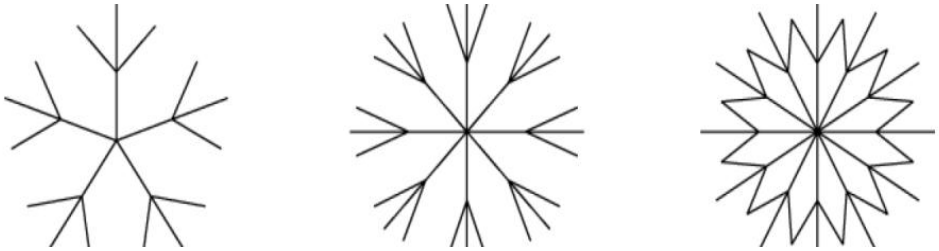
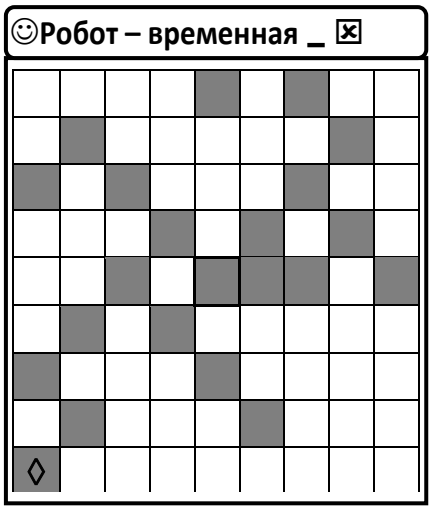
внеурочная деятельность по информатике



Алгоритмика



на КуМире

Дрожжина Е.В.

сборник заданий

по программированию

г. Белгород, 2016 год

ББК 32.97я72

Д75

Автор: Дрожжина Е. В., учитель информатики МБОУ «СОШ №4 г.Шебекино Белгородской области»

Рецензенты:
Лобашова Ю. А., старший методист центра методического обеспечения развития образования, заведующий центром дистанционных образовательных технологийОГАОУ ДПО «Белгородский институт развития образования»
Корнилова Е. А., заведующий кафедрой естественно-математического образования и информационных технологий ОГАОУ ДПО «Белгородский институт развития образования», к. пед. н., доцент
Старовойтов А. С., доцент кафедры информатики, естественнонаучных дисциплин и методик преподавания НИУ «БелГУ» к. ф.-м. н.
Корнилов А. В., заведующий кафедрой физики БГТУ им. В. Г. Шухова, к.ф.-м.н., доцент
Богданова В. А., учитель информатики и ИКТ высшей квалификационной категории МБОУ «Гимназия №3» г. Белгорода
Винникова О. Е., учитель информатики и ИКТ высшей квалификационной категории МАОУ «СОШ №24 с углубленным изучением отдельных предметов» Старооскольского городского округа Белгородской области

Д75 Дрожжина Е.В. Алгоритмика на КуМире: Сборник заданий по программированию в системе КуМир / Е.В.Дрожжина– Белгород, 2016. – 128 с.

Сборник заданий «Алгоритмика на КуМире» рекомендован к использованию в образовательных организациях Белгородской области во внеурочной деятельности (протокол заседания регионального координационного совета по вопросам формирования и функционирования инновационной инфраструктуры в сфере образования департамента образования Белгородской области от 29.06.2016 года №4).

Сборник заданий предназначен для обучения основам программирования с помощью исполнителей системы КуМир. Наличие теории, примеров решения задач, упражнений для самостоятельной работы, заданий экспериментального и исследовательского характера позволяют вести обучение с учетом требований ФГОС.

Данный сборник может быть использован во внеурочной деятельности по информатике в 5-9 классах. Подобранные задания позволяют комплексно и системно подойти к изучению алгоритмов на примерах исполнителей, приобрести навыки проектной деятельности, подготовиться к решению задач ГИА.

© Дрожжина Е.В., 2016.

О программе

КуМир (Комплект Учебных МИРов) - система программирования, предназначенная для поддержки начальных курсов информатики и программирования в средней и высшей школе. Особенности системы КуМир[1]:

- В системе КуМир используется школьный алгоритмический язык с русской лексикой и встроенными исполнителями Робот и Чертёжник.
- При вводе программы КуМир осуществляет постоянный полный контроль ее правильности, сообщая на полях программы обо всех обнаруженных ошибках.
- При выполнении программы в пошаговом режиме КуМир выводит на поля результаты операций присваивания и значения логических выражений. Это позволяет ускорить процесс освоения азов программирования.
- Кумир работает в операционных системах Windows или Linux.
- Система Кумир разработана в ФГУ ФНЦ НИИСИ РАН по заказу Российской Академии Наук и распространяется свободно на условиях лицензии GNU 2.0.

Скачать программу можно по ссылке с официального сайта: https://www.niisi.ru/kumir/index.htm . Версия программы: 1.9.

О сборнике

Сборник заданий «Алгоритмика на КуМире» состоит из трех частей. В первой части изучение основ программирования идет с помощью исполнителей: Черепахи, Водолея, Кузнечика и Чертежника. Вторая часть включает в себя задания для исполнителя Робот. В третьей части собраны задания, не предполагающие использование исполнителей. Каждая из частей начинается изучением самых простых приемов программирования и заканчивается разбором более сложных алгоритмических задач. В конце первой и второй части подобраны экспериментальные и исследовательские задания. Логика построения каждой части позволяет повторять пройденный материал, изучая его заново в среде нового исполнителя.

Практические работы в сборнике заданий предполагают ознакомление с новым материалом и содержат вопросы и задания. В нумерации практической работы первая цифра означает номер

раздела, а вторая – номер в разделе. Вопросы и задания в практических работах располагаются в порядке изучения материала. Названия программ выделены жирным шрифтом. Ответы на вопросы могут быть письменными или устными, групповыми или индивидуальными. Пиктограмма ✎ означает, что ответ будет письменным, а значок 💬 предполагает устный ответ, а 💬 - устное обсуждение вопроса. Значок 💻 означает, что задание нужно выполнить с помощью компьютера.

Введение

Термин «**Алгоритм**» происходит от слова ***Algorithmi:*** так на латинском языке звучало имя *хорезмского* математика **ибн Мусы аль-Хорезми**, одного из крупнейших средневековых персидских учёных IX века, астронома, географа и историка, трактат которого в Средние века был распространен в Европе. Тогда ***алгоритмом*** называлось десятичное счисление и искусство счета. Аль-Хорезми впервые представил алгебру как самостоятельную науку об общих методах решения линейных и квадратных уравнений, дал классификацию этих уравнений[2].

В настоящее время ***алгоритмом*** называют последовательность четко сформулированный указаний **исполнителю**, которым он должен следовать для достижения результата или решения задачи[3].

Вы уже знакомы с *правилами* сложения, вычитания, умножения, деления чисел. Эти правила можно назвать ***алгоритмами***. *Алгоритмизации* поддаются многие инженерные и экономические задачи. В повседневной жизни мы также пользуемся ***алгоритмами.*** Например, такими как переход дороги или оказание первой помощи.

✎**Вопрос 1.** Составьте алгоритм заварки чая из следующих команд, расположив их в правильной последовательности: а) взять стакан; б) определить, когда вода закипит; в) включить чайник; г) налить воду в стакан; д) положить чайный пакетик в стакан; е) налить в чайник воду.

Вопрос 2. Теперь составьте свой алгоритм приготовления чая. Какие команды должен понимать исполнитель вашего алгоритма? Можно ли поменять местами команды в предложенном вами алгоритме?

Сегодня мы познакомимся с исполнителями алгоритмов системы КуМир. В отличие от человека, эти исполнители понимают не все слова русского языка, а только специальные **команды**. В системе КуМир существует 5 исполнителей: **Кузнечик**, **Черепаха**, **Водолей**, **Чертежник** и **Робот**. Нумерация заданий будет вестись по каждому исполнителю отдельно и начинаться с первой буквы исполнителя. Поскольку Черепаха и Чертежник начинаются с одной буквы, то для Черепахи будут использоваться номера **Ч**, а для Чертежника – **ЧТ**. В задачах без использования исполнителей нумерация будет начинаться с **А**. Практические работы включают вопросы и задания.

Часть 1. Исполнители: Черепаха, Кузнечик, Водолей и Чертежник

Раздел 1. Использование Пульта исполнителя

Исполнитель Кузнечик

Практическая работа №1.1.Управление Кузнечиком.

Цель: познакомиться с возможностями исполнителя Кузнечик. Откройте среду исполнителя Кузнечик.

Запустите КуМир. Нажмите **Миры**. Вызовите **Кузнечик - Пульт** и сверните окно КуМира. Оставьте только окна **Пульта** и **Кузнечика** (рис.1)

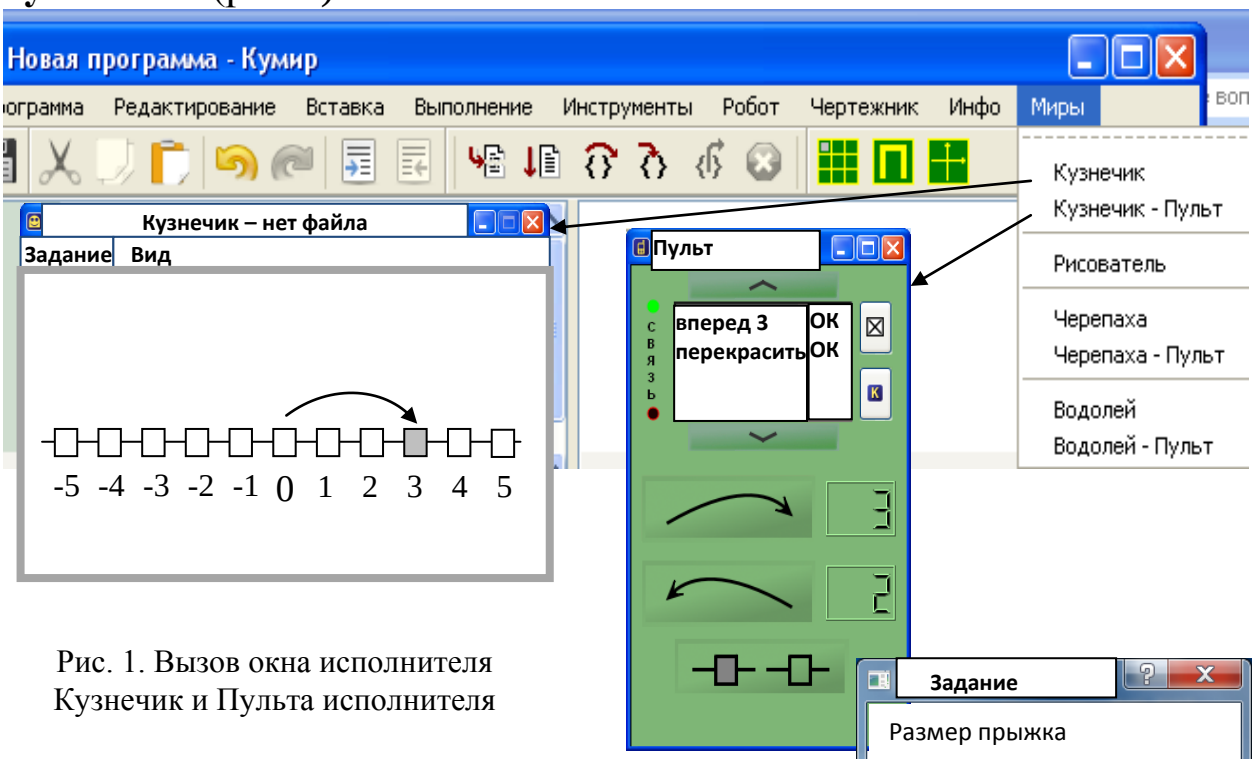
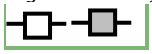


Рис. 1. Вызов окна исполнителя Кузнечик и Пульта исполнителя

Задайте *новое задание*: **Вперед 3. Назад 2** (рис.2). Исследуйте **Пульт** управления.

Вопрос 1. Какие команды входят в **систему команд исполнителя** Кузнечик?

Вопрос 2. Что получится, если нажать на кнопку перекрасить  дважды?

Вопрос 3. Что означает на **Пульте** кнопка с крестиком? 

Итак, вы познакомились с возможностями программы: научились использовать **Пульт** управления, видеть в

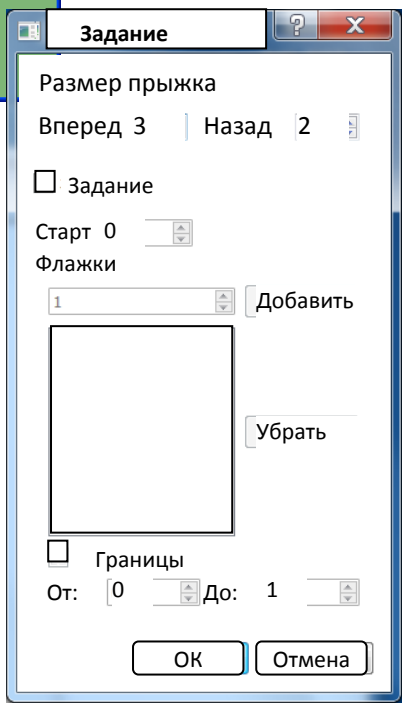


Рис. 2. Окно нового задания исполнителя Кузнечик

окне Кузнечика результат выполнения команд, задавать ***новое задание***.

Задание №К1. С помощью команд Кузнечика **вперед 3**, **назад 2** перекрасьте точки: 0,1,2,3,4,5,6. **Старт 0.**

Задание №К2. С помощью команд Кузнечика в**перед 5**, н**азад 2** перекрасьте точки: 0,1,2,3,4,5,6. **Старт 0.**

Задание №К3. Пусть Кузнечик побывает по одному разу в точках 1, 2, 3, 4, 5 и перекрасит их, не выходя за пределы отрезка от 0 до 5. Добавьте Флажок и выставьте **границы 0 и 5** (рис.2). Команды Кузнечика: **вперед 3**, н**азад 2**. **Старт 0.**

Задание №К4. а) С помощью команд Кузнечика **вперед 5**, н**азад 3** перекрасьте точки: 0,1,2,3,4,5,6. **Старт 0.**

б) Задайте команды: **вперед 11, назад 5**. Перекрасьте все точки от 1 до 10. **Старт 0.**

Задание №К5. Задайте команды: в**перед 12**, н**азад 3**. Перекрасьте все возможные точки от 1 до 10. Какие точки удалось перекрасить? Сможет ли Кузнечик попасть в точку с координатой 1? **Старт 0.**

Здание №К6. Кузнечик последовательно выполнил команды задания. В какой точке оказался Кузнечик, если он начинал движение из точки 0? **а)** вперед 5, назад 3, вперед 5, назад 3, вперед 5, назад 3.

б) вперед 5, вперед 5, вперед 5, назад 3, назад 3, назад 3.

в) назад 3, вперед 5, вперед 5, вперед 5, назад 3, назад 3.

Задание №К7. Вычислите точку, в которой оказался Кузнечик, если до начала программы он находился в точке 0 и выполнил следующие команды:

а) 3 команды **вперед 5** и три команды **назад 3?**

б) 2 команды **вперед 3** и 3 команды **назад 2?**

в) 5 команд **вперед 8** и 8 команд **назад 5**?

Задание №К8. а) Задайте условия: в**перед 8**, н**азад 5, старт 0**. С помощью **Пульта** помогите Кузнечику попасть в точку с координатой 1. Подтвердите результаты программного эксперимента математически. Посчитайте, сколько команд в**перед 8** и сколько команд **назад 5** пришлось делать Кузнечику?

б) выполните задание для команд: **вперед 9**, н**азад 5**; **в**) в**перед 12**, **назад 5; г**) **вперед 15,** н**азад 6**.

Задание №К9. Задайте команды: в**перед 3**, н**азад 2**. с**тарт 0**. С помощью **Пульта** управления запишите алгоритм для Кузнечика, состоящий не более чем из ***семи*** команд, который позволит Кузнечику перекрасить:

а) точки 0, 2 и 4; **б)** точки 2 и 5;
в) точки 4 и 5; **г)** точки 5 и 7.

Задание №К10. Задайте условия: **вперед 4**, **назад 3**. **старт 0**. С помощью **Пульта** управления запишите алгоритм для Кузнечика, состоящий не более чем из ***семи*** команд, который позволит Кузнечику перекрасить:

а) точки 0, 2 и 5; **б)** точки 1 и 6; **в)** точки 2 и 6;
г) точки 5 и 6; **д)** точки 6 и 9.

Задание №К11. Определите, какие точки будут закрашены в результате работы программы, если Кузнечик начинал своё движение из точки с координатой ***0***.

а)	б)	в)
использовать Кузнечик **алг К11_а** **нач** .перекрасить . вперед 3 . назад 2 . перекрасить **кон**	использовать Кузнечик **алг К11_б** **нач** . вперед 3 . перекрасить . назад 2 . перекрасить **кон**	использовать Кузнечик **алг К11_в** **нач** .вперед 3 . перекрасить . перекрасить . назад 2 **кон**

Задание №К12. Определите, в какой точке будет находиться Кузнечик после исполнения каждого из алгоритмов, представленных блок – схемами на рисунке 3. Команда **повтори**, означает что действия, записанные в блок – схеме, будут выполнены несколько раз. Начало движения - точка с координатой 0.

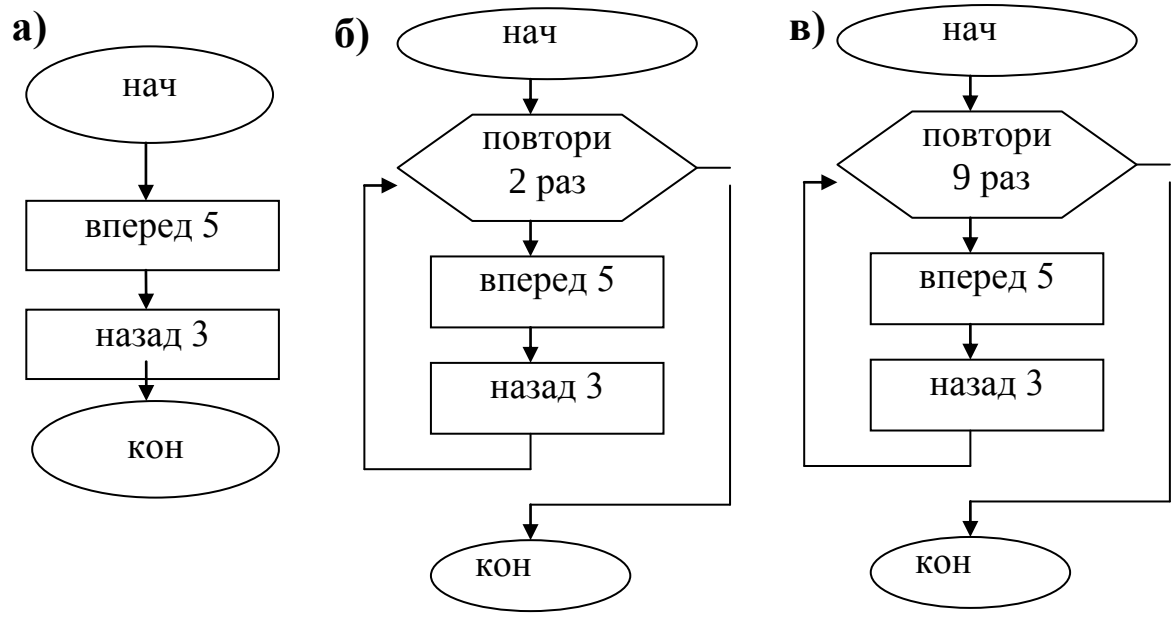


Рис. 3. Блок-схемы к заданию № К12

Задание №К13. Кузнечик умеет выполнять команды **вперед 5**, **назад 3, перекрасить.**

Известно, что после выполнения алгоритма Кузнечик перекрасил заданные точки (рис. 4).

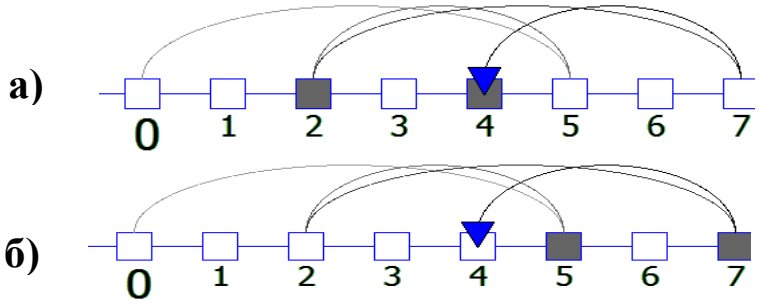


Рис. 4. Результат работы исполнителя Кузнечик

Запишите команды алгоритма в блок – схему, представленную на рисунке 5:

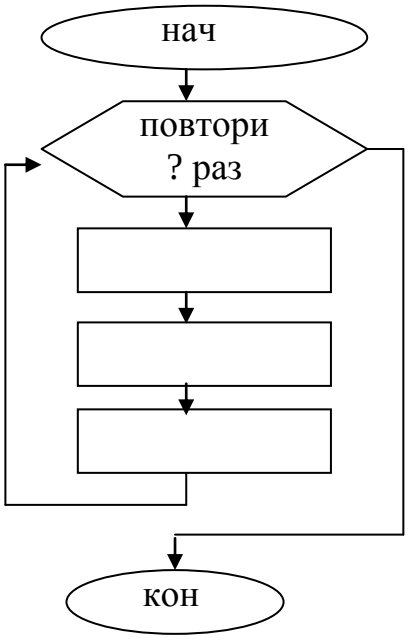


Рис. 5. Блок-схема к заданию №К13

Исполнитель Черепаха

Практическая работа №1.2. Управление Черепахой.

Цель: познакомиться с возможностями исполнителя Черепаха.

Запустите КуМир. Нажмите пункт меню Миры. Вызовите **Пульт** Черепахи (рис. 6) и **окно** Черепахи. Изучите возможности исполнителя с помощью **Пульта**.

✎**Вопрос 1.** Какие команды входят в **систему команд исполнителя** Черепаха?

💻**Задание 1.** Придумайте фигуру, для рисования которой Черепаха будет использовать не более пяти команд. Нарисуйте на эту фигуру с помощью **Пульта** на поле Черепахи. Запишите команды программы в тетрадь.

Рис. 6. Пульт управления исполнителем Черепаха

Задание №Ч1. С помощью **Пульта** нарисуйте фигуры, представленные на рисунке 7:

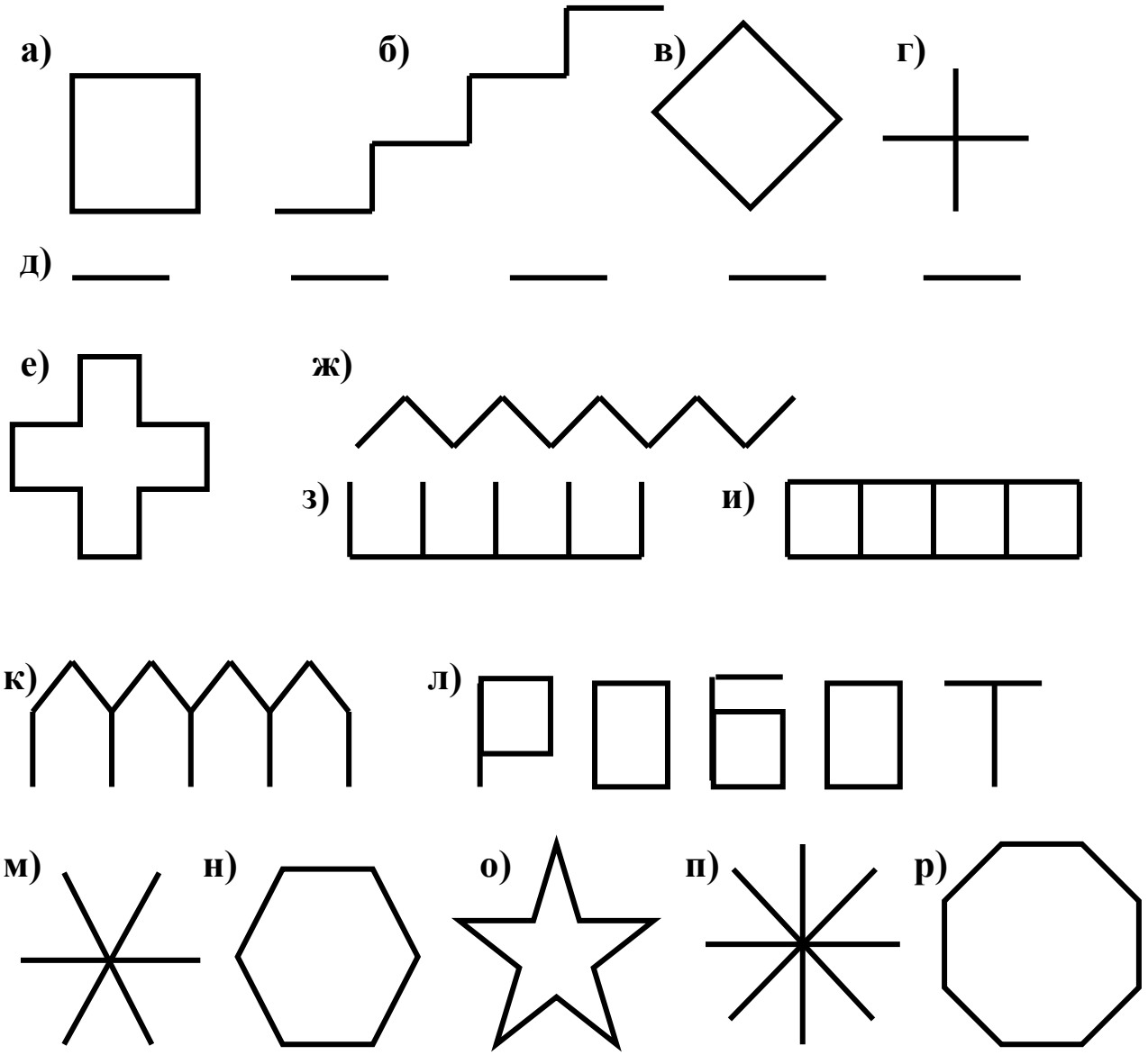


Рис. 7. Задания для исполнителя Черепаха

Задание №Ч2. Многоугольник называется ***правильным***, если у него все стороны и все углы одинаковые. Нарисуйте правильные многоугольники с помощью **Пульта** исполнителя Черепаха:

1) квадрат, **2)** восьмиугольник; **3)** шестиугольник;
4) треугольник; **5)** десятиугольник; **6)** пятиугольник.

В тетрадь запишите градусные меры углов, на которые должна поворачиваться Черепаха при рисовании указанных фигур, и количество поворотов при рисовании фигуры.

№ фигуры	Вид правильного многоугольника	Градусная мера поворота	Количество поворотов
1.	***квадрат***	***90°***	***4 раза***

Исполнитель Водолей

Практическая работа №1.3. Запуск Водолея и назначение задания.

Цель: познакомиться с возможностями исполнителя Водолей.

Задание 1. Запустите КуМир. Нажмите Миры. Вызовите **Пульт** Водолея (рис. 8) и **окно** Водолея (рис 9).

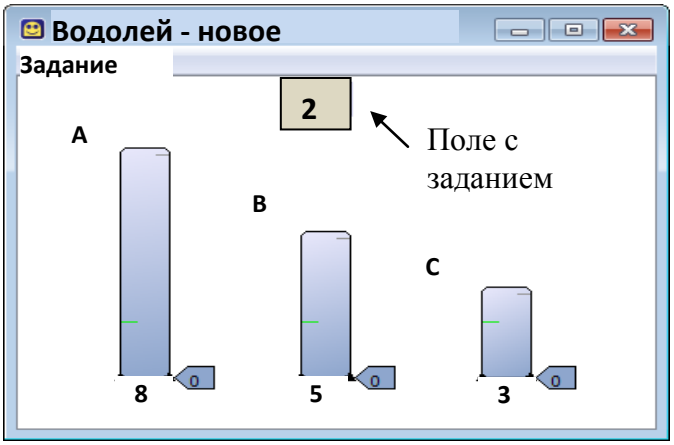


Рис. 9. Окно Водолея

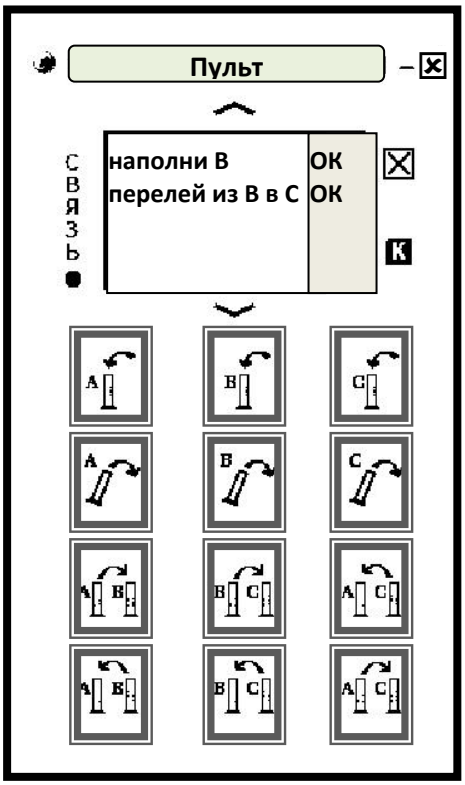


Рис 8. Пульт Водолея

Задание 2.Выставьте для Водолея новое *задание*: Размер сосудов: 8,5 и 3 литра. Отмерить 2 литра (рис. 10).

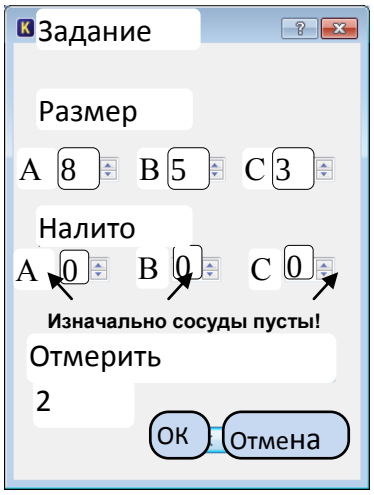


Рис. 10. Назначение нового задания для Водолея

Если вы отмерите 2 литра, то поле с заданием (рис. 9) станет зеленого цвета.

Для решения этой задачи выполните с помощью **Пульта** команды: **наполни B, перелей из B в C**. Убедитесь, что поле ответа стало зеленым.

Задание №B1. Размер сосудов: 8, 5 и 3 литра. Отмерить **а)** 4 литра; **б)** 7 литров, **в)** 1 литр.

Задание №B2. Размер сосудов: 8, 4 и 3 литра. Отмерить **а)** 5 литров; **б)** 6 литров; **в)** 7 литров.

Задание №B3. Размер сосудов: 5, 3 и 0 литров. Отмерить 4 литра. Запишите количество команд, которое вам потребовалось для выполнения задания.

Задание №B4. Отмерить 1 литр с помощью сосудов: **а)** 7 и 2 литра; **б)** 5 и 2 литра; **в)** 11 и 2 литра.

Задание №B5. Составьте задачу для Водолея, для решения которой потребуется не менее **а)** трех команд; **б)** четырех команд; **в)** пяти команд.

Раздел 2. Написание программ

Первая программа

Программа – это ***алгоритм***, написанный на языке исполнителя.

Практическая работа №1.4. Первая программа.

Цель: научиться **в точности** переписывать программу и запускать ее на исполнение.

Вопрос 1: Подумайте, что нарисует Черепаха после выполнения программы **первая программа.kum** (рис.11)? Ответ нарисуйте в тетрадь.

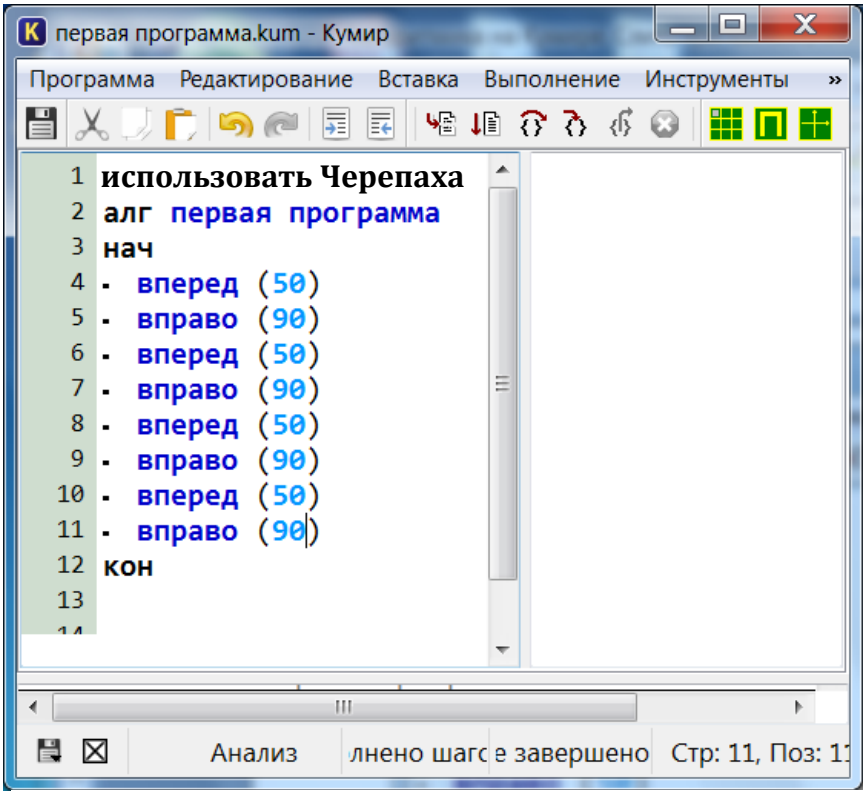


Рис. 11. Первая программа.

Вопрос 2. Какой исполнитель выполнил программу?

Вопрос 3. Какие служебные слова написаны в начале программы? Каким служебным словом заканчивается программа?

Вопрос 4. По справке КуМир определите систему команд исполнителя Черепаха. Вызовите меню Инфо → Описание Миров → Черепаха.

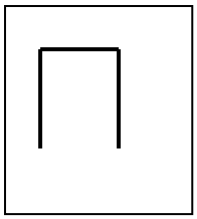
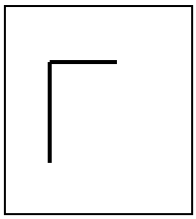
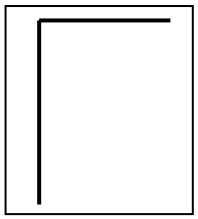
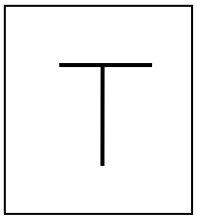
Задание 1. Запустите КуМир**.** Перепишите в точности программу.

Замечание: *точки перед командами показывают уровень вложенности команд алгоритма. В коде программы они появляются автоматически, и ставить их специально не нужно*. В этой программе команды **вперед (50)** и **вправо (90)** отделены от начала строки одной точкой. Они вложены в блок «**начала – конца** алгоритма». С другими уровнями вложенности мы познакомимся позднее.

Отправьте программу на выполнение командой «выполнить непрерывно».  Должно появиться окно Черепахи с результатом работы программы.

Вопрос 4. Что нарисовала Черепаха в результате работы программы? Совпал ли этот результат с тем, который предположили вы?

Задание №Ч3. Исправьте эту программу так, чтобы она рисовала: **а)** букву «п»; **б)** букву «г»; **в)** букву «г» в два раза большего размера; **г)** букву «т».

а)  **б)**  **в)**  **г)** 

Задание №Ч4. Запустите написанную на экране программу на пошаговое исполнение кнопкой «ШАГ»  . Нажимайте эту кнопку после каждого шага. Это позволит вам проследить процесс выполнения программы по шагам. Исправьте программу для рисования буквы Т в два раза большего размера.

Задание №Ч5. Напишите программы для рисунков задания №Ч1 (рис.7).

Практическая работа №1.5. Повторяющиеся части.

Цель: научиться выделять в программе повторяющиеся части, без ошибок переписывать программу и отправлять ее на исполнение**.**

В среде программирования можно попросить исполнителя выполнить последовательность команд несколько раз. Это уменьшает **код** программы. Его становится легче понять и прочитать. Компьютер понимает что команды, записанные между **нц** (*началом цикла*) и **кц** (*концом цикла*), нужно выполнить столько раз, сколько определено в *условии* цикла.

```
использовать Черепаха
алг повторяющиеся части
нач
. нц 4 раз
. . вперед (50)
. . вправо (90)
. кц
кон
```

Рис. 12. Код программы «**повторяющиеся части**».

Команды, которые расположены между ***началом*** и ***концом*** цикла, называют ***телом цикла.*** Для удобства чтения **кода** они отделены (рис.12) двумя точками. Это означает, что они вложены в блок «**начала – конца цикла**». В нашем примере *тело цикла* составляют две команды**: вперед (50)** и **вправо (90).**

Вопрос 1.Как вы думаете, что нарисует Черепаха, выполнив программу **повторяющиеся части** (рис.12)?

Вопрос 2.Запишите команды программы **повторяющиеся части** в блок – схему (рис.13).

Задание. Перепишите код программы без ошибок в КуМир. Запустите программу на исполнение.

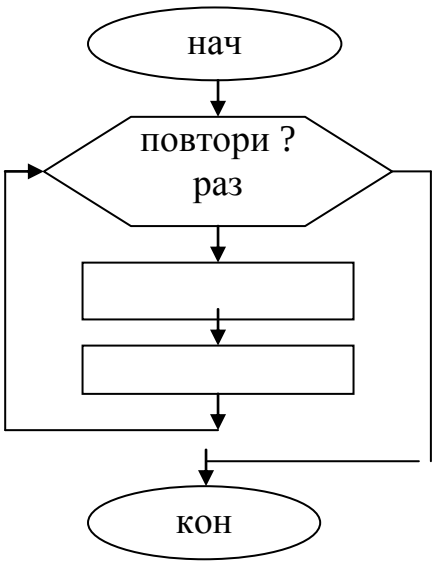


Рис. 13. Блок-схема к вопросу 2 Пр.раб №1.5

Задание №Ч6. Исправьте программу **повторяющиеся части** так, чтобы она рисовала **а)** большой квадратик; **б)** букву П; **в)** букву Г.

Задание №Ч7. Исправьте программу **повторяющиеся части** так, чтобы она рисовала:

а) восьмиугольник;
б) шестиугольник;
в) десятиугольник;
г) треугольник.

Задание №Ч8. Поставьте компьютерный эксперимент: исправьте в программе **повторяющиеся части** количество циклов и угол поворота на ***любые*** числа. Для остановки программы служит кнопка «**Прервать**» (рис.14).

Рис. 14. Кнопка «**Прервать**»

Могут получиться такие рисунки (рис.15):

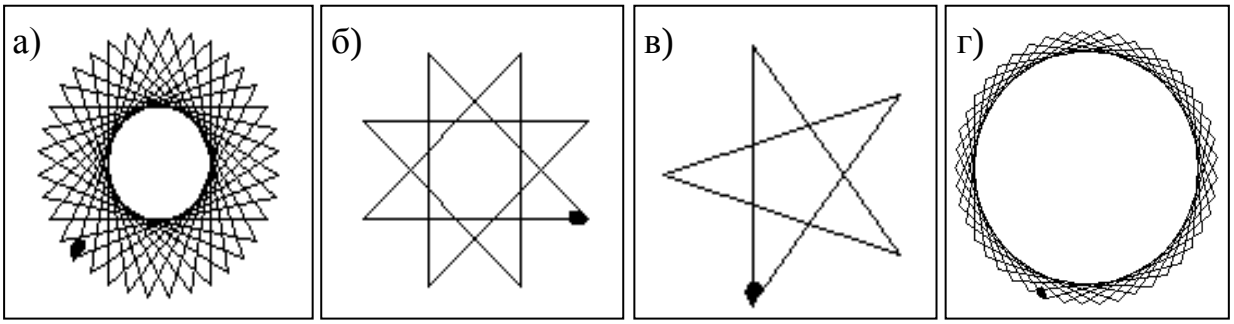


Рис. 15. Возможные рисунки программ при изменении параметров угла поворота и количества повторений в программе **повторяющиеся части**

Циклические алгоритмы

Цикл – это повторение некоторых действий [3].

Практическая работа №1.6. ***Циклические*** **алгоритмы.**

Цель**:** понять назначение алгоритмической структуры «**цикл**» и правила записи этой алгоритмической конструкции.

Ответьте устно на вопросы по окну программы на рисунке 16.

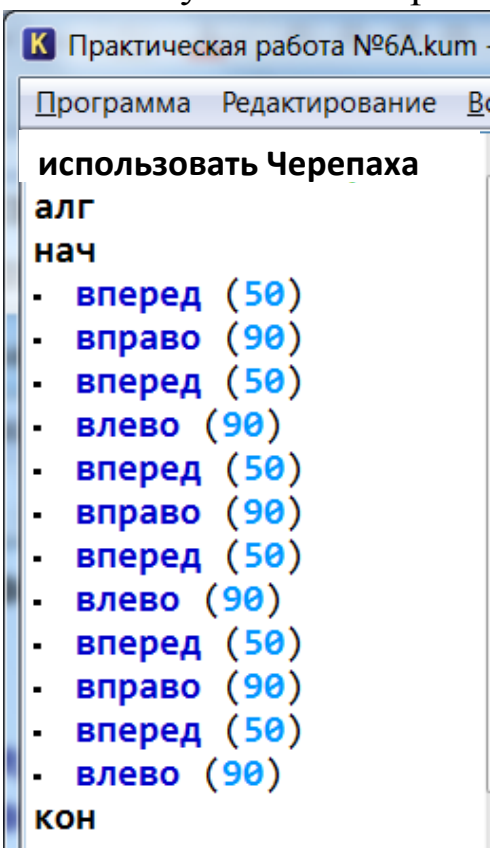


Рис. 16. Код программы **Практическая работа №6A.kum.**

Вопрос 1. Как называется файл с программой и какое расширение он имеет?

Вопрос 2. Какие команды понимает Черепаха?

Вопрос 3. Можно ли записать **вперёд** вместо **вперед**?

Вопрос 4. Что означает команда **вправо (90)?**

Вопрос 5. Что нарисует Черепаха в результате работы программы (рис. 16)?

Вопрос 6. Есть ли в этой программе повторяющиеся части алгоритма?

Вопрос 7. Сколько команд повторяется?

Вопрос 8. Какие команды повторяются?

Вопрос 9. Запишите команды программы **Практическая работа №6A.kum** в блок – схему циклического алгоритма на рисунке 17.

Задание 1. Запустите среду КуМир. Перепишите код программы **Практическая работа №6A.kum** (рис.16) в систему КуМир. Запустите программу.

Задание 2. Теперь запишем ЭТОТ же алгоритм с использованием ***цикла*.**

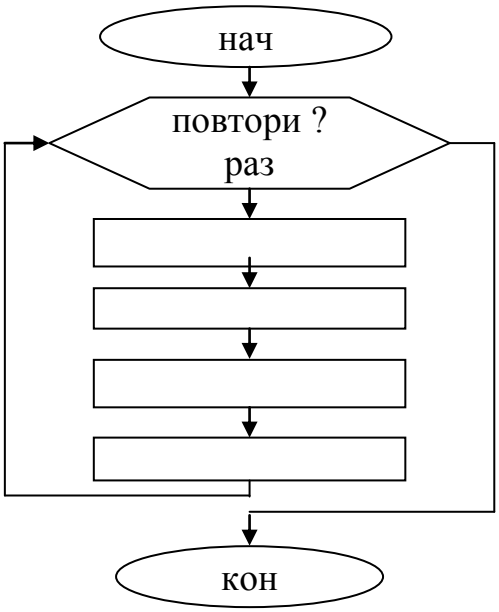


Рис. 17. Блок- схема к вопросу 9 **практической работы 1.6**

Неполный код программы представлен скрине экрана **«Практическая работа №6Б.kum»** (рис.18)**.** Команды тела цикла запишите самостоятельно.

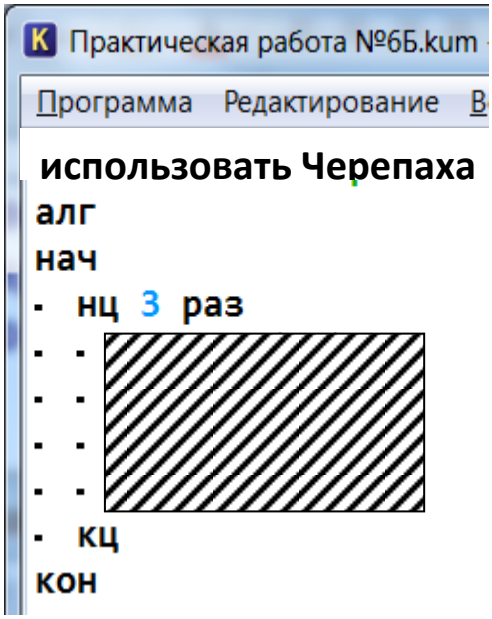


Рис. 18. Неполный код программы **Практическая работа №6Б.kum**

Запустите программу. Убедитесь, что эта программа рисует то же самое, что и программа **Практическая работа №6А.kum**.

Задание №Ч9. Проведите компьютерный ***эксперимен***т, изменяя количество повторений цикла и команды внутри цикла программы **Практическая работа №6Б.kum**. Получите оригинальные рисунки с помощью циклов. Некоторые из них могут напомнить звездочки.

Задание №Ч10. Выберите понравившиеся рисунки для Черепахи из задания **Ч1** и напишите для них программы с использованием цикла.

