**Информатика 9**

**Конструирование алгоритмов**

*заметки*

**Планируемые образовательные результаты**

- *предметные* – представления о методах конструирования алгоритма; умения представлять план действий формального исполнителя по решению задачи укрупненными шагами (модулями), осуществлять детализацию каждого из укрупненных шагов формального исполнителя с помощью понятных ему команд;

- *метапредметные* – умение самостоятельно планировать пути достижения целей; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; умение оценивать правильность выполнения учебной задачи;

- *личностные* – алгоритмическое мышление, необходимое для профессиональной деятельности в современном обществе.

**Решаемые учебные задачи:**

1) познакомить учащихся с методом конструирования алгоритмов – методом пошаговой детализации;

2) рассмотреть пример разработки алгоритма методом пошаговой детализации для исполнителя Робот;

3) рассмотреть понятие и пример вспомогательного алгоритма;

4) рассмотреть понятие и пример рекурсивного алгоритма.

**Основные понятия, изучаемые на уроке:**

- алгоритм;

- последовательное построение алгоритма;

- вспомогательный алгоритм;

- формальные параметры;

- фактические параметры;

- рекурсивный алгоритм.

**Используемые на уроке средства ИКТ:**

- персональный компьютер (ПК) учителя, мультимедийный проектор, экран;

- ПК учащихся.

**Электронные образовательные ресурсы**

- презентация «Конструирование алгоритмов».

- система программирования Кумир (скачать по ссылке <https://www.niisi.ru/kumir/dl.htm>)

**Особенности изложения содержания темы урока**

***1. Организационный момент (1 минута)***

Приветствие учащихся, сообщение темы и целей урока.

***2. Повторение (5 минут)***

1) проверка изученного материала - вопросы (32-34) к §3.4;

2) визуальная проверка выполнения домашнего задания в РТ № 155-159;

3) рассмотрение заданий, вызвавших затруднения при выполнении домашней работы.

***3. Изучение нового материала (20 минут)***

Новый материал излагается в сопровождении презентации «Конструирование алгоритмов».

**1 слайд** — название презентации;  
**2 слайд** — ключевые слова;

- последовательное построение алгоритма

- вспомогательный алгоритм

- формальные параметры

- фактические параметры

- рекурсивный алгоритм

**3 слайд** — последовательное построение алгоритма;

Существуют различные методы конструирования (разработки, построения) алгоритмов. Давайте познакомимся с одним из них — методом последовательного построения (уточнения) алгоритма. Иначе он называется методом разработки «сверху вниз», нисходящим методом

или методом пошаговой детализации.

Процесс последовательного построения алгоритма выглядит следующим образом.

На первом шаге мы считаем, что перед нами совершенный исполнитель, который «всё знает и всё умеет». Поэтому достаточно определить исходные данные и результаты алгоритма, а сам алгоритм представить в виде единого предписания — постановки задачи.

**4 слайд** — последовательное построение алгоритма *(схема)*;

Если исполнитель не обучен исполнять заданное предписание, то необходимо представить это предписание в виде совокупности более простых предписании (команд). Для этого:

- задачу разбивают на несколько частей, каждая из которых проще всей задачи;

- решение каждой части задачи формулируют в отдельной команде, которая также может выходить за рамки системы команд исполнителя;

- при наличии в алгоритме предписании, выходящих за пределы возможностей исполнителя, такие предписания вновь представляются в виде совокупности ещё более простых предписаний.

Процесс продолжается до тех пор, пока все предписания не будут понятны исполнителю.

