



Алгоритмика и программирование

Конспект занятия по теме «Алгоритмы и исполнители».

Данная тема является частью курсов

«Основы программирования в среде Visual Basic» и

«Основы программирования в среде Delphi»

Программа курсов, подбор дидактических материалов и практических работ сделаны преподавателем Тюкавиной Татьяной Михайловной.



Цель занятия:

Познакомить школьников 5-7 классов с основными понятиями темы «Алгоритмы и исполнители»

Задачи:

- познакомить школьников с определением Алгоритм со свойствами алгоритмов;
- познакомить школьников с понятием Исполнитель алгоритма и программа;
- познакомить школьников с формами записей алгоритмов;
- Рассмотреть примеры алгоритмов.

Термин «алгоритм» произошел от имени великого математика IX века **Мухаммеда аль-Хорезми** (на латыни – Algorithmi). Он разработал правила выполнения 4-х действий арифметики. Многие годы понятие алгоритма использовалось для описания правил решения математических задач. Но в жизни каждый из нас ежедневно решает разные задачи, используя тот или иной алгоритм (план действий, инструкцию).

Алгоритм – это определенная последовательность действий, выполнение которых позволяет получить решение поставленной задачи.

Каждый алгоритм создается автором (человеком или группой людей) и рассчитан для выполнения конкретным *исполнителем*.

Исполнители алгоритмов – это люди или какие – либо устройства (компьютеры, роботы, автоматы). **Алгоритм, написанный для компьютера, называется программой.**

Примеры:

Приготовление коктейля; разговор по телефону-автомату или оплата мобильного по электронной карточке; поиск слова в словаре; инструкция по использованию медицинского прибора.



Примером алгоритма может служить последовательность действий сбора в школу юного программиста Пети Торопыжкина.

Вскочить с постели; надеть рубашку; надеть пиджак; надеть правый ботинок; посмотреть на часы; если вскочил на несколько минут раньше, то лечь подремать, в противном случае надеть брюки; надеть левый носок; сходить умыться; если портфель не собран, то собрать, в противном случае позавтракать; надеть правый носок; надеть галстук; надеть левый ботинок; если холодно, то снять ботинки; надеть валенки; надеть пальто; надеть куртку; надеть шапку; выбежать из дома; вернуться назад; взять портфель; выбежать из дома; забежать за Вовкой; если он дома и еще завтракает, то, если есть время, позавтракать вместе с ним и бежать в школу вместе, в противном случае бежать в школу одному.

В большинстве случаев успех любой деятельности зависит от степени продуманности действий, их последовательности и возможных вариантов.

Составление алгоритма бессмысленно, если не учитываются *возможности его исполнителя*. Например, прочесть алгоритм решения уравнения может и первоклассник, а выполнить его навряд ли. С другой стороны, малыш 4-х лет не может прочесть алгоритм (правила) поведения за столом во время еды, но выполнить их может, если ему рассказать их. Для подобных алгоритмов надо учитывать степень детализации.

Исполнителем может быть стиральная машина, компьютер, видеомагнитофон и т.п.

Исполнителя характеризуют:

- **Среда** (место обитания);
- **Система команд (СКИ)** (т.е. команды, которые исполнитель *понимает и может* выполнить);
- **Элементарные действия** (действие исполнителя при выполнении *одной* команды);
- **Отказы** (возникают у исполнителя, если команда *не понятна* исполнителю или вызывается при недопустимом состоянии среды)

Какими свойствами должны обладать алгоритмы?

1. Понятность для исполнителя (т.е. команды только из СКИ).
2. Дискретность (прерывность, раздельность) – т.е. последовательное выполнение простых (или ранее определенных) шагов (этапов): выполнив одну команду, исполнитель может приступить к следующей.
3. Определенность (детерминированность) – т.е. каждое правило алгоритма должно быть четким, однозначным. Тогда выполнение алгоритма носит механический характер и не требуются дополнительные указания или сведения о решаемой задаче.
4. Результативность (или конечность) – алгоритм должен приводить к решению задачи за конечное число шагов.
5. Массовость (универсальность, вариативность) – применимость допустимых исходных данных.