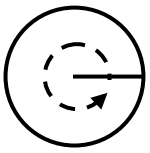
Рисование многоугольников и снежинок

Практическая работа №1.7. Рисование многоугольников и снежинок.

Цель: научиться вычислять градусы поворота для рисования правильных многоугольников с заданным количеством сторон.

Вам уже известно, что многоугольник называется ***правильным***, если у него все стороны и углы одинаковые.

Вопрос 1. Вспомните, сколько градусов в полной окружности?



Задание 1. Измерьте или вычислите углы поворота Черепахи при рисовании следующих фигур:

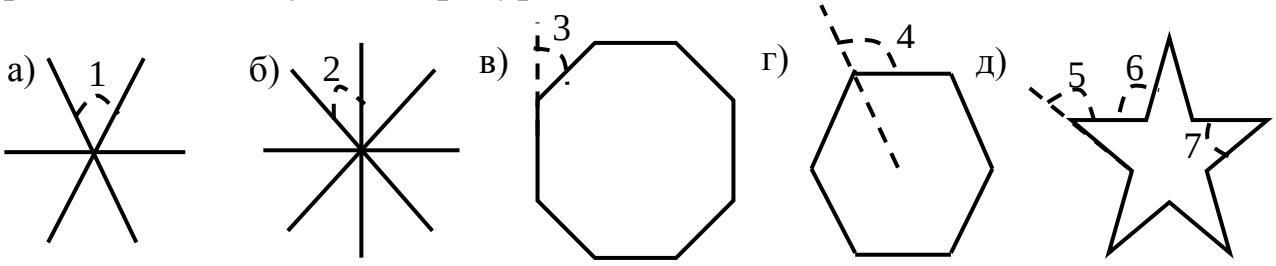


Рис. 19. Рисунки к **заданию 1** практической работы №1.7

Рассмотрим правила составления алгоритмов для рисования многоугольников на примере базовой программы рисования правильного четырехугольника. Посмотрим на код алгоритма **квадрат** (рис. 20).

```
использовать Черепаха
алг квадрат
нач
. нц 4 раз
. . вперед (50)
. . вправо (90)
. кц
кон
```

Рис. 20. Код программы **квадрат.**

Вопрос 1. В какую сторону смотрит Черепаха до выполнения программы?

Вопрос 2.Сколько раз Черепаха поворачивалась на 90 градусов в ходе выполнения алгоритма **квадрат**?

Вопрос 3.Вернулась ли Черепаха в исходное положение?

Вопрос 4. Куда будет смотреть Черепаха после прохождения всей программы? (вверх, вниз, вправо, влево)

Вопрос 5. Что нарисует Черепаха, если количество циклов изменить на **6?** В какой вершине квадрата, представленного на рисунке 21, она окажется? Куда будет смотреть: вверх, вниз, вправо или влево?

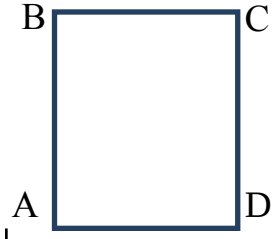


Рис. 21. Квадрат.

Вопрос 6. Что нужно исправить, чтобы Черепаха нарисовала большой квадрат?

Вопрос 7. Что нужно исправить, чтобы Черепаха нарисовала восьмиугольник?

Задание 2. Наберите код программы **квадрат.** Убедитесь, что программа работает.

Задание 3. Исправьте эту программу так, чтобы она рисовала:

а) восьмиугольник; **б)** шестиугольник;

г) треугольник; **д)** пятиугольник;

е) десятиугольник.

Задание 4. а) Нарисуйте снежинку из десяти лучиков. Для этого в программу для рисования десятиугольника вставьте только **одну** команду.

б) Нарисуйте снежинку из 20 лучиков.

Задание №Ч11. Нарисуйте 5- ти конечную звезду**.** Задайте количество повторений цикла **5**. В теле цикла запишите команды **вперед (100); вправо (144)**. Подберите угол поворота для других звезд.

Трассировка программы

Трассировка – это процесс пошагового выполнения программы.

Вы уже выполняли задания типа: «***что нарисует Черепаха»*** или «***в какой точке окажется Кузнечик»***. Это и есть ***трассировка.*** Каждый программист, когда пишет программу, знает, зачем нужна та или другая команда, что получится после того, как исполнитель выполнит эту команду.

Интересно, что на заре программирования был популярен способ написания программы типа «***лапша***». При таком способе считалось «вершиной мастерства» написать программу, которая непонятно как работает (но работает). Сам программист через некоторое время не мог разобраться, для чего нужна та или другая команда. И, как следствие, редактировать эти программы для решения задач было невозможно. От этого способа очень быстро отказались. В настоящее время «хорошим тоном» в программировании считается написание **программного кода** без лишних команд, с понятными комментариями.

Практическая работа №1.8. Трассировка программы.

Цель: научиться выполнять трассировку несложных программ.

Рассмотрим трассировку на примере следующей программы **Практическая работа №1.8.kum** (рис.22) для Водолея с начальными условиями: ёмкости трёх сосудов 8, 5 и 3 литра соответственно. Изначально сосуды пусты. Нужно определить, сколько литров будет в каждом сосуде после выполнения программы.

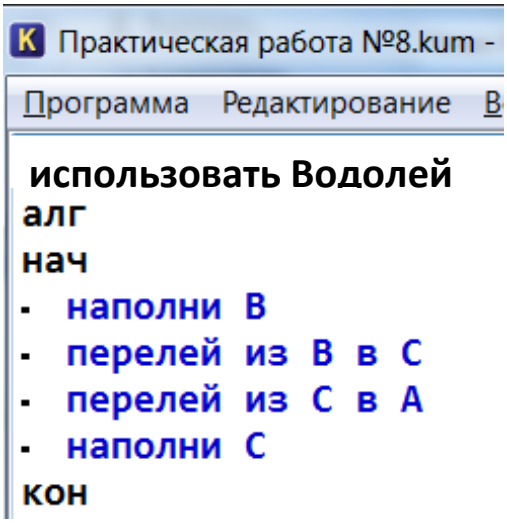


Рис. 22. Код программы **Практическая работа №1.8.kum**

Выполним трассировку программы с помощью рисунков сосудов. Заметим, что в программе 4 команды.

✎**Задание 1**. Рассмотрите рисунок 23, демонстрирующий работу программы **Практическая работа №8.kum**.

Обратите внимание на то, что на **втором** шаге «**перелей из B в C**» жидкость в сосуде **B** осталась в количестве 2-х литров. Это важно отметить на рисунке (рис. 23).

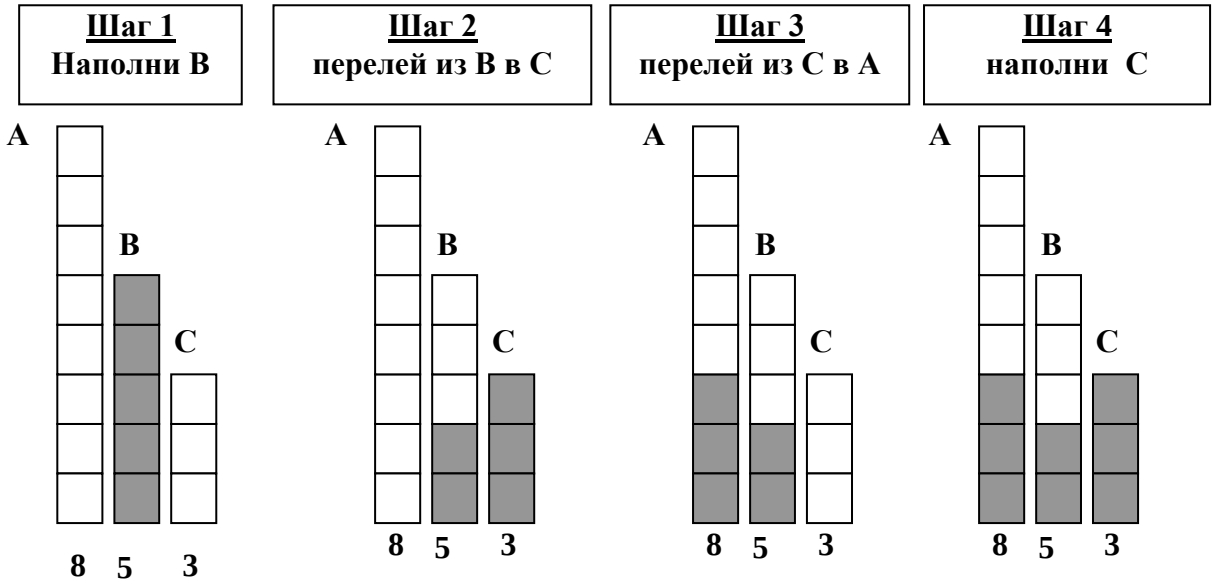


Рис.23. Трассировка программы практической работы №8

На **третьем** шаге «**перелей из C в A**» сосуд **C** стал пустым, а сосуд **B** мы не трогали, поэтому в нем налито столько, сколько и было. Это тоже надо отметить на рисунке.

На **четвертом** шаге мы наполнили сосуд **C**. Сосуды **A** и **B** остались с прежними значениями.

Таким образом, получаем **ответ**: **A=3, B=2, C=3.**

Задание 2. Перерисуйте рисунки трассировки программы **Практическая работа №8** в тетрадь. Запишите ответ.

Задание 3. Запишите программу **Практическая работа №8.kum** на компьютере. Задайте начальные условия для трех сосудов: **8, 5** и **3** литра. Получить **6**. Изначально сосуды пусты.

Выполните пошаговую трассировку программы с помощью кнопки «ШАГ»  . Сверьте результаты работы программы на каждом шаге.

Задание 4. Допишите в программу **одну команду** так, чтобы в каком- либо сосуде стало **6** литров.

Задание №B6. Определите, сколько литров будет налито в сосуды **A, B** и **C** в результате работы программы задания B6 на рисунке 24, если ёмкости сосудов имеют размеры 8, 4 и 3 литра? Оформите решение в тетради по образцу рисунка 23, демонстрирующего пошаговую работу программы.

В ответе запишите, сколько литров жидкости получилось в каждом сосуде. Изначально сосуды пусты.

```
использовать Водолей
алг задание В6_а
нач
. наполни В
. перелей из В в С
. перелей из В в А
. вылей С
. наполни В
. перелей из В в С
. перелей из В в А
кон
```

```
использовать Водолей
алг задание В6_б
нач
. наполни В
. наполни А
. перелей из А в С
. вылей В
. вылей С
. перелей из А в В
. перелей из В в С
кон
```

```
использовать Водолей
алг задание В6_в
нач
. наполни А
. перелей из А в В
. перелей из В в С
. перелей из С в А
. наполни С
. перелей из С в В
. перелей из В в С
кон
```

Рис. 24. Коды программ к заданию№В6.

Задание №В7. Даны три сосуда с ёмкостями 8, 4 и 3 литра. Изначально сосуды пусты. Напишите программу, состоящую не более чем из четырех команд, позволяющую Водолею получить 2 литра в любом из сосудов.

Проверьте свое решение на компьютере.

Задание №В8. Даны три сосуда с ёмкостями в 7, 5, 8 литров. Изначально сосуды пусты. Нужно отмерить 2 литра. Составьте алгоритм и проверьте его на компьютере.

Задание №В9. Сосуды **А**, **В** и **С** имеют размеры **8**, **5** и **3** литра соответственно. Изначально сосуды пусты. Определите, сколько литров будет налито в сосуд **А** при выполнении программы **задание_В9** (рис.25)**.**

а) после первого цикла?

б) после второго цикла?

в) после третьего цикла?

г) посчитайте количество команд при выполнении программы исполнителем.

```
использовать Водолей
алг задание_В9
нач
. нц 3 раз
. . наполни В
. . перелей из В в С
. . перелей из В в А
. . вылей С
. кц
кон
```

Рис. 25. Код программы **задание_В9.**

Задание №К14. Перекрасьте точки, которые перекрасил Кузнечик (рис.27), выполнив программу **задание_К14** (рис.26).

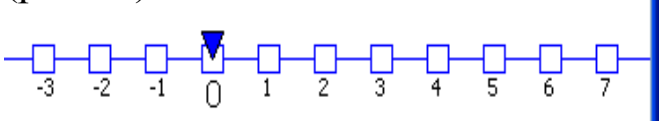


Рис. 27.Чиловая прямая исполнителя Кузнечик.

```
использовать Кузнечик
алг задание_К14
нач
. нц 4 раз
. . вперед 3
. . назад 2
. . перекрасить
. кц
кон
```

Рис. 26. Код программы **задание_К14.**

Отладка программы

Отладкой программы называют процесс исправления ошибок. Ошибки бывают ***синтаксические*** и ***алгоритмические***. К синтаксическим ошибкам относятся такие, как неправильное написание команд или переменных. Например, слово **вперёд** для исполнителя Черепаха будет непонятным. Компьютер сразу видит, что он не знает, как выполнить такую команду, и может написать: ***такой команды нет***.

С алгоритмическими ошибками все намного сложнее: программа вроде бы работает, но делает не то, что хотелось. Для того чтобы исправить такую ошибку, нужно хорошо понимать ***алгоритм*** решения этой задачи.

Практическая работа №1.9. Отладка программы.

Цель: научиться определять синтаксические ошибки и исправлять их.

Вопрос 1. Ты уже знаком с исполнителями Черепахой и Кузнечиком. Распредели команды, которые понимают исполнители, на две колонки: для Черепахи и для Кузнечика.

вперед (90), назад 2, перекрасить, поднять хвост, назад (90), вправо (60), опустить хвост, вперед 5, перекрасить, влево (50).

Вопрос 2. Какие синтаксические ошибки допущены при написании программы, представленной на рисунке 28.

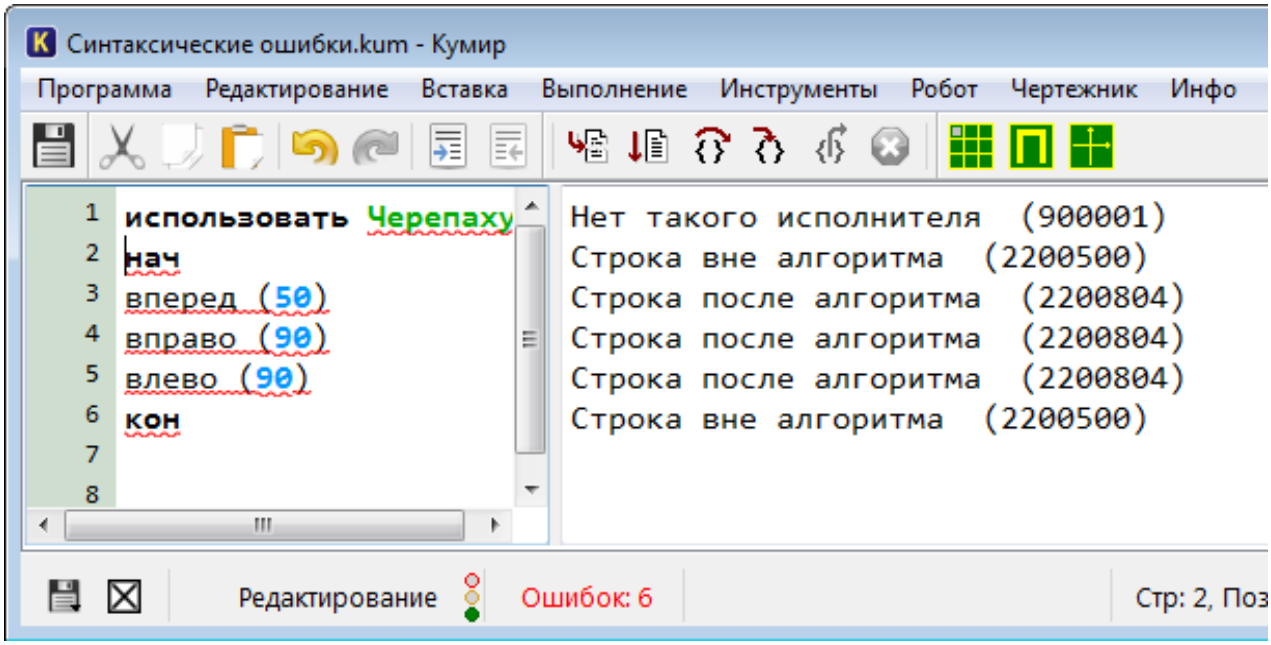


Рис.28. Синтаксические ошибки

Задание 1. Запишите синтаксически ***правильную*** программу в тетрадь. Нарисуйте результат.

Задание 2. Назовите, какие ошибки допущены в программе, представленной на рисунке 29, исправьте их и запишите ***правильную*** программу на компьютере.

```
использовать Черепаха
алг синтаксические ошибки
нач
· нц 2 раза
· · вперёд (90)
· · вправо (90)
· кц
кон
вправо (360)
```

```
Имя не может нач. с цифры  (6102)
Нет такого алгоритма  (2705)


Строка вне алгоритма  (2200500)
```

Рис.29. Синтаксические ошибки.

Задание 3. Расскажите, с какими ошибками вы встречались при написании программ?

Использование Пульта для написания программ

Задание №В10. Сосуды **A**, **B**, **C** ёмкостью по **8** литров. В сосудах А и В налито 5 и 3 литра соответственно, сосуд **C- пустой.**

Напишите с помощью **Пульта** программу, которая ***поменяет*** количество жидкости в сосудах.

В результате работы программы в сосуде **A** должно быть налито **3** литра, а в сосуде **B – 5** литров.

Рис. 30. Пульт Водолея

Перенесите программу из **Пульта** в КуМир специальной кнопкой (рис. 30). Учтите, что код программы будет перенесен в место, где стоит курсор. Убедитесь, что программа работает.

Выполните пошаговое исполнение программы.

Задание №В11.Подумайте, какой исполнитель поможет нам решить следующую задачу, и решите ее**.**

Однажды Лунтик пришел к Пчеленку в гости и застал его за интересным занятием. Пчеленок переливал мед из одного ведерка в другое и что – то записывал в тетради.

- Что ты делаешь? – спросил Лунтик.

- Ищу оптимальный алгоритм! – гордо ответил Пчеленок.

- Для чего? – спросил Лунтик.

- Для того, чтобы отмерить 4 литра, имея всего два сосуда емкостью 3 и 5 литров.

- Можно и мне попробовать? – спросил Лунтик.

Помогите Лунтику решить задачу. Считаем, что запасы меда в улье неограниченны.

а) Решите задачу с тремя сосудами размерами **3, 5** и **8** литров. Отмерить **4.** Подумайте, смогли ли вы найти оптимальный алгоритм? Перенесите программу в КуМир и выполните пошаговую реализацию.

б) Решите задачу с двумя сосудами. Для этого задайте размеры сосудов: **3, 5** и **0** литров. Отмерить **4.** Оптимальный алгоритм состоит из 6 команд.

Задание №В12. Решите задачу. *Как – то раз для проведения своего волшебства волшебнику Мерлину потребовалось отмерить ровно 7 литров воды из живого источника. Только вот задача: у него были сосуды ёмкостью 8 литров и 5 литров. Сможет ли волшебник справиться с таким заданием? Помогите вашему любимому герою решить эту задачу.*

Задайте размеры сосудов: 8, 5 и 0 литров. Отмерить 7. Постарайтесь найти:

а) алгоритм, состоящий из 8 команд;

б) оптимальный алгоритм. Посчитайте количество команд в вашем оптимальном алгоритме.

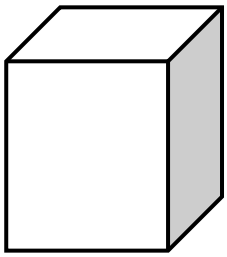


Рис.31. Куб.

Задание №Ч12. Запишите программу в тетрадь для рисования фигуры куба (рис.31) с помощью команд исполнителя Черепаха. При затруднении воспользуйтесь **Пультом.**

Задание №Ч13. Создайте с помощью Пульта Черепахи собственный рисунок. Перенесите программу в КуМир. Запустите на исполнение. Расскажите о трудностях, которые возникли в результате работы.

Задание №К15. Для Кузнечика задайте *новое задание*: **вперед 5, назад 2**. С помощью **Пульта** напишите программу, которая позволяет перекрасить точки, указанные в задании. Перенесите программу из **Пульта** в КуМир. Исправьте синтаксические ошибки, запустите программу.

а) точки 1, 3, 5; **б)** точки 0,2, 4, 6; **в)** все точки от 0 до 10.

Задание №К16. Задачи №К15 перепишите с использованием цикла.

Самостоятельное написание программ

Задание №Ч14. Код программы к заданию представлен на рисунке 32.

а) Определите, что нарисует Черепаха после выполнения программы?

б) Постройте блок – схему к программе

в) Перепишите программу **задание_Ч14** без использования конструкции цикла.

```
использовать Черепаха
алг задание_Ч14
нач
. влево (90)
. нц 2 раз
. . вперед (50)
. . вправо (90)
. . поднять хвост
. . вперед (50)
. . вправо (90)
. . опустить хвост
. кц
кон
```

Рис.32. Код программык **заданию №Ч14**

Задание №Ч15. Для следующих программ (рис.33):

а) определите тип алгоритма (*линейный* или *циклический*).

б) Нарисуйте рисунки, которые получит Черепаха после выполнения алгоритма.

в) Среди программ найдите те, которые дадут одинаковый результат.

а)

```
использовать Черепаха
алг Ч15_а
нач
. вперед (50)
. вправо (120)
. вперед (50)
. вправо (120)
. вперед (50)
. вправо (120)
кон
```

б)

```
использовать Черепаха
алг Ч15_б
нач
. вперед (50)
. назад (50)
. вправо (120)
. вперед (50)
. назад (50)
. вправо (120)
. вперед (50)
. назад (50)
. вправо (120)
кон
```

в)

```
использовать Черепаха
алг Ч15_в
нач
. нц 3 раз
. . вперед (50)
. . назад (50)
. . вправо (120)
. кц
кон
```

г)

```
использовать Черепаха
алг Ч15_г
нач
. нц 3 раз
. . вправо (120)
. . вперед (50)
. кц
кон
```

Рис. 33. Коды программ к заданию №Ч15.

г) Проверьте свои решения на компьютере.

Задание №Ч16. Для каждой из трех программ (рис.35).

а) Определите результат работы программы.

б) Определите команды тела цикла и количество повторений.

в) Постройте блок – схему на основе блок-схемы, представленной на рисунке 34.

г) Для проверки своих ответов каждую из программ перепишите на компьютер, выполните ее и посмотрите на результат работы.

д) Перепишите каждую из программ с использованием **цикла.** Запустите на выполнение. Убедитесь, что полученный рисунок не изменился.

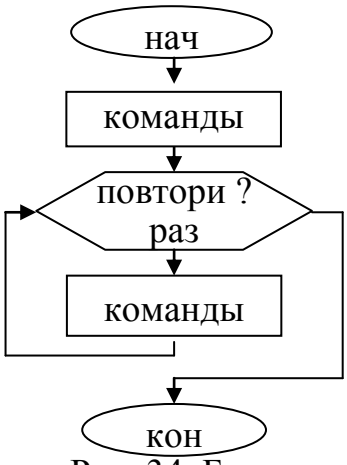


Рис. 34. Блок-схема к заданию №Ч16.

```
1.  использовать Черепаха
2.  алг программа_1
3.  нач
4.  . вперед (90)
5.  . назад (90)
6.  . вправо (90)
7.  . вперед (90)
8.  . назад (90)
9.  . вправо (90)
10.. вперед (90)
11.. назад (90)
12.. вправо (90)
13.. вперед (90)
14.. назад (90)
15.. вправо (90)
16. кон
```

```
1.  использовать Черепаха
2.  алг программа_2
3.  нач
4.  . поднять хвост
5.  . вправо (90)
6.  . назад (200)
7.  . опустить хвост
8.  . вперед (50)
9.  . поднять хвост
10.. вперед (50)
11.. опустить хвост
12.. вперед (50)
13.. поднять хвост
14.. вперед (50)
15.. опустить хвост
16.. вперед (50)
17.. поднять хвост
18.. вперед (50)
19. кон
```

```
1.  использовать Черепаха
2.  алг программа_3
3.  нач
4.  . вперед(50)
5.  . назад (50)
6.  . вправо(90)
7.  . поднять хвост
8.  . вперед (50)
9.  . влево (90)
10.. опустить хвост
11.. вперед(50)
12.. назад (50)
13.. вправо(90)
14.. поднять хвост
15.. вперед (50)
16.. влево (90)
17.. опустить хвост
18. кон
```

Рис. 35. Программы к заданию №Ч16.

Задание№Ч17. Рисование фигур с повторяющейся частью.
Напишите программу для рисования снежинок с рисунка 36:

а) **б)** **в)**

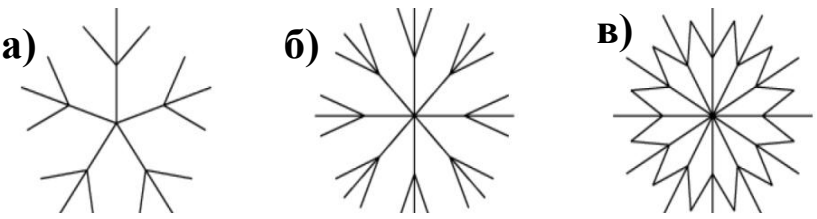


Рис. 36. Рисунки к заданию №Ч17. Снежинки.

Задание №Ч18. Придумайте сами фигуру с повторяющейся частью и напишите для нее программу.

Использование переменных

Переменная – это ячейка памяти компьютера, которая может принимать любое значение. *Ячейка* памяти имеет ***имя***. Его нужно задать, чтобы и компьютер, и программист понимали, к какой области памяти они обращаются.

Имя можно задать в КуМире ***латинскими или русскими*** буквами. Имя не может начинаться с цифры и содержать символы, кроме двух специальных знаков: **@** и **_**. Имена переменных не должны совпадать со служебными словами. *Латинские буквы компьютер пишет курсивом*.

Переменные могут быть **целыми** [**цел n**], **дробными (вещественными)** [**вещ n**], **литерными** [**лит n**], символьными [**сим n**] и логическими [**лог n**]. Компьютер должен знать, сколько памяти ему отвести под переменную. Для этого в начале программы переменную нужно *объявить*. Делают это после служебного слова **нач.** Пример смотри в программе **переменные** (рис.37)**.**

```
использовать Черепаха
алг переменные
нач
. цел n
. n:=4
. нц n раз
. . вперед (50)
. . вправо (360/n)
. кц
кон
```

Рис. 37. Код программы **переменные.**

Значение переменной можно присвоить в программе командой присваивания, например, **n:=4**. Здесь перед знаком равно стоят две точки. В программировании говорят: «**ячейке n присвоить 4**». Слово «равно» программисты употребляют обычно, когда что-либо сравнивают. Например, значения в ячейках **a** и **b**.

Дробная часть вещественного числа пишется через точку. Пример: **n:=2.5**

Вопрос 1. Прочитайте справочный мануал по языку КуМир в разделе «Имена, величины и выражения» и ответьте на вопрос: какие имена можно задать переменным? Ответ объясните.

а) Тегеран43; **б)** многоугольник; **с)** сторона; **д)** алг; **е)** Алг1.

Практическая работа №1.10. Использование переменных.

Цель: научиться объявлять переменные и использовать их в программе.

💻Задание 1. Использование переменных. Задание значений переменным в коде программы.

Найдите в коде программы строку, в которой задается значение переменной. Перепишите без ошибок код программы **переменные** (рис. 37) в КуМир. Запустите программу на выполнение.

Исправьте команду **n:=4** на **n:=6.** Посмотрите результат работы программы. Сделайте вывод.

💻Задание 2. Запрос значения переменной с клавиатуры.

В предыдущей программе вы могли изменять значение переменной **n** внутри самой программы. Теперь мы научимся вводить

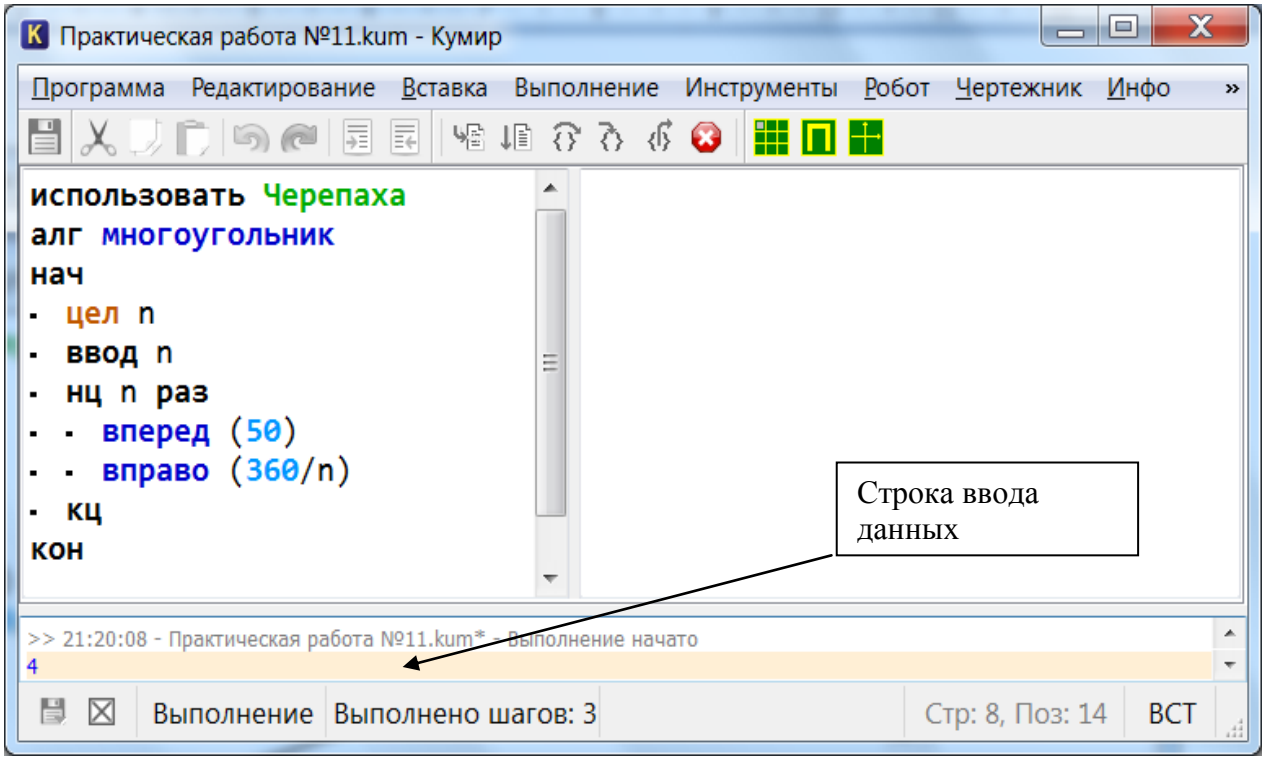


Рис.38. Строка ввода данных в языке КуМир

данные с клавиатуры. Запуская такую программу несколько раз, мы сможем менять **входные данные** и получать разные рисунки.

Перепишите код программы **многоугольник** (рис. 38) в систему КуМир и запустите ее. Обратите внимание, после запуска программы внизу появляется строка для ввода данных (рис.38). Введите число ***4*** и нажмите **Enter**. Посмотрите, что получилось.

Теперь запустите программу еще раз, введите число ***5*** и посмотрите, что получилось. Запустите программу еще несколько раз, изменяя числа при вводе. Сделайте вывод на основе эксперимента.

Вопрос 2. Зачем нужны переменные? Как удобнее задавать значения переменных в самой программе или с клавиатуры?

Задание 3. Пояснения к вводу переменных.

К **вводу** переменных нужно писать пояснение. Исправьте код

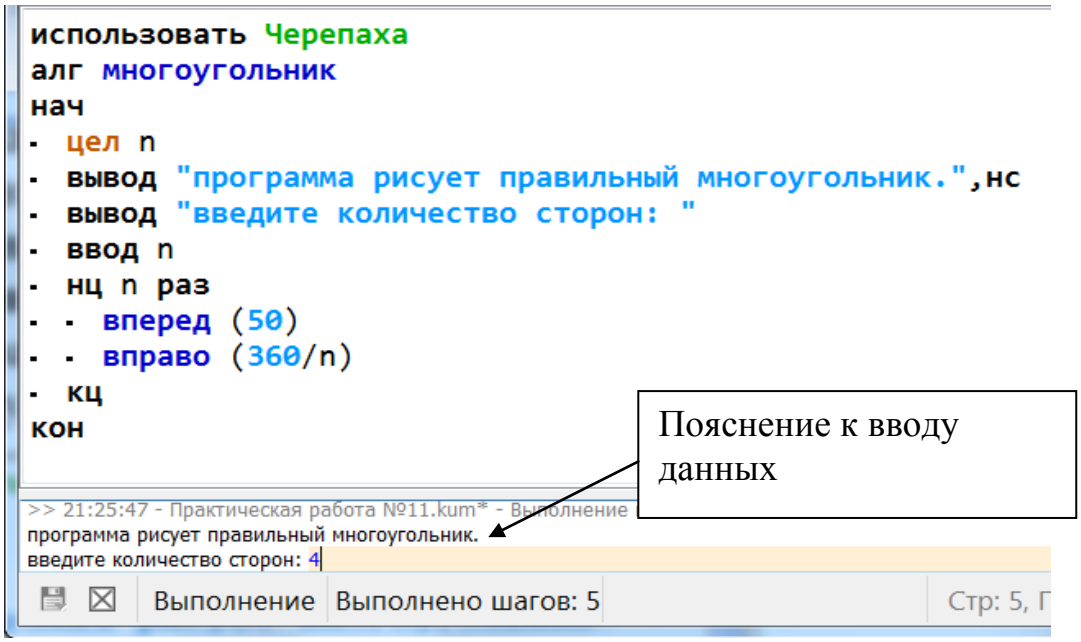


Рис.39. Пояснение к вводу данных

программы **многоугольник**, как показано на рисунке 39 и запустите ее. Сделайте вывод о необходимости писать пояснения к вводу данных в программу. Запишите вывод в тетрадь или обсудите проблему написания пояснений в классе.

Задание №Ч19. Исправьте код программы **многоугольник** так, чтобы Черепаха рисовала Снежинку, а количество лучей задавалось с клавиатуры.

Подпрограммы

Подпрограмма – это участок кода компьютерной программы, имеющий имя. Разберем использование **подпрограмм** на примере.

Поставим задачу: написать программу для рисования **цветка**, состоящего из трех квадратов, равномерно распределенных по кругу (рис.40).

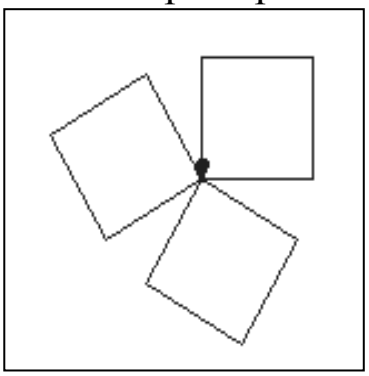


Рис.40. Цветок из трех квадратов

Вспоминаем, что в окружности **360** градусов. Значит, после каждого квадрата необходимо задать угол поворота в **120** градусов.

Словесно можно сформулировать алгоритм так: «рисуем квадрат, поворачиваем на 120 градусов. Повторяем так три раза». Без вспомогательной программы этот алгоритм можно реализовать так, как вы видите в коде программы **цветок1** (рис. 41).

Запишем этот же алгоритм с помощью подпрограмм. Подпрограмму для рисования квадрата назовем ***многоугольник*** и запишем ее как вспомогательную.

В основной программе запишем цикл, в котором 3 раза вызовем процедуру *многоугольник* и сделаем поворот вправо на 120 градусов.

```
использовать Черепаха
алг цветок1
нач
. нц 4 раз
. . вперед (50)
. . вправо (90)
. кц
. вправо (120)
. нц 4 раз
. . вперед (50)
. . вправо (90)
. кц
. вправо (120)
. нц 4 раз
. . вперед (50)
. . вправо (90)
. кц
. вправо (120)
кон
```

Рис. 41. Код программы **цветок1**

Практическая работа №1.11. Подпрограммы.

Цель: практически реализовать программу, использующую вспомогательный алгоритм.

Задание 1. Запишите в точности код программы **цветок2** (рис.42) в систему КуМир.

Запустите программу на исполнение. Убедитесь, что она работает. Она должна нарисовать три квадрата с поворотом каждого на 120° (рис.40).

```
использовать Черепаха
| начало основной программы
алг цветок2
нач
. нц 3 раз
. .многоугольник
. . вправо (120)
. кц
кон
| начало вспомогательной
| программы
алг многоугольник
нач
. нц 4 раз
. . вперед (50)
. . вправо (90)
. кц
кон
```

Рис. 42. Код программы **цветок2.**

Задание 2. В ***основной*** программе исправьте количество **циклов – 5, градус поворота -72**. Должен получиться рисунок из пяти квадратов, расположенных равномерно по кругу (рис.43).

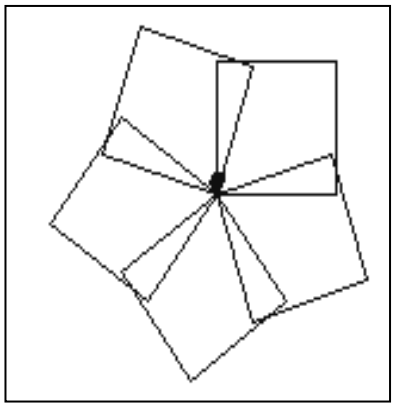


Рис. 43. Цветок из пяти квадратов.

Вопрос 1. В чем преимущество подпрограмм?

Задание №Ч20. Исправьте код программы так, чтобы получился:

а) цветок из шести квадратов;

б) цветок из 8 квадратов;

в) цветок из 10 квадратов.

Помните, что в окружности **360** градусов. Распределите многоугольники равномерно по кругу.

Задание №Ч21. Напишите вспомогательную программу **шестиугольник**. Нарисуйте цветок из 5 лепестков шестиугольника.

Задание №Ч22. Определите, какой многоугольник используется для лепестка и сколько лепестков в цветке по рисунку. Напишите программу, повторяющую рисунок (рис.44).

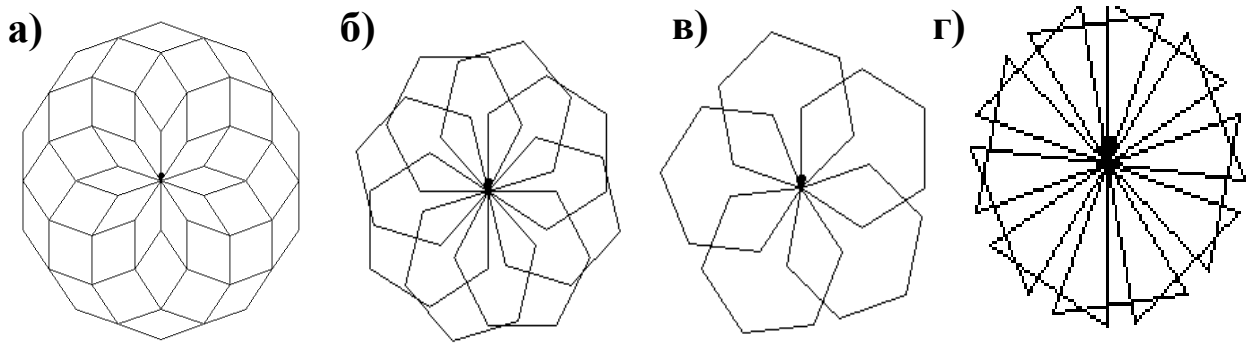


Рис. 44. Рисунок к заданию №Ч22.

Задание №Ч23. Напишите вспомогательный алгоритм лесенка. С помощью вспомогательного алгоритма напишите программы для следующих рисунков (рис. 45):

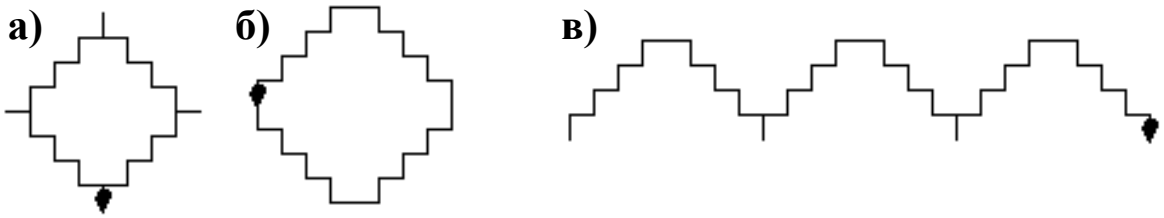


Рис. 45. Рисунки к заданию №Ч23.

Задание №Ч24*. С помощью вспомогательной программы нарисуйте следующую фигуру, состоящую из пяти квадратов (рис. 46):

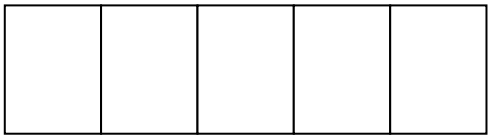


Рис. 46. Задание №Ч24.

Задание №Ч25*. Замостите площадь с помощью правильного многоугольника. Подумайте, какие многоугольники могут подойти для решения этой задачи.

Случайное число

В любом языке программирования есть **функция** получения случайного числа. Рассмотрим ее использование на примере задачи: пусть Черепаха нарисует 10 снежинок, случайным образом расположив их на поле. В КуМире есть ограничение на выход за границы поля, поэтому будем располагать рисунки поближе к центру.

Совестно алгоритм можно сформулировать так: рисуем Снежинку, поворачиваем Черепаху на *случайный* угол, перемещаем вперед на *случайное* число. Так повторяем 10 раз.

Применим команду получения случайного числа ***rnd(100),*** которая выдает ***дробное*** число от 0 до 99. ***Если*** в начале программы переменные **a** и **b** объявлены как целые, то нужно привести дробное число к целому. Это можно сделать, записав

снежинки10.kum - Кумир

Программа Редактирование Вставка Выполн

```
1  использовать Черепаха
2  алг снежинки10
3  нач
4  . цел a
5  . цел b
6  . цел c
7  . поднять хвост
8  . нц 10 раз
9  . . a:=int(rnd(100))
10 . . b:=int(rnd(90))
11 . . вперед (a)
12 . . вправо (b)
13 . . снежинка
14 . кц
15 кон
16 алг снежинка
17 нач
18 . нц 6 раз
19 . . опустить хвост
20 . . вперед (4)
21 . . назад (4)
22 . . вправо (60)
23 . . поднять хвост
24 . кц
25 кон
```

Рис. 47. Код программы **снежинки10.kum**.

команду: **a:=int(rnd(90)).**Это означает, что от дробного *случайного* числа мы *берем целую часть* ***int*** и присваиваем ячейке **a** это значение.

Ключевое слово ***int*** происходит от английского *integer* – целый. Функция ***int*** не округляет число по правилу математического округления, а просто берет его целую часть.

Практическая работа № 1.12. Случайное число.

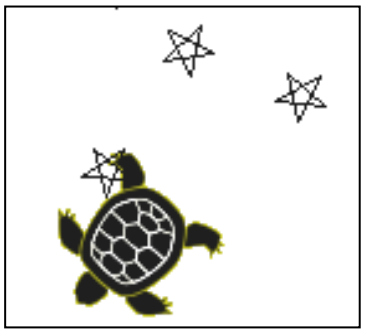


Рис. 48. Звездочки.

Цель: на примере программы ознакомиться с тем, как компьютер получает случайные числа.

💻**Задание 1.** Запишите программу **снежинки10.kum** (рис. 47) в среде программирования КуМир. Запустите программу на исполнение. Убедитесь, что программа работает.

💻**Задание 2.** Исправьте программу **снежинки10**, так чтобы она рисовала: **а)** снежинки из пяти лучиков; **б)** поворачивала бы на угол от 0° до 45°; **в)** рисовала звездочки (рис. 48). Тело цикла для пятиконечной звездочки: [**вперед 8, вправо 144].**

Разветвляющиеся алгоритмы

Практическая работа №1.13. Разветвляющиеся алгоритмы.

Цель**:** познакомиться с алгоритмической конструкцией ветвления.

Решим следующую задачу: пусть Черепаха нарисует **снежинку** или **многоугольник** по нашему выбору, сделанному с клавиатуры. Для выбора рисунка введем переменную ***с***. Пусть для рисования снежинки ***с*=1**, а для рисования многоугольника ***с*=2.** Составим ***блок – схему*** к программе (рис.49).

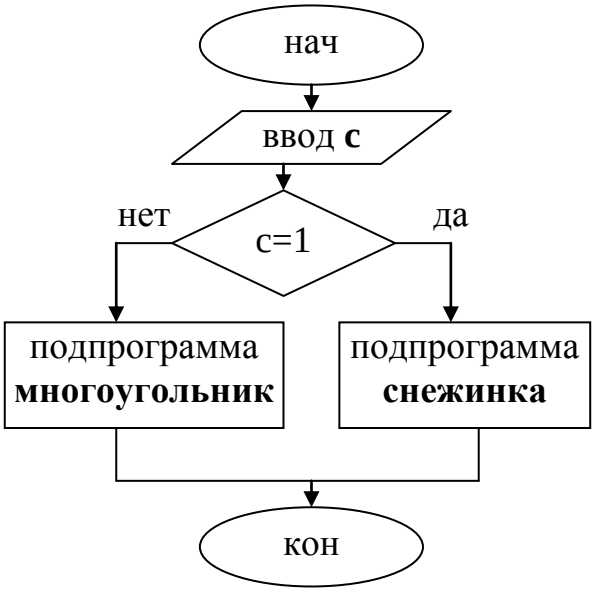


Рис. 49. Блок – схема к п.р. №1.13

💬**Вопрос 1.**Посмотрите на код программы **рисунок на заказ**. Как реализована алгоритмическая конструкция ***«условие»?***

Задание1.Реализуйте программу **рисунок на заказ** (рис.50) на компьютере. Допишите процедуру **многоугольник** самостоятельно.

```
использовать Черепаха
алг рисунок на заказ
нач
. цел с
. вывод "введине 1, если будем рисовать снежинку, ",нс
. вывод "введите 2, если будем рисовать многоугольник: ",нс
. ввод с
. если с=1 то
. . . снежинка
. . иначе
. . . многоугольник
. все
кон
алг снежинка
нач
. нц 6 раз
. . вперед (60)
. . назад (60)
. . вправо (60)
. кц
кон
```

нс – новая строка

Рис. 50. Код программы **рисунок на заказ**.