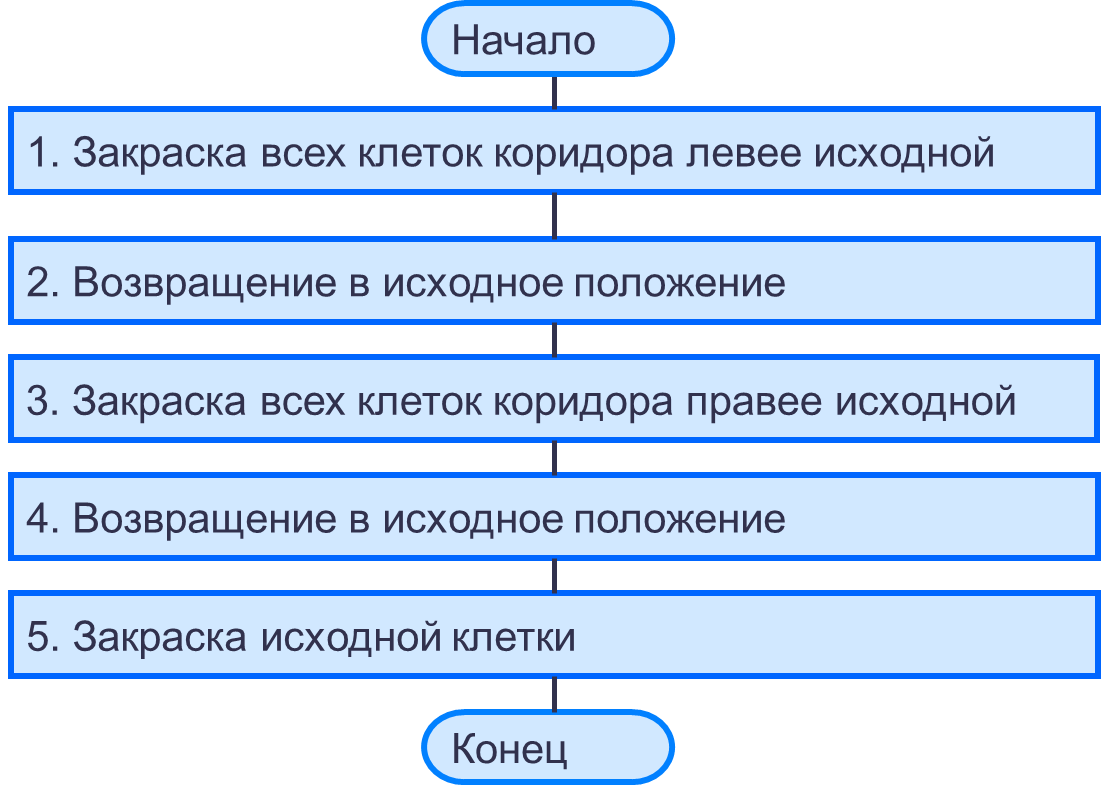
Объединяя полученные предписания в единую совокупность выполняемых в определённой последовательности команд, получаем требуемый алгоритм решения исходной задачи.

**5 слайд** — разработка алгоритма методом последовательного уточнения для исполнителя Робот;

Известно, что Робот находится где-то в горизонтальном коридоре. Ни одна из клеток коридора не закрашена.

Составим алгоритм, под управлением которого Робот закрасит все клетки этого коридора и вернётся в исходное положение.

**6 слайд** — укрупнённый план действий Робота;



**7 слайд** — детализация плана действий Робота;

1. Чтобы закрасить все клетки коридора, находящиеся левее Робота, прикажем Роботу шагнуть влево и выполнить цикл-ПОКА:

влево

**нц пока** сверху стена **и** снизу стена

закрасить; влево

**кц**

Под управлением этого алгоритма Робот закрасит все клетки коридора, находящиеся левее от него, и окажется на клетке рядом с левой границей коридора.

**8 слайд** — детализация плана действий Робота;

2. Командой *вправо* вернём Робота в коридор. Наша задача — вернуть Робота в исходную точку. Эта точка имеет единственный отличительный признак — она не закрашена. Поэтому пока занимаемая Роботом клетка оказывается закрашенной, будем перемещать его вправо.

вправо

**нц пока** клетка закрашена

вправо

**кц**

Под управлением этого алгоритма Робот окажется в исходной клетке.

**9 слайд** — детализация плана действий Робота;

3. Выполнив команду *вправо*, Робот пройдёт исходную клетку и займёт клетку правее исходной. Теперь можно закрашивать клетки коридора, расположенные правее исходной.

вправо

**нц пока** сверху стена **и** снизу стена

закрасить; вправо

**кц**

**10 слайд** — детализация плана действий Робота;

4. Так как, выполнив предыдущий алгоритм, Робот оказался правее коридора, командой *влево* вернём его в коридор. Возвращение в исходную точку обеспечивается алгоритмом:

влево

**нц пока** клетка закрашена

влево

**кц**

5. По команде *закрасить* Робот закрашивает исходную точку.

**11 слайд** — программа для Робота;

Полностью программа управления Роботом выглядит так:

**алг**

**нач**

влево

**нц пока** сверху стена **и** снизу стена

закрасить; влево

**кц**

вправо

**нц пока** клетка закрашена

вправо

**кц**

вправо

**нц пока** сверху стена **и** снизу стена

закрасить; вправо

**кц**

влево

**нц пока** клетка закрашена

влево

**кц**

закрасить

**кон**

**12 слайд** — вспомогательный алгоритм;

При построении новых алгоритмов нередко возникают ситуации, когда в разных местах алгоритма необходимо выполнение одной и той же последовательности шагов обработки данных. Для такой последовательности шагов создают отдельный алгоритм, называемый вспомогательным.