В какой форме записываются алгоритмы?

- Словесная (запись на словесном языке)



- Графическая (язык стрелок)



- Блок - схемы



Алгоритм Гаусса

Великий немецкий математик Карл Гаусс (1777 – 1855) придумал алгоритм быстрого нахождения суммы ряда чисел от 1 до 100:

1. Подсчитать количество пар чисел от 1 до 100
2. Сложить первое и последнее числа
3. Умножить количество полученных пар чисел на получившуюся сумму

Выполнение

1. 50
если 1-е число > 1 ,
тогда число пар = $(\text{последнее число} - \text{первое число} + 1) / 2$
2. $1 + 100 = 101$
3. $50 * 101 = 5050$

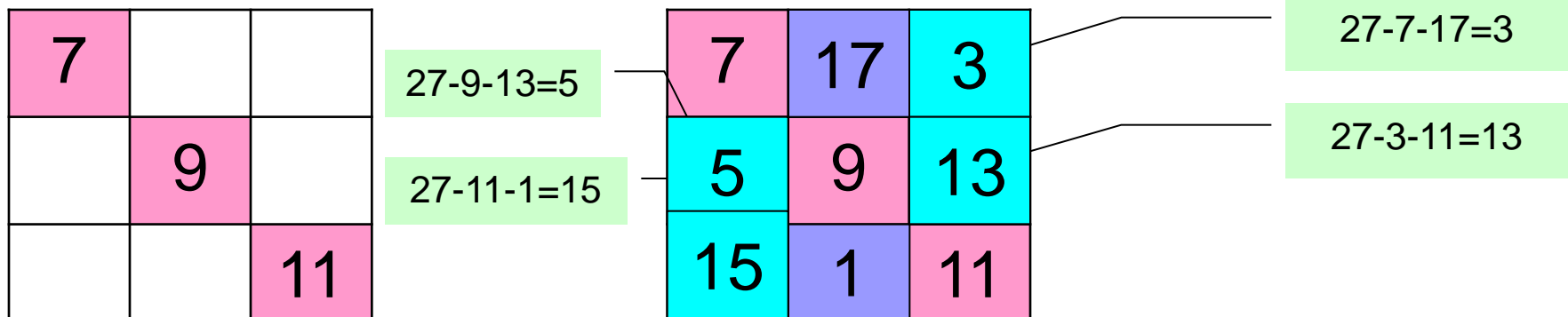
Задание

1. С помощью алгоритма Гаусса найди сумму ряда чисел от 1 до 20
2. Найди сумму ряда чисел от 4 до 13

Алгоритм составления магических квадратов

Магические – это такие *квадраты*, в которых *сумма* чисел в любом направлении равна одному и тому же числу. Чтобы составить такой квадрат, надо действовать по следующему алгоритму:

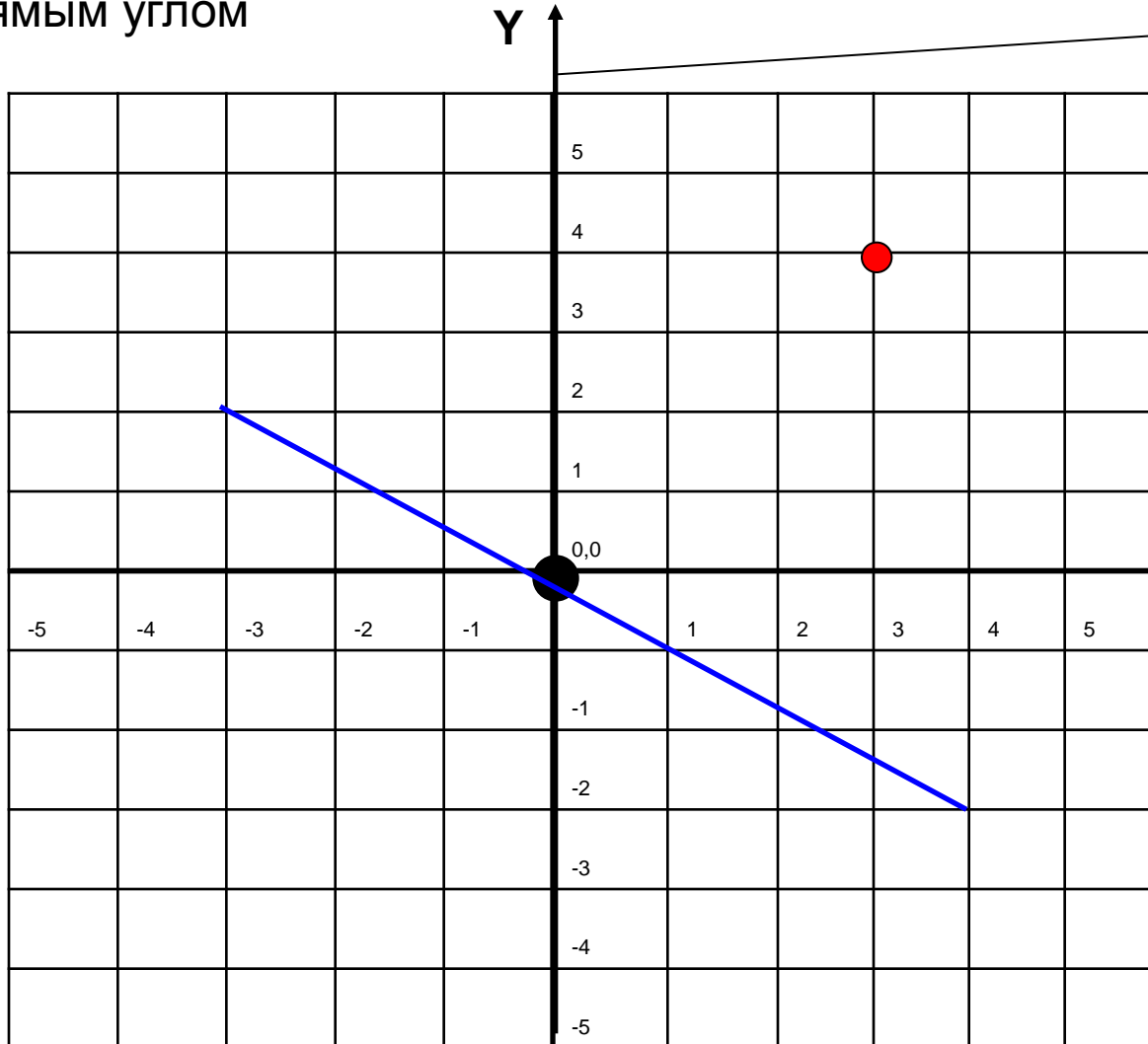
1. Подобрать 9 таких чисел, чтобы разность между соседними числами была равна одному и тому же числу
(например: 1, 3, 5, 7, 9, 11, 13, 15, 17, разность между соседними числами=2)
2. В этом ряду подчеркнуть вторую тройку чисел (7,9,11)
3. Сложить числа из этой тройки (получается *сумма магического квадрата*: $7+9+11=27$)
4. Расположить эту тройку чисел по любой диагонали в квадрате



5. Рядом с наименьшим числом (7) расположить самое большое число ряда (17)
6. Самое маленькое число ряда (1) поместить рядом с большим из трех подчеркнутых чисел (11)
7. Заполнить оставшиеся клеточки квадрата числом, недостающим до магической суммы.