Вопрос 1. Что выдаст программа, если ввести **с**=5?

Задание2. Объявите переменную ***с*** литерной: **лит с**. Исправьте условие запроса и пояснение к вводу данных с клавиатуры. Условие запроса исправьте так: **если с="снежинка".**

Вопрос 2. Переменные какого типа (целые или литерные) легче использовать для ввода данных?

Длина пути для Черепахи

При рисовании фигуры Черепаха проходит некоторый путь. Поставим задачу подсчета длины всего пути, который пройдет Черепаха при выполнении программы. Рассмотрим задачу рисования квадрата. Без цикла эта программа выглядит так (см. алгоритм **путь1,** рис 51):

```
использовать Черепаха
алг путь1
нач
. вперед (50)
. вправо (90)
. вперед (50)
. вправо (90)
. вперед (50)
. вправо (90)
. вперед (50)
. вправо (90)
кон
```

Рис. 51. Код программы **путь1.**

Посчитаем, сколько единиц прошла Черепаха? Введем в программу переменную ***S***, которая будет хранить значение пройденного

пути. Команда **вперед (50)** добавляет к пройденному пути 50 единиц.

Вопрос 1. Сколько единиц к пройденному пути добавляет команда **вправо (90)?**

Посчитаем пройденный путь **S:=50+50+50+50**. То есть Черепаха прошла путь в **200** единиц.

Рассмотрим программу рисования квадрата с использованием цикла (см. алгоритм **путь2,** рис. 52). Добавим в программу целую переменную ***S*** для подсчета пройденного пути. Будем увеличивать значение этой ячейки на 50 единиц после каждой команды **вперед (50).** Код программы **путь3** представлен на рисунке 53.

```
использовать Черепаха
алг путь2
нач
. нц 4 раз
. . вперед (50)
. . вправо (90)
. кц
кон
```

Рис. 52. Код программы **путь2.**

```
1.  использовать Черепаха
2.  алг путь3
3.  нач
4.  . цел S
5.  . нц 4 раз
6.  . . вперед (50)
7.  . . S:=S+50
8.  . . вправо (90)
9.  . кц
10. . вывод S
11. кон
```

Рис. 53. Код программы **путь3.**

Важно! Команда **S:=S+50** означает, что в *оперативную память* занесли значение переменной **S**, выполнили действие **S+50**. Полученный результат записали в ячейку **S**. На рисунке 50 представлена схема выполнения команды присваивания.

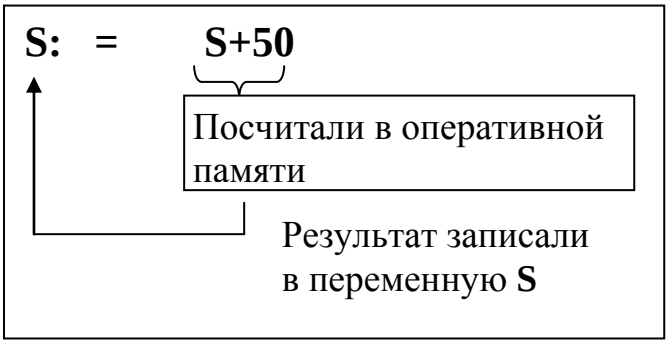


Рис. 54. Схема выполнения команды присваивания.

Задание 1. Запустите программу **путь3** на исполнение.

Как видно, программа не была выполнена до конца и выдала ошибку: «неопределенное значение» в строке 7 (рис.53).

Это произошло потому, что компьютер «не понял» к какому значению переменной ***S*** нужно прибавлять 50 первоначально.

Правило: любой переменной нужно присвоить начальное значение!

Вопрос 1. По коду программы **путь3** на рисунке 49 определите номер строки, после которой в программе нужно вставить команду **S:=0,** позволяющую задать начальное значение переменной ***S***?

Блок – схема программы для подсчета пути Черепахи представлена на рисунке 55.

Как мы знаем, ***блок – схема*** это наглядное представление алгоритма. По *блок – схеме* можно определить, в какой последовательности будут выполняться команды.

Вспомним, что часть команд, которая выполняется несколько раз, называется ***телом цикла***, а то, *сколько* раз будут выполняться команды – ***условием цикла***.

Согласно правилам записи блок – схем, **команды** записываются в ***прямоугольные блоки***, **условие** выполнения цикла в ***шестигранник***, а **вывод** и **ввод** в ***параллелограмм***. **Начало** и **конец** на *блок – схеме* обозначаются словами **нач** и **кон** и обводятся в ***овал***.

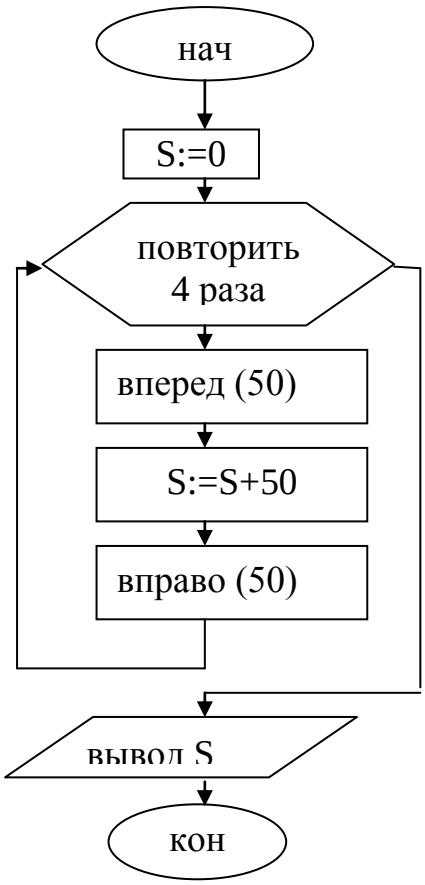


Рис. 55. Блок-схема к программе **путь3**.

Практическая работа №1.14. Длина пути Черепахи.

Цель и задачи работы определите самостоятельно.

Задание 1.Запишите исправленный код программы **путь3**, подсчитывающий длину пути Черепахи на компьютер (рис.53).

Задание 2. Исправьте программу **путь3** для рисования и подсчета длины пути:

а) правильного треугольника;

б) снежинки из трех лучей. Для снежинки путь, пройденный туда и обратно, сложите.

Цикл внутри цикла

Практическая работа №1.15. Цикл внутри цикла.

Цель: познакомиться с алгоритмом, использующим два вложенных друг в друга цикла.

Поставим **задачу**: нарисовать цветок из **5** лепестков, формой которого является шестиугольник со стороной **50** единиц. Посчитать пройденный Черепахой путь.

Запишем алгоритм с помощью блок- схемы (рис.56). На схеме хорошо видно, что цикл рисования лепестка в виде шестиугольника **вложен** в цикл рисования пяти лепестков, равномерно распределенных по кругу с градусом поворота равным **360/5.**

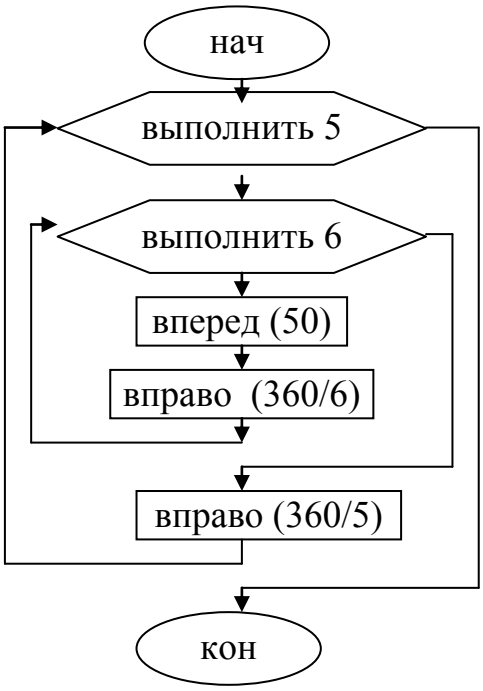


Рис.56. Блок-схема алгоритма **цветок3.**

Вопрос 1. Посчитайте, какой путь проходит Черепаха, рисуя один лепесток? Рисуя цветок из пяти лепестков?

Вопрос 2. Посмотрите на код программы **цветок3** (рис.57). Сколькими точками отделено тело внутреннего вложенного цикла? Тело внешнего цикла?

Задание 1. Наберите код программы и убедитесь, что она работает.

Задание 2. В код программы **цветок3** (рис.53) вставьте три команды: **вывод S**, **S:=S+50**, **S:=0**, так чтобы программа выводила пройденный Черепахой путь.

Задание 3. Исправьте код программы **цветок3** для рисования цветка из шести лепестков квадратов, стороны которых 50 единиц. Найдите длину пути Черепахи при рисовании цветка.

Замечание: если вы получили в ответе число: S= 6*(4*50)=1200, то ваша программа работает правильно.

Задание 4. Посчитайте пройденный путь при рисовании цветка из 10 десятиугольников.

```
использовать Черепаха
алг цветок 3
нач
цел S
. нц 5 раз
. . нц 6 раз
. . . вперед (50)
. . . вправо (360/6)
. . кц
. . вправо (360/5)
. кц
кон
```

Рис. 57. Код программы **цветок3**

Исполнитель Чертежник

Что умеет Чертежник? Возможно, вы уже знакомы с понятием координат на плоскости. В математике используется так называемая декартова система координат (рис.58), в которой положение каждой точки определяется двумя числами – координатами: **X** и **Y**.

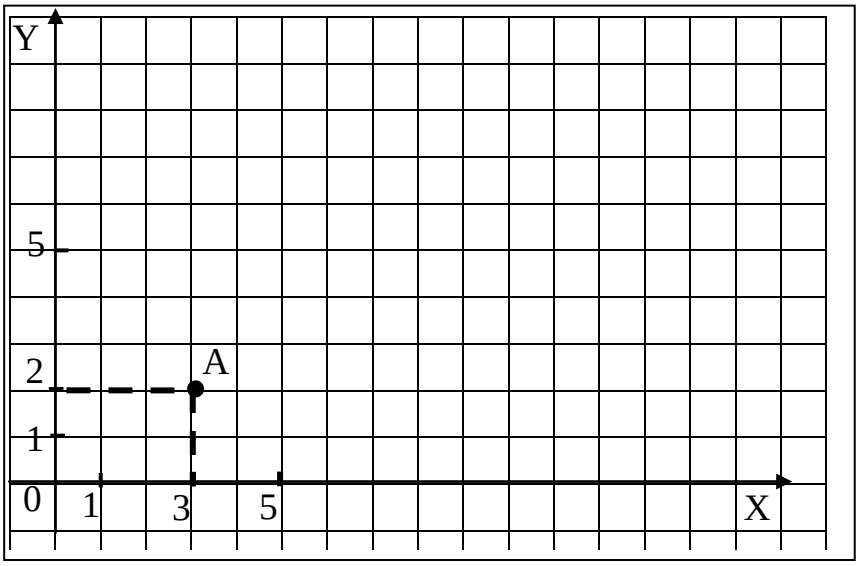


Рис.58.Декартоваясистема координат

Точка А имеет координаты по оси **X** – 3 (ось абсцисс), по оси **Y** – 2 (ось ординат). Записывают так: **A(3,2).** Первую координату точки отмеряют по оси **X**, а вторую – по оси **Y.** На координатной плоскости обязательно отмечают единичный отрезок и по оси **X** и по оси **Y**. Заметим, что эти единичные отрезки могут быть разными.

Вопрос 1. Приходилось ли вам отыскивать точку на плоскости? Возможно, вы искали место в кинозале? Можно ли сравнить поиск места в зале с координатами точки на плоскости?

Вопрос 2. Вспомните, где и с какими графиками вы встречались в жизни. Предложите ситуацию, в которой будут использоваться разные отрезки по оси абсцисс и по оси ординат.

Практическая работа №1.16. Окно Чертежника.

Цель: изучить систему команд и окно исполнителя Чертежник.

Система команд исполнителя Чертежник включает 6 команд:

- опустить перо
- поднять перо
- сместиться в точку (вещ x, y)
- сместиться на вектор (вещ dX, dY)
- установить цвет (лит цвет)
- надпись (вещ ширина, лит текст)

Допустимые цвета: ”черный”, ”белый”, ”красный”, ”оранжевый”, ”желтый”, ”зеленый”, ”голубой”, ”синий”, ”фиолетовый”.

В программе **команды_Чертежника** (рис.59) записан образец их использования.

```
использовать Чертежник
алг команды_Чертежника
нач
. вещ x,y,dx,dy
. x:=1; y:=2; dx:=3; dy:=4
. установить цвет ("черный")
. сместиться в точку (1,1)
. надпись (1,"A(1,2)")
. поднять перо
. сместиться в точку (x,y)
. опустить перо
. сместиться на вектор (dx,dy)
. надпись (1,"B(4,6)")
кон
```

Рис.59. Окно программы **команды_Чертежника**

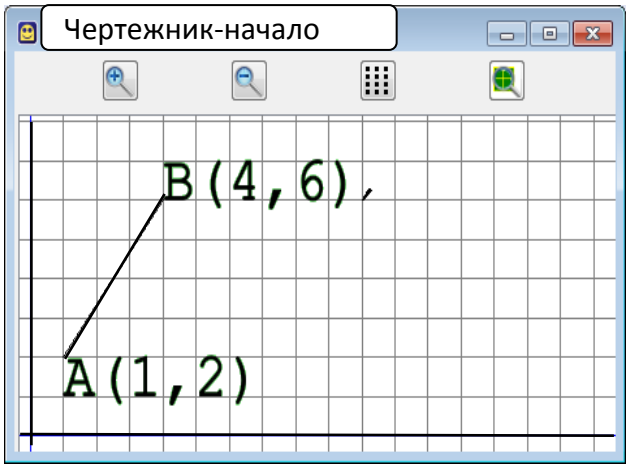


Рис.60. Окно Чертежника

Вопрос 1. Какие кнопки на панели окна КуМир (рис.60) вам знакомы?

Вопрос 2. Укажите кнопку, позволяющую показать окно Чертежника (рис. 205).

Задание 1. Подпишите назначение кнопок в окне Чертежника (рис.61).

1--  . 2 -  . 3 -  . 4 - 

Рис. 61. Кнопки окна Чертежника.

Вопрос 3. По коду программы **команды Чертежника.kum** ответьте на вопросы.

а) Назовите координаты точки, из которой Чертежник начинал движение?

б) Назовите координаты точки, с которой исполнитель начал оставлять видимую линию?

в) Определите, скольким клеткам равен единичный отрезок по оси **X** и по оси **Y**?

г) В какой точке находится перо Чертежника по окончании программы?

д) Как Чертежник выполнил команду: **сместиться на вектор (3,4)?**

е) Какими должны быть значения **a** и **b** в команде **сместиться на вектор (a,b)** для того чтобы Чертежник из точки **B(4,6)** сместился в точку **A(1,2)?** В начало координат?

Практическая работа №1.17. Рисование по точкам.

Цель: научиться составлять программу для чертежа по заданным координатам точек.

Задание 1. Даны координаты точек на плоскости. Выполните чертеж по координатам точек на листе в клеточку (рис. 62). Соедините точки в заданной последовательности. Помните, первое число отмеряется по оси ***OX***, второе – по оси ***OY***.

A(3,1), B(1,3), C(7,3), D(5,6), E(8,11),F(8,3),G(14,3),H(14,1).

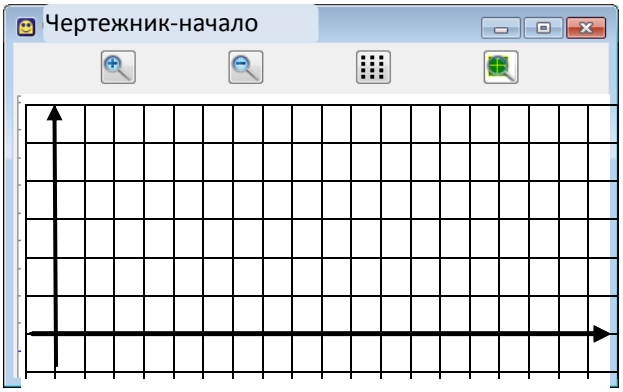


Рис.62. Клетчатое поле.

Соедините точки **A** и **H** так, чтобы контур рисунка стал непрерывным.

Вопрос 1. Посмотрите на код начала программы **парусник** на рисунке 63. Почему команда **опустить перо** стоит после того, как Чертежник сместился в точку (3,1)?

Задание 2. Запишите программу **парусник** в среде КуМир.

Допишите код программы (рис. 63) для рисования всего рисунка практической работы 1.17 задания 1. Соединяйте точки в заданной последовательности. Для завершения рисунка соедините точки начала и конца чертежа так, чтобы контур был непрерывным.

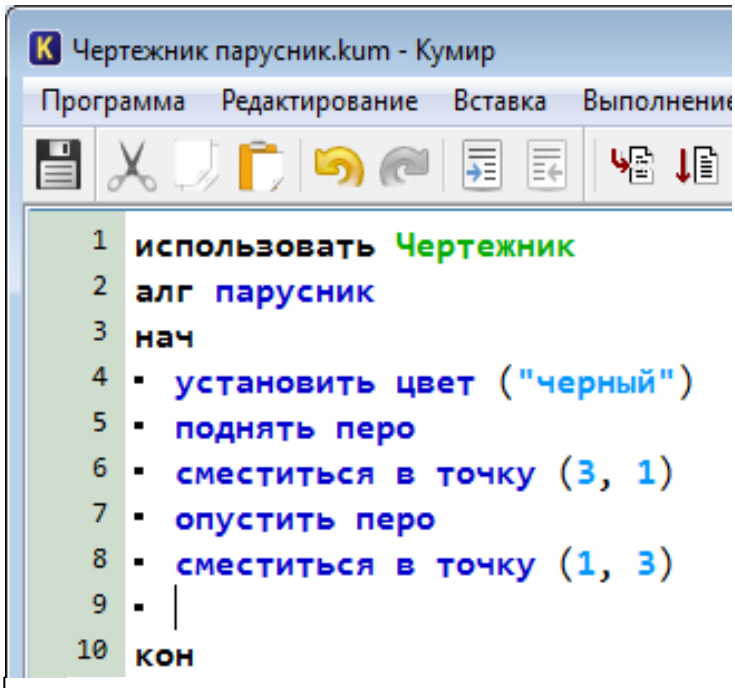


Рис. 63. Код программы **парусник**.

Задание №Чт1. Даны координаты точек на плоскости. Выполните чертеж по координатам точек на листе в клеточку. Соедините точки в заданной последовательности. Помните, первое число – по оси ***OX***, второе – по оси ***OY***. Проверьте свой чертеж на компьютере с помощью программы. Программу напишите самостоятельно.

A(3,0), B(1,2), C(1,3), D(4,3), E(5,2), F(6,2), G(7,4), H(8,4), I(8,8), J(9,8), K(9,4), L(10,4), M(10,2), N(11,2), O(12,3), P(14,3), Q(12,0).

Соедините точки **A** и **Q** так, чтобы контур рисунка стал непрерывным.

Задание №Чт2. Самостоятельно напишите программы для рисунков парусника и ракеты (рис. 64 а, б).

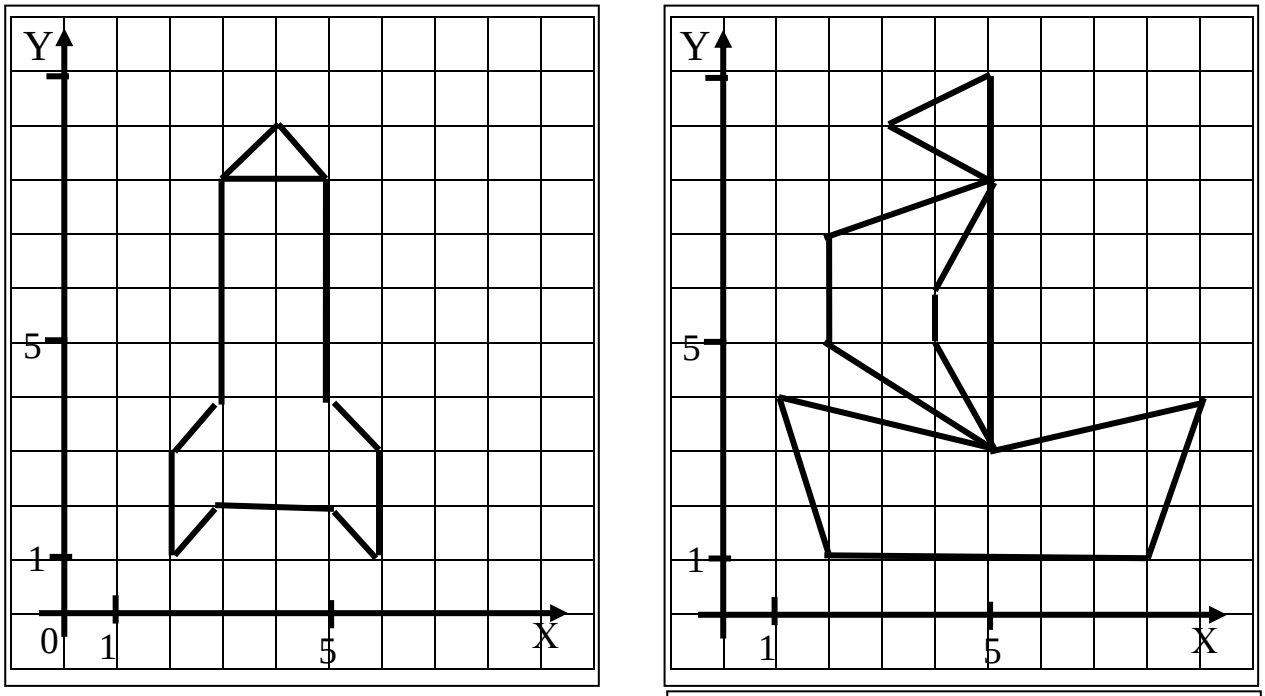


Рис.64. а) Ракета. б) Парусник

Задание №Чт3. Начертите в тетради свой рисунок в декартовых координатах и напишите к нему программу.

Практическая работа №1.18. Рисование с помощью векторов.

Цель: научиться составлять программы с использованием команды «**сместиться на вектор**».

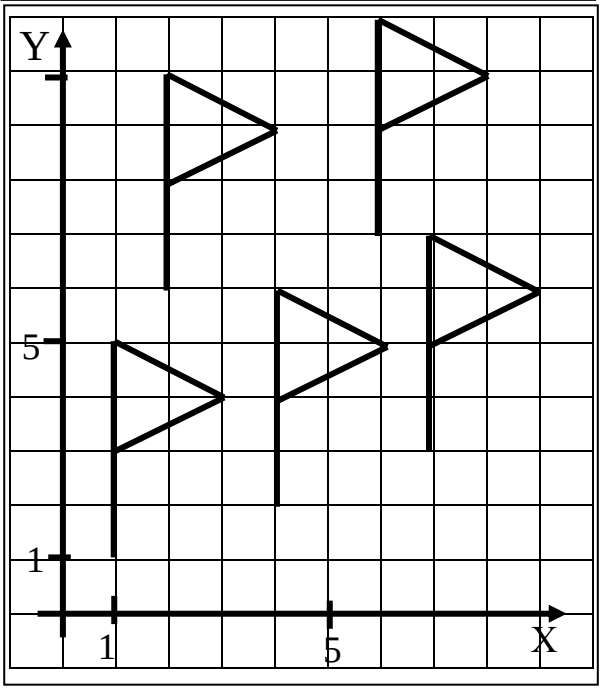


Рис. 65. Поле с флажками.

Напишем программу для следующей задачи. На поле расположены несколько флажков одинакового размера (рис.65). Напишем программу для этого рисунка.

Вопрос 1. Предложите свою идею: что бы могли отмечать флажками на поле?

Вы уже знакомы с понятием вспомогательного алгоритма. С использованием подпрограммы **флажок** код основной программы **поле** для трех флажков мог бы выглядеть, например, так как это представлено на рисунке 66.

```
использовать Чертежник
алг поле
нач
. поднять перо
. сместиться в точку (1,1)
. флажок
. сместиться в точку (5,2)
. флажок
. сместиться в точку (9,3)
. флажок
кон
```

Рис. 66. Код программы **поле.**

Вопрос 2. Предположите, почему в коде основного алгоритма команда **флажок** подчеркнута?

Напишем подпрограмму **флажок** с использованием команды **сместиться на вектор (*a*,*b*)**.

Вопрос 3. По коду подпрограммы **флажок** (рис.67) определите, в какой последовательности будут нарисованы фрагменты (*a*, *b*, *c*) флажка, представленного на рисунке 68?

```
алг флажок
нач
. опустить перо
. сместиться на вектор (0,4)
. сместиться на вектор (3,-1)
. сместиться на вектор (-3,-1)
. поднять перо
кон
```

Рис. 67. Код подпрограммы

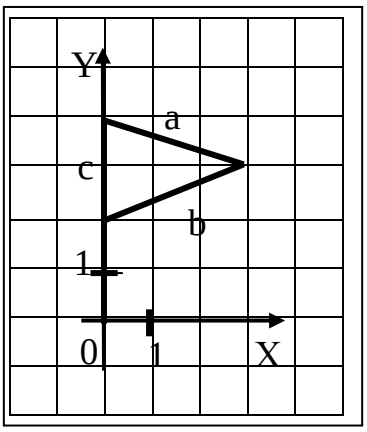


Рис. 68. Флажок.

Задание 1. Наберите код программы **поле** и подпрограммы **флажок**. Запустите программу, убедитесь, что она работает.

Задание 2. Измените код программы **поле** (рис.66) так, чтобы она рисовала все пять флажков, расположенных на поле (рис.65).

Задание 3. Измените код подпрограммы **флажок** так, чтобы она рисовала флажки другой формы (рис. 69).

Задание 4. На месте флажков «посадите» Ёлочки (рис. 70).

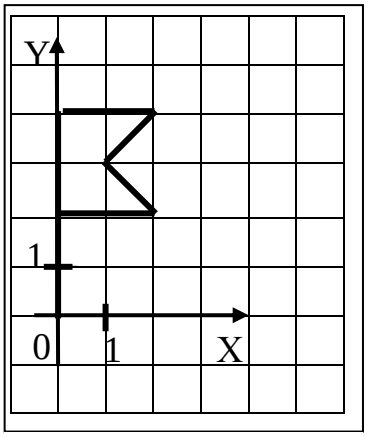


Рис. 69. Флажок другой формы.

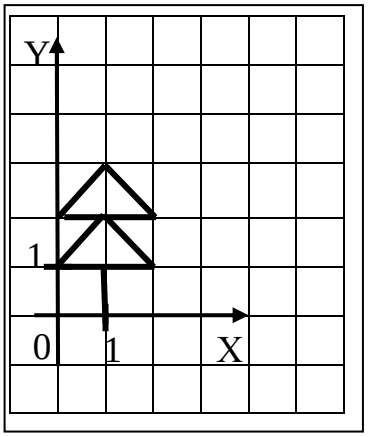


Рис. 70. Ёлочка

Задание №Чт4. С помощью подпрограммы и цикла нарисуйте следующий орнамент (рис.71).

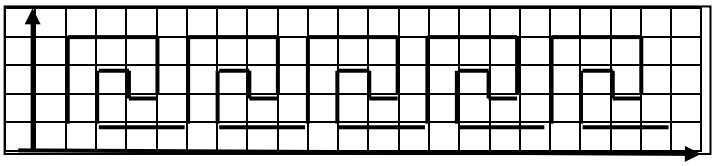


Рис. 71. Орнамент

Указание. Сначала напишите подпрограмму для рисования одной повторяющейся части рисунка.

Задание №Чт5. Напишите программы для фигур на рисунке 72.

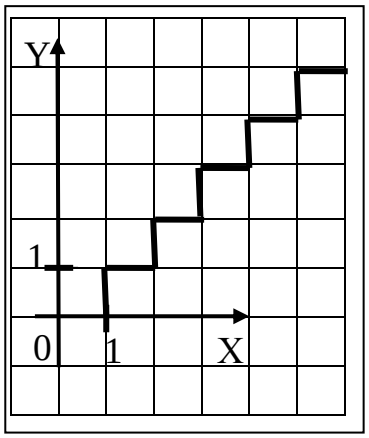


Рис. 72 а. Лесенка.

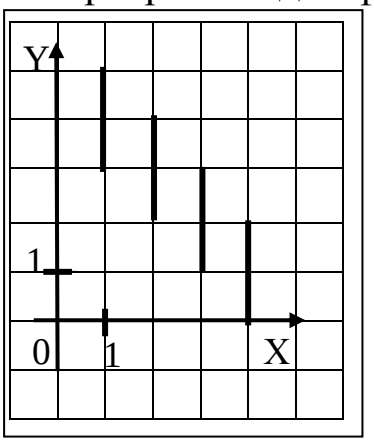


Рис. 72 б. Столбики.

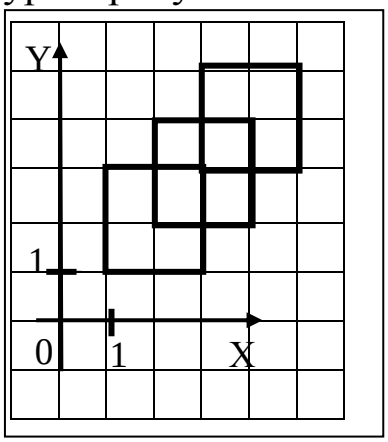


Рис. 72 в. Квадратики

Задание Чт6. Придумайте свой рисунок и напишите программу. Ответьте на вопрос: есть ли в вашей программе повторяющиеся части, циклы или подпрограммы?

Раздел 3. Расчетные графические задания

Ёлочка

Нарисуйте ёлочку на листе в клетку. Измерьте углы поворота для **Черепахи.** Определите значения для команд **вперед**и **вправо**. Поле Черепахи имеет размеры примерно 400 на 400 точек.

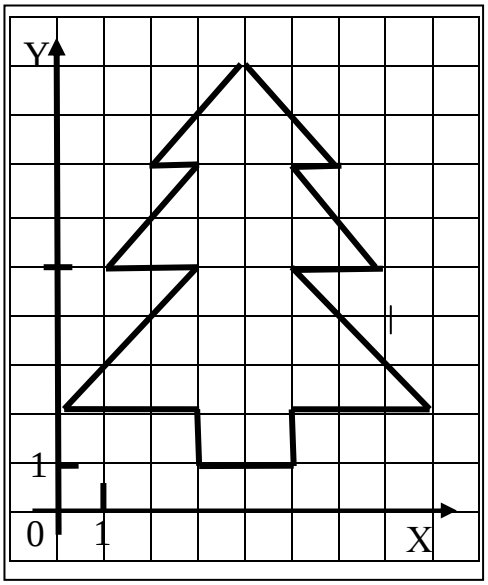


Рис. 73. Ёлочка.

Задание. а) Напишите программу для рисования Ёлочки (рис.73) и проверьте ее на компьютере. Обязательно напишите программу так, чтобы Ёлочка была симметричной, то есть левая и правая часть отражалась зеркально относительно ствола.

б) Напишите программу для рисования этого же рисунка для исполнителя **Чертежник**. Сравните возможности исполнителей.

Орнамент

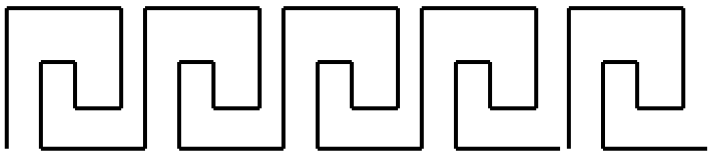


Рис. 74. Орнамент.

Нарисуйте в тетради в клетку орнамент (рис. 74) и напишите программу (для исполнителя Черепаха) для его рисования. Проверьте программу на компьютере. При затруднении воспользуйтесь **Пультом.**

Придумайте и нарисуйте свой орнамент и напишите для него программу. Исполнителя выберите самостоятельно.

Звезда

Измерьте углы пятиконечной звезды (рис.75). Подпишите углы на чертеже. Рассчитайте углы поворота для **Черепахи.** Напишите программу для рисования пятиконечной звезды.

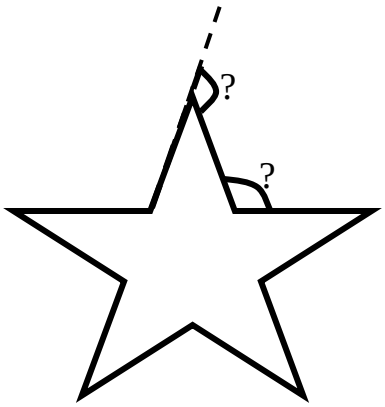


Рис. 75.Звезда.

Напишите программу для рисования звезд с нечетным количеством вершин. Используйте цикл с количеством повторений k и углом поворота (180-180/k).

Самый длинный луч

Нарисуем снежинку, лучи которой будем получать случайным образом с помощью датчика случайных чисел. Поставим задачу: найти самый длинный луч для этой снежинки.

На рисунке 76 представлена блок- схема алгоритма. Запишем программу для рисования снежинки (рис.77).

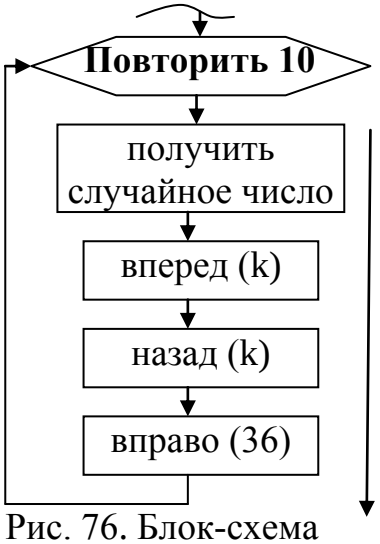


Рис. 76. Блок-схема алгоритма **лучи**.

```
использовать Черепаха
алг лучи
нач
. вещ k
. нц 10 раз
. . k:=rnd(50)+20
. . вперед (k)
. . назад (k)
. . вправо (360/10)
. кц
кон
```

Рис. 77. Код программы **лучи**.

Функция ***rnd(1)*** позволяет получить случайное число от 0 до 1. Например, ***rnd(50)*** *получает случайное вещественное число от 0 до 50.*Заметим, что крайние значения 0 и 50 маловероятны.

Задание 1. Посмотрите на программу **лучи** и ответьте на вопросы:

а) какая команда в программе определяет длину луча?

б) в каком диапазоне чисел получаются значения для длины луча?

Исправим эту программу так, чтобы она ***запоминала*** длину самого большого луча и выводила её на экран.

Для этого нам потребуется переменная, в которой будет храниться самое большое значение длины луча. Назовем ее **m**.

Словесно алгоритм можно сформулировать так: если полученное компьютером *случайное значение* больше того,

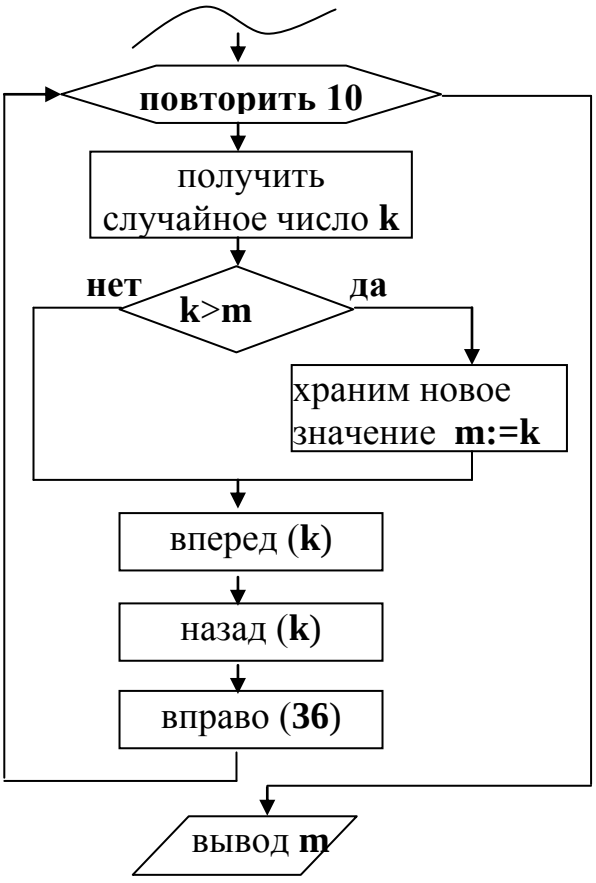


Рис. 78. Блок-схема алгоритма **самый длинный луч**.

которое хранится в переменной **m**, то значение переменной **m** заменим на новое, бóльшее значение. Будем рисовать следующий луч. На рисунке 78 представлена блок – схема алгоритма.

Остался вопрос: чему должно быть равно **m** в начале программы? Выберите ответ из вариантов:

а) максимальному числу, которое может встретиться **(60);**

б) минимальному (**20);**

в) нулю.

Ответьте на этот вопрос самостоятельно. При затруднении обратитесь к учителю.

Программу для нахождения размера самого длинного луча напишите самостоятельно. Правильное написание команды ветвления смотрите в справочнике КуМир или в практической работе №1.13 «**Разветвляющиеся алгоритмы**».

Эксперимент с датчиком случайных чисел.

Запишем программу для нахождения самого длинного луча, изменив количество лучей, например, на 100. С помощью этого приема определим, каким может быть самое большое число, полученное по формуле: **k:=rnd(50)+20**

```
использовать Черепаха
алг случайный луч
нач
. вещ k, m
. m:=0
. нц 100 раз
. . k:=rnd(50)+20
. . вперед (k)
. . назад (k)
. . вправо (360/10)
. . если k>m то m:=k все
. кц
. вывод (m)
кон
```

Рис. 79. Код программы **случайный луч**

Запустите программу **случайный луч** (рис. 79) несколько раз и запишите самое большое число, которое смог получить компьютер по этой формуле.

Задание №Ч26. Измените программу так, чтобы компьютер искал самый короткий луч снежинки.

Задание №Ч27. Напишите программу для поиска средней длины луча снежинки.

Задание №Ч28. Напишите программу для рисования бенгальских огней (рис. 80). Длины лучей получите с помощью датчика случайных чисел.

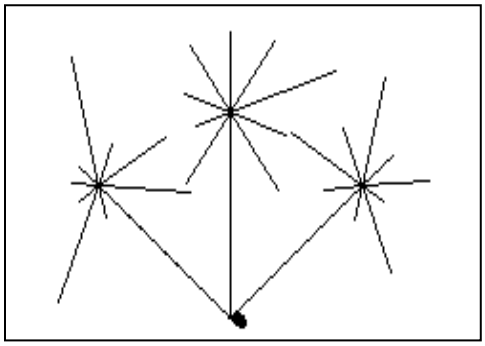


Рис. 80. Бенгальские огни.

Передача значения переменной в процедуру

Запишем программу **цветок4**, которая будет рисовать цветок из лепестков многоугольника.

Количество лепестков **L** и ***количество*** сторон многоугольника **M** будем вводить с клавиатуры. Для этого при **объявлении** *вспомогательного алгоритма- процедуры* укажем, что у него есть **аргумент**:

алг многоугольник (арг цел M)

Вызывать *процедуру из основной программы* будем так:

многоугольник (M)

💻**Запустите** программу **цветок4** (рис.81) на компьютере и нарисуйте цветок из 5 лепестков, форма которых – шестиугольник.

```
использовать Черепаха
алг цветок4
нач
. цел M,L
. вывод "введите количество лепестков: "
. ввод L
. вывод "введите количество сторон многоугольника: "
. ввод M
. нц L раз
. . многоугольник (M)
. . вправо (360/L)
. кц
кон
алг многоугольник (арг цел M)
нач
. нц M раз
. . вперед (50)
. . вправо (360/M)
. кц
кон
```

Рис. 81. Код программы **цветок4.**

Локальные и глобальные переменные

Программа **процедуры_и_функции** (рис. 82) демонстрирует использование *локальных* и *глобальных* переменных, правила объявления *процедур* и *функций*, *аргументов* и *результата* вспомогательного алгоритма.

Программа вычисляет путь, который проходит Черепаха, рисуя цветок из **M** лепестков, состоящих из **L** -угольников.

Прочитайте комментарии к коду программы **процедуры и функции.** В программе используются следующие переменные: **S** – ответ на поставленную задачу (длина пути), **PL** – периметр многоугольника, **PM**- длина пути для цветка из состоящего из **M** многоугольников.

Задание №Ч29.Организуйте ввод длины стороны многоугольника с клавиатуры.

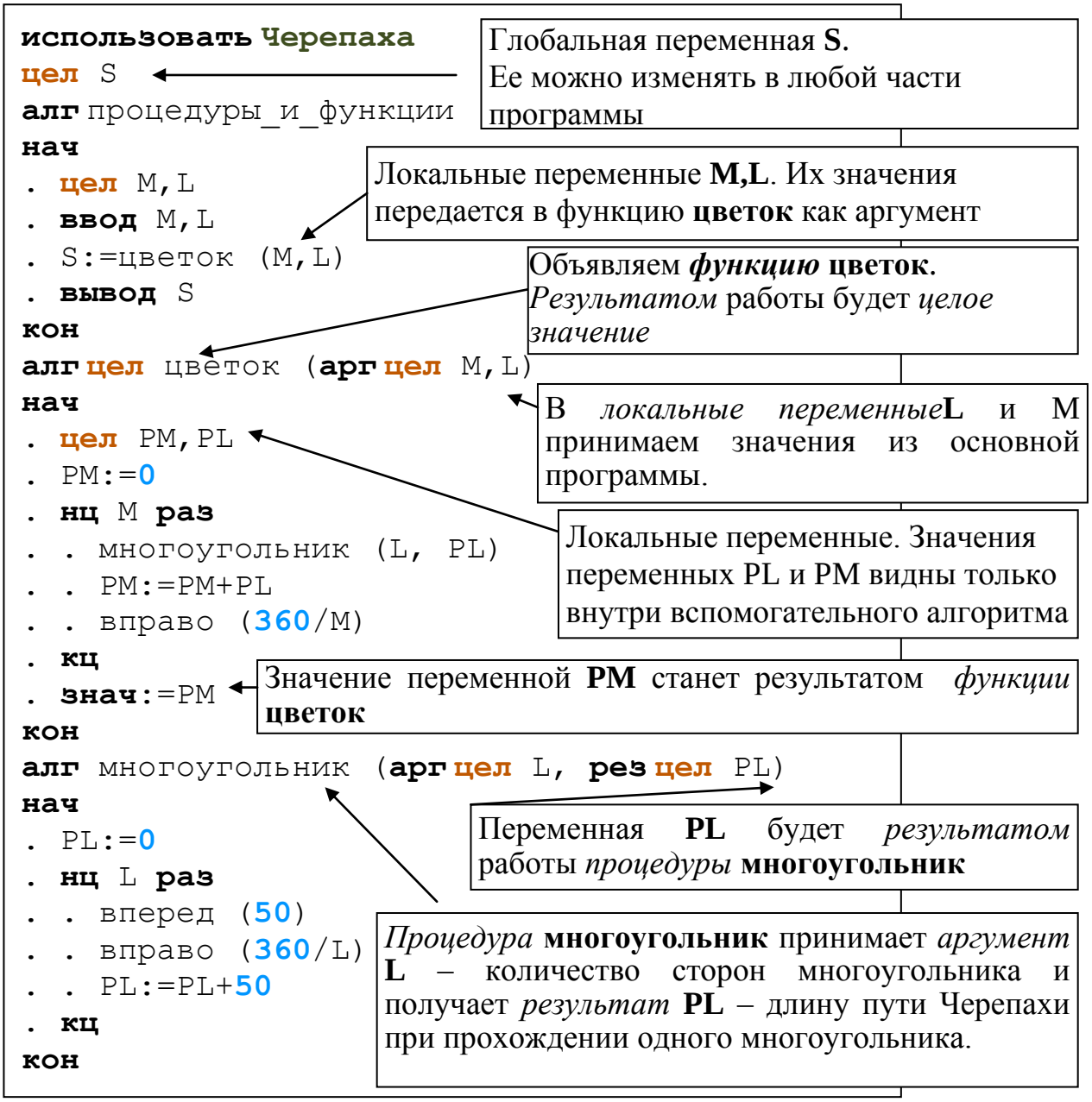


Рис. 82. Код программы с использованием локальных и глобальных переменных.

Задание №Ч30. Вычислите длину пути Черепахи в программе **процедуры_и_функции** (рис.82) с помощью математических расчетов не используя подсчет в цикле.

Рекурсия

Рекурсия - это такая процедура, которая вызывает сама себя. Это самые сложные программы. Рассмотрим пример неуправляемой *рекурсии*.