*Вспомогательный алгоритм* - алгоритм, целиком используемый в составе другого алгоритма.

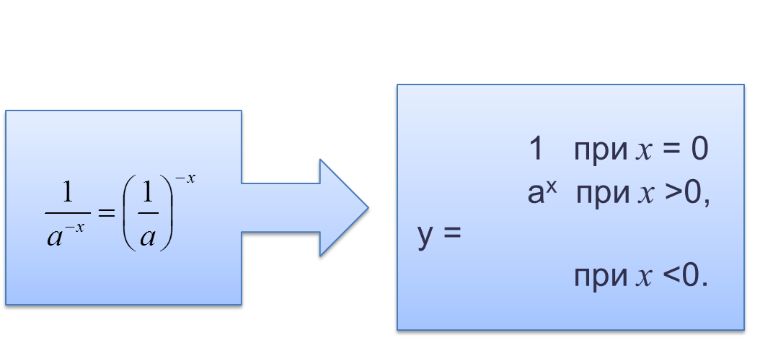
Вспомогательный алгоритм делает структуру алгоритма более простой и понятной.

**13 слайд** — алгоритм вычисления степени;

Построим алгоритм вычисления степени *y* = a*x*, где *x* - целое число, *a* ≠0.

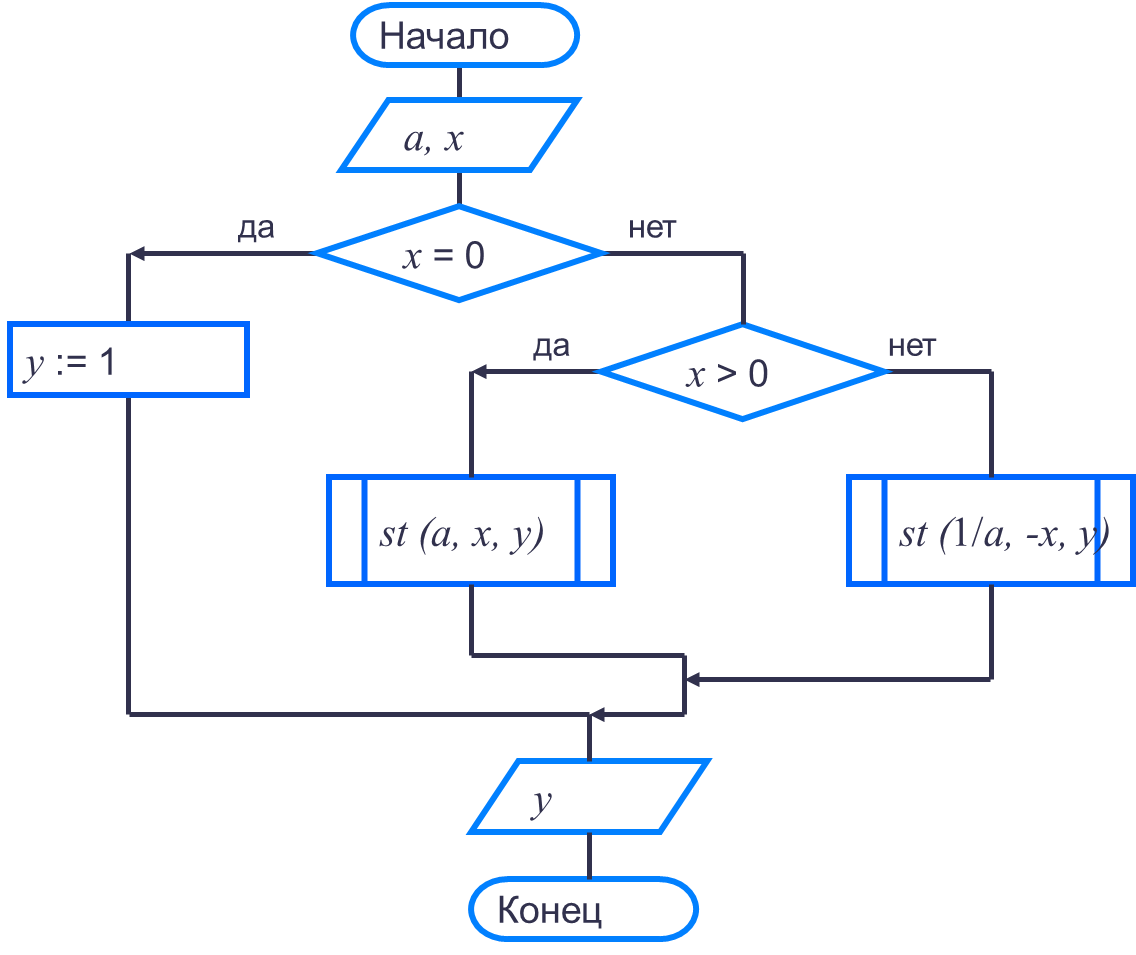
По определению степени с целым показателем:





Обозначим алгоритм возведения числа в степень *st(a, n, y)*. Это вспомогательный алгоритм.

