

Автор:

Лабанова Виктория Александровна

ученица 8 класса

БОУ «СОШ №4» г. Калачинска Омской области

Руководитель:

Лабанова Татьяна Викторовна

НАУЧНО-ПРАКТИЧЕСКАЯ РАБОТА

ТЕМА: «РЕШЕНИЕ ЗАДАЧ СРЕДСТВАМИ PASCAL ABC»

Сегодня невозможно представить специалиста в любой области, который бы не использовал в своей работе компьютерные технологии. Каждому приходится работать с информацией, а обрабатывать ее легче с помощью компьютера.

В этом году на уроках математики мы познакомились с понятием степени с натуральным показателем, научились возводить числа в степень с натуральным показателем. Мы выучили таблицу основных степеней, но все-таки довольно долго приходится возводить числа в степень.

Для увеличения скорости обработки информации эффективно использование компьютера. Почти любая задача с его помощью решается мгновенно.

На уроках информатики я узнала о языках программирования. Мне стало интересно, как с их помощью можно решить задачу возведения числа в степень с натуральным показателем, задачу сравнения двух степеней с натуральным показателем, задачу выполнения арифметических действий со степенями.

Я провела опрос среди 50 учащихся нашей школы и выяснила, что только 4% опрошенных на начальном уровне владеют средствами языка программирования **Pascal ABC**. Все они ответили, что в школе очень мало

времени отводиться на изучение языков программирования, потому я решила начать свое знакомство с языками программирования уже сегодня.

Тема исследования: решение задач средствами Pascal ABC.

Объект исследования: среда Pascal ABC.

Предмет исследования: возведение числа в натуральную степень средствами языка программирования Pascal ABC.

Цель исследования: написать программы на языке Pascal ABC для возведения числа в степень с натуральную показателем, сравнения степеней с натуральным показателем, выполнения арифметических действий со степенями.

В данной работе я рассматриваю следующие **задачи:**

- изучить понятие «язык программирования»,
- познакомиться с историей программирования,
- познакомиться со структурой программ на Pascal ABC,
- изучить основные конструкции языка Pascal ABC,
- разработать алгоритмы для решения задач,
- написать программы для решения задач и протестировать их.

Гипотеза: если изучить языки программирования, то написанные программы позволят быстро решать поставленные задачи.

Решение поставленных задач определило **методы исследования:**

- Изучение и анализ специальной и научно-популярной литературы.
- Работа в сети Интернет.
- Опрос учащихся.

Глава 1. Теоретические основы языка программирования.

1.1. Языки программирования

Язык программирования — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под её управлением^[1].

На сегодняшний день существует несколько сотен языков программирования, каждый из которых применяется в своей области.

С помощью языка программирования записывается алгоритм по определенным правилам.

Каждый язык программирования имеет свой алфавит – набор символов, который можно использовать для написания программы на этом языке. Набор правил, описывающий комбинации символов алфавита называется *синтаксис языка программирования*. Правила, определяющие смысл команд, называются *семантикой языка*. Программа, написанная на языке программирования, состоит из команд (*операторов*), задающих последовательность действий. Эти действия выполняются над некоторыми объектами. Объектами могут быть числа, текстовые строки, переменные и другие. Языки отличаются друг от друга множеством допустимых объектов и набором операций, которые можно выполнять над этими объектами^[1].

По особенностям построения языки программирования делятся на *машинно-ориентированные* (ассемблеры), которые используются для создания драйверов, программирования различных устройств и *универсальные*, предназначенные для решения различных задач. Среди универсальных языков наиболее распространенными являются: **Ада, Basic, Java, Pascal** .

1.2. Язык программирования Pascal



Язык программирования Pascal был разработан в 1968-1971 гг. Никлаусом Виртом (рис 1) в Цюрихском институте информатики (Швейцария). Первоначальная цель разработки языка PASCAL - необходимость инструмента "для обучения программированию как систематической дисциплине"^[2].

Данный язык программирования позволял решать разнообразные задачи, поэтому получил широкое распространение. Сегодня данный язык считается наиболее распространенным, он имеет множество диалектов, расширений.

Язык Pascal получил свое название в честь французского математика, физика, литератора и философа Блеза Паскаля.

Рис. 1

1 <https://ru.wikipedia.org/wiki>

2 <http://pascalabc.net/>

В нашей школе в рамках знакомства с языками программирования изучается система **PascalABC.NET**, которую совместно разработали российские и немецкие программисты.

1.3. Программирование основных конструкций языка Pascal

На сайте <http://pascalabc.net/> я познакомилась с основными символами языка, типами данных, некоторыми операторами, структурой программы в среде PascalABC.NET.

Паскаль включает следующий набор основных символов: 26 латинских строчных и 26 латинских прописных букв, пробел, знак подчеркивания, 10 цифр, знаки операций: + — * / = <> <> <= >=: = @, ограничители: ., ' () [] (.) { } (* *) .::; спецификаторы: ^ # \$; служебные (зарезервированные) слова.

В **PascalABC.NET** имеются следующие операции: *, /, +, -, =, >, <, <>, <=, >=.