В алгоритме **первый рекурсивный** (рис.83) процедура

```
использовать Черепаха
алг первый рекурсивный
нач
. рекурсия
кон
алг рекурсия
нач
. вперед (150)
. вправо (77)
. рекурсия
кон
```

Рис. 83. Код алгоритма **первый рекурсивный**

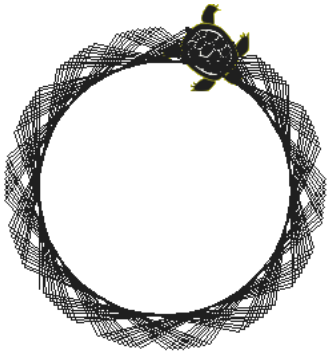


Рис. 84. Результат работы алгоритма **первый рекурсивный**

«**рекурсия**» вызывает сама себя. Условие выхода из процедуры отсутствует. Программа будет работать, пока вы не нажмете кнопку **Стоп** или не закончится оперативная память (рис. 84).

В алгоритме **второй рекурсивный** (рис. 85) Черепаха будет рисовать квадрат, пока не выйдет за границы поля (рис. 86).

```
использовать Черепаха
алг второй рекурсивный
нач
. цел k
. k:=200
. квадрат (k)
кон
алг квадрат (арг цел k)
нач
. нц 4 раз
. . вперед (k)
. . вправо (90)
. кц
квадрат (k-5)
кон
```

Рис. 85. Код программы **второй рекурсивный.**

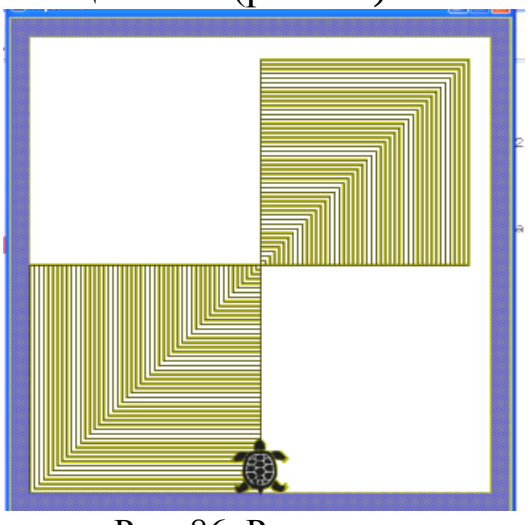


Рис. 86. Результат работы алгоритма **второй рекурсивный.**

От первой она отличается тем, что в процедуре **квадрат** есть аргумент: **целое k.**

При каждом вызове процедуры **k** уменьшается на 5 единиц.

Работа программы закончится, когда Черепаха выйдет за границы поля и программа выдаст **ошибку**.

А вот пример программы с *управляемой рекурсией*: «**рекурсия с выходом**» (рис. 87). **Если** значение переменной k становится меньше -200, то в программе предусмотрен **выход** из процедуры: иначе она рисует **квадрат**, т.е. вызывает сама себя. Поворот на один градус: **вправо (1)** делает рисунок очень красивым!

```
использовать Черепаха
алг рекурсия с выходом
нач
. цел k
. k:=200
. квадрат (k)
кон
алг квадрат (арг цел k)
нач
. если k<-200 то выход
. . иначе
. нц 4 раз
. . вперед (k)
. . вправо (90)
. кц
. вправо (1)
. квадрат (k-5).
. все
кон
```

Рис. 87. Код программы **рекурсия с выходом.**

Фрактальная графика

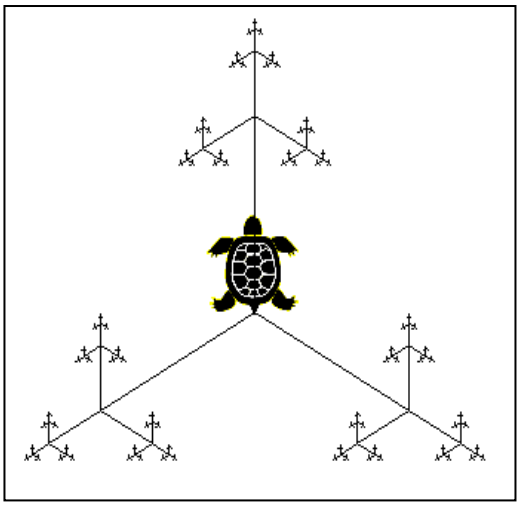


Рис. 88. Фрактал снежинка из трех лучиков.

— математическое множество, обладающее свойством самоподобия [4]. В живой и неживой природе встречаются объекты, составленные из нескольких частей, каждая из которых подобна всей фигуре целиком. Такие явления становятся объектами изучения разных наук. Это снежинки и морозные рисунки на окнах, кораллы, цветная капуста, кровеносная система и бронхи людей и животных и другие. В сети Интернет можно найти много красивых моделей, обладающих свойством ***рекурсивности***: каждая часть модели является уменьшенной копией целой (рис. 88).

Задание 1. Найдите в коде программы **фрактал снежинка** (рис. 89) строку, в которой подпрограмма вызывает саму себя. Какое условие выхода из рекурсии здесь поставлено?

💻**Задание 2.** Наберите без ошибок код программы **фрактал снежинка** (рис.89). В коде программы Черепаха прорисовывает даже самые мелкие детали.

Проведите эксперимент: измените в программе **фрактал снежинка** условие выхода из рекурсии:

а) на k>20;

б) на k>40.

Как изменился результат работы программы?

💻**Задание 3.** Измените код программы **фрактал снежинка** так, чтобы она рисовала снежинку из шести самоподобных частей (рис. 90).

```
использовать Черепаха
вещ S
алг фрактал снежинка
нач
. вещ k
. S:=0
. k:=120
. снежинка (k)
. вывод (S)
кон
алг снежинка (арг вещ k)
нач
. нц 3 раз
. . вперед (k)
. . если k>10 то снежинка (k/3)
. . все
. . назад (k)
. . вправо (360/3)
. кц
кон
```

Рис. 89. Код программы **фрактал снежинка**.

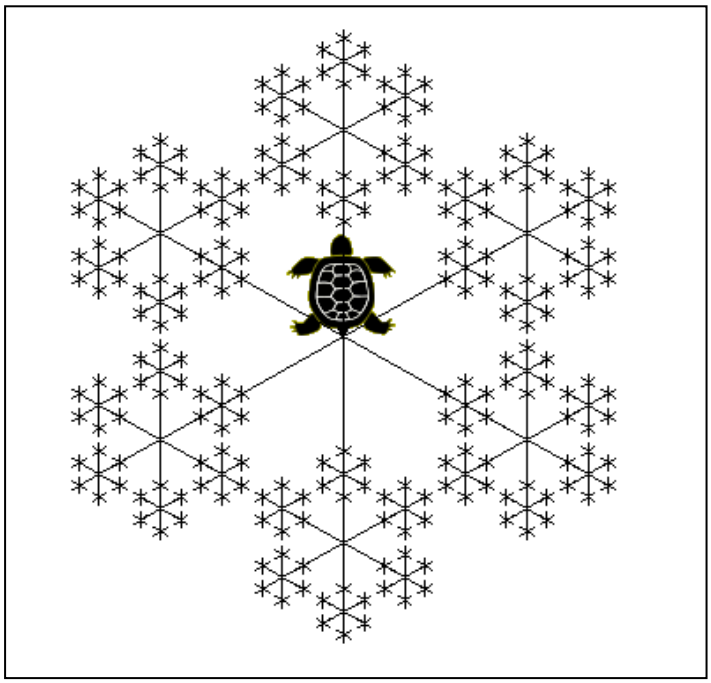


Рис. 90**. Фрактал снежинка** из шести самоподобных частей.

💻**Задание 4.** Посчитайте путь, пройденный Черепахой при рисовании рекурсивных снежинок с разными параметрами. Для решения этой задачи в коде программы есть заготовка: объявлена **глобальная** переменная **S вещественного** типа. Ее значение можно будет изменять в любом месте программы.

Кривая Коха

Кривая Коха — фрактальная кривая, описанная в 1904 году шведским математиком ***Хельге фон Кохом***[5].

Кривая Коха является типичным **геометрическим фракталом**. Процесс её построения выглядит следующим образом: берём единичный отрезок, разделяем на три равные части и заменяем средний интервал равносторонним треугольником без этого сегмента (рис.91а). В результате образуется ломаная, состоящая из четырех звеньев длины 1/3. На следующем шаге повторяем операцию для каждого из четырёх получившихся звеньев (рис. 91б, 91в) и т. д.. Предельная кривая и есть кривая Коха (рис. 91г).

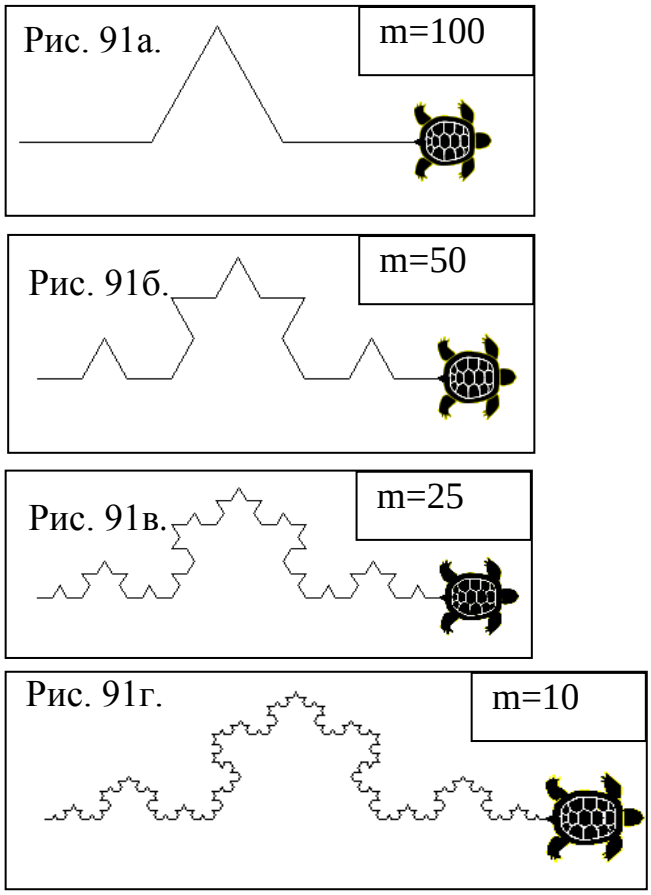


Рис. 91. Кривая Коха.

```
1 использовать Черепаха
2 вещ S
3 алг кривая Коха
4 нач
5 . вещ k
6 . вправо (90)
7 . поднять хвост
8 . назад (200)
9 . опустить хвост
10 . k:=100
11 . Кох (k)
12 кон
13 алг Кох (арг вещ k)
14 нач
15 . цел m=20
16 . если k>m то
17 . . . Кох (k/3)
18 . . иначе вперед (k)
19 . все
20 . влево (60)
21 . если k>m то
22 . . . Кох (k/3)
23 . . иначе вперед (k)
24 . все
25 . вправо (120)
26 . если k>m то
27 . . . Кох (k/3)
28 . . иначе вперед (k)
29 . все
30 . влево (60)
31 . если k>m то
32 . . . Кох (k/3)
33 . . иначе
34 . . . вперед (k)
35 . все
36 кон
```

Рис. 92. Код программы кривая Коха.

Задание 1. Посчитайте длину кривой Коха. Для этого во второй строке программы **кривая Коха** (рис. 92) введена ***глобальная*** вещественная переменная **S**. В начале программы инициализируйте ее: присвойте **S:=0**. После каждой команды **вперед (k)** запишите строку: **S:=S+k.**

Выведите значение переменной **S** на экран. Изменяется ли длина кривой при изменении **m**?

Вопрос 1. Какую максимальную длину для кривой Коха вы смогли получить?

Рисунки в полярных координатах

Вы уже знакомы с декартовыми координатами из курса математики (рис. 93). И уже знаете, что месторасположение любой точки определяется двумя координатами: **X** и **Y**.

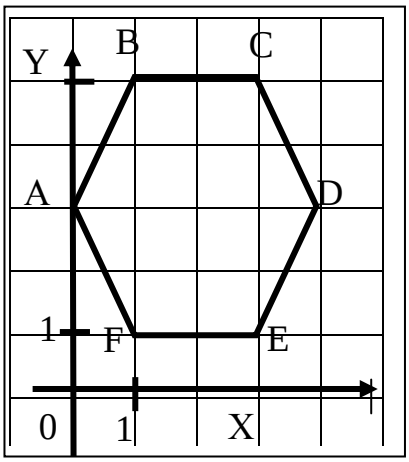


Рис. 93. Декартовы координаты.

Вопрос 1. Назовите координаты точек **A, B, C, D, E, F** на рисунке 93.

Оказывается, что определять местоположение точки на плоскости можно и совершенно другим способом! Попробуем нарисовать точку, первая координата которой – расстояние от центра, а вторая – угол поворота относительно некоторого луча на плоскости.

Задание 1. Нарисуйте в тетради луч и назовите его **ON**. Отметьте следующие точки и соедините их линией по порядку. Помните, что первое число – расстояние от центра, а второе – угол от луча **ON** (рис. 94).

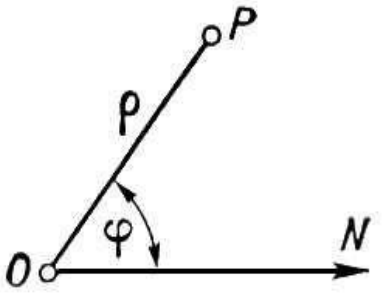


Рис. 94. Полярные координаты.

(2, 0); (2, 45); (2, 90); (2, 135); (2, 180); (2, 225); (2, 270); (2, 315); (2, 360).

Если вы сделали правильно, то у вас получился правильный восьмиугольник.

Заметим, что Черепаха умеет поворачиваться на угол и отмерять расстояние. Однако она не умеет ставить точку. Но мы можем ее научить её это делать подпрограммой, например, такой, алгоритм которой представлен на блок – схеме рисунка 95.

Теперь рассмотрим программу для рисования окружности (рис.96).

Указание: для более быстрого построения используйте изменение угла поворота **вправо (5)**.

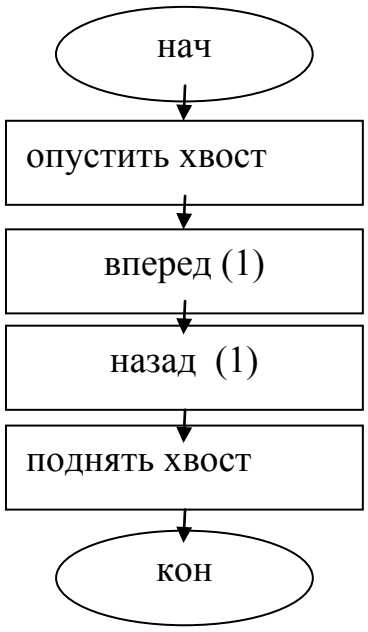


Рис. 95. Блок-схема подпрограммы **точка**.

```
использовать Черепаха
алг окружность
нач
. поднять хвост
. нц 360 раз
. . вперед (50)
. . точка
. . назад (50)
. . вправо (1)
. кц
кон
алг точка
нач
. опустить хвост
. вперед (1)
. назад (1)
. поднять хвост
кон
```

Рис. 96. Программа для построения окружности.

Вопрос 2.На какое число при угле поворота в 5 градусов можно изменить количество повторений цикла в программе?

Вопрос 3. Наверное, в ходе экспериментов с рисованием многоугольников у вас уже получались фигуры, похожие на окружность. Действительно, при увеличении количества сторон правильный многоугольник приближался к форме окружности. Расскажите о преимуществах и недостатках обоих способов построения окружности: с помощью программы **точка** и с помощью программы **многоугольник.**

Круг и квадрат

Решим задачу. Нарисуем квадрат и попробуем точно вписать в него круг (рис.97).

Задание 1. Наберите код программы **многоугольник 360** (рис. 98). Изменяя параметр **вперед (3.1)** подберите его так, чтобы круг был как можно точнее вписан в квадрат. Помните, что дробные числа пишут через

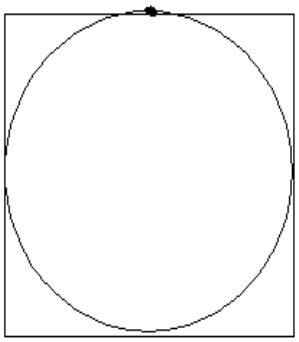


Рис. 97. Круг вписан в квадрат.

точку!

```
использовать Черепаха
алг многоугольник360
нач
. нц 4 раз
. . вперед (200)
. . вправо (90)
. кц
. вперед (100)
. нц 360 раз
. . вперед (3.1)
. . вправо (2)
. кц
кон
```

Рис. 98. Код программы **многоугольник360.**

Вопрос 1.Какое значение стороны многоугольника стало лучшим в ходе эксперимента?

Вопрос 2.Теперь рассчитайте количество повторений цикла так, чтобы Черепаха не проходила по одному месту дважды.

Задание 2. Решим эту же задачу с помощью другой программы: **окружность360** (рис.99).

Здесь под заштрихованным прямоугольником три команды, позволяющие правильно расположить окружность внутри квадрата. Запишите их самостоятельно.

Процедура **точка** описана в программе как вспомогательная.

Вопрос 3. Расскажите о преимуществах и недостатках обоих способов построения окружности.

```
использовать Черепаха
алг окружность360
нач
. нц 4 раз
. . вперед (200)
. . вправо (90)
. кц
. поднять хвост
.
.
.
. нц 360 раз
. . вперед (100)
. . точка
. . назад (100)
. . вправо (2)
. кц
кон
алг точка
нач
. опустить хвост
. вперед (1)
. назад (1)
. поднять хвост
кон
```

Рис. 99. Код программы **окружность 360.**

Спираль

Посмотрите на программу рисования «**спираль**» (рис.100). При каждом повороте, в отличие от рисования окружности, увеличивается расстояние от центра. Если алгоритм **точка** заменить на алгоритм **звездочка**, то получится спиралевидная галактика (рис. 101).

```
использовать Черепаха
алг спираль
нач
. вещ r
. r:=1
. поднять хвост
. нц 360 раз
. . вперед (r)
. . точка
. . назад (r)
. . вправо (5)
. . r:=r+1
. кц
кон
```

Рис. 100. Код программы

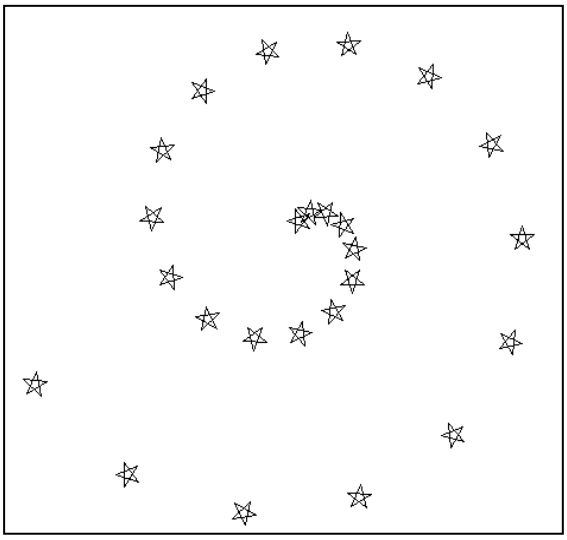


Рис. 101. Результат работы программы **спираль.**

Раздел 5. Исследовательские работы

Синтаксические ошибки

Записывайте в конце тетради синтаксические ошибки, с которыми вы встретились в Кумире.

Как исполнитель понимает «*палочку*» в начале строки?

Для чего в программе нужны пояснения (комментарии)?

Свойства чисел

Определите, сможет ли Кузнечик перекрасить **все точки от 0 до 10** при следующих начальных заданиях (да/нет): в таблицу запишите точки, которые удалось закрасить.

Образец выполнения задания в ячейке [**вперед 2**, **назад 2**].

	вперед 2	**вперед 3**	**вперед 4**	**вперед 5**
назад 2	0,2,4,6,8,10: нет			
назад 3				
назад 4				
назад 5				

Задайте свои значения **a** и **b** для команд [**вперед a]** и [**назад b]**. Разница между числами **a** и **b** должна быть больше 3.

Предположите, какими свойствами должны обладать числа **a** и **b** для команд [**вперед a]** и [**назад b]** для того, чтобы Кузнечик смог попасть в точку с координатой, равной единице?

Запишите вывод в тетрадь. На основании вывода скажите, можно ли перекрасить точки от 0 до 10 при начальных условиях:

а) вперед 32, назад 23; **б) вперед 11, назад 15**;

в) вперед 40, назад 18; **г) вперед 2000, назад 2017**?

Паутинки и звездочки

Напишите программы разных рисунков для исполнителя Черепаха. Исследуйте, какие фигуры получаются при изменении параметров **a** и **b:**

а) с телом цикла: [**вперед (a); вправо (b)];**

б) с телом цикла: [**вперед (a);вправо (b); вперед (c);влево (d)**].

Замощение плоскости

Самая простая фигура для замощения плоскости – квадрат (рис. 102).

Для исполнителя Черепаха напишите программу, которая выполняет замощение плоскости квадратом.

Рис. 102. Замощение плоскости.

Подумайте, какими еще правильными многоугольниками можно замостить плоскость? Какими другими фигурами можно замостить плоскость? Сделайте наблюдение о тротуарной плитке под вашими ногами.

Напишите самостоятельно программы для замощения плоскости. Оформите свое исследование. Напишите цель, задачи и гипотезу. Приложите фотографии и коды программ, которые удалось реализовать.

Формула звезды

Напишите программы для рисования звезд (рис.103). Выявите закономерность рисования звезд разного типа. Организуйте ввод количества вершин звезды с клавиатуры.

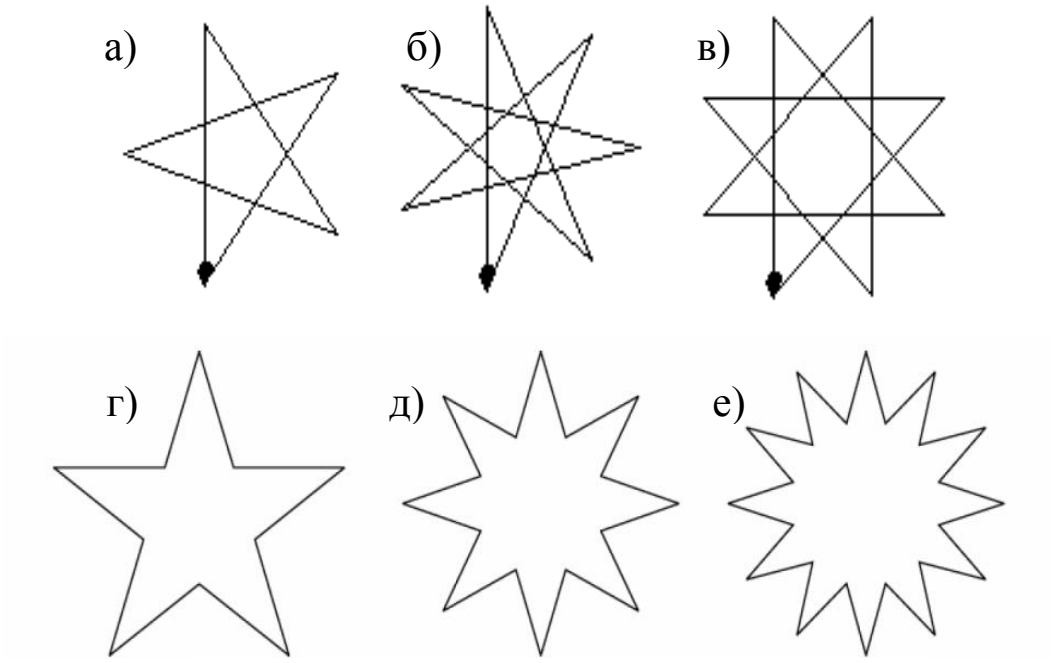


Рис. 103. Звёзды.

Магазин Вини - Пуха

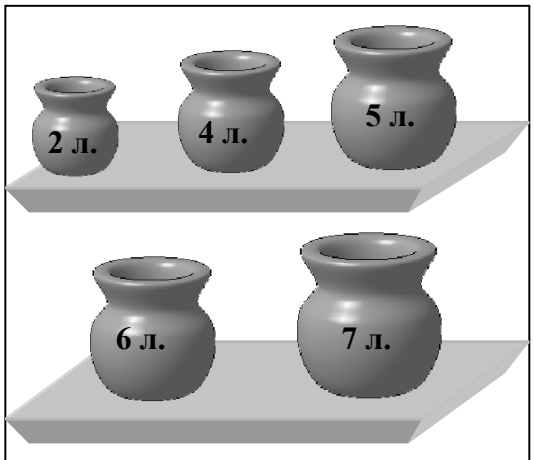


Рис. 104. Лавка магазина.

Вини- Пухи Пятачок решили открыть свой магазин по продаже меда. За мерными горшками поехали в лавку. Дорога была далекой, а денег было только на два горшка. К их удивлению в лавке были только горшки в 2, 4, 5, 6 и 7 литров (рис. 104).

Наши герои задумались: какие меры нужно взять? Отмерять придется меры от 1 одного до 8 литров

включительно. Мед можно будет переливать в бочку большого размера, а такая в магазине имеется.

Винни – Пуху и Пяточку хотелось, чтобы покупатели ожидали в очереди как можно меньше, и начали думать: какие меры будут оптимальными сточки зрения количества переливаний при продажах.

Помогите Пяточку и Винни – Пуху выбрать **две лучшие** меры. Докажите, что этот выбор будет правильным. Напишите оптимальные алгоритмы нахождения каждой меры от 1 до 8 литров.

Проверьте свое решение с помощью исполнителя Водолей. Первый сосуд возьмите на 20 литров - он будет моделировать бочку большого размера.

Решите задачу, при условии что в лавке были горшки 4,5,6 и 7 литров. Взять можно только два.

Сформулируйте свою задачу для исследовательской работы.

Полярная роза

В XVIII в. итальянский геометр Гвидо Гранди (1671—1742) создал розы[6]. Нет, вовсе не те прекрасные цветы, о которых вы, наверное, подумали. Розы Гранди радуют нас правильными и плавными линиями, но их очертания не каприз природы — они предопределены специально подобранными математическими зависимостями. Эти зависимости были подсказаны самой природой, ведь в большинстве случаев абрис листа или цветка представляет собой кривую, симметричную относительно оси. Вот формула Полярной розы $R=sin(\alpha*\varphi)$ полярных координатах.

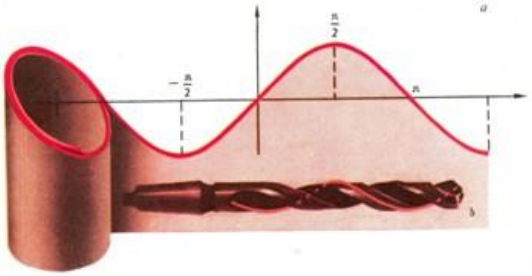


Рис. 105. График функции синуса[6].

Возможно, вы уже слышали о функции синуса (рис. 105). В математике это звучит так: синус – это отношение прилежащего (к углу) катета к гипотенузе. Например, в Египетском треугольнике (рис. 106) со сторонами 3, 4, 5 синус угла А равен 3/5, а синус угла В равен 4/5.

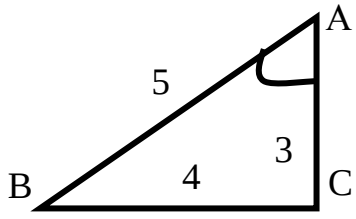


Рис. 106. Египетский треугольник.

График функции синус периодический, представлен на рисунке 107.

Эта *периодичность* и легла в основу построения *красивых кривых*, называемых «***Полярными розами***».

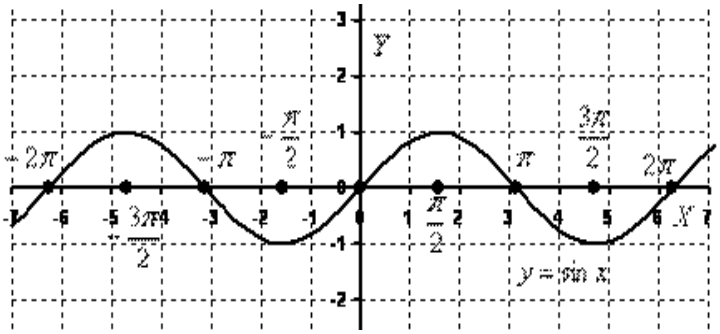


Рис. 107. График функции синус.

Конечно, математик Гвидо Ганди в XVIII веке все вычисления проводил вручную. Оцените примерно, сколько времени ему потребовалось для вычисления координат одной точки? Рисования одного цветка?

Напишите программу для построения в полярных координатах кривой, в которой расстояние до от точки до центра вычисляется по формуле: **R=sin(α*φ).** Постройте **цикл** изменения угла φ от 0 до 360 градусов. Для перевода в радианы пользуйтесь соотношением:

1 радиан =57,296 градусов, то есть указанная формула преобразуется в следующую: **R=sin(α*φ/57.296).** Нарисуйте цветы при разных значениях **α.** Подумайте, в каких пределах значений находятся значения функции синуса? Во сколько раз необходимо увеличить радиус R в формуле цветка для того, что бы этот цветок хорошо был виден на поле Черепахи?

Указание. Для построения цветка используйте линии, выходящие из центра. Подумайте, во сколько раз нужно увеличить радиус **R** при построении линий на поле Черепахи.

Игра Баше с Черепахой

Правила игры: на исходной позиции 15 спичек (рис. 108). Игроки ходят по очереди. Можно брать одну, две или три спички. Выигрывает тот, кто делает последний ход, то есть забирает последние (одну, две или три) спички.

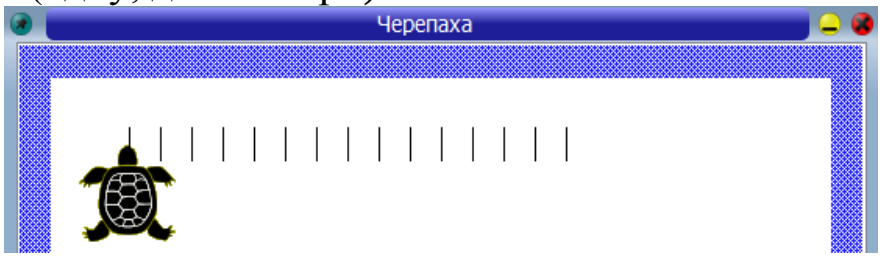


Рис. 108. Исходная позиция игры Баше.

Вопрос1. Какие вспомогательные алгоритмы используются в программе игра Баше (рис. 109)?

Вопрос2. По коду программы определите, что выполняет Черепаха при исполнении а) алгоритма **в начало**; б) алгоритма **ставь спички?**

Вопрос3. По коду основной программы определите:

а) в какой строчке задается начальное условие: количество спичек, равное 15?

б) В какой переменной хранится текущее количество спичек?

в) До каких пор выполняется цикл ходов в игре? Назовите строчку этого условия.

г) В каком случае программа выдает сообщение: «вы нарушили правила»?

д) Определите, в каких строчках программы Черепаха принимает решение о том, сколько спичек взять?

е) В какой строчке используется оператор получения случайного числа?

ж) В каком диапазоне Черепаха получает случайное число?

з) Опишите выигрышную стратегию в игре. Определите, кто выигрывает – первый игрок или второй при правильной стратегии?

и) Измените начальное количество спичек и количество спичек, которые игрок может взять одним ходом. Опишите выигрышную стратегию при новых правилах.

```
1  использовать Черепаха
2  цел с, я, ч
3  алг основной
4  нач
5  . с:=15
6  . в начало
7  . ставь спички
8  . нц пока с>0
9  . . вывод "ваш ход "
10 . . ввод я
11 . . если я <1 или я >3 то
12 . . . . вывод "вы нарушили правила!"
13 . . . . выход
14 . . все
15 . . с:=с-я
16 . . если с>0 то
17 . . . . ставь спички
18 . . . иначе вывод " ты выиграл"
```

```
. . . . выход
. . все
. . если с/4-int(с/4)>0 то ч:=с-int(с/4)*4
. . . иначе ч:=int(rnd(3)+1) все
. . вывод "беру ",ч,"осталось ",с-ч, нс
. . с:=с-ч
. . если с>0 то
. . . . ставь спички
. . . иначе
. . . . вывод " ты проиграл"
. . все
. кц
кон

алг в начало
нач
. поднять хвост
. вправо (90)
. назад (200)
. влево (90)
. вперед (200)
. опустить хвост
кон

алг ставь спички
нач
. нц с раз
. . опустить хвост
. . вперед (20)
. . назад (20)
. . вправо (90)
. . поднять хвост
. . вперед (20)
. . влево (90)
. кц
. влево (90)
. вперед (с*20)
. вправо (90)
. назад (50)
кон
```

Рис. 109. Код программы «**Игра Баше**».

Часть 2. Исполнитель Робот

Раздел 1. Роботы в нашей жизни

Что умеет Робот? Робот – это автоматическое устройство, действующее по специальной программе и получающее информацию о внешнем мире от различных датчиков[8]. Сейчас становится все больше роботов, способных заменить людей в разной деятельности.

Говоря словами российского ученого в области робототехники, академика РАН по отделению нано технологий и информационных технологий И.М.Макарова: «Люди получили верного помощника, способного не только выполнять опасные для жизни человека работы, но и освободить человечество от однообразных рутинных операций» [9].

Наш Робот – учебный, он существует в определенной обстановке. Начальная обстановка Робота называется стартовой. Поле Робота имеет максимальные размеры: 10 строк и 16 столбцов, ограниченных стенами. На поле могут стоять стены. Клетка поля может быть **закрашенной** или чистой, иметь **радиацию** и (или) **температуру**.

Робот в системе КуМир умеет измерять уровень радиации и температуры в клетке, определяет, есть ли около него стена, умеет закрашивать клетку, в которой уже побывал. Мы научимся программировать Робота.

Теперь обо всем по порядку.

Система команд исполнителя Робот

С Роботом связаны три окна: окно наблюдения за Роботом, окно редактирования ***стартовой*** обстановки, окно **Пульта.**

Стартовая обстановка — особая обстановка, в которую принудительно помещается Робот в начале выполнения программы.

Система команд исполнителя Робот включает следующие типы команд: управления, измерения и определения. Команды управления **вверх, вниз, вправо, влево, закрасить** позволяют Роботу передвигаться по полю и закрашивать клетки. Значения команд **температура** и **радиация** имеют *вещественный* тип и позволяют Роботу измерять температуру и радиацию. Команды *определения* имеют *логический* тип и позволяют Роботу определить, есть ли стена и закрашена ли клетка, на которой стоит Робот.

Использование Пульта Робота

Практическая работа №2.1. Стартовая обстановка.

Цель: научиться редактировать ***стартовую*** обстановку Робота.

Исправим ***стартовую*** обстановку так, чтобы она была без стен и без закрашенных клеток. Робот должен стоять в левом верхнем углу.

💻**Задание 1.** Вызовите на экран (рис. 110) окно редактирования ***стартовой*** обстановки Робота (рис. 111).

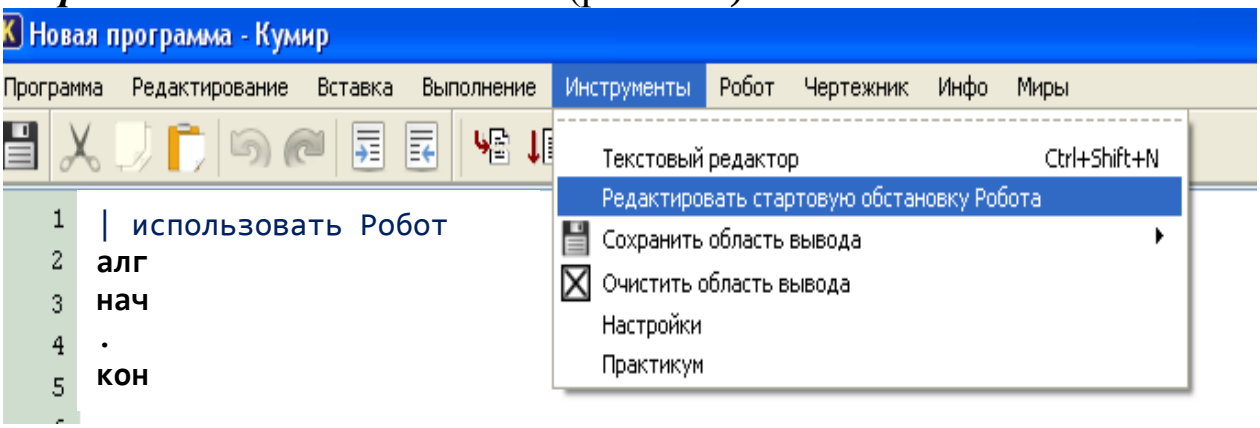


Рис. 110. Вызов окна редактирования Стартовой обстановки Робота

В пункте меню ***Обстановка*** (рис. 111а) можно задать количество строк и столбцов обстановки.

В пункте меню ***Новая обстановка*** можно выставить размеры поля (рис. 111 б).

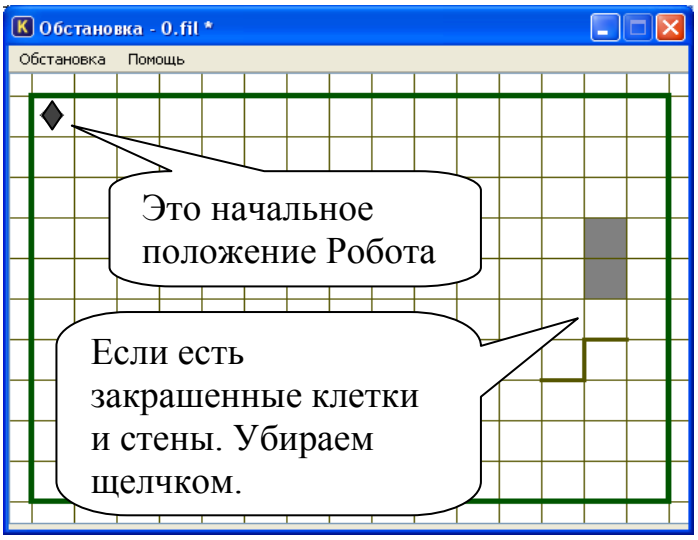


Рис. 111 а. Редактирование стартовой обстановки Робота. Редактирование стен.

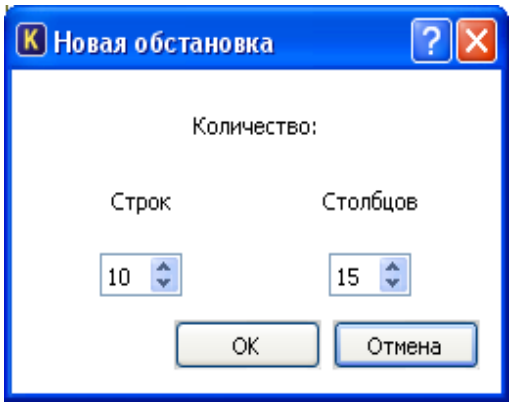


Рис. 111 б. Редактирование стартовой обстановки Робота. Задание размеров поля.

После того, как вам удалось отредактировать ***стартовую*** обстановку, нажмите пункт меню **Робот, Вернуться в стартовую обстановку** (рис.112).

Важно! *Робот будет считать* ***стартовой*** *обстановкой ту, которая* ***открыта*** *или сохранена последней!*

Нажмите кнопку «**Показать окно Робота**» (рис.113).

Задание 2. Расположите 4 окна на экране: среды КуМир, **Пульта**, поля Робота и ***стартовой*** обстановки так, чтобы было удобно (рис.114).

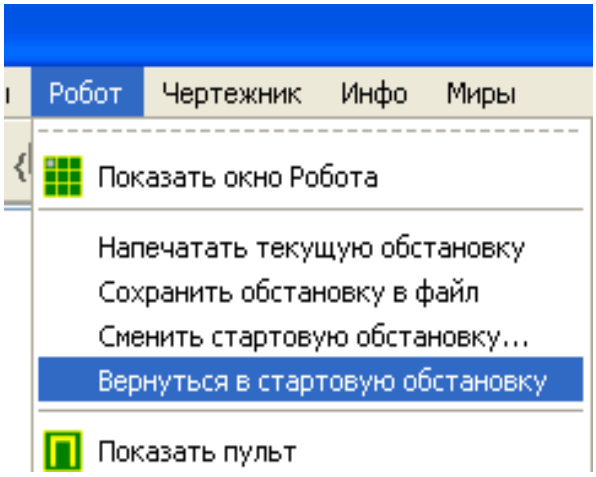


Рис. 112. Возвращение в стартовую обстановку

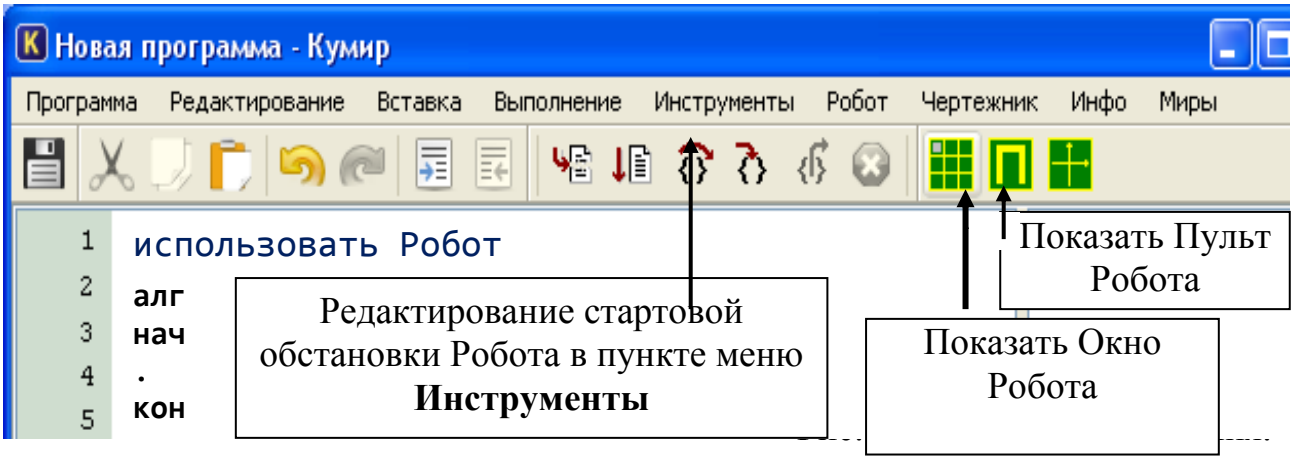


Рис. 113. Окно программы КуМир. Окна исполнителя Робот.

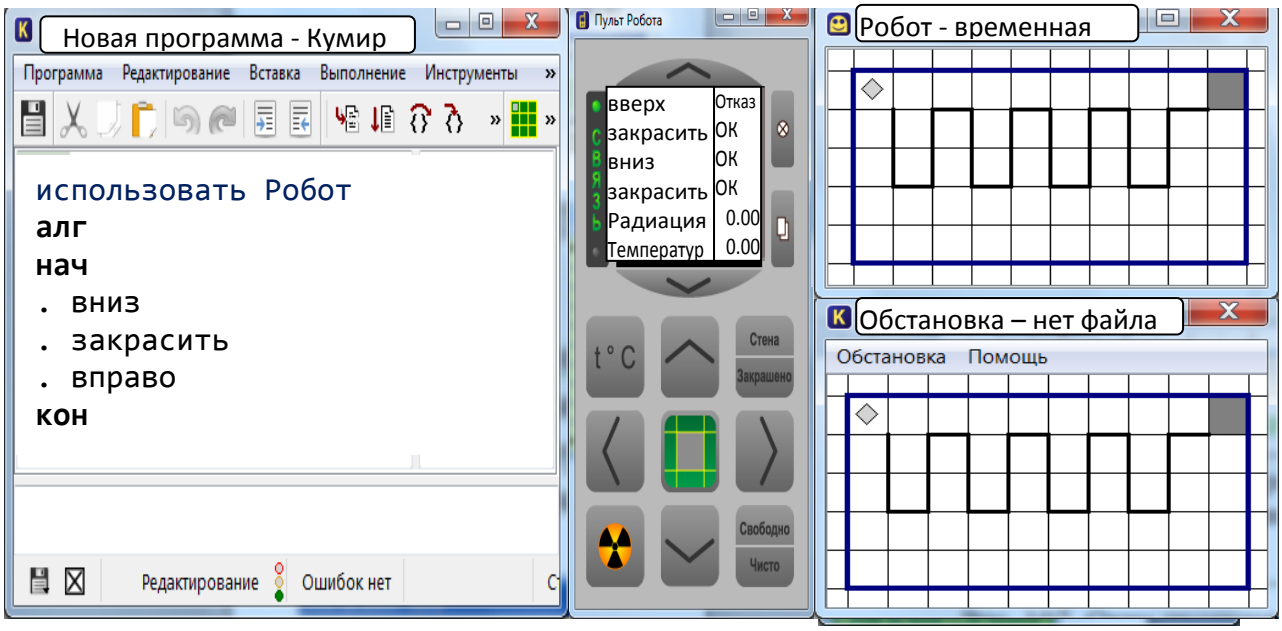


Рис. 114. Окна для работы с исполнителем Робот.

Практическая работа №2.2. Использование Пульта.

Цель: научиться использовать **Пульт** для Робота и переносить программу в систему КуМир.

Вопрос 1. На рисунке 115 определите кнопки «скопировать в буфер обмена», «вставить из буфера», «пошаговая реализация программы», «непрерывное выполнение программы».

Рис. 115. Кнопки управления.

Задание 1. С помощью **Пульта** Робота нарисуйте первую букву вашего имени с помощью закрашивания клеток.

Задание 2. Нажмите на **Пульте** Робота кнопку копирования программы в буфер обмена.

Появится окно с надписью «Текст программы скопирован в буфер обмена» (рис.116).

