нц для k от 1 до 6
. x[k]:=x0+p*cos(k*пи/3);y[k]:=y0+p*sin(k*пи/3)
кц
графика
нц для k от 1 до 6
. точка(x[k],y[k])
кц
кон

```

Задача. Нарисовать правильный шестиугольник.

```

алг шестиугольник
дано
надо !нарисовать правильный шестиугольник
нач вещтаб x[1:6],y[1:6], цел p,x0,y0,k, вещ пи
x0:=100;y0:=40;p:=30 !центр и радиус

```

```

пи:=3.14
нц для к от 1 до 6
. x[k]:=x0+r*cos(k*пи/3);y[k]:=y0+r*sin(k*пи/3)
кц
графика
поз(x[6],y[6])
нц для к от 1 до 6
. линия(x[k],y[k])
кц
кон

```

Задача. Нарисовать правильный шестиугольник и провести в нем диагонали.

Одно из возможных решений:

```

алг диагонали
дано
надо !нарисовать правильный шестиугольник с диагоналями
нач вещтав x[1:6],y[1:6], цел p,x0,y0,k,l, вещ пи
x0:=100;y0:=40;p:=30 !центр и радиус
пи:=3.14
нц для к от 1 до 6
. x[k]:=x0+r*cos(k*пи/3);y[k]:=y0+r*sin(k*пи/3)
кц
графика
нц для к от 1 до 6
. нц для л от 1 до 6
. . поз(x[k],y[k])
. . линия(x[l],y[l])
. кц
кц
кон

```

Массивы часто используются также в вычислительных задачах. Приведем несколько таких задач.

Задача. В массиве  $a[1:10]$  получить 10 чисел, кратных трем.

```

алг числа
дано
надо !получить 10 чисел, кратных трем
нач целтав a[1:10], цел k

```

```

нц для к от 1 до 10
. a[k]:=3*k
кц
кон

```

Задача. Вычислить сумму элементов массива. Использовать массив, полученный при решении предыдущей задачи.

```

алг сумма
дано
надо !получить сумму элементов массива
нач целтаб a[1:10], цел к, с
нц для к от 1 до 10
. a[k]:=3*k
кц
с:=0
нц для к от 1 до 10
. с:=с+a[k]
кц
кон

```

**ЗАДАНИЯ.**

1. Что будет изображено в результате работы программы

```

алг отрезки
дано
надо !нарисовать отрезки
нач вещтаб x[1:5], цел к
нц для к от 1 до 5
. x[k]:=15*k
кц
графика
нц для к от 1 до 5
. поз(x[k],0)
. линия(x[k],x[k])
кц
кон

```

2. В массиве T[1:7] находятся значения температуры за 7 дней недели. Вычислить среднюю температуру.
3. В массиве T[1:7] находятся значения температуры за 7 дней недели. Определить максимальную температуру.

4. В массиве  $D[1:5]$  находятся значения длины 5 прямоугольников, в массиве  $Ш[1:5]$  – значения ширины. В массиве  $П[1:5]$  получить значения площади.
5. Две бригады школьников работали на школьном заводе и получили зарплату. Зарплата участников первой бригады хранится в массиве  $B1[1:5]$ , второй – в массиве  $B2[1:7]$ . Определите, в какой бригаде средний заработок выше.

## 2.2. Вспомогательные алгоритмы

Программа

```

алг ряд1
  дано: !черепашка в левом верхнем углу
  надо: !нарисовано 5 квадратов  $a \times a$ ; 20 шагов от края
  нач цел  $a, k, l$ 
   $a := 20$ 
  черепашка
  вперед(20)
  налево(45)
  опустить_хвост
  нц для  $k$  от 1 до 5
  . нц для  $k$  от 1 до 4
  . . вперед( $a$ )
  . . направо(90)
  . кц
  . поднять_хвост
  . вперед( $2 \times a$ )
  . опустить_хвост
  кц
кон

```

рисует 5 квадратов  $20 \times 20$  в одном ряду. Такая программа выглядит нагляднее, если рисование одного квадрата выделить в самостоятельную часть программы:

```

алг ряд2
  дано !черепашка в левом верхнем углу
  надо !нарисовано 5 квадратов  $a \times a$ ; 20 шагов от края
  нач цел  $k$ 
  черепашка
  вперед(20)

```

```

налево(45)
опустить_хвост
нц для к от 1 до 5
. квадрат
кц
кон
алг квадрат
дано !черепашка в л.в. углу квадрата лицом на восток
надо !нарисован квадрат аЖа; встали в л.в. угол след.
нач цел а,к
а:=20
нц для к от 1 до 4
. вперед(а)
. направо(90)
кц
поднять_хвост
вперед(2Жа)
опустить_хвост
кон

```

Часть программы, выполняющая самостоятельную функцию и оформленная специальным образом, так что ее можно использовать как единое целое, называется **ВСПОМОГАТЕЛЬНЫМ АЛГОРИТМОМ**. В данном случае таким вспомогательным алгоритмом является алгоритм **КВАДРАТ**.

Наш алгоритм записан так, что всегда в результате его выполнения будет нарисован квадрат определенного размера, а именно  $20 \times 20$ . Можно оформить алгоритм иначе, обеспечив возможность рисовать квадраты разного размера:

```

алг ряд3
дано !черепашка в левом верхнем углу
надо !нарисовано 5 квадратов аЖа; 20 шагов от края
нач цел к
черепашка
вперед(20)
налево(45)
опустить_хвост
нц для к от 1 до 5
. квадрат(10Жк)
кц
кон

```

```

алг квадрат(цел к)
  дано !черепашка в л.в. углу квадрата лицом на восток
  надо !нарисован квадрат ажа; встали в л.в. угол след.
нач цел к
  нц для к от 1 до 4
  . вперед(а)
  . направо(90)
  кц
  поднять_хвост
  вперед(2Жа)
  опустить_хвост
кон

```

При исполнении такой программы будут нарисованы 5 квадратов разного размера со сторонами 10, 20, 30, 40 и 50 единиц.

Можно передавать не один, а несколько параметров. Так, при рисовании правильного многоугольника естественно иметь 2 параметра – длину стороны и количество вершин. При вычислении угла, на который нужно поворачиваться при рисовании многоугольника, следует вспомнить, что сумма всех внутренних углов многоугольника равна  $180 \times (n-2)$ , количество таких углов  $n$ , значит, внутренний угол равен  $180 \times (n-2) / n$ , а черепашка должна поворачиваться на угол, смежный с внутренним углом многоугольника.

```

алг ряд4
  дано !черепашка в левом верхнем углу
  надо !нарисовано 5 многоугольников
нач цел к
  черепашка
  вперед(20)
  налево(45)
  опустить_хвост
  нц для к от 1 до 5
  . многоугольник(15,6)
  кц
кон
алг многоугольник(цел а,н)
  дано !черепашка в л.в. углу n-угольника
  надо !нарисован n-угольник; встали в л.в. угол след.
нач цел к,у
  у:=180*(н-2)/н
  нц для к от 1 до н

```



```

. вперед (a)
. направо (180-y)
кц
поднять_хвост
вперед (2Жa)
опустить_хвост
кон

```

**ЗАДАНИЯ.**

1. Что будет изображено при выполнении программы