Плоскость координат

На листе бумаги поставим точку и через неё проведем две прямые линии под прямым углом



Координатная ось Y

Координатная ось X

Координаты точки на плоскости указываются парой чисел X, Y
Например **(3, 4)**

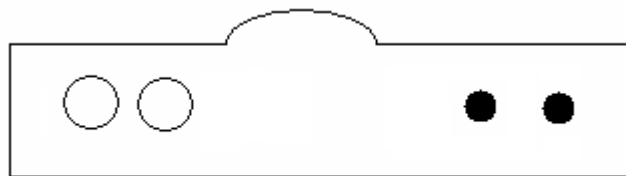
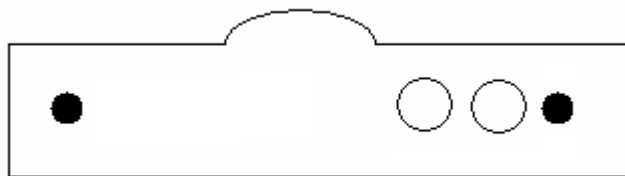
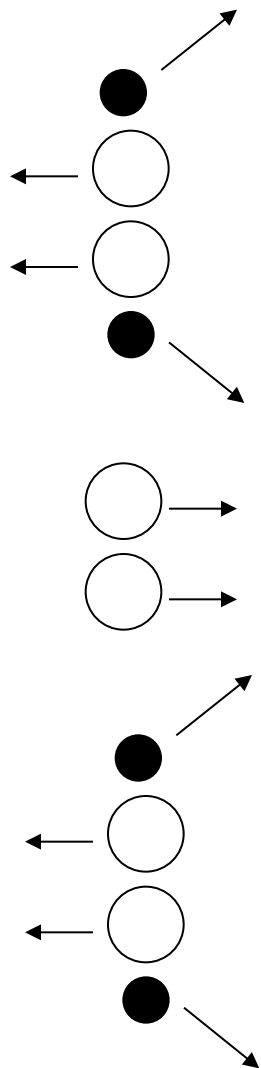
Линия на плоскости задается 2 парами координат X_1, Y_1 и X_2, Y_2
Например **(-3, 2, 4, -2)**

Этапы решения задач на компьютере.

Программа составляется программистом. Он не объясняет исполнителю (компьютеру) цель задачи и смысл команд. Программист должен понимать ограниченность возможностей компьютера как исполнителя и предусмотреть все тонкости команд. Процесс решения задачи состоит из следующих этапов:

1. Постановка задачи – смысл, ограничения, определение исходных данных, определение класса задачи (функциональная, вычислительная);
2. Выбор метода решения (построение модели задачи), выбор языка программирования;
3. Разработка алгоритма – самый творческий и трудный этап;
4. Кодирование алгоритма на языке программирования;
5. Реализация программы на компьютере – запуск и отладка (проверка) программы на конкретных вариантах данных;
6. Анализ полученных результатов – вывод о всех этапах решения.

Выполнение задания №1



В узком и очень длинном желобе находится 4 шарика: 2 черных в левом конце желоба и 2 белых чуть большего диаметра в правом конце. В стенке желоба есть небольшая ниша, в которой может поместиться только один шарик (любой). Левый край желоба закрыт, а в правом конце есть отверстие, через которое может пройти только черный шарик. Как выкатить из желоба все черные шарики? Брать шарики из желоба не разрешается

*Используется
графическая форма
(язык стрелок)
записи алгоритма*

Выполнение задания №2

Имеется бочка и 2 кувшина емкостью 3 литра и 8 литров.
Исполнитель Джинн может набирать воду из реки в каждый кувшин,
выливать из него воду и определять, налита ли вода в кувшине доверху.
Составьте алгоритм, выполнив который Джинн наберет в бочку из реки
7 литров воды.