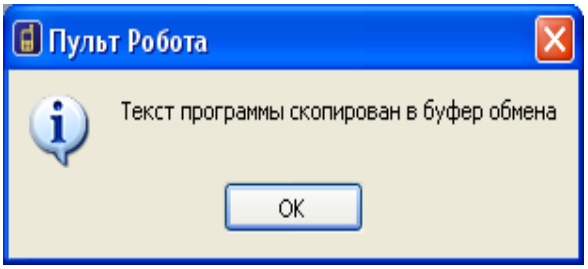


Рис. 116. Диалоговое окно копирования команд на **Пульте** Робота в буфер обмена.

Важно! Вставка из буфера обмена будет в окно КуМир произведена в позицию курсора!

Нажмите кнопку «вставить из буфера».

Запустите программу на пошаговую реализацию.

Вопрос 2. По коду программы **первая программа** окна **Новая программа** (рис.117) определите, какие клетки закрасит Робот. Стартовая обстановка: поле без стен и закрашенных клеток. Робот в верхнем левом углу.

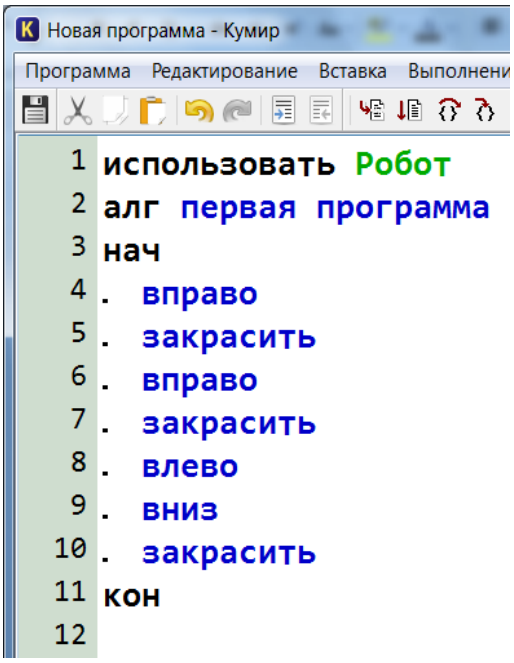


Рис. 117. Код программы **первая программа**.

Начертите в тетради таблицу 5 на 5 клеток (рис. 118) и выполните трассировку программы. Проверьте своё решение на компьютере.

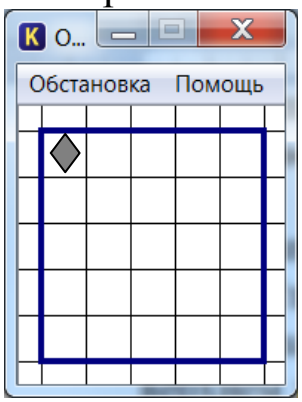


Рис. 118. Поле без стен для Робота 5 на 5 клеток. Робот стоит в левом верхнем углу.

Задание №Р1. а) С помощью **Пульта** напишите программы для рисования одной из букв. Скопируйте текст кода программы в КуМир. Запустите программу. Проследите за работой Робота в окне Робота.

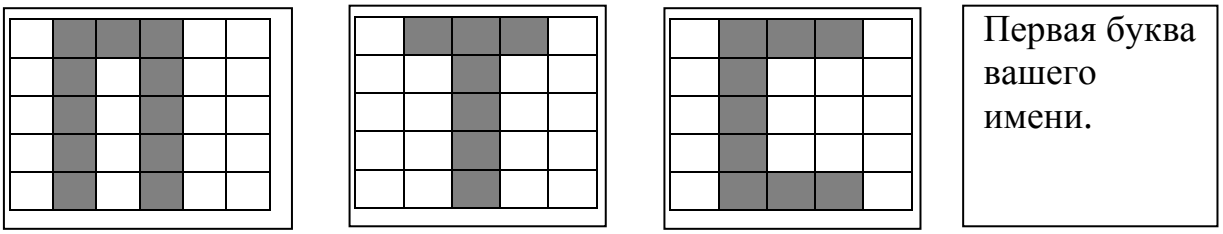


Рис. 119. Буквы на клетчатом поле.

б) По образцу программы **первая программа** (рис. 117) напишите программу для рисования других букв. Запустите ее на исполнение.

Практическая работа №2.3. Где радиация? Работа в паре.

Цель**:** научиться создавать новую ***стартовую*** обстановку, задавать ***Радиацию*** в определенной точке поля. С помощью **Пульта** научиться определять точки с радиацией и закрашивать их.

Первый ученик должен подготовить поле, а второй – с помощью **Пульта** найти и закрасить точки с радиацией.

Размер поля и критический уровень радиации обговариваются участниками заранее.

Задание 1. Первый ученик. Готовит ***стартовую*** обстановку (рис.120).

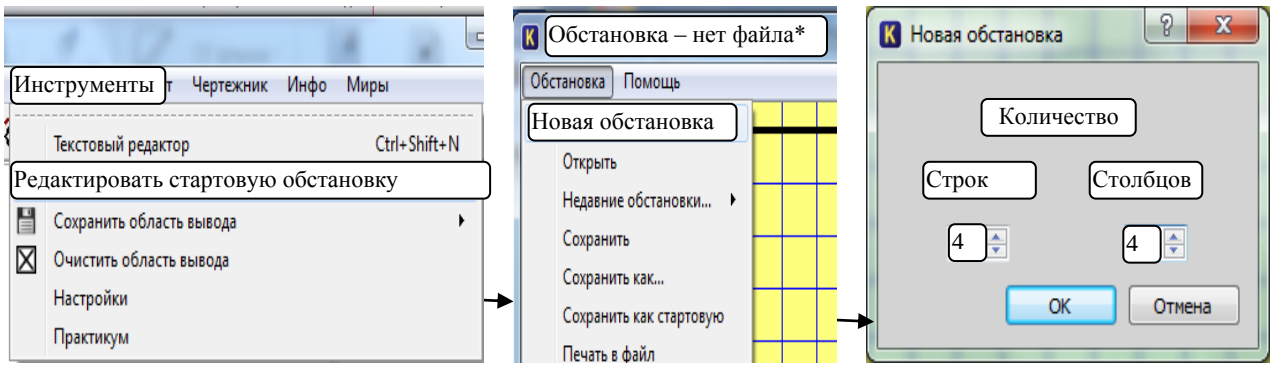


Рис. 120. Подготовка стартовой обстановки.

В окне ***стартовой*** обстановки щелчком правой кнопки мыши вызываем окно редактирования температуры и радиации. Выставляем значение радиации в клетках (рис. 121).

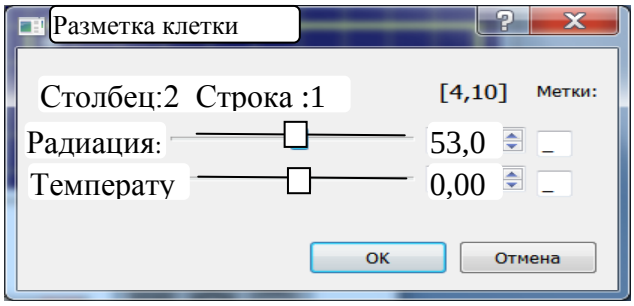


Рис. 121. Выставление радиации в клетке поля Робота.

Обстановку теперь нужно ***свернуть*** (открытая обстановка считается ***стартовой***) или сохранить в файл под именем Робот1.fil (последняя сохраненная обстановка станет ***стартовой***, если нет открытой обстановки).

💻**Задание 2. Второй ученик**. Возвращает Робота в ***стартовую*** обстановку. Вызывает окна: Робота и **Пульт.**

С помощью **Пульта** второй ученик обходит все клетки, измеряет радиацию и закрашивает клетки с радиацией, большей критического уровня.

💬**Задание 3. Первый ученик**. Проверяет, все ли клетки с радиацией закрасил второй ученик правильно.

💬**Задание 4. В паре** подготавливают сообщение, на тему: «В каких опасных ситуациях Робот может заменить Человека».

Практическая работа №2.4. Система команд исполнителя.

Цель: выяснить, какие команды понимает исполнитель Робот.

💻**Задание 1**. Известно, что Робот понимает пять команд: **вверх**, **вниз**, **вправо**, **влево**, **закрасить**. В справочнике языка КуМир найдите эти команды Робота. Есть ли в **системе команд исполнителя** команда «**очистить клетку**»? Запишите все команды Робота в тетрадь.

💻**Задание 2.** С помощью закрашенных клеток, используя команды Робота, придумайте рисунок и напишите для него программу.

💻**Задание 3**. Напишите программу для написания букв слова РОБОТ.

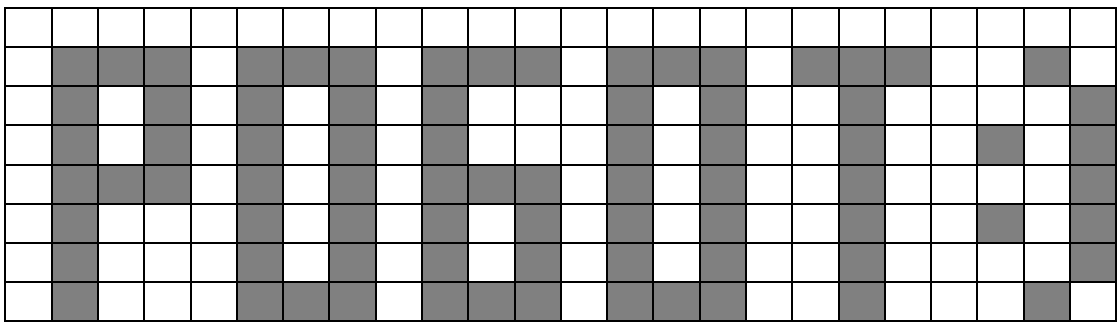


Рис.122. Слово РОБОТ на клетчатом поле.

Раздел 2. Написание программ
Циклы со счетчиком
Практическая работа №2.5. Циклы со счетчиком.

Цель**:** научиться правильно переписывать программы и запускать их на исполнение. На практике познакомиться с алгоритмической конструкцией ***«цикл со счетчиком».***

💻**Задание 1.** Используйте стартовую обстановку без стен и без закрашенных клеток. Робот в левом **верхнем** углу (рис. 123).

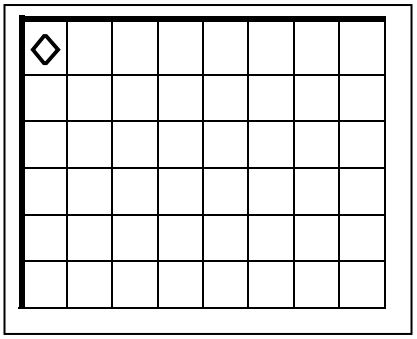


Рис. 123. Поле Робота.

```
использовать Робот
алг цикл
нач
- нц 4 раз
- - вниз
- - вправо
- - закрасить
- кц
кон
```

Рис. 124. Код программы **цикл.**

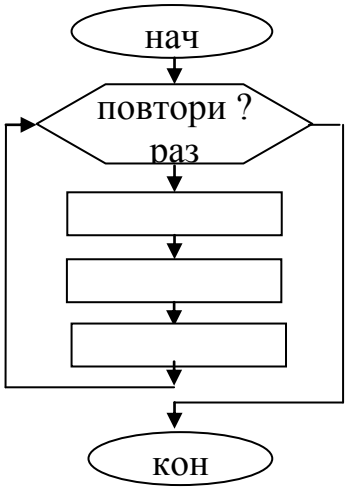


Рис. 125. Заготовка блок- схемы к программе **цикл.**

Предположите, что нарисует Робот после выполнения программы **цикл** (рис. 124)? Зарисуйте предварительный рисунок в тетрадь.

🖉**Задание 2.** Зарисуйте в тетрадь блок – схему (рис.125) программы **цикл** (рис.124), заполнив команды тела цикла.

💻**Задание 3.** Перепишите программу **цикл** (рис.124) в систему КуМир.

Важно! *Во время переписывания* программы у вас будут появляться сообщения об ошибках. Примерно такие, как показаны на рисунке 126.

Это программа - ***транслятор*** подчеркивает ошибки потому, что видит, что у алгоритма есть начало **нач**, но нет конца **кон**, а у алгоритмической конструкции цикла есть начало - **нц 4 раз**, но нет конца цикла **кц.**

Кроме того, для удобства чтения транслятор делает отступы и помечает их точками. Если вы ***допишете программу до конца***, то транслятор эти точки поставит автоматически и уберет информацию о ***текущих*** ошибках.

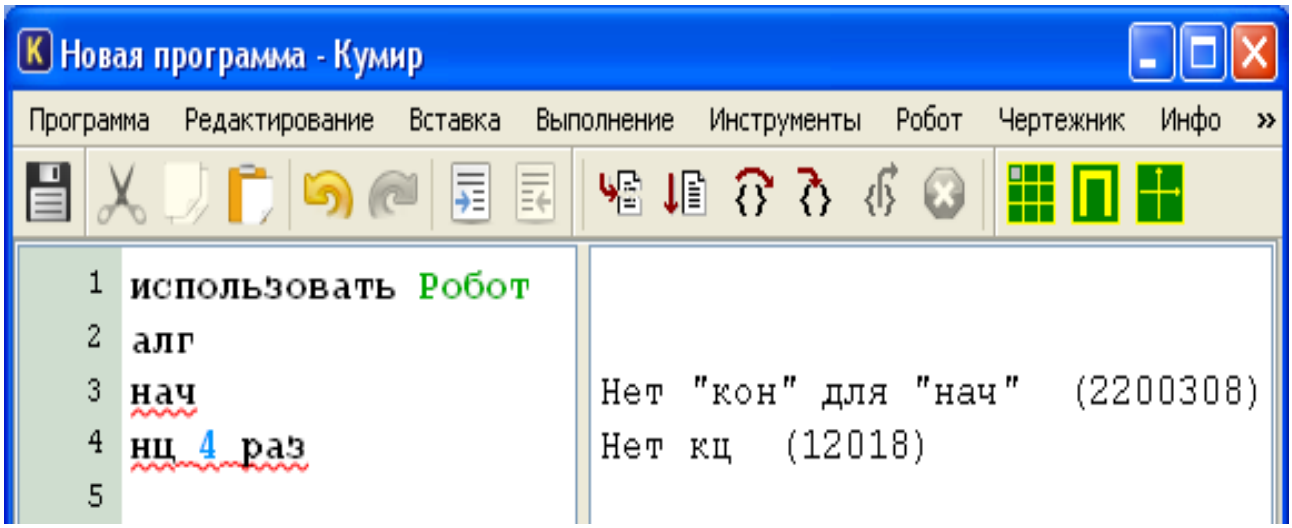


Рис.126. Ошибки, обнаруженные компилятором во время написания программы.

Намного удобнее при наборе программы сразу вставлять шаблон алгоритмической конструкции **«начало и конец»**. Для этого нужно нажать пункт **меню Вставка** и выбрать конструкцию **алг - нач-кон** (рис.127).

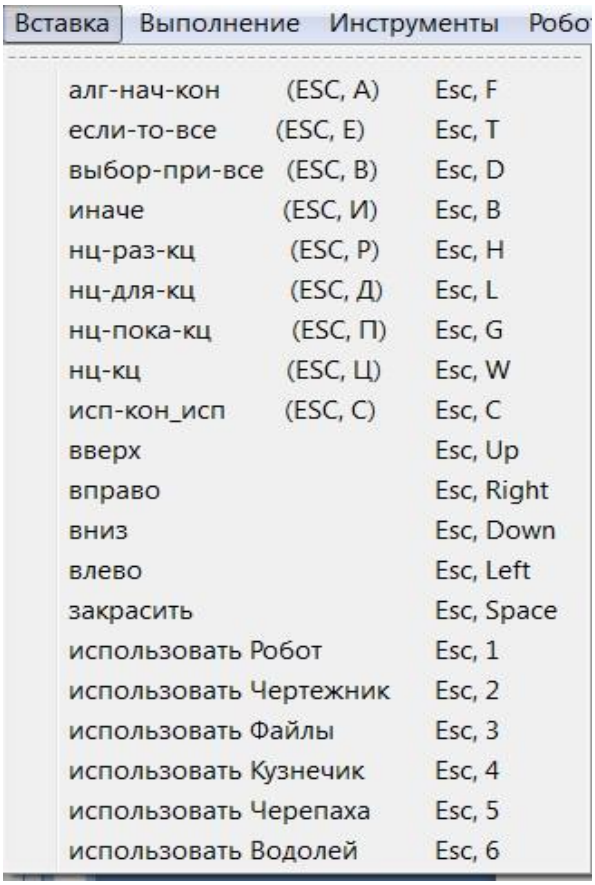


Рис.127. Команды Вставки

Вы увидите, что между началом и концом программы автоматически появилась точка. Теперь вставите конструкцию цикла **нц-раз-кц.**

Поставьте курсор в нужное место и запишите недостающие строки программы. Убедитесь, что ошибок нет.

Отправьте программу на исполнение кнопкой  . Посмотрите, что нарисовал Робот. Оцените, правильно ли вы сделали предположение.

Очень интересно отправить программу на ***пошаговое исполнение.*** В этом случае вы сможете увидеть, как Робот исполняет каждую команду. Сделайте это кнопкой «ШАГ»:  .

Задание №Р2. Предположите, что нарисует Робот в результате работы программы **задание_Р2** (рис.128)**.** Стартовая обстановка: поле без стен, Робот в левом верхнем углу (рис. 129)**.**

```
использовать Робот
алг задание_Р2
нач
. нц 4 раз
. . вниз
. . вправо
. . закрасить
. кц
кон
```

Рис. 128. Код программы **задание_Р2.**

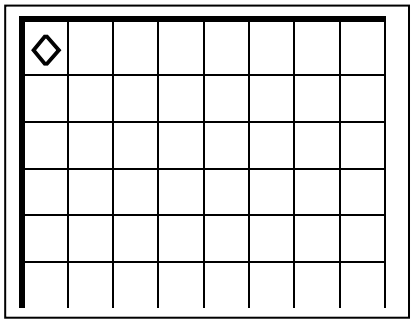


Рис. 129. Стартовая обстановка к программе задание_Р2.

Запишите программу **задание_Р2** на компьютере в системе КуМир.

Убедитесь, что программа работает правильно.

Исправьте программу так, чтобы Робот закрасил клетки так, как это показано на рисунке 130.

Подумайте, какими командами отличаются программы для этих рисунков?

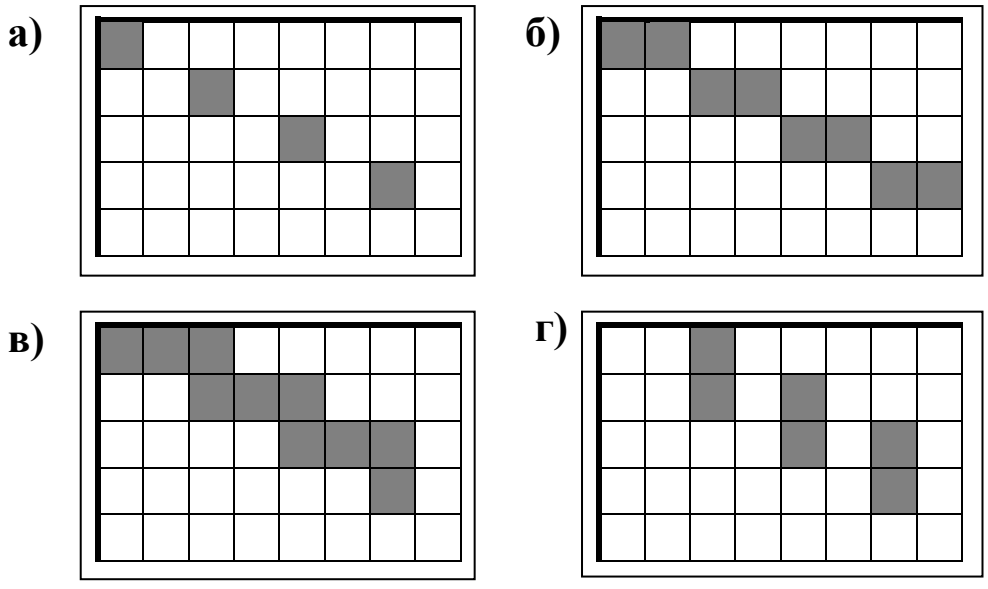


Рис. 130. Рисунки к **заданию № Р2**.

Задание №Р3. Напишите программы, реализующие следующие рисунки (рис.131):

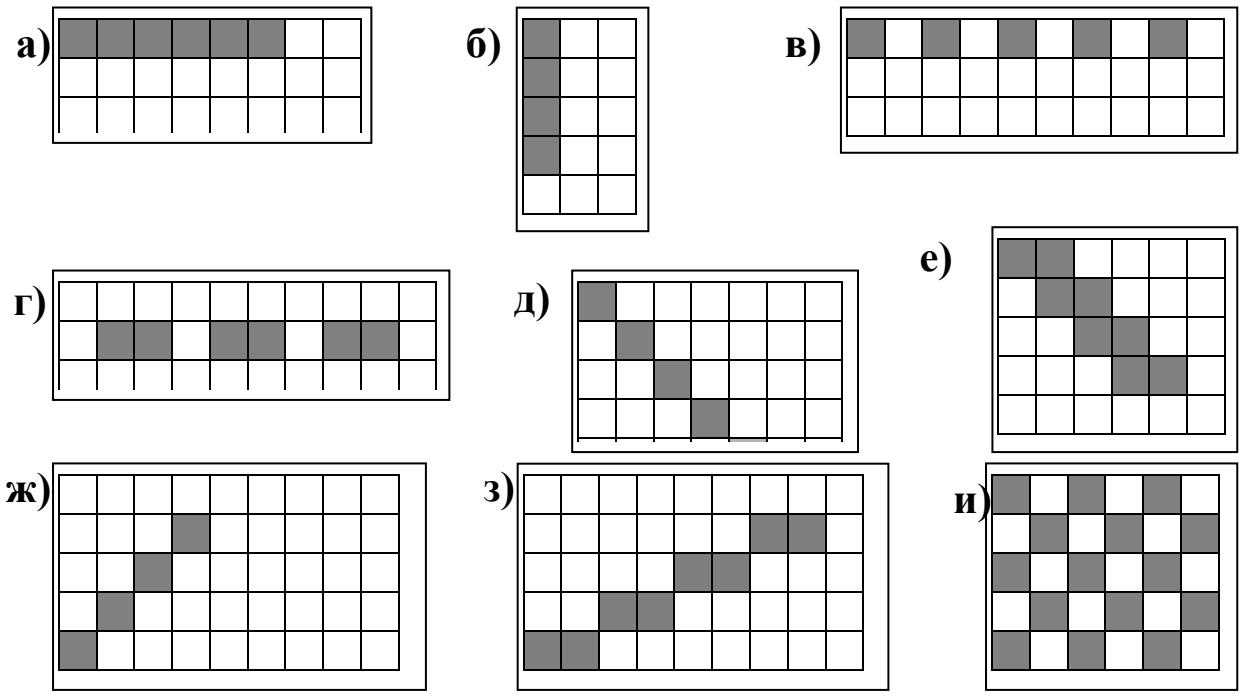


Рис. 131. Рисунки к заданию № Р3.

Задание №Р4. Проверим себя. Посмотрите на программу **задание_Р4** (рис. 132). Стартовая обстановка: поле без стен, Робот в левом верхнем углу.

Предположите, какие клетки закрасит Робот в результате работы программы?

Как изменится рисунок при изменениях в программе, указанных в заданиях а), б), в), г)?

```
использовать Робот
алг задание_Р4
нач
. нц 4 раз
. . закрасить
. . вправо
. . вниз
. кц
кон
```

Рис. 132. Код программы **задание_Р4**.

Ответ нарисуйте в тетради в клеточку.

а)	б)	в)	г)
использовать Робот **алг Р4_а** **нач** **нц 4 раз** вправо вниз закрасить **кц** **кон**	**использовать Робот** **алг Р4_б** **нач** **нц 4 раз** вправо закрасить вниз **кц** **кон**	**использовать Робот** **алг Р4_в** **нач** закрасить **нц 4 раз** вправо вниз **кц** **кон**	**использовать Робот** **алг Р4_г** **нач** **нц 4 раз** вправо вниз **кц** закрасить **кон**

Проверьте свои предположения на компьютере.

Робот закрашивает прямоугольник

Практическая работа №2.6. Цикл внутри цикла.

Цель: на практике ознакомиться с алгоритмической конструкцией «**цикл внутри цикла**».

Задача. Закрасить поле размером **8** на **6** клеток (рис.133). Стартовая обстановка: поле без стен и закрашенных клеток, Робот в левом верхнем углу. Конечное положение Робота может быть любым.

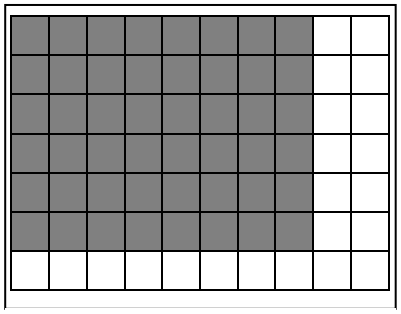


Рис. 133. Поле размером 8 на 6 клеток, которое закрасил Робот.

Напишем программу, с помощью которой можно закрасить **одну** полосу.

```
использовать Робот
алг одна полоса
нач
. нц 8 раз
. . закрасить
. . вправо
. кц
. нц 8 раз
. . влево
. кц
кон
```

Рис. 134. Код программы **одна полоса**.

В программе **одна полоса** (рис. 134) Робот сначала двигается вправо и закрашивает клетки, затем возвращается влево, но клетки уже не закрашивает. Для закрашивания всего поля размером 8 (по горизонтали) и 6 (по вертикали) клеток мы можем скопировать эти два цикла **6** раз, поставив после прохода каждой **полосы** команду **вниз.**

Вопрос 1: посчитайте, сколько строк будет занимать такая программа?

В программировании, если некоторая часть кода повторяется, то ее лучше оформить с помощью ***цикла.***

Для этой программы будем использовать конструкцию «***цикл внутри цикла***». Блок – схема алгоритма представлена на рисунке 135.

Правильное написание конструкции «цикл внутри цикла» смотрите в коде программы **закрашиваем прямоугольник** (рис. 136).

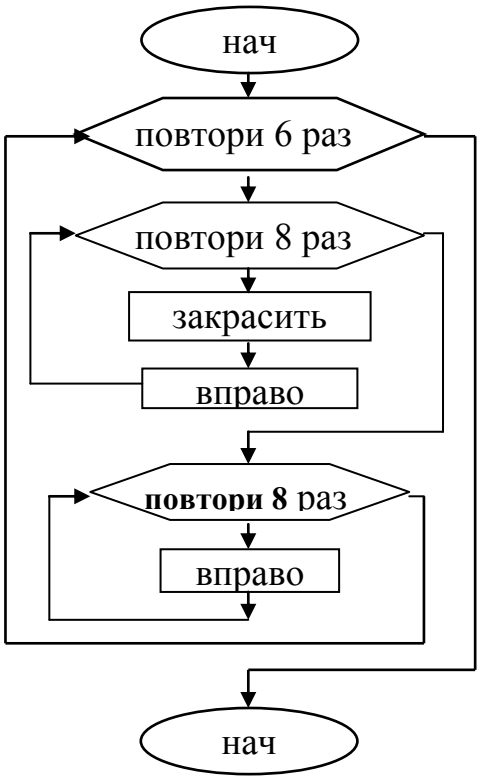


Рис. 135. Блок –схема алгоритмической конструкции «цикл внутри цикла».

Задание 1. Перепишите программу **закрашиваем прямоугольник** (рис. 136) в КуМир. Запустите программу, убедитесь, что она работает.

```
использовать Робот
алг закрашиваем прямоугольник
нач
. нц 6 раз
. .нц 8 раз
. . . закрасить
. . . вправо
. . кц
. . нц 8 раз
. . . влево
. . кц
. . вниз
. кц
кон
```

Рис. 136. Код программы **закрашиваем прямоугольник.**

Задание 2. Исправьте программу так, чтобы Робот закрашивал прямоугольник со сторонами 7 (по горизонтали) и 5 (по вертикали) клеток.

Вопрос 2. Почему конструкция «цикл внутри цикла» предпочтительней копирования блоков программы?

Задание №Р5. Напишите программу для обхода Роботом поля размером 8 на 6 клеток по заданному пути (рис. 137). Все клетки поля 8 на 6 нужно закрасить.

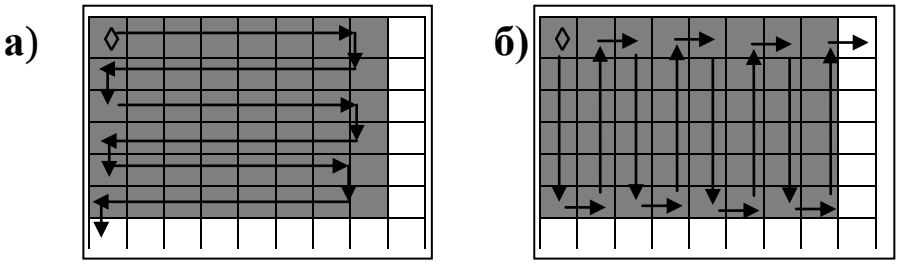


Рис. 137. Траектории обхода поля к заданию №Р5.

Задание №Р6. Напишите программу для представленных ниже рисунков (рис.138). Стартовая обстановка без стен и закрашенных клеток. Робот в левом верхнем углу. Используйте конструкцию «цикл внутри цикла».

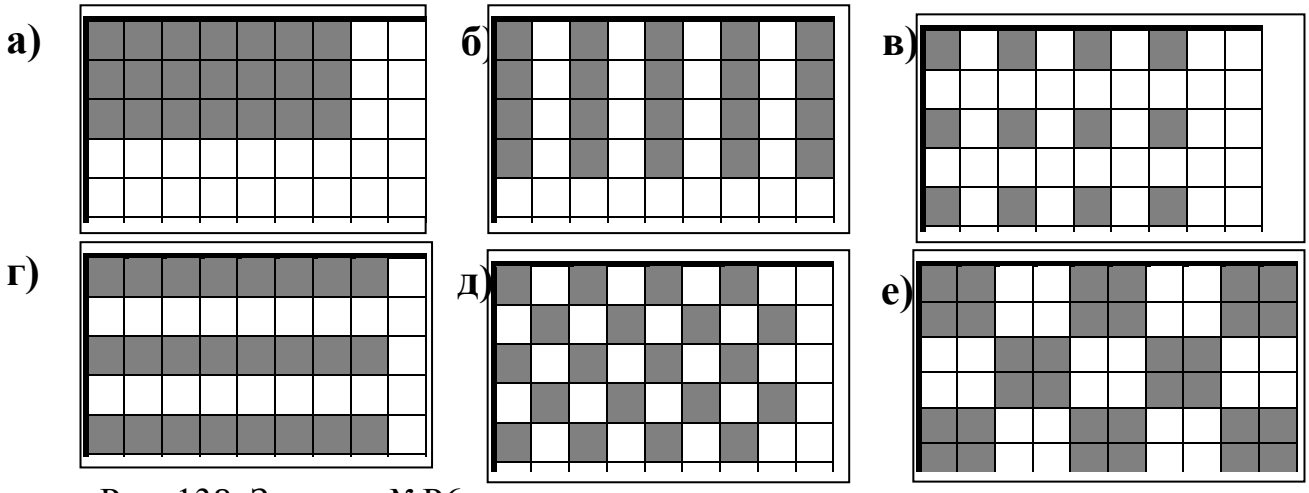


Рис. 138. Задание №Р6.

Умный Робот ищет стену

Мы с вами научились пользоваться циклом со счетчиком типа: ***нц-раз-кц***. Этот цикл удобен и необходим, если мы знаем точно, сколько раз будем выполнять команды тела цикла. На примере практической работы **«цикл с условием»** рассмотрим использование ***цикла,*** команды тела которого выполняются в зависимости от условия.

Практическая работа №2.7. Робот доходит до стены и останавливается.

Цель**:** на практике посмотреть, как Робот может использовать цикл с условием, для того что бы проанализировать обстановку и остановиться у стены.

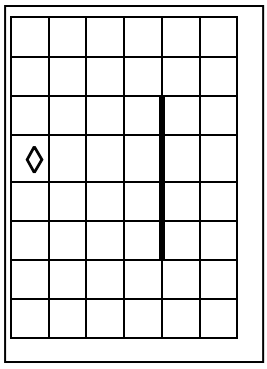


Рис. 139. Стартовая обстановка к п.р.№2.7.

Задача. Пусть у нас есть следующая стартовая обстановка: на поле есть вертикальная стена, Робот стоит слева от стены, но ***не известно на каком расстоянии от стены.*** Возможное положение Робота и стены представлено на рисунке 139. Требуется написать программу, позволяющую Роботу дойти до стены и не разбиться.

Решение. Если Робот будет все время двигаться вперед, то он разрушится. Будем программировать ***умного Робота***, который **не** будет идти вперед, если впереди стена.

Воспользуемся конструкцией ***цикла с условием.*** ***Словесно*** алгоритм можно сформулировать так: двигаемся вправо пока **не стена**, иначе– останавливаемся. На рисунке 140 представлена блок – схема этого алгоритма.

Вопрос 1. Назовите команды ***тела*** цикла и команды ***условия*** цикла.

Вопрос 2. Определите, какие команды есть у исполнителя Робот в среде КуМир для реализации этого алгоритма. Для этого в справочнике языка КуМир найдите команды ***проверки условия*** для Робота.

Вопрос 3. Почему для реализации этого алгоритма мы не можем воспользоваться циклом со счетчиком: **нц – раз – кц**?

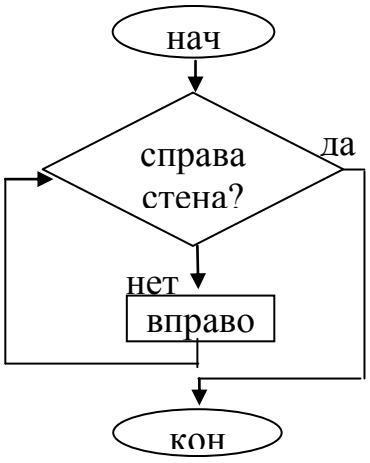


Рис. 140. Блок-схема к п.р. №2.7.

Рассмотрим алгоритмическую конструкцию **цикла с условием** типа: ***нц-пока-кц***. Команды тела цикла выполняются

до тех пор, пока ***условие цикла*** **истинно** (рис.141)**.**

Вопрос 4. Чем отличаются алгоритмические конструкции, представленные в блок-схемах на рисунках 140 и 141?

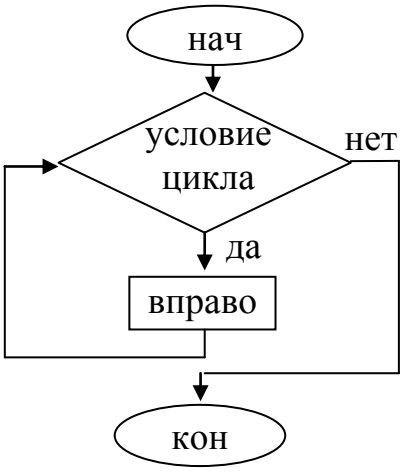
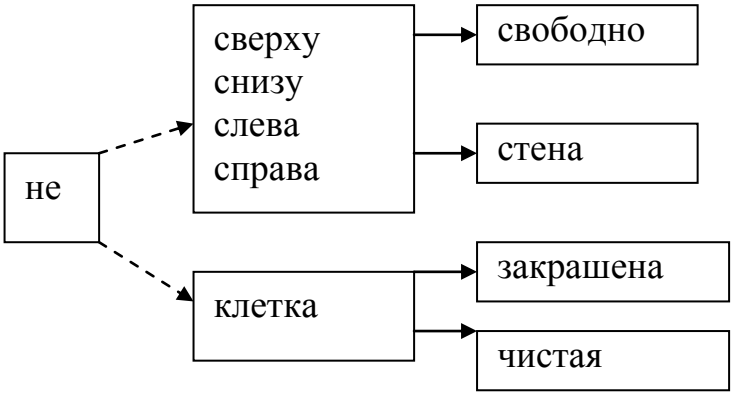


Рис. 141. Алгоритмическая конструкция цикла с условием.

Итак, нам необходимо правильно сформулировать условие цикла. Условие цикла является *логическим* выражением. Результатом выполнения условия может быть *истина* или *ложь*. В зависимости от этого исполнитель будет выбирать следующую команду выполнения. В блок – схеме это стрелки «да» и «нет», выходящие из условия.

Для написания условий в КуМир, как и в любом другом языке программирования, есть определенные правила, которые можно найти в справочнике языка. В КуМир можно пользоваться следующим конструктором условий (рис.142):

Конструктор условий



Отрицание «**не**» ставится в условие по необходимости.
Примеры условий:
а) *сверху свободно,*
б) ***не*** *клетка чистая,*
в) *слева стена*.

Рис. 142. Конструктор условий

Вопрос 5. С помощью **конструктора условий** (рис.142) сформулируйте *условие* цикла для решения задачи «Робот доходит до стены…». Блок – схема алгоритма представлена на рисунке 141.

Задание 1. Создайте стартовую обстановку: на поле есть вертикальная стена. Робот находится слева от стены на некотором расстоянии (рис.139).

Расположите на экране **три** окна: системы КуМир, Робота и стартовой обстановки, так чтобы было удобно (рис. 143).

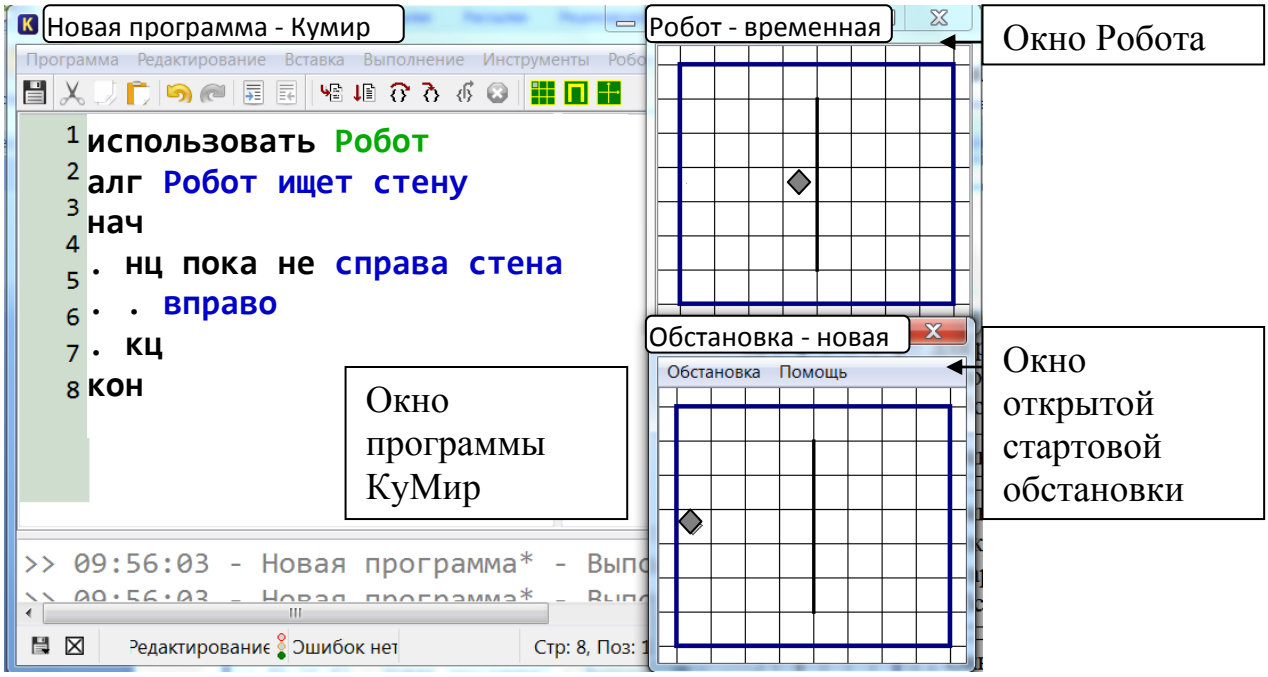


Рис. 143. Расположение окон для работы с исполнителем Робот на экране.

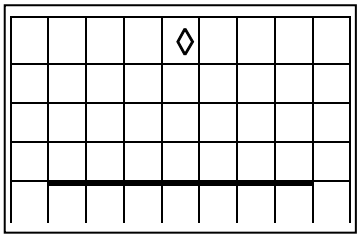
Запишите программу **Робот ищет стену** (рис. 143) без ошибок. Запустите программу на реализацию. Убедитесь, что программа работает.

При написании программы можно пользоваться пунктом меню Вставка, **нц-пока-кц** (рис. 127) или горячими клавишами (рис.144).

Вставка	**Рус.**	**En.**
алг-нач-кон	(Esc,А)	Esc,F
если-то -все	(Esc,Е)	Esc,T
выбор-при-все	(Esc,В)	Esc,D
иначе	(Esc,И)	Esc,B
нц-раз-кц	(Esc,Р)	Esc,H
нц-для-кц	(Esc,Д)	Esc,L
нц-пока-кц	(Esc,П)	Esc,G
нц-кц	(Esc,Ц)	Esc,W

Рис. 144. Горячие клавиши

Задание №Р7. Измените стартовую обстановку и напишите программу **Робот доходит до стены** при следующих стартовых условиях (рис.145):

а) 

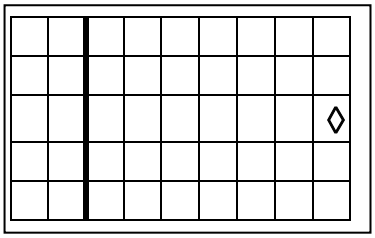
б) 

Рис. 145. Задание №Р7.

Задание №Р8. Выберите **равные** условия цикла и запишите в тетрадь:

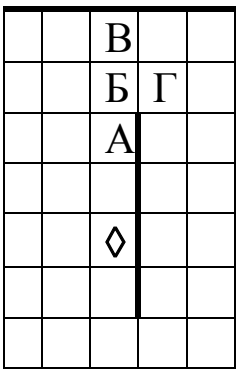
А) не справа стена = справа свободно
Б) сверху свободно = не сверху стена
В) не слева стена = сверху стена
Г) не клетка закрашена = клетка чистая
Д) не клетка чистая = клетка закрашена
Е) не клетка чистая = сверху свободно
Ж) снизу стена = сверху свободно

Робот идет вдоль стены

Практическая работа №2.8. Умный Робот движется вдоль стены.

Цель**:** найти алгоритмы, помогающие Роботу остановиться в заданной точке, двигаясь вдоль стены.

Стартовая обстановка: допустим, Робот стоит около стены слева, как показано на рисунке 146.



Требуется написать программу так, чтобы Робот двигался вдоль стены вверх и остановился около стены в последней клетке. На нашем рисунке это клетка **А.** Длина стены заранее неизвестна.

Рис. 146. Стартовая обстановка к п.р.№2.8

Рассмотрим программы, представленные на рисунке 147.

```
использовать Робот
алг алгоритм_1
нач
. нц пока справа стена
. . вверх
. кц
кон
```

```
использовать Робот
алг алгоритм_2
нач
. нц пока не сверху стена
. . вверх
. кц
кон
```

Рис. 147. Алгоритмы к п.р.№2.8

Вопрос 1. В какой точке остановится Робот при выполнении **алгоритма_1**?

Вопрос 2. При выполнении **алгоритма_2**?

Вопрос 3. Как можно исправить программу, чтобы Робот остановился в точке **А**?

Задание 1. Создайте стартовую обстановку, она должна соответствовать условию задачи. Запишите программу **алгоритм_1** на компьютере в системе КуМир. Запустите ее на исполнение. Убедитесь, что она работает.

Задание 2. Допишите в программу **алгоритм_1** одну команду так, чтобы в результате работы Робот оказался в точке А.

Задание 3. Исправьте программу так, чтобы Робот оказался в точке **Г**.

Задание №Р9. Напишите программы для решения задач, представленных на рисунке 148. Условие и требуемый результат показаны на схемах. Стартовая обстановка: Робот стоит около стены, размеры которой неизвестны. Точное положение Робота заранее не определено. Необходимо закрасить клетки, указанные в задании. Положение Робота по окончании программы может быть любым. Проверьте правильность работы программы, изменив размер стены и (или) положение Робота.