```

алг целое
дано
надо
нач цел к
  черепашка
  вперед (60)
  налево (135)
  нц для к от 1 до 4
  . часть (20)
  кц
кон
алг часть (цел a)
дано
надо
нач цел к
  опустить_хвост
  вперед (a)
  налево (45)
  нц для к от 1 до 3
  . вперед (a)
  . назад (a)
  . направо (45)
  кц
  налево (90)
  назад (a)
  направо (90)
  поднять_хвост
  вперед (3Жa)
  налево (90)
кон

```

2. Нарисуйте 4 ряда по 5 квадратов в каждом ряду. Используйте вспомогательный алгоритм РЯД.
3. Нарисуйте цветок, имеющий несколько одинаковых лепестков. Алгоритм ЛЕПЕСТОК сделайте вспомогательным.
4. Нарисуйте елочку, используя вспомогательный алгоритм ВЕТКИ (либо два вспомогательных алгоритма – ЛЕВАЯ\_ВЕТКА и ПРАВАЯ\_ВЕТКА).
5. Используя вспомогательный алгоритм МНОГОУГОЛЬНИК с двумя параметрами – количеством вершин и длиной стороны, – нарисуйте рядом правильный треугольник, квадрат и правильный пятиугольник при помощи одного цикла.

### 2.3. Вспомогательные алгоритмы типа функция

Все вспомогательные алгоритмы, которые мы рассматривали до сих пор, сводились к определенной последовательности действий исполнителя. Вызов был равносителен выполнению группы предписаний.

Существует другой способ использования вспомогательных алгоритмов, а именно такой, когда в результате вызова вспомогательного алгоритма вырабатывается некоторое числовое значение, а в вызывающем алгоритме удобно работать с именем алгоритма, как с числом.

**Задача.** В правильном шестиугольнике найти диагональ максимальной длины.

Здесь нам понадобится алгоритм ДЛИНА, который по четырем числам – координатам концов – будет вычислять длину отрезка.

```

алг макс_диагональ
  дано
  надо !найти макс. диагональ правильного шестиугольника
  нач вещтаб x[1:6], y[1:6], цел p, x0, y0, k, l, вещ pi, m
  графика
  x0:=100; y0:=40; p:=30 !центр и радиус
  pi:=3.14
  нц для k от 1 до 6
  . x[k]:=x0+p*cos(k*pi/3); y[k]:=y0+p*sin(k*pi/3)
  кц
  m:=0
  нц для k от 1 до 6
  . нц для l от 1 до 6
  . . поз(x[k], y[k])

```

```

. . линия(x[l],y[l])
. . если m < длина(x[k],y[k],x[l],y[l])
. . . то m:=длина(x[k],y[k],x[l],y[l])
. . все
. кц
кц
кон
алг вещ длина(цел x1,y1,x2,y2)
дано !координаты двух точек
надо !определить длину отрезка
нач
. знач:=sqrt((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2))
кон

```

Задача. В лабиринте для РОБОТа имеется некоторое количество горизонтальных стенок, не примыкающих к внешним стенам. Требуется пройти лабиринт и подсчитать общую длину внутренних стенок.

Возможное решение:

```

алг общая_длина_стен
дано !робот в левом верхнем углу
надо !робот прошел лабиринт; длина стенок подсчитана
нач цел длина
робот
длина:=0
нц пока на_юге_свободно
. длина:=длина+длина_в_ряду
. шаг_на_юг
кц
кон
алг цел длина_в_ряду
дано ! робот в начале ряда
надо !робот в начале след.ряда, длина стен в ряду подсчитана
нач цел к
к:=0
нц пока на_востоке_свободно
. шаг_на_восток
. если не на_юге_свободно
. . то к:=к+1
. все
кц
нц пока на_западе_свободно

```

```

. шаг_на_запад
кц
знач:=к
кон

```

Разберем задачу, когда в качестве параметра передаются литерные переменные.

Задача. Определить, каких букв в данном слове больше – о или е. Использовать вспомогательный алгоритм КОЛИЧЕСТВО\_БУКВ с двумя параметрами – слово, буква.

```

алг каких_букв_больше
  дано
  надо !определить, каких букв в слове больше – о или е
  нач лит а, у, цел к1, к2
  а:='головокружение'
  к1:=количество_букв(а, 'о')
  к2:=количество_букв(а, 'е')
  если к1>к2
  .то у:='о больше'
  .иначе у:='е больше'
  все
кон
алг цел количество_букв(лит слово, буква)
  дано !слово и буква
  надо !определить, сколько указанных букв в этом слове
  нач цел к, лит б, в
  знач:=0
  нц для к от 1 до длин(слово)
  . если слово[к:к]=буква
  . .то знач:=знач+1
  . все
  кц
кон

```

**ЗАДАНИЯ.**

1. Что будет изображено при выполнении программы:

```

алг кое_что
  дано
  надо
  нач цел к, л

```

```

робот
л:=мин(В,5)
нц для к от 1 до л
. шаг_на_восток
. закрасить
кц
кон
алг цел мин(цел а,б)
дано
надо
нач цел к
если а<б
. то к:=а
. иначе к:=б
все
знач:=к
кон

```

2. Обойти прямоугольное препятствие. Вычислить периметр и площадь препятствия, используя вспомогательные алгоритмы ДЛИНА и ШИРИНА.
3. Найти максимальную радиацию в лабиринте, используя вспомогательный алгоритм МАКСРАДИАЦИЯ\_В\_РЯДУ.
4. Определить количество гласных букв в слове, используя вспомогательный алгоритм КОЛИЧЕСТВО\_БУКВ.

#### 2.4. Конструкция выбора

Если нам нужно организовать ветвление не на две возможности, а на три или больше, то применяют конструкцию выбора.

Задача. Радиация в лабиринте меняется от 0 до 7. При этом клетки с радиацией меньше 2 не вызывают тревоги; от 2 до 5 – вернутся на учет; свыше 5 – требуют немедленной обработки. Требуется пройти по лабиринту и подсчитать количество клеток каждого типа; клетки третьего типа закрасить.

Возможное решение:

```

алг клетки_трех_типов
дано !робот в левом верхнем углу лабиринта
надо !количество клеток разного типа подсчитано

```

```

нач цел к1, к2, к3
  робот
  к1:=0; к2:=0; к3:=0
  ряд(к1, к2, к3)
  нц пока на_юге_свободно
  . шаг_на_юг
  . ряд(к1, к2, к3)
  кц
  к1:=к1
  к2:=к2
  к3:=к3
кон
алг ряд(цел к1, к2, к3)
  дано !робот в левой клетке ряда
  надо !ряд пройден; счетчики увеличены; робот слева
нач
  клетка(к1, к2, к3)
  нц пока на_востоке_свободно
  . шаг_на_восток
  . клетка(к1, к2, к3)
  кц
  нц пока на_западе_свободно
  . шаг_на_запад
  кц
кон
алг клетка(цел к1, к2, к3)
  дано !робот в текущей клетке
  надо !один из счетчиков увеличен
нач
  выбор
  . при радиация<=1
  . . к1:=к1+1
  . ---
  . при радиация>1 и радиация<6
  . . к2:=к2+1
  . ---
  . при радиация>=6
  . . к3:=к3+1
  . . закрасить
  . ---
  все
кон

```

Конструкция выбора вставляется по нажатию ESC/B (B – от "выбор"); дополнительные условия вставляются по нажатию ESC/C (C – от "случай"). Для удаления одной ветки нужно установить курсор на соответствующую строку ПРИ и нажать SHIFT/DEL; для удаления всей конструкции – на строку ВЫБОР и нажать SHIFT/DEL.

Конструкция выбора не допускает возможности ИНАЧЕ, очень удобной в некоторых случаях. Для реализации такой возможности в качестве последнего условия используют всегда верное условие, например,  $a=a$ .

#### ЗАДАНИЯ.

1. Что будет результатом работы следующей программы:

```
алг куку
  дано
  надо
  нач лит а, в, цел к
  а := 'тригонометрия'
  в := ''
  нц для к от 1 до длин(а)
    выбор
      . при а[к:к] = 'о'
      . . в := в + 'а'
      . ____
      . при а[к:к] = 'р'
      . . в := в + а[к:к] + а[к:к]
      . ____
      . при а[к:к] = 'и'
      . . в := а[к:к] + в
      . ____
      . при а = а
      . . в := в + а[к:к]
      . ____
    все
  кц
кон
```

2. В кинозале 30 рядов по 20 мест. Цена одно места в рядах с 1 по 5 – 30 коп.; с 6 по 15 – 40 коп.; с 16 по 25 – 50 коп.; с 26 по 30 – 40 коп. Напишите программу подсчета выручки на сеанс, применяя конструкцию выбора.

3. В таблице P[1:10] содержатся сведения о выполнении планов предприятий в процентах (например, P[1]=98, P[2]=102 и т.д.). Подсчитайте, сколько предприятий
- не выполнило;
  - выполнило;
  - перевыполнило план.

#### 2.5. Исполнитель ЧЕРТЕЖНИК

Исполнитель ЧЕРТЕЖНИК понимает следующие предписания:

```
чертежник(0),
чертежник(1)
поднять_перо
опустить_перо
сместиться_в_точку(x,y)
сместиться_на_вектор(dx,dy)
```

Он работает в прямоугольнике 12x8; параметр 0 или 1 определяет, находится ли начало координат в левом нижнем углу этого прямоугольника или в середине его. В начале работы перо находится в точке (0,0) и поднято. В случае чертежник(0) координаты изменяются от (0,0) до (12,8); в случае чертежник(1) – от (-6,-4) до (6,4).

Задача. Нарисовать квадрат 2x2 с центром в начале координат.

Возможное решение:

```
алг квадрат
  дано
  надо !нарисован квадрат 2x2
нач
  чертежник(1)
  сместиться_в_точку(1,1)
  опустить_перо
  сместиться_на_вектор(0,-2)
  сместиться_на_вектор(-2,0)
  сместиться_на_вектор(0,2)
  сместиться_на_вектор(2,0)
кон
```



#### ЗАДАНИЯ.

1. Что будет изображено при выполнении программы:

```
алг контур
  дано
  надо
  нач
    чертежник(0)
    сместиться_в_точку(0,0)
    опустить_перо
    сместиться_на_вектор(0,5)
    сместиться_на_вектор(5,0)
    сместиться_на_вектор(-1,-1)
    сместиться_на_вектор(1,-1)
    сместиться_на_вектор(-5,0)
  кон
```

2. Нарисовать 3 "концентрических" квадрата с центром в начале координат. Использовать вспомогательный алгоритм КВАДРАТ с одним параметром – стороной квадрата.
3. Прямоугольник 10x5 развить на клетки 1x1.
4. Нарисовать шахматную доску. Черные клетки изобразить с помощью горизонтальных линий с очень мелким шагом.

#### 2.6. Исполнитель ПУТНИК

Исполнитель ПУТНИК работает в прямоугольном лабиринте, разбитом на клетки, и понимает предписания

```
путник
вперед      - движение
до_упора
направо     - повороты
налево
кругом
на_восток
на_запад
на_север
на_юг
```

Кроме того, у него есть средства контролировать свое положение в

лабиринте. Он понимает вопросы

```
вперед_свободно
справа_свободно
слева_свободно
```

и выработывает ответы ДА или НЕТ в зависимости от ситуации.

Дополнительные предписания:

```
нарисуй(' ') – рисует указанный символ в текущей клетке
тек_символ – возвращает значение символа, записанного
              в текущей клетке
```

У ПУТНИКА имеется режим редактирования лабиринта со следующими возможностями:

1. ESC/Л – переход в режим редактирования лабиринта.
2. BS – поворот ПУТНИКА против часовой стрелки.
3. INS/DEL – установка/стирание стенки рядом с текущей клеткой, в направлении, указываемом курсором.
4. RETURN – выйти из режима редактирования.

ЗАДАНИЯ.

1. Что будет результатом выполнения программы в лабиринте без внутренних стен:

```
алг маршрут
  дано
  надо
нач
  путник
  нарисуй('X')
  вперед
  нц пока тек_символ<>'X'
  . до_упора
  . направо
кц
кон
```

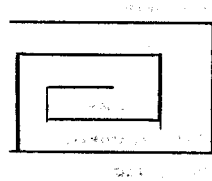
2. Что произойдет, если в программе "маршрут" убрать предписание "вперед"?
3. Провести путника по всем клеткам лабиринта без внутренних

стен.

4. Устроить из стен прямоугольное препятствие. Обвести путника вокруг препятствия.

### 3. Задачи

1. С помощью исполнителя ЧЕРЕПАШКА нарисуйте снежинку. Помните, что в природе снежинки состоят не из четырех или восьми, а из шести симметричных частей.
2. Разлините с помощью ЧЕРЕПАШКИ тетрадь в клеточку; тетрадь для первого класса.
3. Напишите программу рисования "прямоугольной" спирали, начиная ее с центра поля:



4. Напишите программу рисования морской звезды. Она состоит из большого числа лучей одинаковой длины, выходящих из одной точки.
5. Нарисуйте забор, состоящий из досок с заостренными концами.
6. С помощью исполнителя ГРАФИКА нарисуйте шахматную доску.
7. Изобразите движение двух точек навстречу друг другу по прямой.
8. Изобразите движение четырех точек из разных углов экрана к центру.
9. Движение точки по окружности можно изобразить при помощи следующей программы:

```
алг круг
  дано
  надо !изобразить движение точки по окружности
  нач вещ x0, y0, r, пи, x, y, a
  графика
  x0:=100; y0:=40; r:=30
  пи:=3.1416
  нц для a от 0 до 2*пи шаг 0.1
  . x:=x0+r*sin(a)
  . y:=y0+r*cos(a)
  . точка(x, y)
```

кц  
кон

При этом следует иметь в виду, что окружность задается формулами  $x=\sin(t)$ ,  $y=\cos(t)$ , а об удобном размещении на экране мы заботимся сами, подбирая значения переменных  $x_0$ ,  $y_0$  (центр окружности) и  $r$  (радиус).

Изменив программу "круг", изобразите движение точки по спирали.

10. Изобразите движение двух точек по одной окружности так, чтобы они находились по разные стороны от центра окружности.
11. Нарисовать пирамидку, состоящую из уменьшающихся закрашенных прямоугольников. Использовать массивы  $X, Y$  для координат левого верхнего угла прямоугольников и массив  $L$  для длин прямоугольников.
12. Нарисовать 5 олимпийских колец. Использовать массивы  $X, Y$  для центров окружностей.
13. Задать в таблицах  $X$  и  $Y$  координаты вершин пятиугольника. Нарисовать этот пятиугольник и другой, у которого вершины расположены в серединах сторон исходного.
14. Нарисовать в верхнем левом углу кусочек солнышка и от него – расходящиеся лучи. Координаты концов лучей поместить в массивы  $X, Y$ .
15. В массиве  $B[1:8]$  хранятся сведения о направлении ветра в течение месяца следующим образом: элемент  $B[1]$  равен количеству дней, в течение которых дул северный ветер,  $B[2]$  – северо-восточный,  $B[3]$  – восточный и т.д. При помощи исполнителя ЧЕРЕПАШКА нарисовать розу ветров. Роза ветров – диаграмма из восьми отрезков, выходящих из одной точки под углом  $45^\circ$  друг к другу; длина каждого отрезка пропорциональна количеству дней, когда дул ветер соответствующего направления.
16. С помощью исполнителя ROBOT пройти лабиринт и закрасить все его клетки. Использовать вспомогательный алгоритм ПРОЙТИ\_РЯД\_И\_ВЕРНУТЬСЯ.
17. Нарисовать график радиации в верхнем ряду лабиринта ROBOTА.
18. Нарисовать графики радиации во всех рядах лабиринта.
19. Закрасить клетки лабиринта в шахматном порядке.
20. Закрасить клетку лабиринта, где радиация максимальна.
21. Нарисуйте план класса, используя вспомогательный алгоритм ПАРТА.
22. Нарисуйте прямоугольный план поселка, где дома двух типов расположены в шахматном порядке. Используйте вспомогательные ал-

горитмы ДОМ1 и ДОМ2.

23. Вокруг некоторой планеты по круговой орбите вращаются 6 одинаковых спутников, по 6 на каждой орбите. Спутники находятся на равном расстоянии друг от друга. Нарисуйте схему расположения планеты и спутников в некоторый момент времени.
24. Вокруг некоторой планеты по трем круговым орбитам вращается 18 одинаковых спутников, по 6 на каждой орбите. Спутники на одной орбите находятся на равном расстоянии друг от друга. Нарисуйте схему расположения планеты и спутников в некоторый момент времени.
25. Напишите программу, которая удваивает все буквы 'р'. Проверьте работу программы на слове 'транспортир'.
26. Проверьте, имеются ли в слове удвоенные буквы.
27. Выясните, какая буква встречается в слове чаще всего.
28. Постройте с помощью исполнителя ЧЕРТЕЖНИК параболу.
29. Напишите с помощью исполнителя ЧЕРТЕЖНИК слово "топот". Используйте вспомогательные алгоритмы "буква\_т", "буква\_о", "буква\_п".
30. В лабиринте для ПУТНИКА установить внутренние стены, примыкающие к внешним. Провести ПУТНИКА по периметру.

## ПРИЛОЖЕНИЕ

### Список команд редактирования

|                                |           |
|--------------------------------|-----------|
| Вставить символ                | INS       |
| Удалить символ                 | DEL       |
| Вставить строку                | SHIFT/INS |
| Удалить строку                 | SHIFT/DEL |
| Вставить конструкцию АЛГ – КОН | ESC/A     |
| НЦ ДЛЯ – КЦ                    | ESC/D     |
| НЦ ПОКА – КЦ                   | ESC/P     |
| ЕСЛИ-ТО-ИНАЧЕ-ВСЕ              | ESC/E     |
| ВЫБОР                          | ESC/B     |
| ПРИ                            | ESC/C     |

Для удаления конструкции нужно установить курсор на первую строку конструкции и нажать SHIFT/DEL. Частичное удаление конструкции возможно лишь в двух случаях:

1. Можно удалить строку ИНАЧЕ в конструкции ЕСЛИ-ТО-ИНАЧЕ-ВСЕ.
2. Можно удалить часть ПРИ в конструкции ВЫБОР.

В обоих случаях нужно установить курсор на указанную строку и нажать SHIFT/DEL.

#### Литература

1. Основы информатики и вычислительной техники: Проб. учеб. для сред. учеб. заведений / А.П.Ершов, А.Г.Кушниренко, Г.В.Лебедев и др.; Под ред. А.П.Ершова. – М.:Просвещение, 1988.

## Содержание

|  | Стр. |
|--|------|
| Предисловие .....                                    | 3    |
| <b>1. Основные конструкции</b>                       |      |
| 1.1. Исполнитель ЧЕРЕПАШКА .....                     | 4    |
| 1.2. Понятие переменной. Оператор присваивания ..... | 9    |
| 1.3. Цикл ДЛЯ .....                                  | 11   |
| 1.4. Исполнитель РОБОТ .....                         | 14   |
| 1.5. Цикл ПОКА .....                                 | 16   |
| 1.6. Ветвление .....                                 | 18   |
| 1.7. Исполнитель ГРАФИКА .....                       | 21   |
| 1.8. Литерные переменные .....                       | 23   |
| <b>2. Дополнительные возможности</b>                 |      |
| 2.1. Массивы .....                                   | 25   |
| 2.2. Вспомогательные алгоритмы .....                 | 29   |
| 2.3. Вспомогательные алгоритмы типа функция .....    | 33   |
| 2.4. Конструкция выбора .....                        | 36   |
| 2.5. Исполнитель ЧЕРТЕЖНИК .....                     | 39   |
| 2.6. Исполнитель ПУТНИК .....                        | 40   |
| <b>3. Задачи</b> .....                               | 42   |
| Приложение. Список команд редактирования .....       | 45   |
| Литература .....                                     | 46   |



Подписано к печати 7.09.89 г. Т13380. Тираж 660 экз.  
Зак. 2160Р. Уч.-изд.л. 2,0. Усл.печ.л. 3,0. Формат 60x90/16.  
Бумага офсетная. Цена 40 к. Изд.№ 287.

Отпечатано с оригинала-макета на ротапринтере в Стделе научно-  
технической информации Научного центра биологических исследо-  
ваний АН СССР в Пущине.