Типы данных в **PascalABC.NET** подразделяются на простые, структурированные, типы указателей, процедурные типы, последовательности и классы.

Операторы.

Оператор присваивания имеет вид: *переменная := выражение*

Оператор присваивания заменяет текущее значение переменной значением выражения.

Составной оператор предназначен для объединения нескольких операторов в один. Он имеет вид:

begin

операторы

end

В **PascalABC.NET** можно описывать переменные внутри составного оператора begin-end в специальном **операторе описания переменной**. Такие описания называются внутриблочными.

Внутриблочное описание имеет одну из форм:

var список имен: тип; или **var** имя: тип := выражение; или **var** имя: тип = выражение; // Для совместимости с Delphi или **var** имя := выражение;

Имена в списке перечисляются через запятую.

Оператор цикла for имеет одну из двух форм:

for переменная := начальное значение **to** конечное значение **do**
оператор или
for переменная := начальное значение **downto** конечное значение **do**
оператор

Кроме того, переменную можно описать непосредственно в заголовке цикла:

for переменная: тип := начальное значение **to** или **downto** конечное значение **do**
оператор или
for var переменная := начальное значение **to** или **downto** конечное значение **do**
оператор

Условный оператор имеет полную и краткую формы.

Полная форма условного оператора выглядит следующим образом:

if условие **then** оператор1
else оператор2

В качестве условия указывается некоторое логическое выражение. Если условие оказывается истинным, то выполняется оператор1, в противном случае выполняется оператор2.

Краткая форма условного оператора имеет вид:

if *условие* **then** *оператор*

Если условие оказывается истинным, то выполняется *оператор*, в противном случае происходит переход к следующему оператору программы.

В случае конструкции вида

if *условие1* **then**

if *условие2* **then** *оператор1*

else *оператор2*

else всегда относится к ближайшему предыдущему оператору **if**, для которого ветка **else** еще не указана. Если в предыдущем примере требуется, чтобы **else** относилась к первому оператору **if**, то необходимо использовать составной оператор:

if *условие1* **then**

begin

if *условие2* **then** *оператор1*

end

else *оператор2*

Оператор выбора выполняет одно действие из нескольких в зависимости от значения некоторого выражения, называемого *переключателем*. Он имеет следующий вид:

case *переключатель* **of**

список выбора 1: оператор1; ...

список выбора N: операторN;

else *оператор0*

end;

Переключатель представляет собой выражение порядкового типа или строкового типа, а списки выбора содержат константы совместимого по присваиванию типа. Как и в операторе **if**, ветка **else** может отсутствовать.

Оператор **case** работает следующим образом. Если в одном из списков выбора найдено текущее значение переключателя, то выполняется оператор,

соответствующий данному списку. Если же значение переключателя не найдено ни в одном списке, то выполняется оператор по ветке **else** или, если ветка **else** отсутствует, оператор **case** не выполняет никаких действий.

Список выбора состоит либо из одной константы, либо для перечислимого типа из диапазона значений вида **a..b** (константа **a** должна быть меньше константы **b**); можно также перечислить несколько констант или диапазонов через запятую. Списки выбора не должны пересекаться.

1.4. Структура программы на Pascal ABC

Программа содержит ключевые слова, идентификаторы, комментарии. Ключевые слова используются для выделения синтаксических конструкций и подсвечиваются жирным шрифтом в редакторе. Идентификаторы являются именами объектов программы и не могут совпадать с ключевыми словами.

Программа на языке **PascalABC.NET** имеет следующий вид:

program *имя программы*;

раздел uses

раздел описаний

begin

операторы

end.

Первая строка называется *заголовком программы* и не является обязательной.

Раздел **uses** состоит из нескольких подряд идущих секций **uses**, каждая из которых начинается с ключевого слова **uses**, за которым следует список имен модулей и пространств имен **.NET**, перечисляемых через запятую.

Раздел описаний может включать следующие подразделы, которые следуют друг за другом в произвольном порядке: раздел описания переменных, раздел описания констант и другие.

Далее следует блок **begin/end**, внутри которого находятся операторы, отделяемые один от другого символом "точка с запятой". Среди операторов

может присутствовать оператор описания переменной, который позволяет описывать переменные внутри блока.

Раздел **uses** и раздел описаний могут отсутствовать.

Глава 2. Решение задач в среде программирования Pascal –ABC.

Для решения задачи возведения числа в натуральную степень я рассмотрела задачу возведения числа в квадрат, а после задачу возведения числа в натуральную степень.

2.1. Задача № 1. Возвести в квадрат заданное число.

Решение.

Для того чтобы предложить пользователям ввести число будем использовать оператор **writeln**.

Чтобы ввести заданное число с помощью оператора **readln** необходимо задать одну переменную, обозначим ее **a**, тип переменные зададим **real**.

Чтобы получить квадрат числа, присвоим переменной **a** значение, равное произведению ее заданного значения на саму себя. То есть **a := a * a**.

Для вывода результата на экран будем использовать оператор **writeln**.