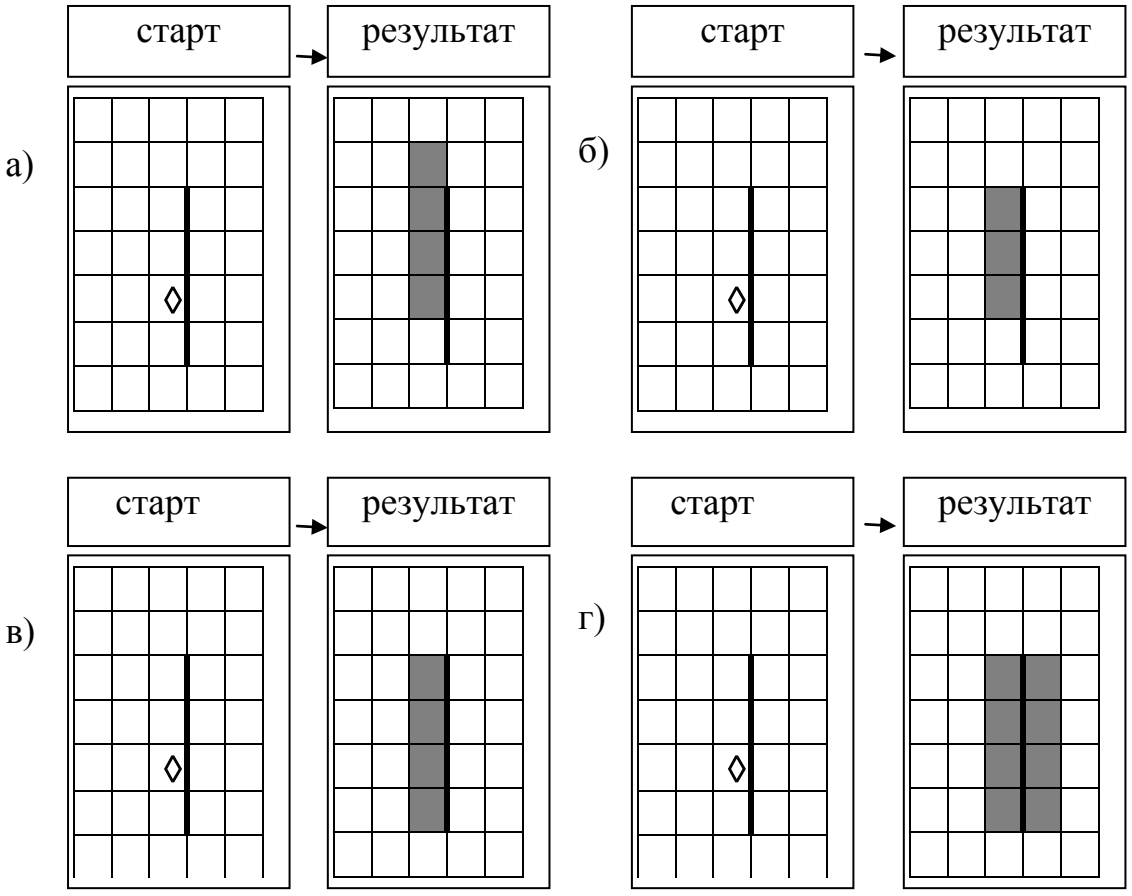


Рис. 148. Задание №Р9.

Задание №Р10. Стартовая обстановка: Робот стоит напротив вертикальной стены, размеры которой неизвестны. Расстояние от

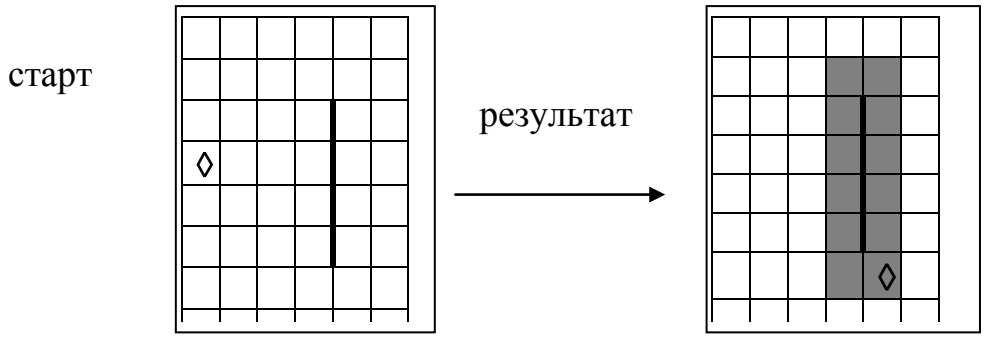


Рис. 149. Задание №Р10.

Робота до стены может быть любым. Задание: Роботу нужно дойти до стены и закрасить все клетки вокруг стены. Пример стартовой обстановки и результата работы программы показан на рисунке 149.

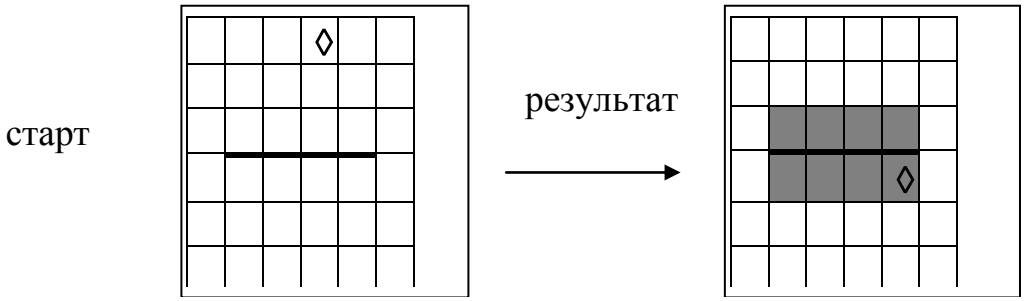


Рис. 150. Задание №Р11.

Задание №Р11. Стартовая обстановка: Робот стоит напротив горизонтальной стены, размеры которой неизвестны. Расстояние от Робота до стены может быть любым. Задание: Роботу нужно дойти до стены и закрасить все клетки вокруг стены. Пример стартовой обстановки и результата работы программы показан на рисунке 150.

Практическая работа №2.9. Умный Робот обходит прямоугольный забор.

Цель: научиться применять циклы с условием для решения задач.

Задача. Написать программу для обхода Роботом пути вокруг прямоугольного забора при заданной стартовой обстановке (рис.151).

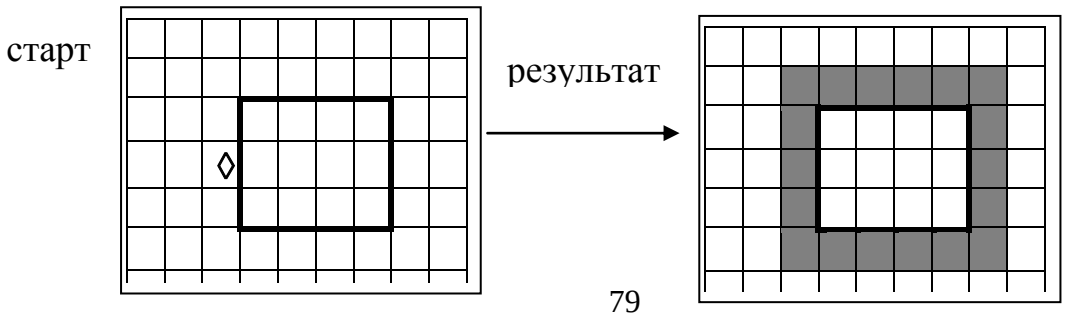


Рис. 151. Стартовая обстановка и задание к п.р.№2.9.

Стартовая обстановка: Робот стоит около стены слева, с наружной стороны забора. Размер прямоугольника неизвестен.

Решение. Посмотрите на код программы **обход забора** (рис. 152)**.** Для удобства программа представлена с номерами строк.

Как видно, в программе дважды используется цикл «пока».

Вопрос 1.Результат работы этой программы представлен на одном из рисунков: **а, б** или **в** (рис.153). Как вы думаете, на каком?

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

Рис. 152. Код программы **обход забора.**

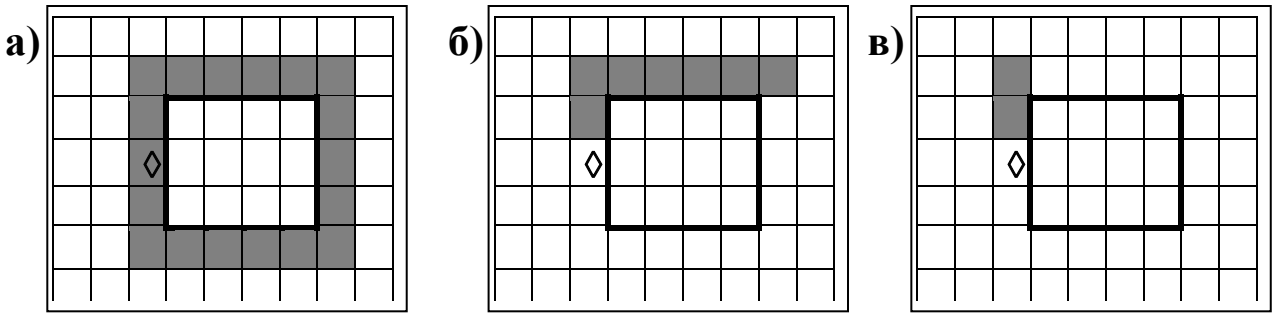


Рис. 153. Рисунки к вопросу 1 по коду программы **обход забора**.

Вопрос 2. Как нужно дописать программу, чтобы она работала правильно? Проверьте правильность предположения на компьютере.

Задание 1. Запишите на компьютере правильно работающую программу для решения задачи закрашивания клеток вокруг забора**.** Для удобства располагайте на экране три окна: системы КуМир, Робота и стартовой обстановки (рис.143).

Задание №Р12. Исправьте стартовую обстановку: Робот стоит напротив горизонтальной стены сверху на **неизвестном** расстоянии от стены. Старт и результат работы показан на рисунке 154.

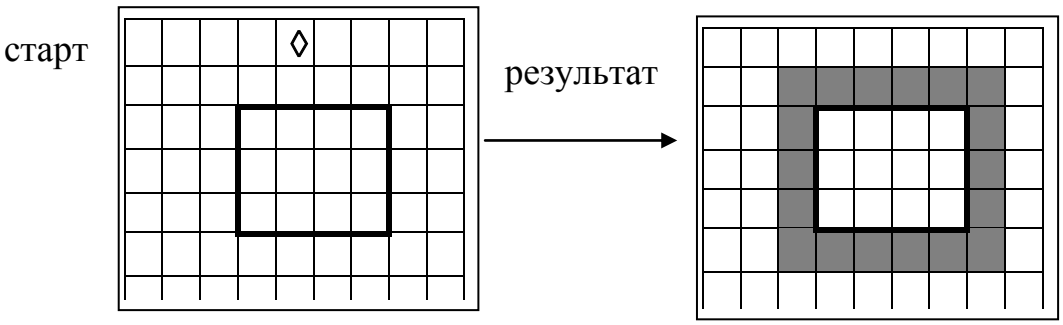


Рис. 154. Задание №Р12.

Задание №Р13. Робот стоит около стены справа, внутри забора.

Напишите программу так, чтобы Робот обошел территорию по внутреннему периметру прямоугольного забора. Размер прямоугольника неизвестен. Разбиться Робот не должен. Старт и результат работы показан на рисунке 155.

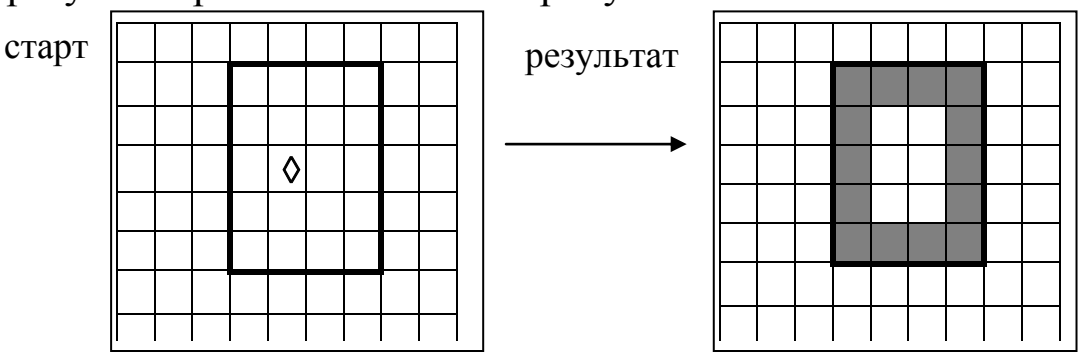


Рис. 155. Задание №Р13.

Задание №Р14. На бесконечном поле имеется лестница, которая поднимается вверх слева направо. Высота каждой ступени — 1 клетка, ширина — 2 клетки. Количество ступенек, ведущих вверх неизвестно. Лестница заканчивается горизонтальной ступенькой. Робот находится в клетке, расположенной в начале спуска. Напишите для Робота алгоритм, закрашивающий все клетки, расположенные непосредственно над ступеньками (рис.156).

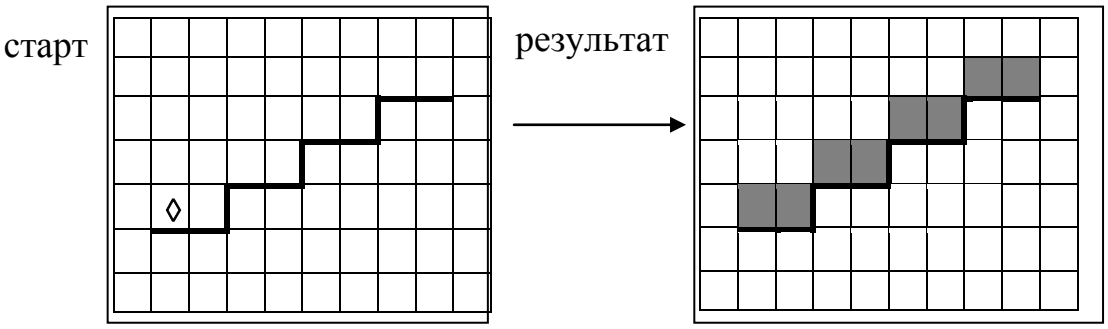


Рис. 156. Задание №Р14.

Логические операции

Для постановки условий в практической жизни мы часто используем союзы «**и**», «**или**», «**не**». Пример предложений с этими союзами вы можете без труда привести самостоятельно. В программировании также нельзя обойтись без сложных (составных) условий.

Логических операций в системе КуМир всего три:

Название	**Форма записи**	**Пример использования**
конъюнкция	и	**справа стена *и* сверху свободно**
дизъюнкция	или	**справа стена *или* слева стена**
отрицание	не	***не*** **справа стена**

Результатом логической операции может быть «**истина**» или «**ложь**».

В **блок – схеме** логические операции пишут внутри **блока условия**: Блок условия используется в командах **ветвления** типа «***если***» и в командах **цикла** типа «***пока***» (рис.157).

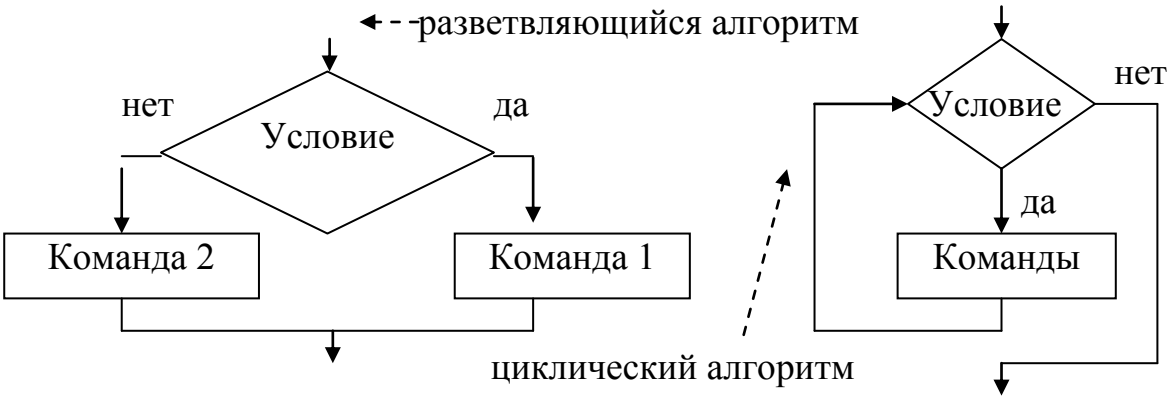


Рис. 157. Использование условия в алгоритмических конструкция ветвления и цикла

Практическая работа №2.10. Логические операции.

Цель: научиться применять логические операции в КуМир для решения задач.

Рассмотрим следующую задачу. На бесконечном поле есть две горизонтальные стены, образующие коридор, шириной в одну клетку. Справа коридор ограничен стеной. В стенах есть проходы, размеры которых неизвестны. Робот стоит слева, в первой клетке коридора. Необходимо закрасить клетки, стоящие напротив проходов в стенах. Старт и результат работы показан на рисунке 158.

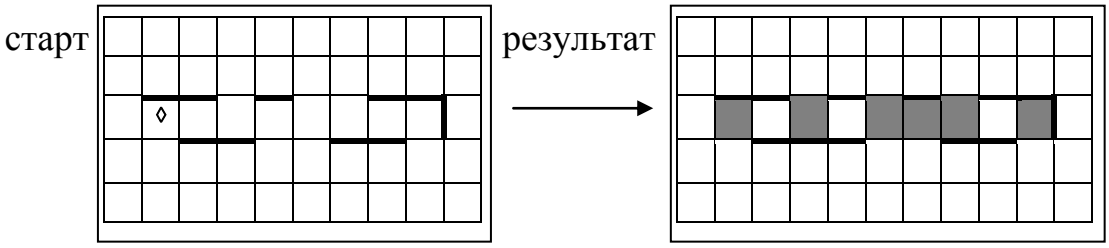


Рис. 158. Задание к практической работе 2.10.

Посмотрим на блок – схему (рис. 159). При решении этой задачи используется следующая алгоритмическая конструкция: в оператор цикла **нц-пока-кц** вложен оператор ветвления **если-то-иначе-все.**

Вопрос 1. Какой ***логический*** оператор используется в команде ветвления?

Задание 1. Перепишите код программы **логические операции** (рис.160). Запустите ее на исполнение.

Вопрос 2. Почему последняя клетка коридора осталась чистой?

Задание 2. Исправьте код программы так, чтобы программа обрабатывала последнюю клетку.

Задание 3. Закрасьте только те клетки, которые ограничены стенами с двух сторон: сверху и снизу.

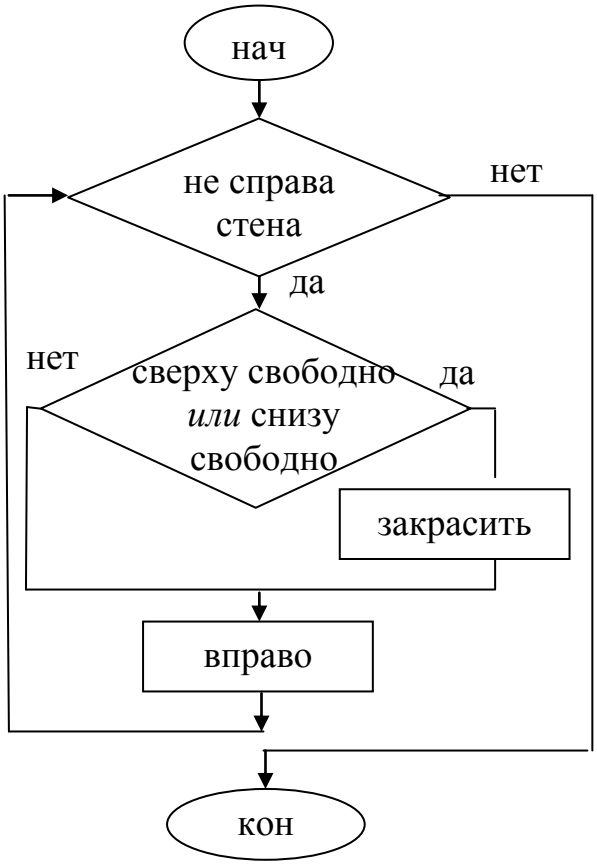


Рис. 159. Блок-схема к п.р. №2.10.

```
использовать Робот
алг логические операции
нач
. нц пока не справа стена
. .если сверху свободно или снизу свободно
. . . то
. . . . закрасить
. . все
. . вправо
. кц
кон
```

Рис. 160. Код программы **логические операции.**

Задание №Р15. Закрасьте клетки сверху ступенек лестницы, ведущей вверх. Размер ступенек неизвестен. Робот стоит на нижней ступеньке. Последняя ступенька – горизонтальная. Страт и результат работы показан на рисунке 161. Количество ступенек может быть любым.

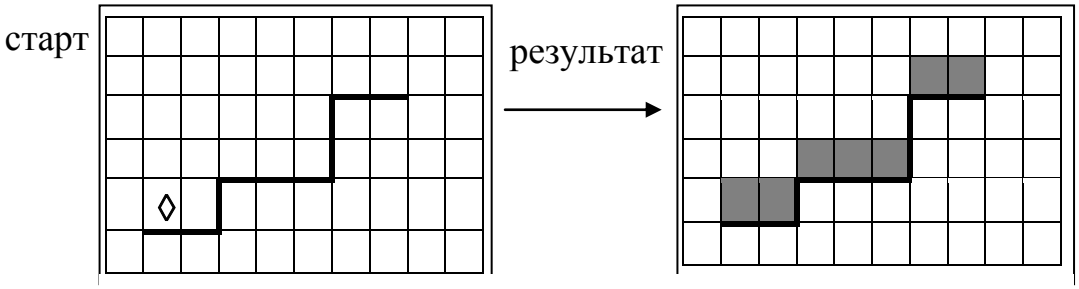


Рис. 161. Задание № Р15.

Обход поля Роботом

Практическая работа №2.11.Обход поля Роботом.

Цель: научиться применять циклы с условием для решения задач.

Задача. Составить алгоритм обхода поля для Робота, ограниченного стенами, размеры которых заранее неизвестны. Стен внутри поля нет. Робот должен закрасить все клетки поля. Стартовая обстановка: Робот стоит в верхнем левом углу.

Мы уже умеем проходить первую строку поля с помощью цикла «пока не справа стена - вправо». Рассмотрим две траектории на рисунке 162. В случае **а** «возвращаемся по следующей строке», а в случае **б** «возвращаемся по той же строке».

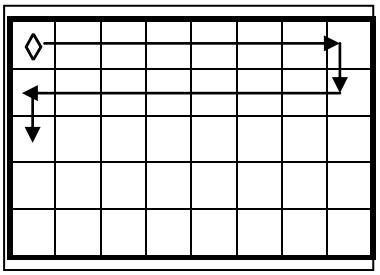
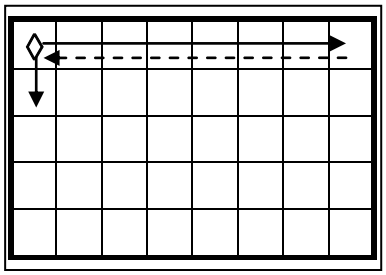
а)  **б)** 

Рис. 162. Возможные траектории обхода поля Роботом.

Вопрос 1. Подумайте, по какой траектории было бы лучше двигаться настоящему Роботу? Обоснуйте ответ.

Напишем программу **обход поля** для случая **А** (рис. 163).

```
 1  использовать Робот
 2  алг обход поля
 3  нач
 4  . закрасить
 5  . нц
 6  . . нц пока не справа стена
 7  . . . вправо
 8  . . . закрасить
 9  . . кц
10  . . нц пока не слева стена
11  . . . влево
12  . . кц
13  . . |закрасить
14  . . вниз
15  . кц при снизу стена
16  кон
```

Рис. 163. Код программы **обход поля.**

В программе используется алгоритмическая конструкция «***цикл внутри цикла***». Здесь мы использовали так называемый цикл «**с постусловием**», то есть проверяющим условие после первого прохождения, как это показано на блок – схеме. В языке КуМир этот цикл реализуется с помощью конструкции: **нц- кц -при** Результат работы этой программы и конечное положение Робота вы видите на рисунке 164.

Вопрос 2. Почему некоторые клетки остались незакрашенными?

Задание 1. Исправьте программу так, чтобы она закрашивала все клетки поля. Запишите программу в КуМир.

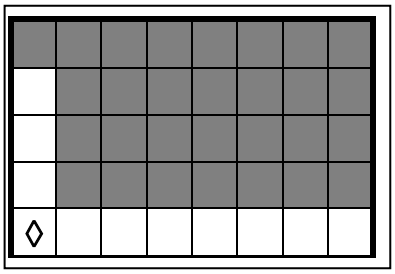


Рис. 164. Результат работы программы **обход поля**.

Задачи с переменными

Практическая работа №2.12. Использование переменных.

Цель: на практике изучить возможности использования переменных в среде Кумир. Оценить необходимость и значимость их использования.

Переменная **–** это ячейка памяти, имеющая собственное имя. Переменные могут быть **целыми** [цел ***a***], **дробными** [вещ ***a***], **символьными** [сим ***a***], **литерными** [лит ***a***] и **логическими** [лог ***a***].

Для начала работы с переменной ее нужно ***объявить***. Делается это для того, чтобы компьютер знал, сколько памяти нужно выделить. Для вещественных (дробных) переменных, например, в КуМир выделяется четыре байта памяти, а для целых – всего два. Переменные объявляются в начале программы.

Задача. Посчитать длину стены. Стартовая обстановка: Робот стоит снизу около стены внизу (рис.165).

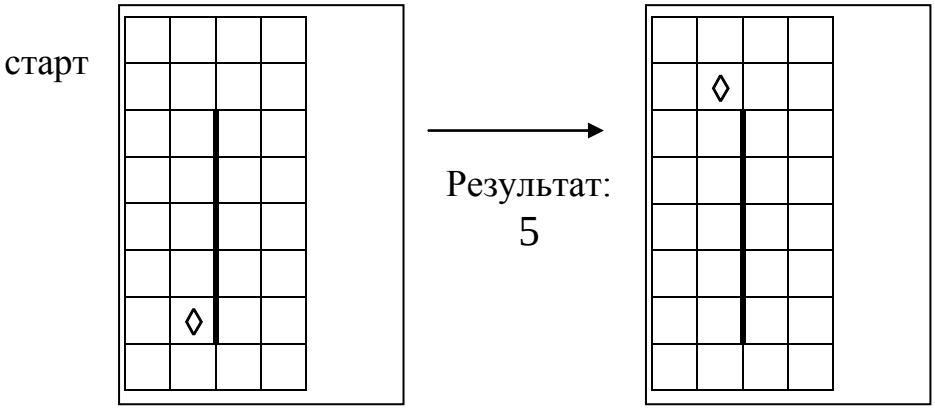


Рис. 165. Задание к п.р.№2.12.

Решение. Посмотрите на код программы **длина_стены.** В начале программы переменная **длина** объявлена как целая переменная: **`цел длина`**.

Команда ***длина:=длина+1*** для компьютера означает, что машина загрузит в оперативную память значение переменной ***длина***, сложит

ее с единицей, а затем запишет новое значение в ту же ячейку памяти под именем «***длина***».
Результат работы программы будет выведен в специальную строку (рис. 166).

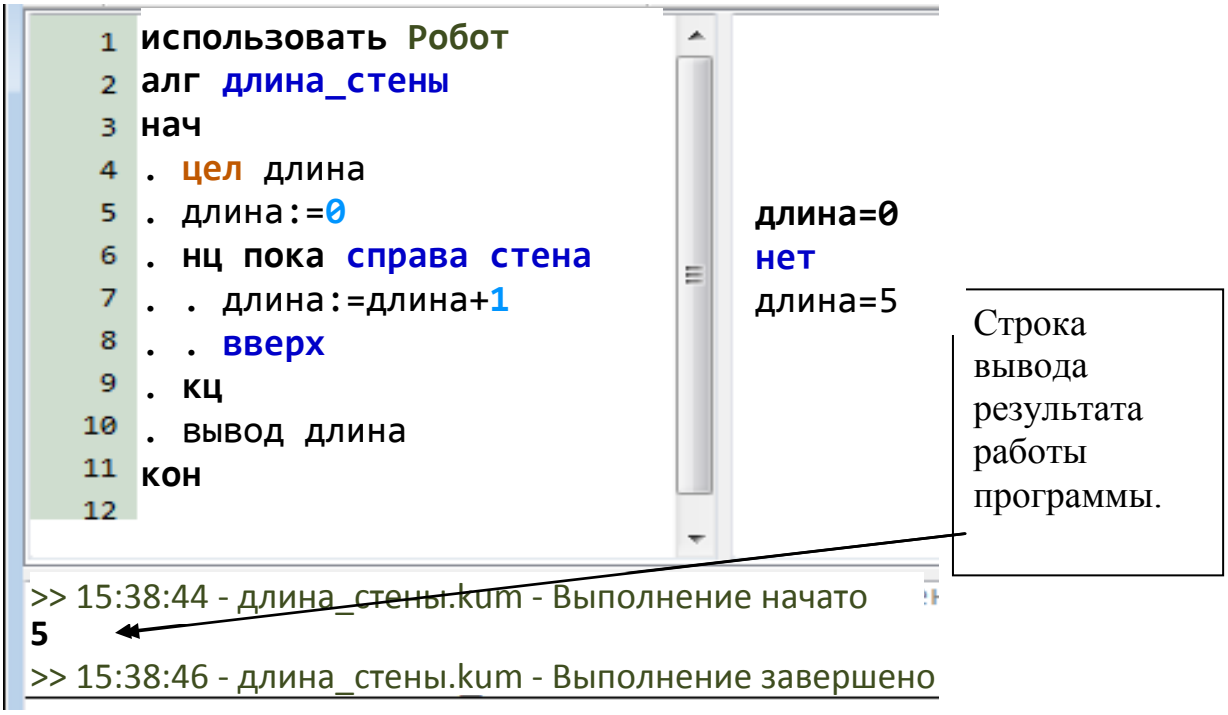


Рис. 166.Скрин экрана с кодом программы **длина_стены.**

Схематично вычисления компьютера можно представить так:

длина **:=** **длина** +1

вычисление в оперативной памяти

Записали новое значение в ту же ячейку

Рис. 167. Схема операции присваивания.

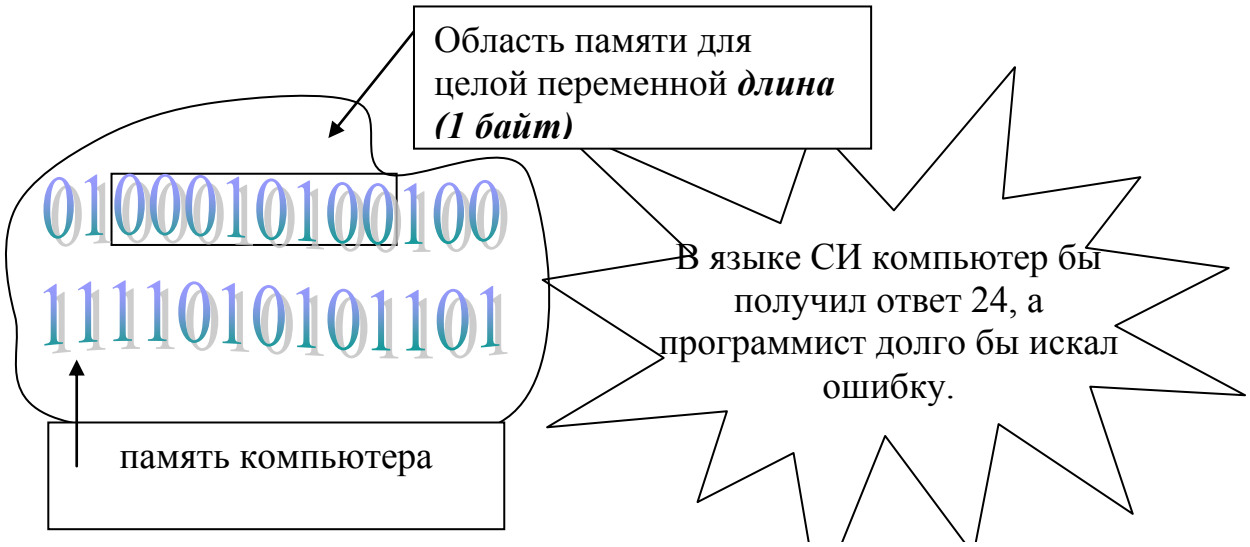
Предыдущее значение ячейки длина при этом стирается, а на его место записывается новое значение. Знак «**:=**» в программировании называется «**присвоить**».

Обратим внимание на строку с номером 5 в программе: ***длина :=0***.
Эта очень важная строка. Если ее убрать, то программа выдаст ошибку:

```
ОШИБКА ВЫПОЛНЕНИЯ: Неопределенное значение (8002)
```

Это значит, что компьютер не может выполнить команду ***длина:=длина+1***, если предыдущее значение переменной ***длина*** не было определено.

Примечание. Возможно, вы слышали про язык «СИ»? Так вот этот бы язык ошибку не выдал, а прибавил бы единицу к тому, что у него случайно было записано в этой области памяти. Схематично можно нарисовать так, как это показано на рисунке :



Раньше в этой области памяти была, например, музыка, а теперь эта область переменной «***длина»,*** где остались записанные нули и единицы, составляющие в нашем примере число 20.

Рис. 168. Область памяти компьютера, отведенная для переменной ***длина.***

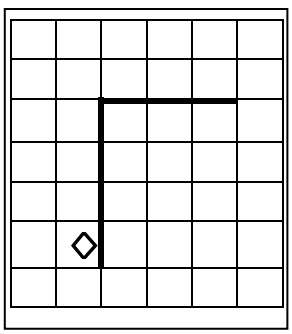
Важно! Язык КуМир сразу приучает начинающих программистов задавать переменным начальные значения!

Задание 1.Запишите код программы **длина_стены** (рис.166) и запустите его на выполнение. Измените длину стены в стартовой обстановке. Убедитесь, что программа точно выдает длину стены.

Задание №Р16. Посчитайте длину изгороди, на рисунке 169.

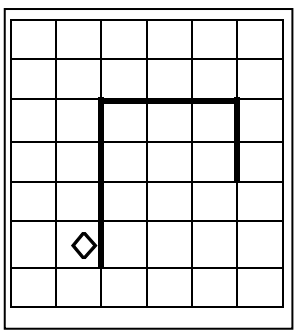
Размеры вертикальной и горизонтальной стен могут быть любыми и заранее неизвестны. Робот стоит слева от вертикальной стены снизу.

А



Ответ 7.

Б



Ответ 9.

Рис. 169. Задание №Р16.

Задание №Р17. На противоположную сторону стены.

Пусть дана стартовая обстановка: Робот стоит около стены слева. Длина стены может быть любой. Роботу необходимо попасть на противоположную сторону стены. Стартовая обстановка и результат работы показаны на рисунке 170.

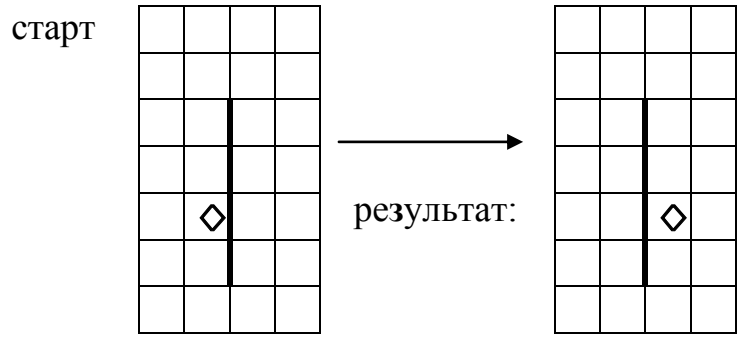


Рис. 170. Задание №Р17.

Решение. Будем обходить стену сверху. Очевидно, без переменной, которая запомнит количество пройденных вверх клеток, нам не обойтись. Назовем эту переменную ***длина.*** Объявим её как ***целую*** переменную. Первым циклом посчитаем количество шагов до конца стены.

Сместимся на одну клетку вправо и пройдем столько шагов, сколько насчитали в первом цикле. Итак, нам осталось перейти на другую сторону стены и пройти количество шагов, равное значению переменной ***длина.***

Запишите и исполните программу **противоположная_сторона**в системе КуМир (рис. 171).

```
1   использовать Робот
2   алг противоположная_сторона
3   нач
4   . цел длина
5   . длина:=0
6   . нц пока справа стена
7   . . длина:=длина+1
8   . . вверх
9   . кц
10  . вправо
11  . нц длина раз
12  . . вниз
13  . кц
14  кон
```

Рис. 171. Код программы **противоположная_ сторона.**

Задание №Р18. а) Закрасьте клетки, как это показано на рисунке 172 а. Робот должен оказаться на противоположной стороне стены и обойти стену сверху. **б)** Обойдите стену снизу и закрасьте клетки, как это показано на рисунке 172 б.

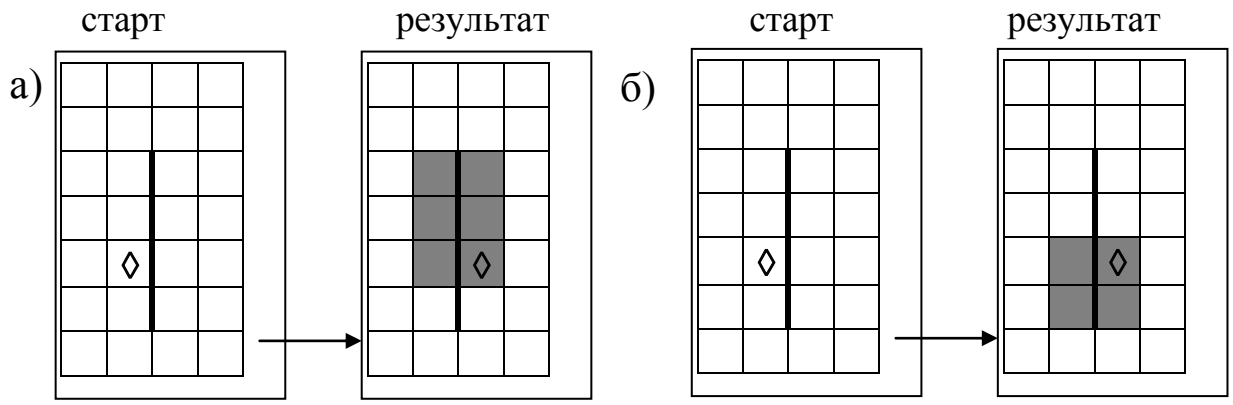


Рис. 172. Задание №Р18.

Задание №Р19. Создайте стартовую обстановку: поставьте одну стену на поле произвольной длины (рис.173). Поставьте Робота около стены слева. Напишите программу: посчитайте длину стены.

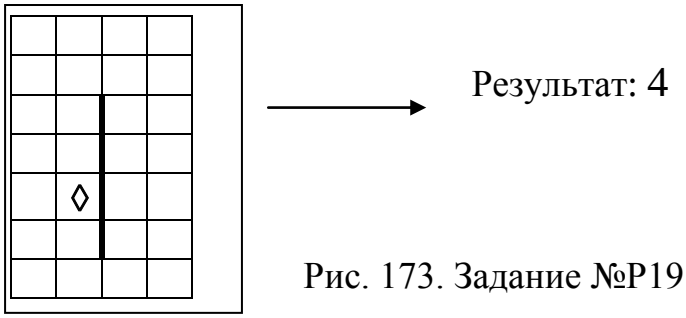
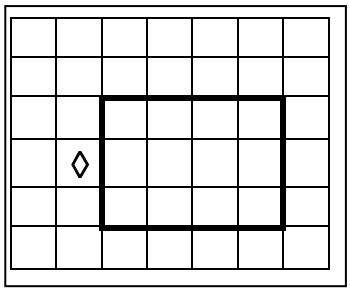


Рис. 173. Задание №Р19.

Задание №Р20. Посчитайте длину прямоугольного забора. Ответ выведите на экран. Робот стоит около забора слева, как показано на рисунке 174. Длины стен и конечное положение Робота могут быть любыми.

Задание №Р21. Посчитайте площадь прямоугольника, ограниченного забором. Ответ выведите на экран. Робот стоит около забора слева, как показано на рисунке 174. Длины стен и конечное положение Робота могут быть любыми.



Результат к заданию №Р20: 14

Результат к заданию №Р21: 12

Рис. 174. Робот стоит около прямоугольного забора слева.

Умный Робот закрашивает клетки с радиацией

Практическая работа №2.13. Закрашиваем клетки с радиацией.

Цель: на практике ознакомиться с алгоритмической конструкцией условия внутри цикла.

Задача. Пусть Робот закрасит все клетки **первой** строки на поле, ***имеющие радиацию***. Создадим для Робота стартовую обстановку, в которой некоторые клетки первой строки на поле будут иметь радиацию. Радиация может принимать значение от 0 до 100 включительно.

Решение. Такую задачу вы уже решали с помощью **Пульта** Робота в практической работе №2.3. Решим ее программными методами без использования **Пульта** для строки любого размера, ограниченной стенами и с любыми значениями радиации в клетке. Робот стоит в первой клетке строки.

◊1	22	0.3	0	5	0	77	**0**

В приведенном примере Робот должен закрасить клетки с номерами: 1, 2, 3, 5, 7.

Сформулируем алгоритм ***словесно*** для первой строки прохождения Робота по полю. *Пока* ***не справа стена****, двигаться* ***вправо****, измерить* ***радиацию****. Если* ***радиация>0****, то клетку* ***закрасить.***

Как видно, в словесном алгоритме есть слово ***если***. Вспомним, что эта конструкция в алгоритмических языках реализуется с помощью *оператора условного перехода*.

Вопрос 1. Найдите в представленной блок – схеме (рис. 175) команду ***ветвления***. Какая форма оператора ветвления (полная или неполная) будет использована в программе?

Вопрос 2. Какие клетки закрасит Робот в результате реализации этого алгоритма, при следующей стартовой обстановке на первой полосе?

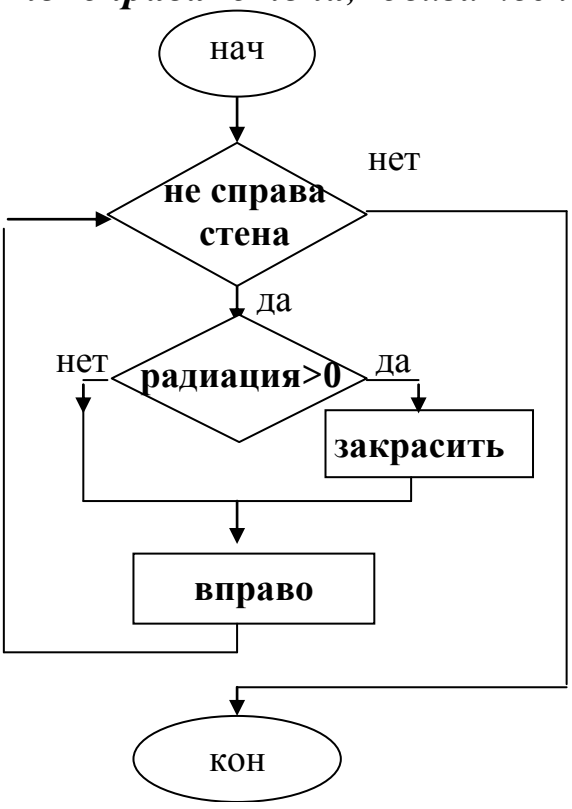


Рис. 175. Блок – схема к п.р.№2.13.

а)

◊1		3		5			

б)

◊	2	3		5			8

Значок ◊ обозначает начальное положение Робота в первой клетке. Цифры означают значение радиации в клетке. В пустых клетках радиация равна нулю (нет радиации).

Вопрос 3. Как нужно изменить алгоритм для того, чтобы он обрабатывал все клетки полосы?

Задание 1. Постарайтесь самостоятельно написать программу.

Важно! В программировании очень часто приходится обращать внимание на начало и конец обработки данных. То есть нужно обязательно **проверить** «*нос* ***и*** *хвост*» программы**.** Программа считается работающей, если нет случаев ее неправильной работы при любых заданных условиях.

Вопрос 4. Посмотрите на код программы **поиск радиации** (рис. 176). Этот код соответствует блок- схеме на рисунке 175. Какие команды составляют тело цикла?

```
1  использовать Робот
2  алг поиск радиации
3  нач
4  . нц пока не справа стена
5  . . если радиация>0 то
6  . . . . закрасить
7  . . все
8  . вправо
9  . кц
10
11 кон
```

Рис. 176. Код программы **поиск радиации.**

Вопрос 5. Найдите в коде программы (рис.176) команду ветвления. Прочитайте (полностью) код этой команды.

Вопрос 6. Назовите служебные слова, которые входят в состав оператора ветвления?

Вопрос 7. Какая форма (полная или неполная) используется в операторе ветвления?

Вопрос 8. Почему перед командой **закрасить** компилятор поставил дополнительные точки?

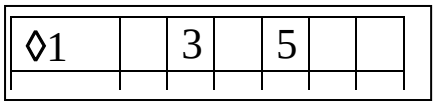
Задание 2. Создайте новую стартовую обстановку без стен, Робот в верхнем левом углу. **Некоторым** клеткам первой строки задайте радиацию (любую, большую нуля). Сделайте это с помощью щелчка левой кнопки мыши на клетке.

Запишите код программы **поиск радиации**, отправьте программу на исполнение. Допишите код программы так, чтобы Робот обрабатывал все клетки первой строки.

Задание 3. Проверьте работоспособность исправленной вами программы в следующих тестовых случаях (рис.177):

а) имеют радиацию 1, 3 и 5 клетка;

б) имеют радиацию все клетки строки.

а) 

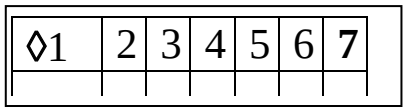
б) 

Рис. 177. Возможные значения радиации для **задания 3а** и **задания 3б**.

Задание №Р22. Стартовая обстановка: Робот стоит в верхнем левом углу внутри поля, ограниченного стенами со всех сторон, других стен на поле нет. Некоторые клетки имеют радиацию. Задание: закрасить все клетки поля с радиацией выше нуля.

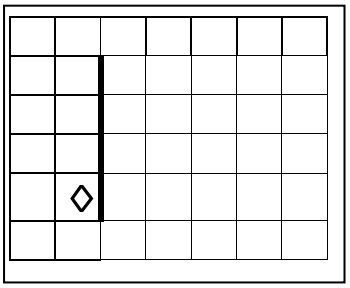
а) в первых двух строках;

б) на всем поле, размеры которого заранее неизвестны.

Задание №Р23. Запишите программу для Робота, который закрашивает все точки с **радиацией >0** вдоль стены, размеры которой заранее неизвестны при стартовой обстановке:

а) Робот стоит около стены слева в нижней клетке.

б) Робот стоит около стены слева.