**14 слайд** — Блок-схема решения задачи;



**15 слайд** — формальные и фактические параметры;

*Формальные параметры*используются при описании алгоритма.

*Фактические параметры*- те величины, для которых будет исполнен вспомогательный алгоритм.

Типы, количество и порядок следования формальных и фактических параметров должны совпадать.

**16 слайд** — схема вызова вспомогательного алгоритма;

Команда вызова вспомогательного алгоритма исполняется следующим образом:

1) формальные входные данные вспомогательного алгоритма заменяются значениями фактических входных данных, указанных в команде вызова вспомогательного алгоритма;

2) для заданных входных данных исполняются команды вспомогательного алгоритма;

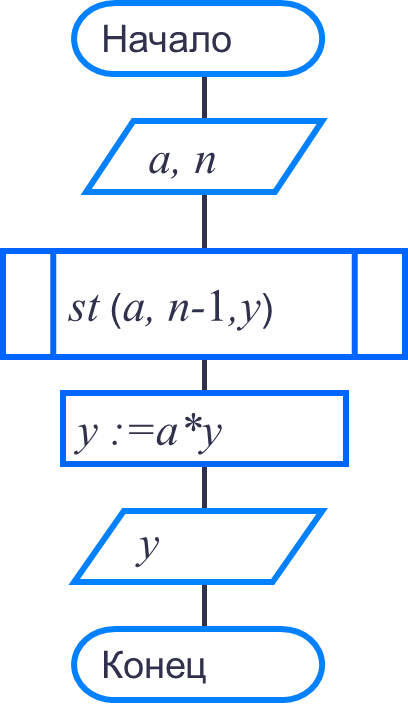
3) полученные результаты присваиваются переменным с именами фактических результатов;

4) осуществляется переход к следующей команде основного алгоритма.

**17 слайд** — рекурсивный алгоритм;

Алгоритм, в котором прямо или косвенно содержится ссылка на него же как на вспомогательный алгоритм, называют рекурсивным.

*Пример.* Алгоритм вычисления степени с натуральным показателем *n* для любого вещественного числа *а*, представленный в виде рекурсивного алгоритма.



**18 слайд** — снежинка Коха;

*Пример.* Рассмотрим алгоритм построения геометрической фигуры, которая называется снежинкой Коха. Шаг процедуры построения состоит в замене средней трети каждого из имеющихся отрезков двумя новыми той же длины.



С каждым шагом фигура становится всё причудливее. Граница снежинки Коха — положение кривой после выполнения бесконечного числа шагов.

**19 слайд** — самое главное.

*Метод последовательного построения алгоритма:*

- исходная задача разбивается на несколько частей, каждая из которых проще всей задачи, и решение каждой части формулируется в отдельной команде;

- если получаются команды, выходящие за пределы возможностей исполнителя, то они представляются в виде совокупности ещё более простых предписаний;

- процесс продолжается до тех пор, пока все предписания не будут понятны исполнителю.

*Вспомогательный алгоритм* - алгоритм, целиком используемый в составе другого алгоритма.

Алгоритм, в котором прямо или косвенно содержится ссылка на него же как на вспомогательный алгоритм, называют *рекурсивным*.

***Вопросы и задания***

**20 слайд** – вопросы и задания;

Вопросы 1-10 к параграфу 3.5

***4. Практическая часть (15 минут)***

Ученики работают с исполнителем Робот в среде КуМир, в которой выполняют задания 5 и 10 к параграфу 3.5.

Для работы необходимо установить на ПК учащихся Систему программирования КуМир (скачать ее можно по ссылке <https://www.niisi.ru/kumir/dl.htm>)

***5. Подведение итогов урока. Сообщение домашнего задания. Выставление оценок (4 минуты)***

**21 слайд** — опорный конспект;  
**22 слайд** — Д/з.

Домашнее задание.

§3.5, вопрос № 1-10 к параграфу;

РТ: № 160-164.