Таким образом, получена следующая программа:

```
var  a: real;

begin
    writeln ('Введите число, квадрат которого необходимо вычислить:');
    readln(a);
    a := a * a;
    writeln('Квадрат числа =', a);
end.
```

2.2. Задача № 2. Возвести заданное число в степень с заданным натуральным показателем.

Решение.

Для того чтобы предложить пользователям ввести число и показатель степени будем использовать оператор **writeln**.

Чтобы ввести основание и показатель степени с помощью оператора **readln** необходимо задать две переменные, обозначим их **a**, **b**, тип переменных зададим **integer**.

Введем еще две переменные **c** – результат и **i** для задания цикла.

Присвоим результату **c** значение 1.

Запустим цикл с помощью оператора **for**, который будет повторяться **b** раз и в котором **c** будем присваивать значение равное произведению **c** на **a**.

После **b** повторений можно выводить результат работы программы.

Для вывода результата на экран будем использовать оператор **writeln**.

Таким образом, получена следующая программа:

```
var  b, i: integer;
      a, c: real;
begin
  writeln ('Введите основание степени:');
  readln(a);
  writeln ('Введите показатель степени:');
  readln(b);
  c:=1;
  for i:= 1 to b do  begin
    c:= c * a;
  end;
  writeln(a,'^',b,'степени =', c);
end.
```

2.3. Задача № 3. Сравнить две степени с натуральным показателем.

Решение.

Для того чтобы предложить пользователям ввести основания и показатели двух степеней будем использовать оператор **writeln**.

По аналогии с задачей 2 вычислим значение двух степеней.

Используя условный оператор **if**, сравним полученные значения. Если значения равны, сообщим об этом пользователю, иначе, если первая степень

больше, сообщим об этом пользователю, в противном случае, сообщим, что вторая степень больше первой.

Для вывода результата на экран будем использовать оператор **writeln**.

Таким образом, получена следующая программа:

```
var b,d,i,j: integer;  
      a,c,m,n: real;  
  
begin  
  writeln ('Введите основание первой степени:');  
  readln(a);  
  writeln ('Введите показатель первой степени:');  
  readln(b);  
  writeln ('Введите основание второй степени:');  
  readln(c);  
  writeln ('Введите показатель второй степени:');  
  readln(d);  
  m:=1;  
  for i:= 1 to b do begin  
    m:= m * a;  
  end;  
  n:=1;  
  for j:= 1 to d do begin  
    n:= n * c;  
  end;  
  if m=n then  
    begin writeln('Степени равны')  
    end  
  else  
    if m > n then  
      begin  
        writeln(a,'в',b,'степени больше, чем',c,'в',d,'степени')  
      end
```

```
end
else
begin
writeln(c,'в',d,'степени больше, чем',a,'в',b,'степени')
end
```

end.

2.4. Задача № 4. Выполнить арифметические действия с двумя степенями с натуральным показателем.

Решение.

Для того чтобы предложить пользователям ввести основания и показатели двух степеней, а также знак арифметического действия, которое необходимо выполнить со степенями будем использовать оператор **writeln**.

По аналогии с задачей 2 вычислим значение двух степеней.

Используя оператор выбора **case**, выполним со степенями арифметическое действие, заданное пользователем.

Для вывода результата на экран будем использовать оператор **writeln**.

Таким образом, получена следующая программа:

```
var
  b,d,i,j: integer;
  a,c,m,n,f: real;
  k:char;
begin
  writeln ('Введите основание первой степени:');
  readln(a);
  writeln ('Введите показатель первой степени:');
  readln(b);
  writeln ('Введите основание второй степени:');
  readln(c);
  writeln ('Введите показатель второй степени:');
  readln(d);
```

```
writeln ('Введите знак арифметического действия, которое необходимо  
выполнить со степенями');  
readln(k);  
m:=1;  
for i:= 1 to b do   begin  
m:= m * a;  
end;  
n:=1;  
for j:= 1 to d do   begin  
n:= n * c;  
end;  
case k of  
'+' : f := m+n;  
'-' : f := m-n;  
'*' : f := m*n;  
else f := m/n;  
end;  
writeln ('Результат:',f);  
end.
```

Заключение

В ходе выполнения исследовательской работы решены задачи исследования, достигнута цель.

Были разработаны алгоритмы и написаны программы возведения числа в квадрат, возведения числа в степень с натуральным показателем, сравнения двух степеней с натуральным показателем, выполнения арифметических действий со степенями с натуральным показателем. Программы были протестированы.

При этом я познакомилась с историей программирования, освоила некоторые операторы среды **Pascal –ABC**.

Написание программы занимает время, но разработанная программа позволяет ускорить процесс решения задач.

Надеюсь, что полученные знания и навыки помогут мне на уроках информатики в 9 классе, а может быть и успешно сдать ОГЭ по предметам математика и информатика.

Литература

1. Семакин И.А., Информатика: Базовый курс /Семакин И.А., Залогова Л., Русаков С., Шестакова Л. – Москва: БИНОМ.,2005. – 105с.
2. TurboPascal: практикум. — СПб.: Питер, 2002. — 256 с.: ил.
3. <http://pascalabc.net/>