а) 

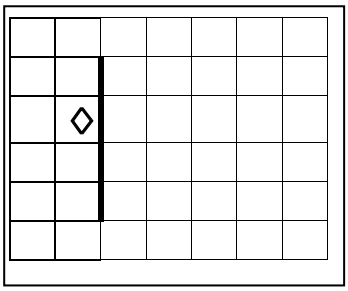
б) 

Рис. 178. Возможные стартовые обстановки для случая ***а*** и ***б*** задания №Р23.

Задание №Р24. Посчитать **количество** клеток с положительной радиацией

а) в полосе;

б) на поле;

в) вдоль стены, размеры которой заранее неизвестны. Робот стоит около стены справа (рис. 178б).

Поиск максимального элемента

Практическая работа №2.14. Максимальная радиация.

Цель: на практике научиться решать задачи с алгоритмической конструкцией условия внутри цикла.

Задача. Найдем максимальное значение радиации и выведем **на экран**.

Решение. Для начала решим задачу поиска ***максимального элемента в горизонтальной полосе***. Пусть по условию задачи в

горизонтальной полосе есть обязательно хотя бы одна клетка с положительной радиацией. Клеток с одинаковой максимальной радиацией может быть одна или несколько.

Для задачи поиска нам понадобится переменная, в которой мы будем хранить максимальное значение. Назовем эту переменную **m**. Алгоритм можно представить в виде блок- схемы (рис.179):

В этой блок- схеме не хватает одной команды: **вывод m.**

Вопрос 1. В каком месте блок – схемы нужно расположить команду **вывод m**, чтобы она правильно выводила значение максимальной радиации на полосе?

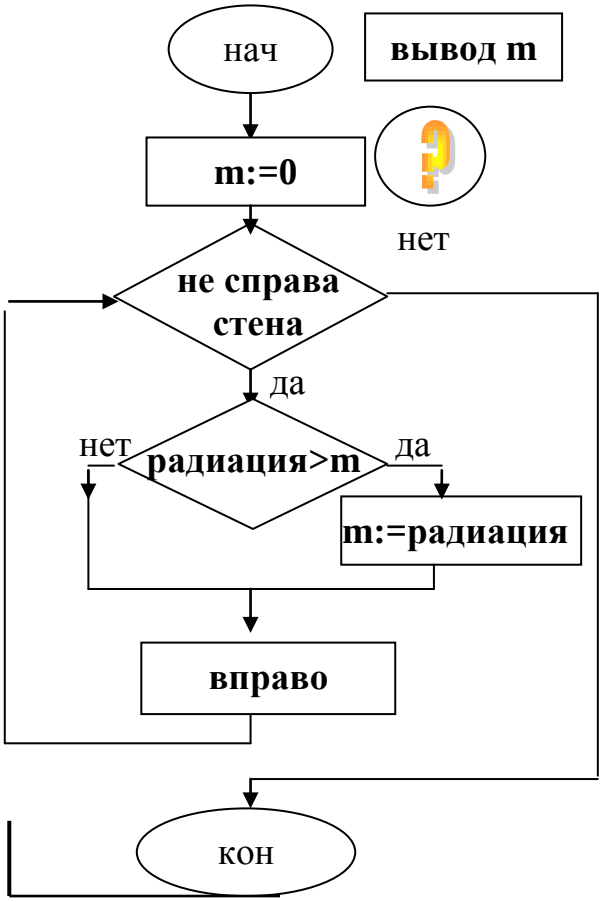


Рис. 179. Блок- схема алгоритма поиска максимального элемента.

Пусть по условию задачи в горизонтальной полосе есть обязательно хотя бы одна клетка с положительной радиацией.

Прочитайте код программы, **поиск максимального** (рис.180) и ответьте на вопросы**.**

```
1  использовать Робот
2  алг поиск максимального
3  нач
4  вещ m
5  . нц пока не справа стена
6  . . если радиация>m то
7  . . . . m:=радиация
8  . . все
9  . вправо
10 . кц
11 . вывод m
12 кон
```

Рис. 180. Код программы **поиск максимального.**

Вопрос 2. Назовите номер строки, после которой необходимо задать начальное значение переменной **m**? Какое начальное значение можно присвоить переменной **m**?

Возможно, вы уже догадались, что даже после этого исправления программа выдает правильные ответы **не во всех случаях**.

Вопрос 3. Выберите из примеров стартовых обстановок те, в которых эта программа будет работать

правильно.

а) | 5 | 0 | 7 | 8 | 0 |

б) | 5 | 0 | 5 | 5 | 0 |

в) | 5 | 0 | 7 | 0 | 5 |

г) | 8 | 6 | 7 | 1 | 8 |

д) | 5 | 0 | 7 | 0 | 8 |

Допишите программу так, чтобы не было случаев ее неправильной работы. Проверьте работу программы на компьютере.

Вопрос 4. Как изменится программа, если нужно найти минимальную радиацию?

Вопрос 5. Как найти минимальную температуру? Замечание температура бывает отрицательной.

Вопрос 6. Как нужно дописать код программы, чтобы в горизонтальной полосе она закрашивала все клетки со значением, равным максимальному?

Задание №Р25. Посчитать количество клеток с радиацией, равной максимальной в горизонтальной полосе.

Задание №Р26. Посчитать разницу между максимальным и минимальным значением радиации в горизонтальной полосе.

Задание №Р27. Посчитать максимальную температуру в горизонтальной полосе.

Задание №Р28. Определить, будет ли точка с максимальной радиацией иметь максимальную температуру. Если точек с максимальной радиацией несколько, то определить, есть ли хотя бы в одной из них максимальная температура.

Задание №Р29. Посчитать суммарную радиацию

а) в полосе;

б) на всем поле;

в)найти среднюю радиацию на горизонтальной полосе.

Робот определяет два или три максимальных значения радиации

Практическая работа №2.15. Два максимальных значения радиации.

Цель: найти алгоритмические отличия решения этой задачи от задачи поиска одного максимального элемента.

Пусть нам требуется написать программу для Робота, который находит два наибольших, но разных, значения радиации на поле.

Воспользуемся тем фактом, что минимальная радиация не может быть отрицательной. Таким образом, если первоначально мы возьмем максимальное значение равное нулю, то после прохождения поля и

измерения радиации либо это значение изменится на большее, либо так и останется равным нулю. Последнее будет означать, что радиации нет на всей полосе.

На рисунке 181 приведена «не идеальная» программа, которая находит два наибольших, **но разных** значения на горизонтальной полосе: **max1**-наибольшее значение и **max2** – значение радиации, следующее за наибольшим.

💻**Задание1**. Перепишите программу **два наибольших значения** (рис. 181) без ошибок и проверьте ее работу на примерах:

а)

5	0	7	0	8

б)

8	6	7	1	8

в)

5	0	7	8	0

💻**Задание2**. Допишите ее так, чтобы не было случаев ее неправильной работы.

✎**Задание3**. Нарисуйте блок – схему исправленной программы в тетрадь.

```
1  использовать Робот
2  алг два наибольших значения
3  нач
4  . вещ max1, max2
5  . max1:=0
6  . max2:=0
7  . нц пока не справа стена
8  . . если радиация> max1 то
9  |нашли клетку с радиацией, большей max1
10 . . . . max2:=max1
11 . . . . max1:= радиация
12 . . . иначе
13 |радиация не больше max1
14 . . . . если радиация> max2 то
15 . . . . . . max2:= радиация
16 . . . . все
17 . . все
18 . . вправо
19 . кц
20 . вывод max1, " ",max2
21 кон
```

Рис. 181. Код программы **два наибольших значения.**

Задание №Р37. Исправьте программу, чтобы она находила два наибольших значения, но они могли бы быть одинаковыми. Проверьте решение на примере Б.

Задание №Р30. Найдите три наибольших значения на полосе.

Задание №Р31. Закрасьте все клетки на поле, значения радиации которых больше или равно значению, следующему за максимальным.

Робот идет по горизонтальной полосе и записывает радиацию в таблицу

Очень часто возникают задачи, в которых Роботу требуется записать данные в таблицу для последующего анализа. В языках программирования для таких однородных данных предусмотрены специальные структуры, называемые ***массивами***. **Таблица** – это очень удобная форма организации хранения данных. Выглядит она так:

№	1	2	3	4	5
значение	5	0	7	8	0

Для каждой ячейки есть свой номер, и по этому номеру можно определить значение ячейки. В языке КуМир такая одномерная таблица объявляется следующим образом: **целтаб k[1:5],** что означает, что таблица будет иметь имя ***k***, в ней будут храниться целые элементы, элементов будет 5 и их номера будут от 1 до 5.

Практическая работа №2.16. Таблицы.

Цель: ознакомиться с правилами обращения с данными в таблицах в языке программирования КуМир.

Задача. Найти максимальное значение радиации в полосе, используя таблицу.

Решение. Пусть Робот пройдет по горизонтальной полосе длиной 5 клеток, запишет данные о радиации в таблицу, а затем по значениям из таблицы найдет максимальный элемент и выдает на экран значение максимальной радиации.

```
использовать Робот
алг работа с таблицей
нач
. вещ таб A[1:5]
. цел i
. вещ m
. нц для i от 1 до 5
. . A[i]:=радиация
. . вправо
. кц
. m:=0
. нц для i от 1 до 5
. . если A[i]>m то m:=A[i]
. . все
. кц
. вывод m
кон
```

Рис. 182. Код программы **работа с таблицей.**

Прочитайте код программы **работа с таблицей**, представленный на рисунке 182. Здесь в программе использован цикл **для i от 1 до 5**. Переменная ***i*** должна быть обязательно целой, а значения радиации – вещественными.

Вопрос 1.Как нужно дописать программу, чтобы она выдавала **номер** клетки с максимальной радиацией?

Вопрос 2.Как организовать запись данных в таблицу, если размеры горизонтальной полосы заранее неизвестны, но есть возможность объявить размеры таблицы заведомо большей длины? Учтем, что для программы КуМир максимальные размеры поля: 16 столбцов на 10 строк.

Задание №Р32. Решите задачи №Р25-№Р29 при помощи таблиц. Для этого при первом проходе Робота по полосе запишите данные в таблицу.

Задание №Р33*. Найдите в полосе самый длинный участок с радиацией большей нуля. Выведите его длину- количество клеток этого участка.

Для решения задачи используйте блок- схему на рисунке 183.

Для удобства проверки клетки, имеющие радиацию, большую нуля, закрашиваются.

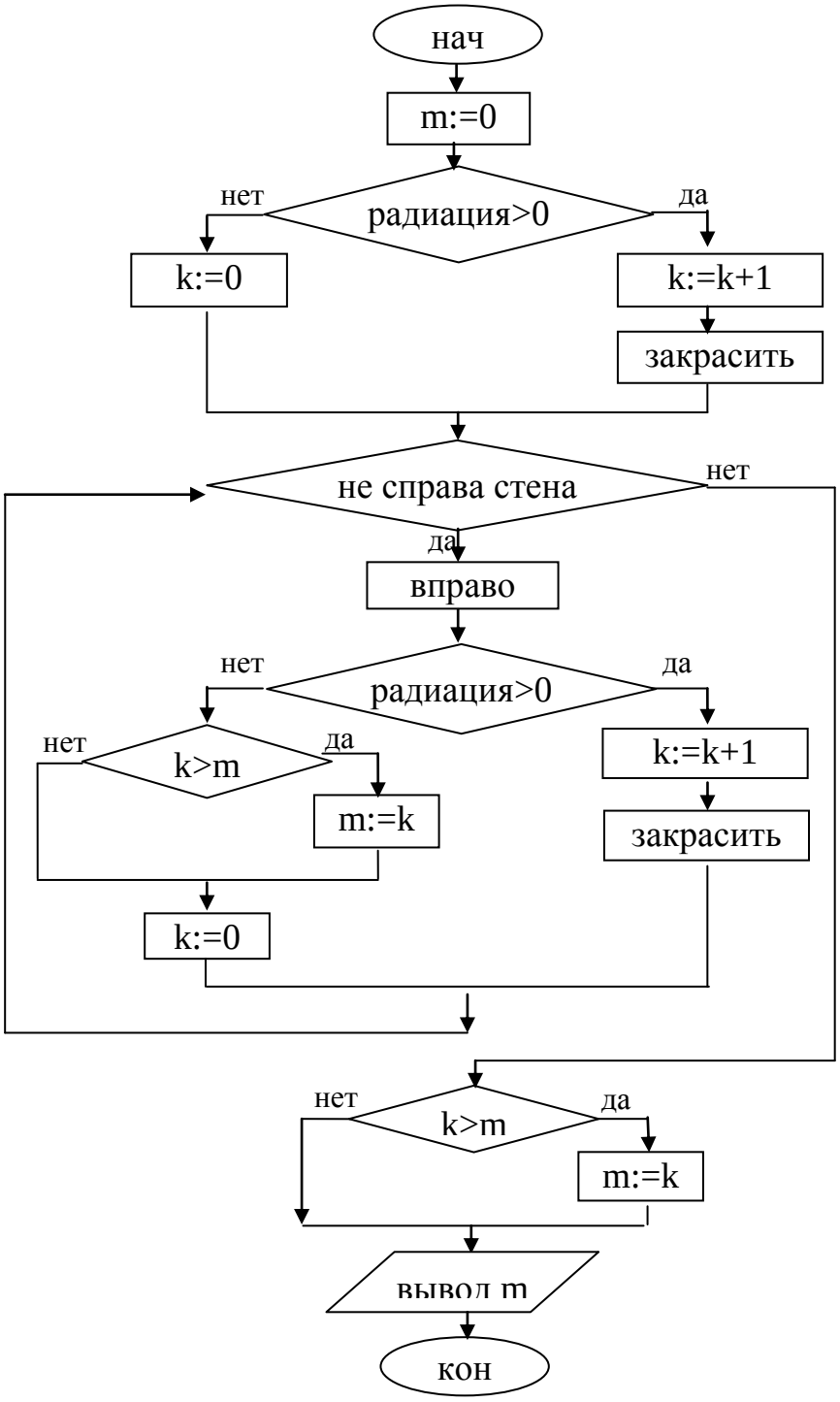


Рис.183. Блок-схема для решения задания №Р33.

Раздел 3. Экспериментальные работы

Программа перевода десятичного числа в двоичное

Запишем программу для Робота, который по десятичному числу будет находить его двоичный код и выводить его на поле в виде закрашенных и незакрашенных клеток. Единица будет соответствовать закрашенной клетке, ноль- незакрашенной.

Для проведения математических расчетов перевода из двоичного числа в десятичное можно воспользоваться таблицей степеней двойки. Десятичное число сотоит из суммы чисел, соответствующих закрашенным клеткам. По этой таблице можно переводить числа от 0 до 255, то есть числа двоичный код которых не превышает одного байта.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Сумма степеней	Двоичный код
128	64	32	16	8	4	2	1		
1	1	1	0	0	0	0	1	128+64+32+1=225	11100001_2=225
0	0	1	0	0	0	1	0	32+2=34	00100010_2=34
1	0	0	1	0	0	0	1	128+16+1=145	10010001_2=145

Рис. 184. Перевод числа из десятичного кода в двоичный и наоборот.

Таблица (рис.184) подходит для перевода из двоичного кода в десятичный и наоборот.

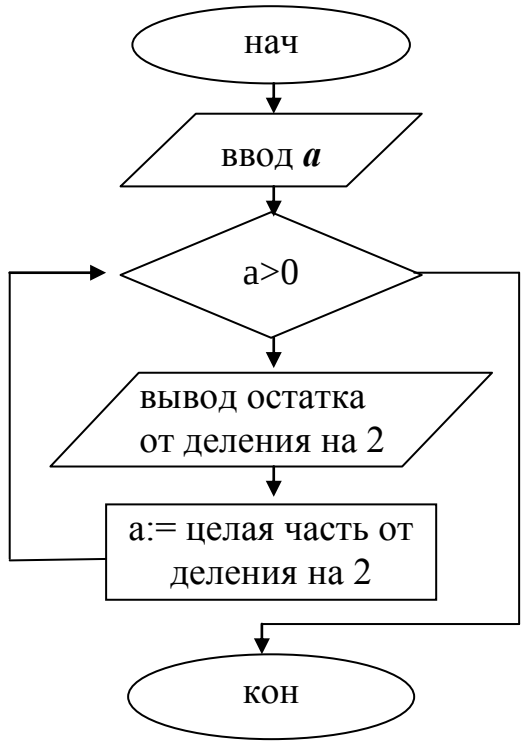


Рис. 185. Блок-схема к программе перевода десятичного числа в двоичный код.

Для написания программы воспользуемся алгоритмом, представленным блок-схемой на рисунке 185, раскладывающим десятичное число в двоичное по остаткам:

Пример:

a	остаток от деления на 2
145	1
72	0
36	0
18	0
9	1
4	0
2	0
1	1

По схеме получаем двоичный код числа.

$145=10010001_2$

Рис. 186. Схема перевода десятичного числа в двоичный код.

Заметим, что если в программе мы сделаем вывод на экран остатка от деления, то двоичный код числа получится «перевернутым», то есть написанным с конца.

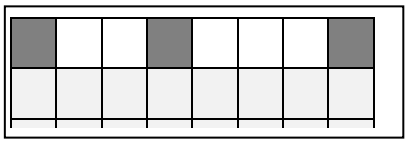
У исполнителя Робот есть возможность начать закрашивать клетки, начиная с **крайней правой** и двигаясь влево, то есть к началу двоичной записи числа. Единице будет соответствовать закрашенная клетка, а нулю – незакрашенная, так как это представлено на рисунке 171.

```
использовать Робот
алг из десятичного в двоичное
нач
. цел a
. ввод a
. нц пока не справа стена
. .вправо
. кц
. нц пока a>0
. . если mod(a,2)=1 то
. . . . закрасить
. . . . влево
. . . иначе
. . . . влево
. . все
. . a:=div(a,2)
. кц
кон
```

Рис.187. Код программы перевода из десятичного числа в двоичное.

В приведенной программе число ***a*** вводится с клавиатуры (рис.187). Если остаток от деления на 2 больше 0, то закрашивается соответствующая клетка и Робот передвигается влево на одну клетку. Иначе Робот двигается на одну клетку влево, но клетка остаётся чистой. В разделе «Теоретические сведения» прочитайте теоретические сведения по использованию операторов ***mod*** и ***div***.

При вводе в программу числа 145 вы должны получить следующий результат (рис.188):



$10010001_2=145$

Рис.188. Результат работы программы перевода из десятичного числа в двоичное.

Задание 1. Переведите с помощью таблицы в двоичную систему десятичные числа: 10, 31, 63, 64, 100, 127, 128, 129, 145, 255.

Задание 2. Наберите программу на компьютере и проверьте правильность перевода.

Задание 3. По наблюдению за записью числа в двоичной системе сделайте вывод о том, как выглядят в двоичном коде четные и нечетные числа.

Програма перевода двоичного числа в десятичное

Запишем программу для Робота, который по двоичному коду, представленному на поле в виде закрашенных клеток, будет получать десятичный код. Вспомним алгоритм перевода двоичного числа в десятичное с использованием таблицы степеней 2:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		
128	64	32	16	8	4	2	1		
	■				■		■	64+4+1=69	01000101_2=69

Рис. 189. Таблица перевода.

Создадим новую стартовую обстановку, в которой закрасим клетки, соответствующие двоичному коду числа. Закрашенные клетки соответствуют единице, незакрашенные – нулю.

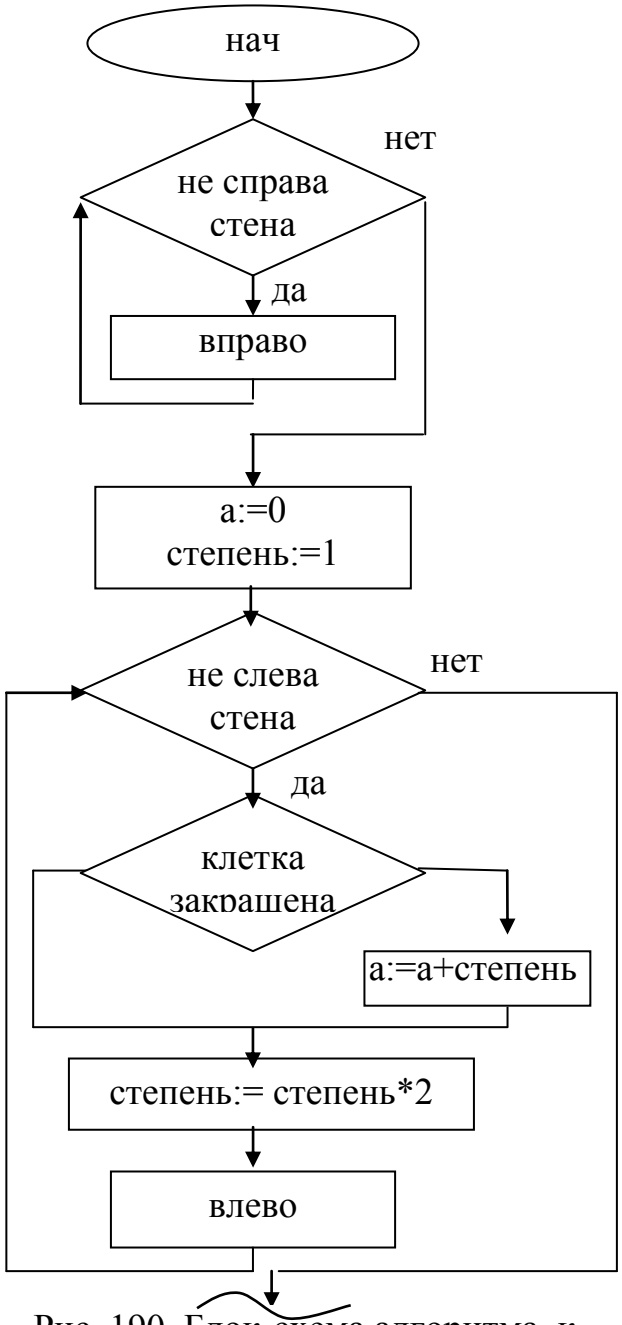


Рис. 190. Блок-схема алгоритма к задаче перевода из двоичного числа в десятичное.

Заметим, что размеры поля Робота имеют ограничения: количество столбцов не может быть более 15.

Вопрос 1: какое максимальное двоичное число можно представить на одной строке поля?

Рассмотрим блок – схему алгоритма. Ответьте на **вопросы.**

а) Для чего предназначен первый цикл?

б) Какие переменные используются в алгоритме?

в) В какой переменной хранится десятичное значение числа, соответствующего двоичному коду раскрашенной полосы?

г) В какой переменной хранится степень числа 2?

д) При каком условии переменная ***а*** увеличивается и на сколько?

е) В каком случае Робот продвигается на 1 клетку влево?

Задание 1. В блок-схеме алгоритма (рис. 190) добавьте команду **вывод *а.*** Напишите программу самостоятельно.

Робот умеет прибавлять 1 и умножать на 2…

Задача. Получить из числа 1 заданное число за наименьшее число команд, если Робот умеет выполнять только две команды: +1, *2. Рассмотрим пример. Нужно получить из числа 1 число 5 за наименьшее число команд. Можно решить задачу так:

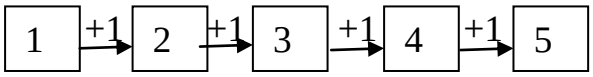


Рис. 191. Первый вариант решения.

Для получения числа нам понадобилось 4 команды.

Решим эту задачу за меньшее число команд:

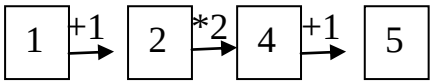


Рис. 192. Второй вариант решения.

Как видно, число 5 можно получить за 3 команды: +1, *2,+1.

```
использовать Робот
алг оптимальный алгоритм
нач
. цел а, флаг
. нц пока не справа стена
. .вправо
. кц
. ввод а
. нц пока а>0
. . если mod(а,2)=1то
. . . . закрасить
. . . . влево
. . . . а:=а-1
. . . иначе
. . . . а:=div(а,2)
. . . . влево
. . все
. кц
. вправо
. нц пока не справа стена
. . вправо
. . если клетка закрашена то
. . . . вывод"+1,"иначе
. . . . вывод"*2,"
. . все
. кц
кон
```

Программа **оптимальный алгоритм** поможет вам найти самое оптимальное решение этой задачи.

Посмотрите на код программы (рис.193) и догадайтесь, каким образом компьютер выполняет эту задачу?

Задание. Проверьте свои предположения на примерах:

а) из 1 получить 5;
б) из 1 получить 8;
в) из 1 получить 15;
г) из 1 получить 81;
д) из 1 получить 100.

Рис. 193. Код программы **оптимальный алгоритм.**

```
>> 15:38:46 – оптимальный алгоритм.kum –
10
*2, *2, +1, *2,
>> 15:38:48 – оптимальный алгоритм.kum -
```

Рис. 194. Результат работы программы **оптимальный алгоритм.**

Раздел 4. Исследовательские работы

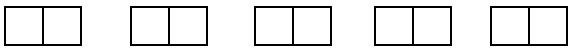
Штрих - код

Постановку задачи начнем с **вопроса**: расскажите, что вы знаете о штрих – коде?

Сегодня мы попробуем создать свой штрих – код с помощью программы закрашивания клеток.

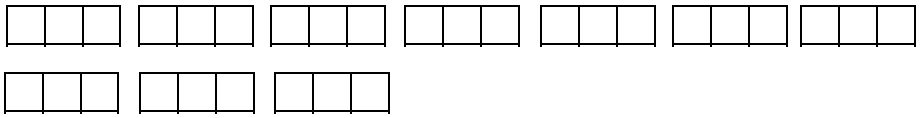
Для начала сформулируем математическую модель задания. Как известно, штрих – код состоит из полос, имеющих разную ширину. Попробуем создать **разные** штрих- коды на горизонтальной полосе.

Задание1. Допустим, в нашем распоряжении всего одна клетка. Ее мы можем закрасить или оставить незакрашенной. Таким образом, на одной клетке можно получить только два кода. Теперь попробуйте создать разные коды, на двух клетках:



Вопрос 1. Сколько **разных** кодов удалось создать в **задании 1**?

Задание 2. Теперь создадим **разные** коды на полосах, состоящих из трех клеток:



Вопрос 3. Сколько **разных** штирх-кодов удалось создать на трех клетках?

Задание 3. Теперь создадим **разные** коды на полосах, состоящих из четырех клеток. Как вы думаете, сколько разных кодов вам удастся создать?

Проверьте ваше предположение, используя лист в клеточку.

Задание 4. Систематизация. Создадим правила упорядочивания штрих - кодов. Занесем упорядоченные штрих коды из **четырех** клеток в таблицу. Путь код, в котором все клетки чистые, имеет номер 0. Первый номер присвоим коду, у которого закрашена последняя клетка, второму – предпоследняя, далее – две последние клетки закрашены. Теперь все четыре комбинации кодов, для которых первые две клетки чистые – исчерпаны. Следующая комбинация – третья клетка с конца – закрашена, а последняя и предпоследняя – чистые.

Продолжите формировать штрих – коды по описанному правилу. Посчитайте, сколько разных штрих – кодов у вас получилось? Используйте таблицу «штрих-код» (рис. 194).

Таблица: «штрих-код»

№	код			
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

Рис. 194. Таблица «штрих-код».

Проверьте себя. Если вы правильно формировали штрих коды, то в последнем столбце закрашенные клетки чередуются через одну. В предпоследнем столбце – две клетки чистые, две – закрашенные. В третьем с конца столбце – четыре клетки чистые, четыре – закрашенные. И, наконец, в первом – первые восемь клеток чистые, вторые восемь клеток – закрашенные.

Сколько разных штрих-кодов, включая комбинации, когда все четыре клетки чистые и все четыре клетки закрашены, получилось?

Задание 5. а) Какое количество разных штрих – кодов можно создать на пяти клетках? **б)** На шести клетках? **в)** При максимальном размере строки поля Робота - на 15 клетках?

Рассмотрим программу формирования штрих – кода при следующем условии: штрих – код состоит из четырех клеток. Алгоритм **мой штрих код** (рис.195) начнет формировать код справа налево от правой стены.

```
использовать Робот
алг мой штрих код
нач
. нц пока не справа стена
. .вправо
. кц
. закрасить
. влево
. закрасить
. влево
. влево
. закрасить
кон
```

Рис. 195. Код программы **мой штрих код.**

Представленная программа соответствует штрих-коду с номером **11** в таблице.

Таким штрих – кодом можно закодировать, например, месяц ноябрь.

Постановка задачи экспериментальной работы: сформулируйте условия **формирования вашего** штрих – кода. Придумайте ситуацию, когда вам может пригодиться двоичное кодирование. Что вы будете кодировать? Какие размеры поля будут использоваться?

Приведем примеры постановки некоторых задач. Пусть ваш штрих – код соответствует месяцу вашего рождения. Какой длиной штрих-кода можно ограничиться при решении этой задачи?

Можно усложнить задание: сформировать штрих – код, соответствующий дню и месяцу вашего рождения. Скорее всего, вам понадобится 9 клеток: пять для даты рождения и четыре – для месяца рождения. Для даты рождения будет необходимо увеличить таблицу кодов. Подумайте как.

Рассмотрим программу распознавания образа одного сформированного кода (рис.198), соответствующего в таблице номеру 11 .

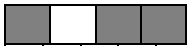


Рис. 196. Штрих – код по номером 11.

Объявим логическую переменную **флаг**. Будем идти от **правой** стены и проверять клетки. Они должны быть такими: закрашена, закрашена, не закрашена, закрашена. Если хотя бы в одном случае будет несовпадение, то переменная **флаг** примет значение «***ложь***» и в конце программы будет выдано сообщение: «**не мой код**».

Ниже приведена программа. В ней есть ошибка. Один из кодов будет неверно распознан. Нарисуйте этот код (рис.197). Подумайте, как можно исправить эту ошибку.

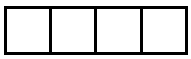


Рис. 197. Заготовка к ответу.

```
1. использовать Робот
2. алг распознавание штрих кода
3. нач
4. . лог флаг
5. . флаг:=да
6. . нц пока не справа стена
7. . . вправо
8. . кц
9. . если клетка чистая то флаг:=нет все
10. . влево
11. . если клетка чистая то флаг:=нет все
12. . влево
13. . влево
14. . если клетка чистая то флаг:=нет все
15. . если флаг =да то вывод "мой код"
16. . . иначе вывод "не мой код"
17. . все
18. кон
```

Рис. 198. Код программы **распознавание штрих-кода.**

💻Задание 6. Напишите программу распознавания своего шрих – кода.

Распознавание образов

Возможно, вы уже слышали о том, что, несмотря на то что компьютер очень быстро делает вычисления, ему по-прежнему тяжело различать образы.

Попробуем решить следующую задачу. Пусть Робот определит: нарисована ли на его поле полоса шириной в одну клетку. Она может быть длиной или короткой, но вокруг ее контура поле должно быть чистым. Полоса должна состоять более чем из одной клетки. Других стен, кроме стен, ограничивающих поля, – нет. **Упростим задачу.** Наша полоса не будет стоять около стены и будет вертикальной. На поле расположена только одна фигура. Рассмотрим примеры.

Пример А	Пример Б	Пример В	Пример Г	Пример Д	Пример Е
да	да	да	нет	нет	нет

Как видно в примере **Г** около контура есть закрашенные клетки, в примере **Д** контур состоит из двух закрашенных полосок, а в примере **Е** палочка очень короткая и по условию задачи не считается **полосой.** Нарисуем возможную блок- схему алгоритма программы распознавания вертикальной полосы (рис. 199).

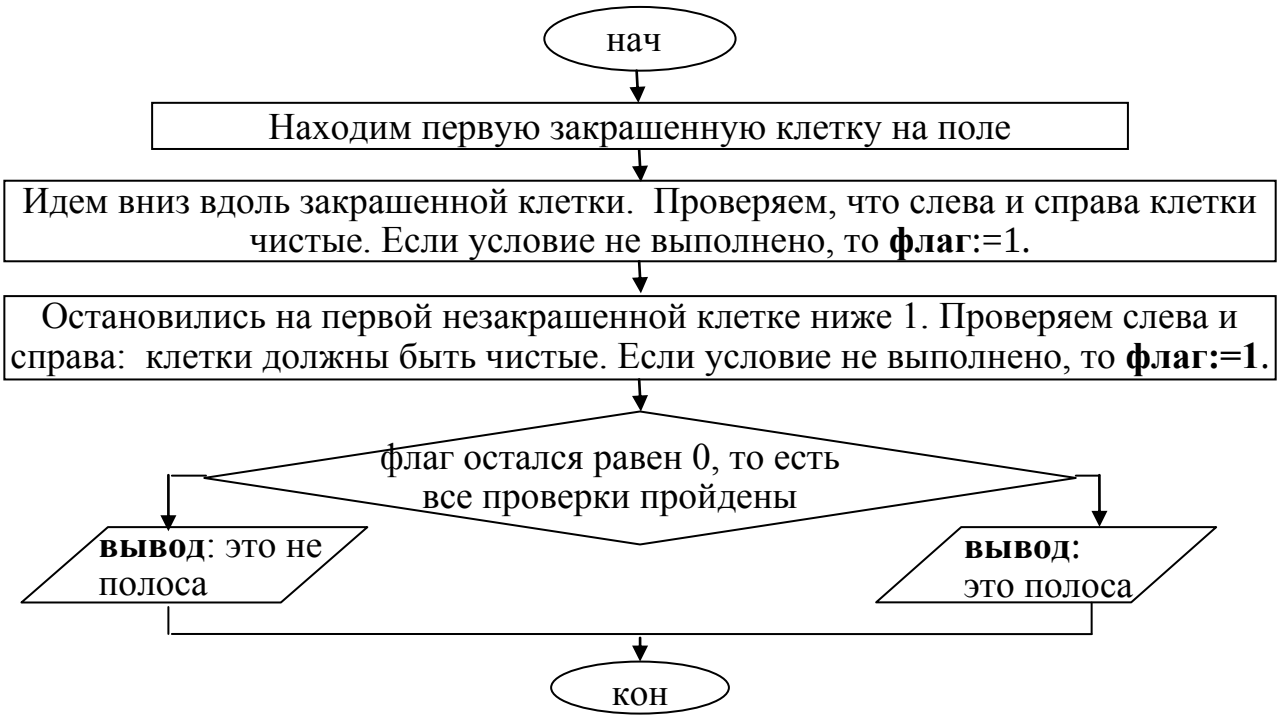


Рис. 199. Блок- схема алгоритма распознавания вертикальной полосы.

```
использовать Робот
цел флаг, нашли
алг распознавание полосы
нач
. флаг:=0
. нашли:=0
. нц пока нашли=0
. . идем_вправо_ищем_закрашенную_клетку
. кц
. вывод "нашли",нс
. | идем по закрашенной полосе и проверяем:
. | слева и справа - чисто
. нц пока клетка закрашена
. . влево
. . проверка
. . вправо
. . вправо
. . проверка
. . влево
. . вниз
. кц
. | остановились на первой незакрашенной
. | клетке ниже полосы
. влево
. проверка
. вправо
. вправо
. проверка
. если флаг=0 то вывод "полоса"
. .иначе вывод "не полоса"
. все
кон

алг проверка
нач
. если клетка закрашена то флаг:=1 все
кон

алг идем_вправо_ищем_закрашенную_клетку
нач
. нц пока не справа стена и нашли=0
. . вправо
. . если клетка закрашена то
. . . . нашли:=1
. . . . выход
. . все
. кц
. если нашли=0 то нц пока не слева стена
. . . . влево
. . . кц
. все
кон
```

Рис. 200. Код программы распознавание образов.

Программа может выглядеть так как показано на скрине экрана (рис.200).

Вопрос 1. В программе есть ошибка: на одном из примеров, приведенных нами выше, будет выдавать неверный ответ. Определите, какой это пример, и исправьте ошибку в программе.

Теперь вы сами увидели, как трудно написать программу распознавания образов, даже для небольшой задачи.

Напишите исследовательскую работу по теме: «Распознавание образов»

Скатерть Улама

Скатерть Улама [10] была открыта случайно в 1963 году. Однажды математику довелось присутствовать на очень длинном и скучном докладе. Чтобы развлечься, он начертил на листке бумаги вертикальные и горизонтальные линии, чтобы заняться составлением шахматных этюдов. Но вместо этого он стал нумеровать клетки: в центре поставил единицу, а затем, двигаясь по спирали, двойку, тройку и т. д.. При этом он машинально отмечал простые числа (рис.201).

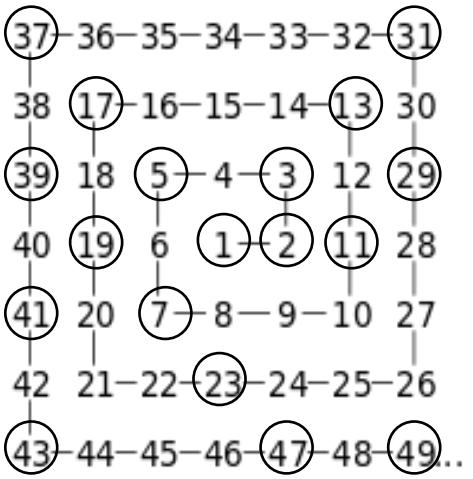


Рис. 201. Скатерть Улама.

Оказалось, что простые числа стали выстраиваться вдоль диагональных прямых. Это заинтересовало Улама, и позже он вместе с Майроном Л. Стейном и Марком Б. Уэллсом продолжил это исследование на ЭВМ MANIAC II Лос-Аламосской лаборатории, использовав магнитную ленту, на которой были записаны 90 млн. простых чисел.

Попробуем нарисовать скатерть на поле Робота. Размеры поля ограничены: максимальное количество строк- 10.

Вопрос 1.Сколько чисел нам удастся отметить на этой скатерти?

Решение задачи будет состоять из двух этапов: решение задачи на определение количества делителей числа, то есть его простоты и задачи обхода поля по заданному направлению.

Для определения простоты числа ***a*** воспользуемся следующим алгоритмом. Пусть переменная **флаг** будет ***логической*** и принимает значение «***истина»***, если число простое. В цикле переберем все числа от 2 до ***a-1***, и если число ***a*** разделится нацело хотя бы на одно из чисел, то оно уже простым являться не будет. Логическая переменная **флаг** в этом случае примет значение «***ложно»***. Блок-схема этого алгоритма представлена на рисунке 202.

Вопрос 2. Какой командой можно определить, делится ли одно число на другое нацело?

Вопрос 3. Количество циклов можно было бы сократить. Подумайте как?

Задание 1. Напишите программу, которая определяет, является ли введенное число простым.

Теперь решим задачу обхода Роботом поля по спирали.

К сожалению, максимальные размеры поля для Робота: 10 на 16 клеток, поэтому мы можем обойти поле по спирали всего 4 раза.

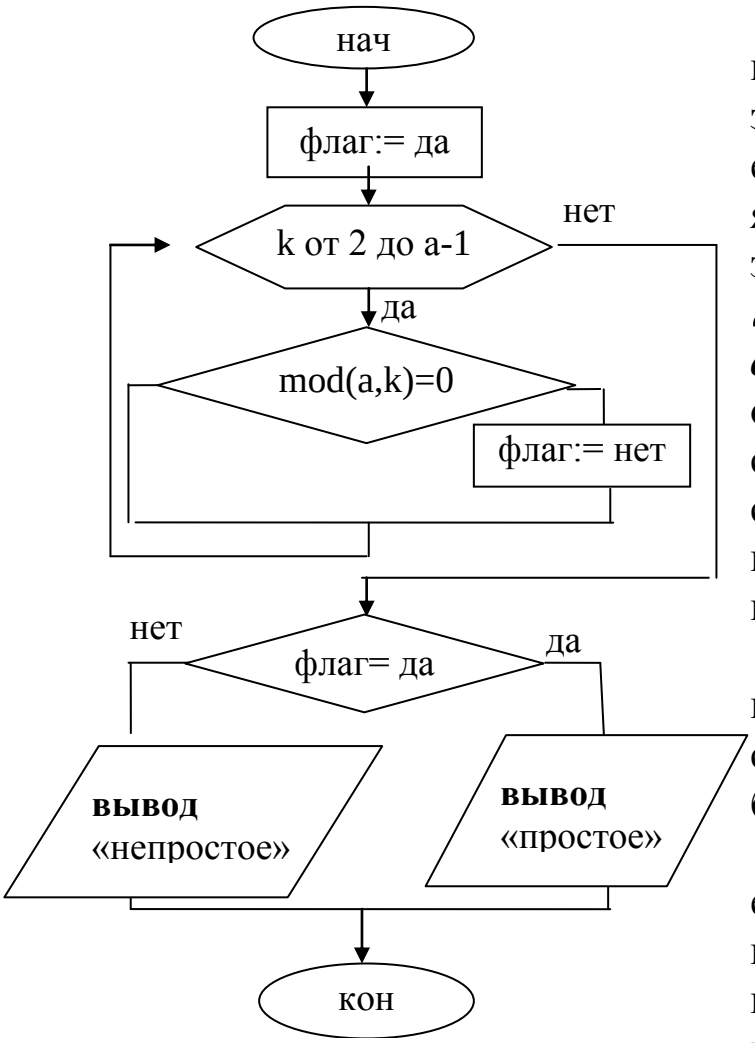


Рис. 202. Блок-схема алгоритма определения простоты числа.

На каждом шаге будет необходимо увеличивать значение переменной ***a*** на единицу, и если число ***a*** является простым, то клетку **закрашивать**

Задание2. Напишите ***отдельную*** программу для обхода поля Роботом по спирали. Выведите на экран самое большое количество клеток, которое смогли получить при таком обходе.

Расположите Робота в центре поля так, чтобы он смог обойти как можно больше клеток.

Для проверки числа на его «простоту» можно написать **функцию**, принимающую логическое значение ***«истина»,*** если число ***a*** простое, и ***«ложь»*** – в противоположном случае. В основной программе «обхода поля по спирали» можно будет использовать команду: **если простое(*a*) то закрасить все.** В подпрограмму будет передаваться значение целой переменой ***a***, статус которой (простое или не простое) необходимо определить.

Задание 3. Напишите программу для Робота, который закрашивает клетки на поле так же, как это делал математик Улам.

Задание 4. Посмотрите, какой закономерностью, на ваш взгляд, обладают простые числа, расположенные по скатерти Улама?

Задание 5. Предложите свою траекторию обхода поля.

Часть 3. Система программирования КуМир

Система программирования КуМир [1] разрабатывалась в начале 80-х годов прошлого века для изучения курса информатики в школе. С тех пор поколения компьютеров изменились, но язык КуМир по - прежнему является одним из лучших для обучения азам программирования школьников. С 2008 года этот язык включен в экзаменационный материал Государственной итоговой аттестации по информатике[11].

Практическая работа №3.1.Изучение справки по языку КуМир.

Цель: научитесь пользоваться справкой в системе.

Вопрос 1. Прочитайте сведения о программе КуМир (рис. 203). На каких условиях распространяется программа?

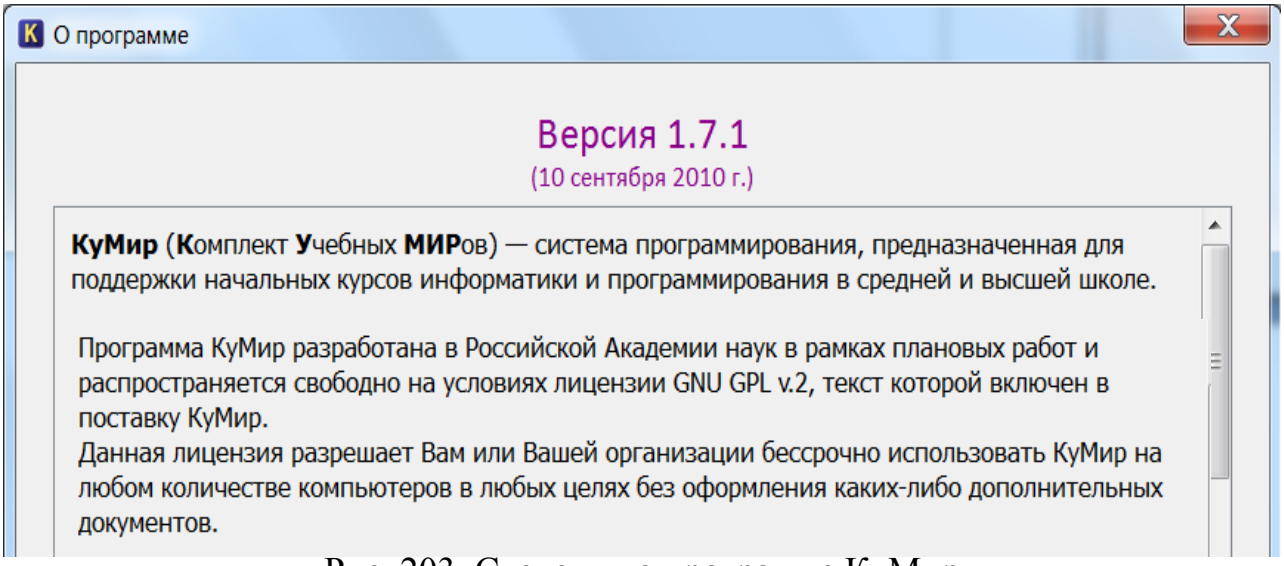


Рис. 203. Сведения о программе КуМир.

Вопрос 2. По окнам на экране (рис.204) определите, какую информацию можно получить о системе КуМир.