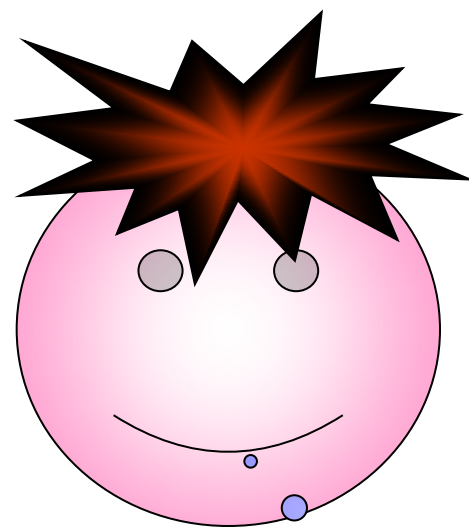
Варианты решения:

1) $3 * 5 - 8 = 7$

2) $8 * 2 - 3 * 3 = 7$

3) $8 / 2 + 3 = 7$

4) $8 - 3 + (8 - 3 - 3) = 7$



Можно
придумат
ь ещё

Выполнение задания №3

1. С помощью алгоритма Гаусса найди сумму ряда чисел от 1 до 500

2. С помощью алгоритма Гаусса найди сумму ряда чисел от 500 до 1000

Выполнение

1. Число пар = 250

$$\{ (X_n - X_1 + 1) / 2 \}$$

2. $1 + 500 = 501$ $\{ X_1 + X_n \}$

3. $250 * 501 = 125250$

Выполнение

1. Число пар = 250,5

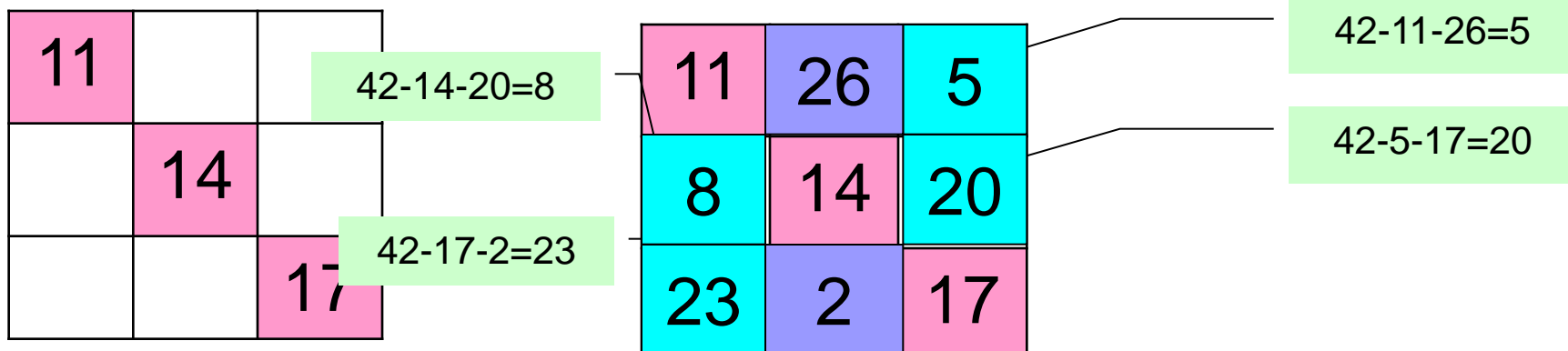
$$\{ (1000 - 500 + 1) / 2 \}$$

2. $500 + 1000 = 1500$

3. $250,5 * 1500 = 375750$

Выполнение задания №4

1. Дан ряд чисел: 2, 5, 8, 11, 14, 17, 20, 23, 26, разность между соседними числами=3)
2. В этом ряду подчеркнуть вторую тройку чисел (11, 14, 17)
3. Сложить числа из этой тройки (получается *сумма магического квадрата*: $11+14+17=42$)
4. Расположить эту тройку чисел по любой диагонали в квадрате



5. Рядом с наименьшим числом тройки (11) расположить самое большое число ряда (26)
6. Самое маленькое число ряда (2) поместить рядом с большим из трех подчеркнутых чисел (17)
7. Заполнить оставшиеся клеточки квадрата числом, недостающим до магической суммы.

Выполнение задания №5