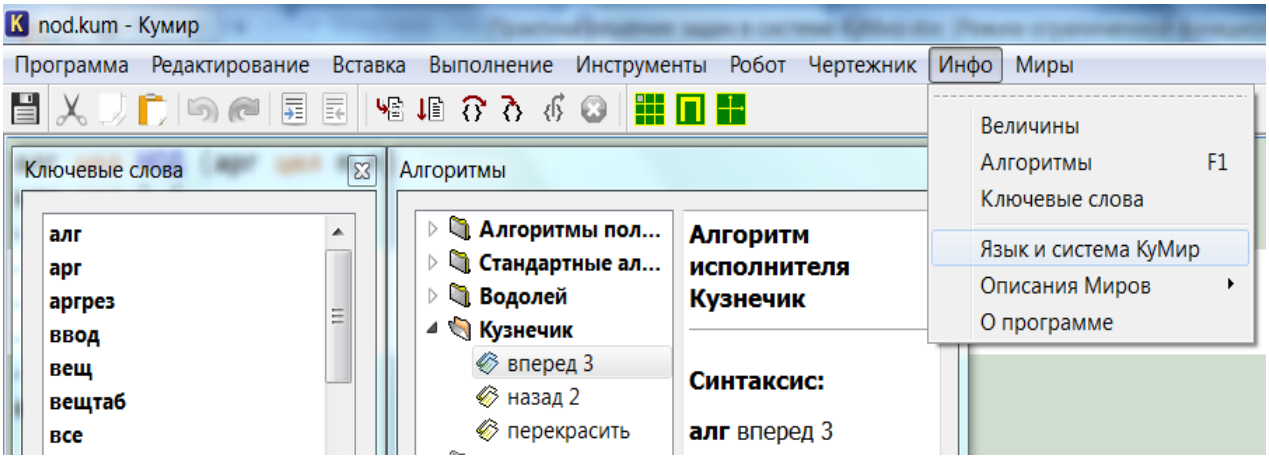


Рис. 204. Информационные и справочные системы КуМир.

Вопрос 3. По справке «Язык и система Кумир» найдите, есть ли возможность для исполнителя Робот очистить клетку, то есть закрашенную клетку сделать незакрашенной?

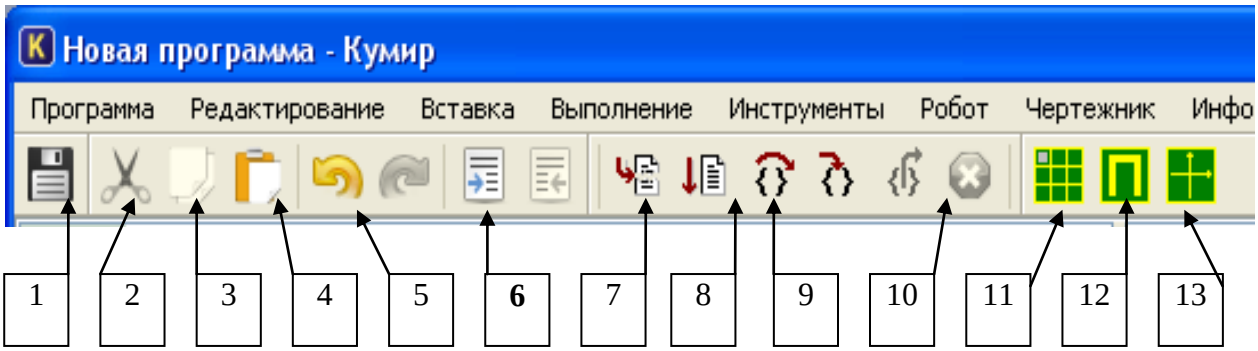


Рис. 205. Панель инструментов окна системы КуМир.

Вопрос 4. Определите назначение кнопок в окне Кумир (рис. 205). **Назначение кнопок для соответствия: А**- вставить содержимое буфера обмена, **Б** -пошаговая реализация программы, **В** - выполнить программу непрерывно, **Г**- скопировать в буфер обмена, **Д** - отменить исправления в программе, **Е**- показать окно Робота, **Ж** - показать **Пульт** Робота, **З** - остановить выполнение программы, **И**- вставить комментарий, **К**- сохранить программу, **Л** – показать окно Чертежника, **М** – вырезать, **Н** – выполнить непрерывно без показа на полях.

Написание программ. Линейные алгоритмы

Практическая работа №3.2. Вычисление площади прямоугольника.

Цель: Научиться выделять входящие и выходящие переменные.

Задача. Написать программу для нахождения площади прямоугольника по двум сторонам.

Решение. Выделим входящие и выходящие параметры. По условию задачи на вход в программе подаются две переменные- стороны прямоугольника. На выходе мы должны получить одно значение – площадь. Назовем переменные для сторон прямоугольника ***a*** и ***b***. Переменную для площади обозначим **S**. Решим задачу для **целых** переменных.

```
алг площадь
нач
- цел a,b,S
- ввод a,b
- S:=a*b
- вывод S
кон
```

```
>> 13:12:27 - Новая программ
8 9
72
```

Рис. 206. Код и результат работы программы **площадь**.

Обратите внимание на **ввод** переменных. Они вводятся через **пробел или через запятую** в одной строке (рис.206).

В коде программы целесообразно писать комментарии. Это помогает определить назначение самой программы, ее блоков, используемых переменных. Также хорошим тоном считается написание пояснений к вводу и выводу переменных. Эта же программа с комментариями может выглядеть так:

```
алг площадь прямоугольника
нач
. | найти площадь прямоугольника
. | по заданным сторонам
. цел a,b,S
. вывод "введите стороны прямоугольника"
. ввод a,b
. S:=a*b
. вывод "площадь прямоугольника: ",S
кон
```

```
>> 16:42:26 - Площадь прямоугольника.kum* -
введите стороны прямоугольника5 6
площадь прямоугольника: 30
```

Рис. 207. Код и результат работы программы **площадь прямоугольника.**

Как видно на рисунке 207, в строке **вывода** появились комментарии, поясняющие условие задачи, пояснения к вводу и выводу.

Используйте эти возможности для написания сложных программ и программ, которыми будут пользоваться другие.

Для того, чтобы проверить, правильно ли работает программа, ее тестируют. В тестах для проверки входящие и выходящие переменные заранее известны. Эту программу можно протестировать на следующих значениях.

	входные данные	выходные данные
тест 1	5 6	30
тест 2	2 3	6

Если программа работает правильно на этих примерах, значит, она правильно будет работать при вводе других чисел.

✎**Задание 1.** Нарисуйте блок – схему к программе **площадь прямоугольника**.

💻**Задание 2.** Перепишите код программы в систему КуМир. Запустите программу и убедитесь, что она работает.

💻**Задание 3.** Посчитайте площадь прямоугольника со сторонами **а)** 15 и 37; **б)** 18 и 37; **в)** 21 и 37; **г)** подберите сторону b так, чтобы при стороне a=37 площадь была бы 888.

💬**Задание 4.** Объясните ошибку, которая возникает при значениях a=100 000 и b=100 000.

Задание №А1. Напишите программу, которая вычисляет периметр прямоугольника. Протестируйте ее работу на своих примерах.

Задание №А2. Напишите программу, которая вычисляет периметр треугольника.

Практическая работа №3.3. Вычисление значения по формуле.

Цель: научиться записывать программы для вычисления значений по формуле с использованием линейных алгоритмов.

Задача 1. Найти время по введенному расстоянию и скорости.

В программе нам необходимо использовать формулу ***t=S/V***. Результат операции деления записывается в переменную ***t***, а это означает, что переменная ***t*** должна иметь **вещественный** тип. Типы переменных ***S*** и ***V*** могут быть как целыми, так и вещественными (рис.208).

```
алг вычисление по формуле
нач
. вещ S,V,t
. вывод "введите расстояние и скорость: "
. ввод S,V
. t:=S/V
. вывод "время: ", t
кон
```

```
введите расстояние и скорость: 22   7
время: 3.142857
```

Рис. 208. Код и результат работы программы **вычисление по формуле.**

Задача 2.Рассмотрим еще одну задачу: найти площадь круга по заданному радиусу по формуле: $S=\Pi R^2$.

Посмотрите на код программы **площадь круга** (рис.209). В языке КуМир нет специальной константы ***ПИ***. Ее значение с точностью до 30 знаков скопировано из Википедии[12]. На настоящее время известны более триллиона знаков числа ***ПИ***, вычисленных с помощью компьютера.

Вопрос 1. С какой точностью ведутся вычисления в языке КуМир?

Вопрос 2. По коду программы определите входящие и выходящие переменные. Объясните целесообразность использования вещественного типа переменных.

Вопрос 3. Догадайтесь по коду **программы площадь** круга или найдите по справке языка КуМир, что означает команда «**»?

Задание 1. Наберите код программы **площадь круга**. Посчитайте площадь круга при радиусе **а)** r=1; **б)** r=2.

```
алг площадь круга
| найти площадь круга
| по заданному радиусу
нач
. вещ r, S, ПИ
. ПИ:=3.14159265358979323846264338327 95
. вывод "введите радиус: "
. ввод r
. S:=ПИ*r**2
. вывод "площадь круга: ", S
кон
```

```
ПИ=3.141593

r=1
S=3.141593
```

Рис. 209. Код и результат работы программы **площадь круга.**

введите радиус: 1
площадь круга: 3.141593

Задание №А3. Найдите площадь кольца между двумя окружностями с заданными радиусами **r1** и **r2**. 0<r1<r2<10000 и общим центром.

Задание №А4. Напишите программу, которая вычисляет длину окружности по известному радиусу по формуле **L=2Пd**, где **d** – диаметр окружности.

Задание №А5. Напишите программу для вычисления

а) объема прямоугольного параллелепипеда по формуле: ***V=a*b*c;***

б) объема шара по формуле: $V=4/3*\pi R^3$.

Задание №А6. Вычислите площадь треугольника АВС (рис.210) по формуле:

$$S = \frac{a}{2} \bullet h$$

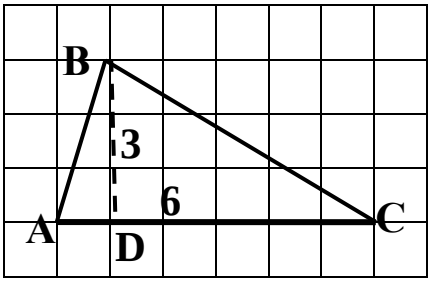


Рис. 210. Задание №А6.

Сверьте ответ, полученный с помощью программы с математическими расчетами.

Задание №А7. Вычислите площадь трапеции ABCD (рис. 211) по формуле:

$$S = \frac{a+b}{2} \bullet h$$

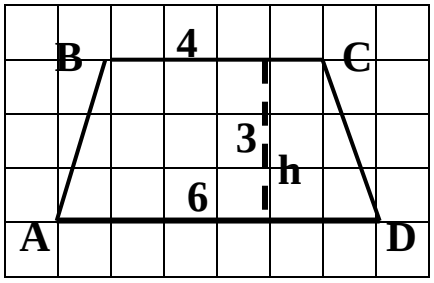


Рис. 211.Задание №А7.

Сверьте ответ, полученный с помощью программы с математическими расчетами.

Задание №А8. Найдите площадь треугольника со сторонами ***a, b, c*** по формуле Герона:

$$S = \sqrt{p\,(p-a)(p-b)(p-c)}, \quad где \quad p = \frac{a+b+c}{2}.$$

Функция корня в системе КуМир записывается как **sqrt(x).**

Задание №А9. Найдите расстояние между двумя точками по формуле:

$R = \sqrt{(x1-x2)^2 + (y1-y2)^2}$, где (x1, y1) и (x2, y2) – координаты точек.

Задание №А10. а) Дана программа (рис.212) для нахождения периметра и площади **Фигуры1** (рис.213)**.** Перепишите программу в КуМир, сделав необходимые пояснения.

```
алг площадь_периметр фигуры
нач
. цел a,b,c,d,S,P
. вывод "введите стороны внешнего прямоугольника"
. ввод a,b
. вывод "введите стороны вырезанного прямоугольника"
. ввод c,d
. P:=(a+b)*2
. S:=a*b-c*d
. вывод "периметр ",P,нс
. вывод "площадь ", S,нс
кон
```

```
введите стороны внешнего прямоугольника10 9
введите стороны вырезанного прямоугольника2 5
периметр 38
площадь 80
```

Рис. 212. Код и результат работы программы **площадь_периметр фигуры.**

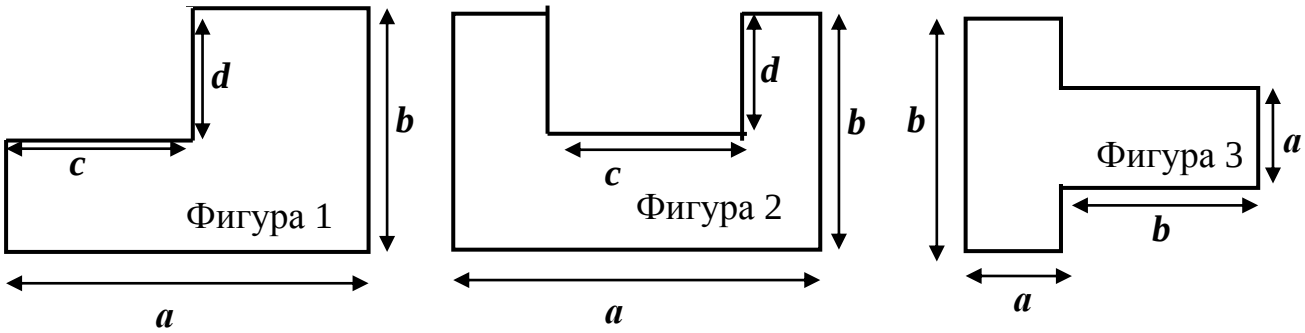


Рис. 213. Фигуры к заданию №А10.

б) Исправьте программу для нахождения периметра и площади **Фигуры 2** (рис.200). Если вы правильно написали программу, то для второй фигуры при **a**=10, **b**=9, **c**=2, **d**=5 у вас получится: **P**=48, **S**=80. Второй тест для проверки программы составьте самостоятельно.

в) Исправьте программу для нахождения периметра и площади **Фигуры 3** (рис.200). **Фигура 3** состоит из двух одинаковых прямоугольников со сторонами **a** и **b**, объединенных в одну фигуру, как показано на рисунке 213. **0<a<b<1000.** Найдите площадь и периметр фигуры.

Задание №А11. В изображении размером ***a*** на ***b*** точек каждая точка кодируется k битами. Найти объем изображения в Мб, зная, что 1 байт= 8 бит, 1 Кб=1024 байт, 1 Мб=1024 Кб.

	входные данные (***a,b,i***)	выходные данные (***Мб***)
Тест 1.	5122048 24	3
Тест 2.	100 100 8	0.009536743

Операторы целочисленного деления

Как вы уже знаете, не всегда можно разделить одно число на другое в целых числах. Такое деление называется делением с остатком. Например: 17:3=5 (2 остаток).

В языке программирования КуМир для такого деления предусмотрены два оператора: ***div*** – взятие целой части числа и ***mod*** – остаток от деления. Прочитайте справку об операторах целочисленного деления **mod** и **div**. Записывают их так:

a:=div(17,3) **b:=mod(17,3)**

В результате выполнения этих команд ячейка ***a*** примет значение целой части от деления числа 17 на 3, а ячейка b – остатка от деления числа 17 на 3. То есть значения ячеек станут: a=5 иb=3.

Решите примеры.

а) a:=14+mod(12,5); **б)** b:=div(10,3)+2*3;

в) c:=div(12,5)+mod(12,5); **г)** d:=div(81,10)+mod(81,10).

Практическая работа №3.4. Операторы целочисленного деления.

Цель: на примере решения задачи научиться использовать операторы целочисленного деления.

Задача. Найти сумму цифр двузначного числа.

Решение. В этом задании на вход в программе подается двузначное число, а на выход выдается сумма его цифр. Подумайте, как можно в этой задаче применить операторы целочисленного деления?

В коде программы **сумма цифр двузначного числа** (рис. 214) используются четыре целочисленные переменные. На самом деле переменные **d** и **e** можно было не использовать. В этом случае можно было посчитать S по формуле: **S:=div(a,10)+mod(a,10).**

```
алг сумма цифр двузначного числа
нач
. цел S,d,e,a
. вывод "введите двузначное число:",нс
. ввод a
. d:=div(a,10)
. e:=mod(a,10)
. S:=d+e
. вывод S
кон
```

Рис. 214. Код и результат работы программы **сумма цифр двузначного числа**.

💻**Задание 1.** Измените код программы **сумма цифр двузначного числа** так, чтобы программа меняла цифры двузначного числа местами. Например, ввод: 81, вывод: 18. **Указание:** в коде программы **сумма двузначного числа** нужно дописать только один знак и одно число. Решение, состоящее из последовательного вывода цифр числа, не принимается.

Задание №А12. Найдите сумму цифр трехзначного числа.

Задание №А13. Поменяйте местами первую и третью цифры трехзначного числа.

Задание №А14. а) На вход программы подаются два числа: минуты и секунды. Получите одно число: количество секунд.

входные данные	выходные данные
1 5	**65**
16 40	**1000**

б) Переведите секунды в минуты и секунды. В таблице даны форматы входных и выходных данных.

входные данные	выходные данные
65	**1 мин. 5 с.**
140	**2 мин. 20 с.**
1000	**16 мин. 40 с.**

Задание №А15. Переведите секунды в часы, минуты и секунды. В таблице даны форматы входных и выходных данных.

входные данные	выходные данные
65	**0 час. 1 мин. 5 с.**
10000	**2 час. 46 мин. 40 с.**

Задание №А16. По заданной неправильной дроби получить смешанное число. На вход в программе подаются два числа: числитель и знаменатель дроби. На выходе нужно получить три числа: целую часть, числитель и знаменатель. Образец вывода:

входные данные	выходные данные
17 5	**3 2/5**
5 17	**0 5/17**

Указание по использованию команды вывода:

```
. вывод целое," ",числитель,"/",знаменатель
```

Задание №А17.Автомат получает из четырехзначного числа новое число по правилу: складывает первую и вторую цифры, а затем третью и четвертую цифры. Две полученных суммы записывает друг за другом без пробелов. Напишите программу для работы этого автомата.

Оператор условного перехода

Оператор условного перехода (ветвления) предполагает выполнение действий в зависимости от условия. Оператор может иметь полную и неполную формы.

Полную форму оператора условного перехода можно представить следующей конструкцией:

Если [условие] **то** [команды] **иначе** [команды] **все**

В неполной форме оператора условного перехода отсутствуют команды ветки иначе. Конструкция неполной формы выглядит так:

Если [условие] **то** [команды] **все**

Оператор ветвления используется в задачах, где необходимо выполнить команду в зависимости от условия.

Практическая работа №3.5. Оператор условного перехода.

Цель: познакомиться с применением оператора условного перехода в задачах.

Рассмотрим, как применяется оператор условного перехода на примере двух задач: нахождения модуля и определения четности введенного числа.

Задача 1. Найти модуль числа.

Вопрос 1. По блок- схеме (рис. 215) составьте словесный алгоритм решения задачи.

Вопрос 2. Найдите в коде программы модуль числа оператор условного перехода.

Вопрос 3. Определите: полная или неполная форма оператора условного перехода, используется в решении **задачи 1.**

Вопрос 4. Найдите в коде программы **модуль числа** (рис. 216) команду ветвления и определите служебные слова этой команды.

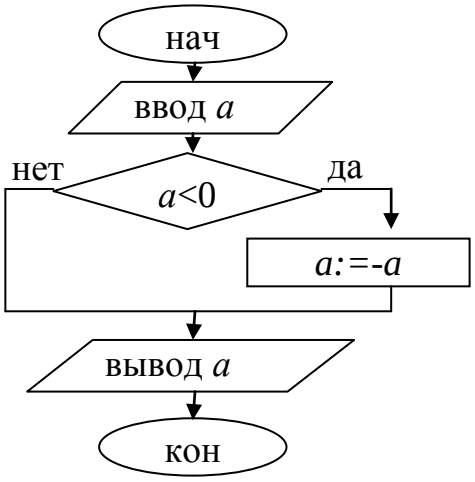


Рис. 215. Блок-схема алгоритма **модуль числа.**

```
алг модуль числа
нач
. вещ a
. вывод "введите число: "
. ввод a
. если a<0то
. . . a:=-a
. все
. вывод a
кон
```

Рис. 216 Код программы **модуль числа.**

Задача 2. Определить, является ли веденное число четным.

Вопрос 5. По блок-схеме алгоритма решения **задачи 2** (рис.217) составьте словесный алгоритм.

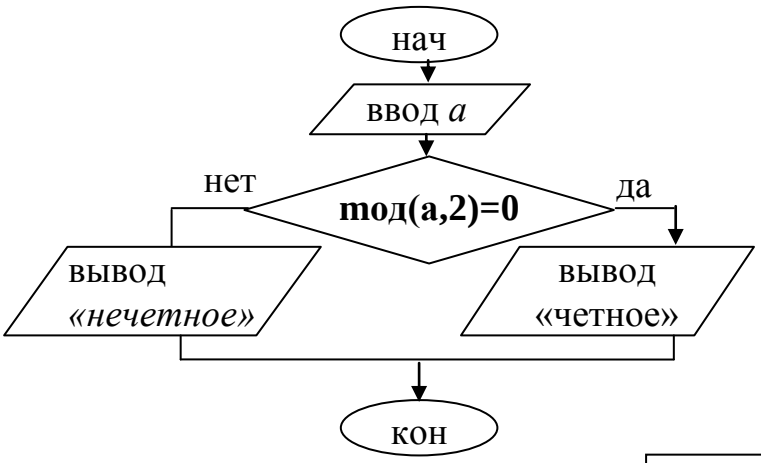


Рис. 217. Блок-схема алгоритма **четность числа.**

Вопрос 6. Найдите в коде программы **четность числа** (рис. 218) команду ветвления и определите служебные слова этой команды.

Вопрос 7. В какой из задач (№1 или №2) практической работы используется полная форма оператора ветвления?

Задание 1. Наберите на компьютере код программы

```
алг четность числа
нач
. цел a
. вывод "введите число: "
. ввод a
. если mod(a,2)=0 то
. . . вывод"четное"
. . иначе
. . . вывод"нечетное"
. все
кон
```

Рис. 218. Код программы **четность числа.**

модуль числа.

💻**Задание 2.** Исправьте программу **модуль числа** для нахождения корня числа, если это число положительное. Если число отрицательное – выдайте сообщение: корень найти нельзя.

Правило использования функции корня найдите в справочнике.

💻**Задание 3.** Наберите на компьютере код программы **четность числа**. Проверьте правильность работы программы.

💻**Задание 4.** Исправьте программу **четность числа** так, что бы она определяла, оканчивается ли введенное число нулем.

Задание №А18. На вход в программу подаются два положительных числа. Определить, могут ли эти два числа быть сторонами квадрата?

Задание №А19. а) Вводятся два разных числа. Вывести на экран наименьшее из них.

б) Вводятся три разных числа. Найти и вывести на экран наименьшее из них.

Задание №А20. На вход в программу подаются два разных числа – значения переменных ***a*** и ***b***. Значения в ячейках ***a*** и ***b*** поставить в порядке возрастания. Вывести значения ячеек ***a*** и ***b***.

Задание №А21. На вход в программу подаются три числа – значения переменных ***a, b*** и ***c***. Значения в ячейках ***a, b*** и **c** поставить в порядке неубывания. Вывести значения ячеек ***a, b*** и ***c***.

Задание №А22.Дверь и ящик. На вход в программу подаются пять чисел: первые два - размеры двери, остальные три – размеры ящика. Определить, пройдет ли ящик через дверь с заданными размерами. Ящик можно поворачивать как угодно, но проносить его можно только параллельно проемам двери.

Задание №А23.Вагон. В плацкартном вагоне 52 места: с 1 по 36 место «купе», остальные – «боковые». Снизу нечетные, сверху – четные. По заданному номеру определить тип места и вывести на экран: «купе» или «боковое», «верхнее» или «нижнее».

Задание №А24. По введенным трем положительным числам определить, можно ли построить треугольник с этими сторонами.

Задание №А25. По введенным трем положительным числам определить, будут ли эти три числа составлять стороны прямоугольного треугольника.

Задание №А26. По трем введенным углам треугольника определить его тип: остроугольный, прямоугольный или тупоугольный. На вход в программу подаются три положительных числа, сумма которых равна 180.

Задание №А27. На вход в программу подаются два положительных числа, являющиеся градусными мерами углов треугольника. Если это возможно, то найти третий угол и определить вид треугольника.
Задание №А28. Вводится трехзначное число, определить, будет ли оно одинаково читаться слева направо и наоборот.
Задание №А29. На вход программы подается трехзначное число. Его нужно преобразовать по следующему правилу: сложить первую и вторую цифры, а также вторую и третью цифры. Записать полученные суммы в порядке невозрастания. Выдать результат работы автомата на экран.
Задание №А30. Решить линейное уравнение вида ***ax=b***. Проверить решение на числах: а) 0 и 0; б) 1 и 0; в)10 и 15; г) 2 и 10. Объяснить подбор тестовых примеров.
Задание №А31. Будильник[16]. Требуется написать программу, определяющую через сколько часов зазвонит будильник. На вход в программу подаются два числа S и T. ($1 \le S, T \le 12$), где S – время завода будильника, T – время звонка будильника.

входные данные	выходные данные
10 12	2

Операторы циклов

Согласно легенде, школьный учитель математики, чтобы занять первоклассников, велел им сложить все числа от 1 до 100. Но не успел он закончить чтение условия задачи, как один из учеников написал на своей грифельной доске ответ и положил на учительский стол.

[13].

Возможно, вы догадались, какой прием для устного счета он использовал.

Решим эту задачу с помощью компьютера.

Практическая работа №3.6. Оператор цикла со счетчиком.

Цель: научиться использовать цикл со счетчиком при решении задач.

Задача. Напишем программу нахождения суммы чисел от 1 до 10.

Решение.

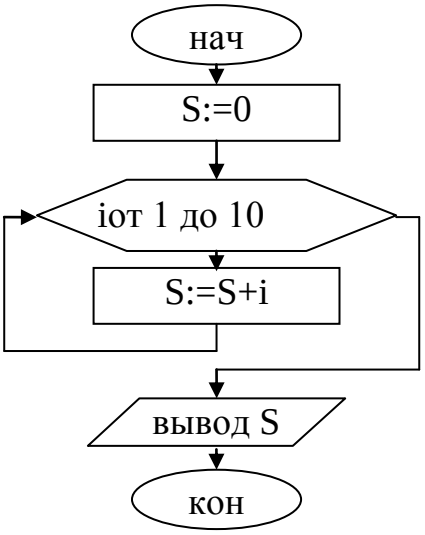


Рис. 219. Блок-схема алгоритма **сумма чисел.**

```
алг сумма чисел
нач
. цел S,i,n
. n:=10
. S:=0
. нц для i от 1 до n
. . S:=S+i
. кц
. вывод S
кон
```

Рис. 220. Код программы **сумма чисел.**

💻**Задание 1**. Запишите программу для подсчета суммы чисел от 1 до 100.

✎**Задание 2.** Посчитайте устно или письменно сумму чисел от 1 до 10 и сравните это значение с тем, которое выдал компьютер (код программы сумма чисел).

С помощью **циклов** решите следующие задачи:

Задание №А33. Найти произведение чисел от 1 до 5.

Задание №А34. Найти сумму квадратов чисел от 1 до 5.

Задание №А35. Найти сумму кубов чисел от 1 до 5.

Задание №А36. Найти сумму дробей: 1+1/2+1/3+1/4+1/5.

Задание №А37. Найти сумму чисел $1+\frac{1}{2^2}+\frac{1}{3^2}+\frac{1}{4^2}+\frac{1}{5^2}$.

Задание №А38. Возвести ***a*** в степень ***n***.

Задание №А39. Выдать квадраты чисел от 10 до 20 на экран.

Задание №А40. Найдите сумму ряда и сравните ее со значением ПИ используя[12].

а) Ряд Мадхавы–Лейбница: $\Pi = 4*\left(\frac{1}{1}-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\ldots\right)$

Для суммы возьмите 100 слагаемых.

б) Формулу Валлиса:

$$\Pi = 2*\left(\frac{2}{1}\bullet\frac{2}{3}\bullet\frac{4}{3}\bullet\frac{4}{5}\bullet\frac{6}{5}\bullet\frac{6}{7}\bullet\frac{8}{7}\bullet\frac{8}{9}\bullet\frac{10}{9}\bullet\frac{10}{11}\ldots\right)$$

Операторы циклов и условий

Практическая работа №3.7. Конструкции циклов и условий.

Цель: на практике научиться использовать вложенные конструкции циклов и условий.

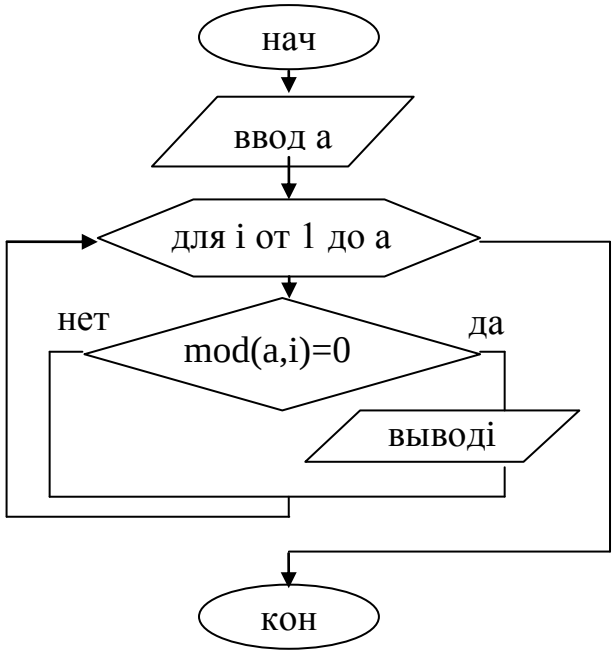


Рис. 221. Блок-схема алгоритма для вывода на экран всех делителей числа.

Задача. Вывести на экран все делители заданного числа, включая само число и единицу. Рассмотрим блок – схему решения этой задачи на рисунке 221.

Вопрос 1. По блок – схеме составьте словесный алгоритм решения задачи.

Вопрос 2. Что выдаст программа при ***a***= 3?

Вопрос 3. Что выдаст программа при ***a***=6?

Вопрос 4. Сколько делителей выдаст программа для простого числа?

Вопрос 5. Сколько циклов совершает компьютер в приведенной программе?

Вопрос 6. Как нужно изменить программу, чтобы она выдавала делители числа не включая само число и единицу?

Вопрос 7. Как можно сократить число циклов для решения этой задачи.

Задание 1. По блок – схеме напишите программу, которая выдает на экран все делители числа ***a***, включая само число и единицу.

Задание №А41. Найти количество делителей числа ***a.***

Задание №А42. Определить, является ли число простым.

Задание №А43. —

). По мере того как натуральные числа возрастают, совершенные числа встречаются всё реже. Неизвестно, бесконечно ли множество всех совершенных чисел. Составьте программу, которая определяет, является ли введенное число совершенным. Найдите в Интернет информацию о совершенных числах и проверьте правильность работы программы.

Задание №А44. Вычислите выходные результаты при следующих входных данных: **а)** 12 и 6; **б)** 13 и 7; **в)** 18 и 12; **г)** 72 и 16 для блок – схем ***а*** и ***б*** на рисунке 222.

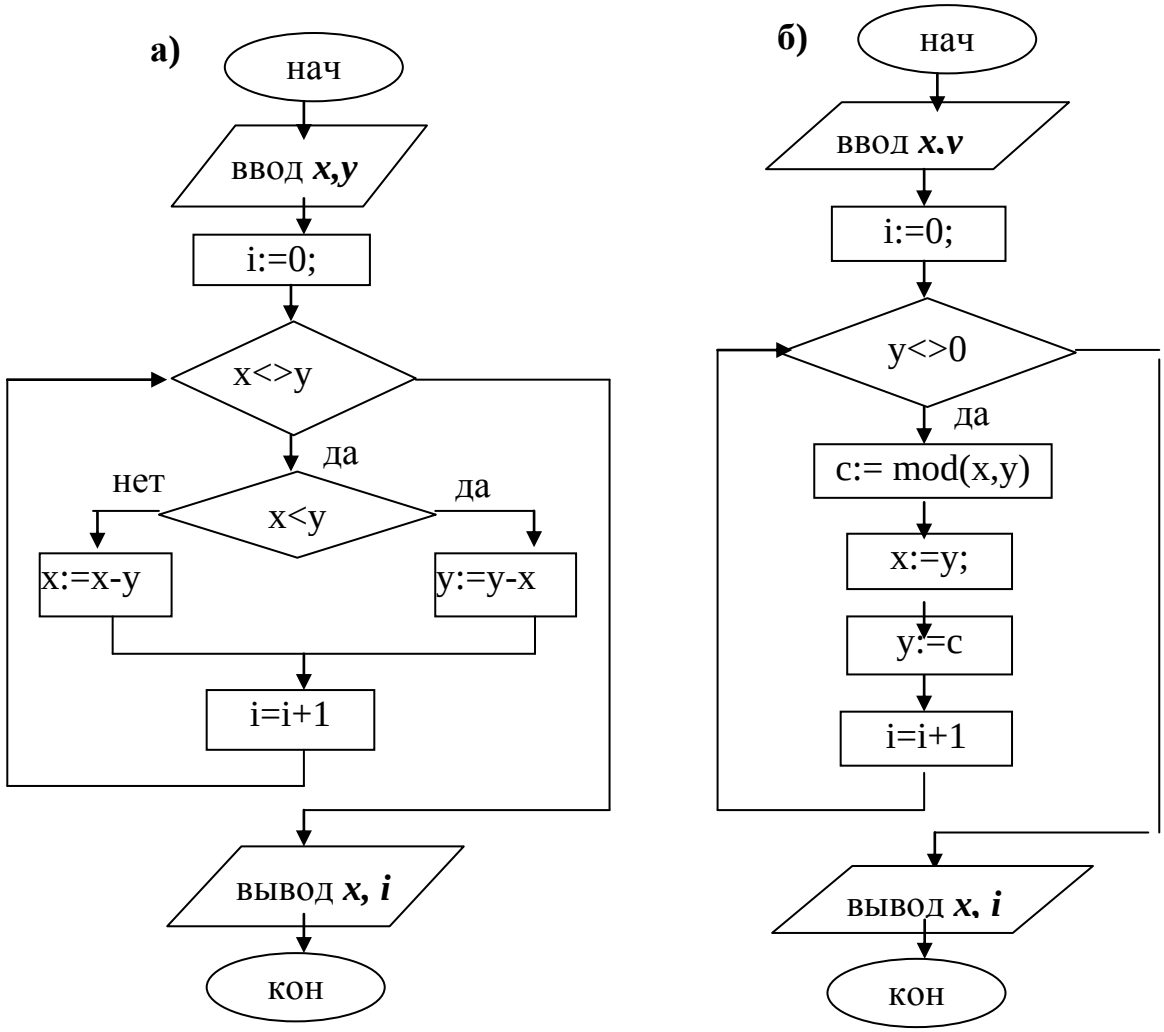


Рис. 222. Блок-схемы к заданию №А44.

Сравните результаты работы двух алгоритмов и сделайте выводы.

Запишите программы, соответствующие блок – схемам А и Б.

Задание №А45. Пифагоровы тройки.

Пифагоровы тройки известны очень давно. Многие древние сооружения использовали замечательное свойство треугольника, сторонами которого является пифагорова тройка: в таком треугольнике есть прямой угол, и он лежит напротив большей стороны. Тройка чисел, удовлетворяющих соотношению: $x^2+y^2=z^2$ является пифагоровой. Самой известной такой тройкой является египетский треугольник, стороны которого имеют длину 3, 4 и 5 единиц. Действительно, $\mathbf{5^2=3^2+4^2}$.

Найдите другие пифагоровы тройки.

Задание №А46. Задача о поимке Льва в Пустыне.

Известную задачу о «поимке Льва в пустыне» шуточными методами решали разные профессиональные сообщества. Математики предложили применить к решению этой задачи метод бинарного поиска[17].

Рассекаем пустыню линией, проходящей с севера на юг. Лев находится либо в восточной части пустыни, либо в западной. Предположим для определенности, что он находится в западной части. Рассекаем ее линией, идущей с запада на восток. Лев находится либо в северной части, либо в южной. Предположим для определенности, что он находится в южной части, рассекаем ее линией, идущей с севера на юг. Продолжаем этот процесс, воздвигая после каждого шага крепкий забор вдоль разграничительной линии. Площадь последовательно получаемых областей стремится к нулю, так что Лев в конце концов оказывается окруженным решеткой произвольно малого периметра.

Напишите программу, подсчитывающую длину забора, который необходимо выстроить для того что бы поймать Льва. Пусть по условию задачи пустыня имеет форму прямоугольника, стороны которого расположены с севера на юг и с запада на восток. Лев будет пойман, если размеры X и Y прямоугольной клетки будут находиться в пределах от 5 до 10 метров включительно: 5<=X<10, 5<=Y<10. Перегораживать пустыню забором необходимо перпендикулярно наибольшей стороне прямоугольника.

Пояснение: забор вокруг пустыни включается в длину всего забора.

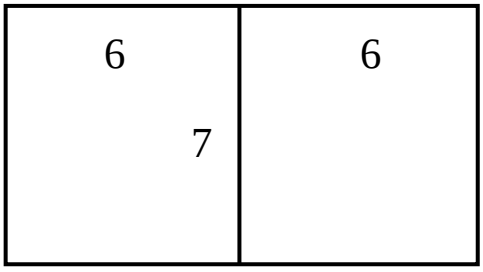


Рис. 223. Рисунок к заданию №А46

Например, если даны размеры пустыни12 и 7 метров, то длина забора:
7+7+12+12+7=45 (метров).

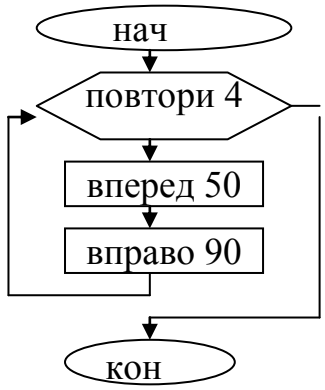
На вход в программе подаются два числа X и Y– длина и ширина пустыни в метрах. 5<=X<10000, 5<=Y<10000.

	входные данные	выходные данные
тест 1	5 5	20
тест 2	7 12	45

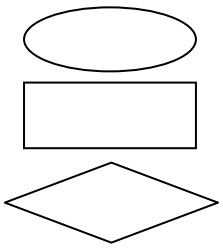
Теоретические сведения

Алгоритм –это последовательность четко сформулированных указаний **исполнителю**, которым он должен следовать для достижения результата или решения задачи [2].

Способы записи алгоритмов

способ записи	**словесный**	**блок-схема**	**программа**
описание	неформальное описание последователь-ности действий на естественном языке	способ записи алгоритмов при помощи графических блоков с указанием последовательности команд	алгоритм, написанный на языке программирования
пример: алгоритм рисование квадрата исполни-телем Черепаха	повтори четыре раза команды: пройди вперед 50 метров, повернуть направо на 90 градусов		использовать Черепаха алг **квадрат** нач нц 4 раз вперед (50) вправо (90) кц кон

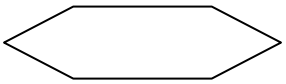
Обозначение блок-схемы



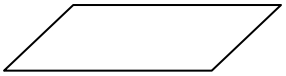
- начало и конец

- действие

- ветвление



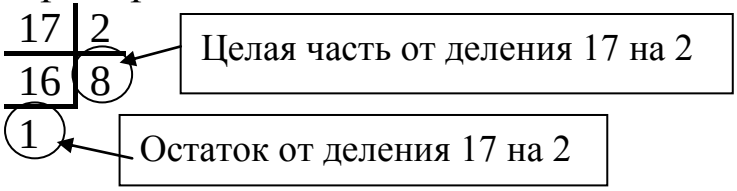
- цикл для повторения несколько раз



- ввод и вывод данных

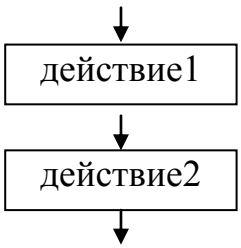
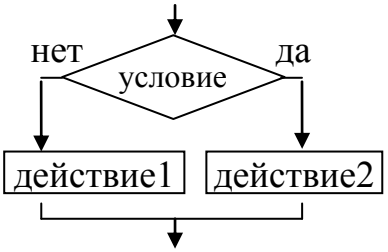
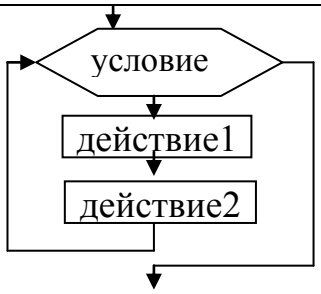
Справка. Операторы *mod* и *div*.

Оператор ***mod(a,2)*** означает взятие остатка от числа ***a*** на 2, а оператор ***div*** – взятие целой части от деления. Лучше всего это видно на примере деления в столбик:



Замечание: mod(a,2)=0, если число четное; mod(a,2)=1, если число нечетное.

Виды алгоритмов

вид алгоритма	следование	ветвление	цикл
описание	команды ***следуют*** одна за другой	команды выполняются в зависимости от ***условий***	команды выполняются ***несколько*** раз
блок - схема			

Арифметические операции для работы с числами

Название операции	Форма записи
сложение	x + y
вычитание	x – y
умножение	x * y
деление	x / y
возведение в степень	x ** y

Стандартные функции для работы счислами

Название операции	Форма записи
корень квадратный	sqrt(x)
абсолютная величина	abs(x)
остаток от деления x на y (x,y – целые)	mod(x,y)
целая часть от деления x на y (x,y – целые)	div(x,y)
целая часть числа x	int(x)
случайное число в диапазоне от 0 до x	rnd(x)
синус	sin(x)
косинус	cos(x)
тангенс	tg(x)

Список использованных источников

1. Система программирования КуМир. [Электронный ресурс]: справка в программе. – Режим доступа: https://www.niisi.ru/kumir/, свободный.

2.Алгоритм [Электронный ресурс]// Википедия. Дата обновления: 22.02.2016.– Режим доступа: http://ru.wikipedia.org/?oldid=76629481, свободный. Загл. с экрана.

3. Фиошин, М.Е. Информатика и ИКТ. 10-11 кл. профильный уровень. [Текст] В 2 ч. Ч.2: 11кл.: учеб. для общеобразоват. учреждений / М.Е. Фиошин, А.А. Рессин, С.М.Юнусов.– М.:Дрофа, 2008. –271.

4. Фрактал [Электронный ресурс]// Википедия. Дата обновления: 12.02.2016. – Режим доступа: http://ru.wikipedia.org/?oldid=76389786, свободный. Загл. с экрана.

5. Кривая Коха [Электронный ресурс] // Википедия. Дата обновления: 18.02.2016. – Режим доступа: http://ru.wikipedia.org/?oldid=76517201,свободный. Загл. с экрана.

6. Гранди, Гвидо [Электронный ресурс]// Википедия. Дата обновления: 18.10.2014. – Режим доступа: http://ru.wikipedia.org/?oldid=66278140, свободный. Загл. с экрана.

7. Энциклопедический словарь юного математика [Текст]/Сост. А. П. Савин. - М.: Педагогика, 1989. - 352 с.

8. Робот [Электронный ресурс]// Википедия. Дата обновления: 30.01.2016.– Режим доступа: http://ru.wikipedia.org/?oldid=76107130, свободный. Загл. с экрана.

9.Макаров, И. М.Робототехника: история и перспективы. [Текст]: научное издание/ Макаров И. М., Топчеев Ю. И. — М.: Наука; Изд-во МАИ, 2003. — 349 с.

10. Скатерть Улама[Электронный ресурс]// Википедия. Дата обновления: 04.12.2015. – Режим доступа: http://ru.wikipedia.org/?oldid=74903397, свободный. Загл. с экрана.

11. Федеральный институт педагогических измерений.[Электронный ресурс].– Режим доступа: http://www.fipi.ru/, свободный.

12. Пи (число) [Электронный ресурс]// Википедия. Дата обновления: 26.03.2016. – Режим доступа: http://ru.wikipedia.org/?oldid=77350026, свободный. Загл. с экрана.

13. Гаусс, Карл Фридрих [Электронный ресурс]// Википедия. Дата обновления: 28.12.2015. – Режим доступа: http://ru.wikipedia.org/?oldid=75380579, свободный. Загл. с экрана.

14. Звонкин, А.К.Алгоритмика. 5-7 класс. [Текст]: учебник для общеобразоват. учеб. заведений / А.К.Звонкин, А.Г.Кулаков, С.К.Ландо, А.Л.Семенов, А.Х.Шень. –3-е изд. –М.:Дрофа, 1998. –304 с.:ил.

15. Гейн, А.Г. Основы информатики и вычислительной техники. [Текст]: проб.учеб. для 10-11 кл. сред. шк. / А.Г. Гейн, В.Г.Житомирский, Е.В. Линецкий и др. – 3-е изд. –М.:Просвещение, 1993. – 254 с.: ил.

16. Кирюхин, В.М. Информатика. Программы внеурочной деятельности учащихся по подготовке к Всероссийской олимпиаде школьников: 5-11 классы [Текст]: программы и планирование/ В.М.Кирюхин, М.С. Цветкова. – М.: БИНОМ. Лаборатория знаний, 2014. –224 с: ил.

17. Двоичный поиск[Электронный ресурс] // Википедия. Дата обновления: 22.07.2016. – Режим доступа: http://ru.wikipedia.org/?oldid=79747830, свободный. Загл. с экрана.

Оглавление

Дрожжина Елена Владимировна

АЛГОРИТМИКА НА КУМИРЕ
Сборник заданий по программированию в системе КуМир

Отпечатано с оригинал-макетов автора
Подписано в печать: 25.09.2016. Формат 60*90 1/16
Бумага офсетная. Печ.л.3,5
Тираж 100 экземпляров. Заказ №__

Отпечатано типографией _____
_____