«Турлатовская средняя школа»
– филиал МБОУ «Александровская средняя школа»

Язык программирования Паскаль и его основные элементы

2019 год

- ***Тип урока:*** обобщения и систематизации знаний и умений.
- ***Цель:*** повторение и закрепление материала по теме «Программирование на языке Паскаль»
- **Задачи урока:**
- Образовательная: повторить и закрепить ранее изученный материал, уметь составлять программы на языке программирования Паскаль;
- Развивающая: развитие алгоритмического мышления, памяти, внимательности, умения применять полученные знания при решении задач различной направленности;
- Воспитательная: развитие познавательного интереса, логического мышления, организованности, ответственности, повысить интерес обучающихся к данной теме.

Языки программирования – это формальные языки, предназначенные для записи алгоритмов, исполнителем которых будет компьютер.

Записи алгоритмов на языках программирования называются программами.

Существуют несколько тысяч языков программирования (Basic, Cg, JOVIAL, Pascal и т.д.).

Язык программирования Паскаль был разработан в 70-х годах прошлого века Никлаусом Виртом (Швейцария). Свое название этот язык получил в честь французского ученого Блеза Паскаля, известного не только своими достижениями в математике, физике и философии, но и созданием первой в мире механической машины, выполнявшей сложение двух чисел.

Никлаус Вирт (род. 15 февраля 1934 года) — швейцарский учёный, специалист в области информатики, один из известнейших теоретиков в области разработки языков программирования, профессор компьютерных наук Швейцарской высшей технической школы Цюриха (ETHZ), лауреат премии Тьюринга 1984 года. Создатель и ведущий проектировщик языков программирования Паскаль, Модула-2, Оберон.

Блез Паска́ль (19 июня 1623 — 19 августа 1662) - французский математик, механик, физик, литератор и философ. Классик французской литературы, один из основателей математического анализа, теории вероятностей и проективной геометрии, создатель первых образцов счётной техники, автор основного закона гидростатики.

Язык Паскаль считается универсальным языком программирования, так как он может применяться для записи алгоритмов решения самых разных задач (вычислительных, обработки текстов, построения графических изображений. Поиска информации и т.д.). Он поддерживает процедурный стиль программирования, в соответствии с которым программа представляет собой последовательность операторов, задающих те или иные действия.

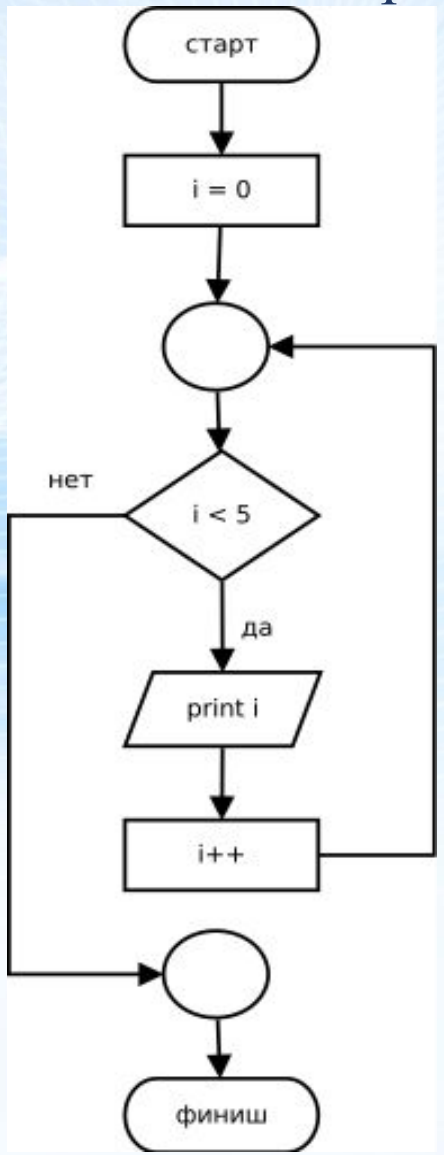
Цикл - разновидность управляющей конструкции в высокоуровневых языках программирования, предназначенная для организации многократного исполнения набора инструкций. Также циклом может называться любая многократно исполняемая последовательность инструкций, организованная любым способом (например, с помощью условного перехода).

Схема – графическое представление определения, анализа или метода решения задачи, в котором используются символы для отображения операций, данных, потока, оборудования и т.д.

Блок-схема – распространенный тип схем, описывающих алгоритмы или процессы , в которых отдельные шаги изображаются в виде блоков различной формы, соединенных между собой.

Перечень основных элементов блок-схем

Элемент блок-схемы	Наименование	Содержание
	Блок вычислений (вычислительный блок)	Вычислительные действия или последовательность действий
	Логический блок (блок условия)	Выбор направления выполнения алгоритма в зависимости от некоторого условия
	Блок ввода-вывода данных	Общее обозначения ввода (вывода) данных (вне зависимости от физического носителя)
	Начало (конец)	Начало или конец алгоритма, вход или выход в подпрограмме
	Процесс пользователя (подпрограмма)	Вычисление по стандартной программе или подпрограмме
	Блок модификации	Функция выполняет действия, изменяющие пункты (например, заголовок цикла) алгоритма
	Соединитель	Указание связи прерванными линиями между потоками информации в пределах одного листа
	Межстраничные соединения	Указание связи между информацией на разных листах



Существует три вида алгоритмов: линейный, разветвляющийся и циклический. Мы с вами приступим к изучению последнего типа алгоритмов.

В языке Pascal, как и в большинстве языков программирования, существует три типа циклических конструкций.

Любой цикл состоит из тела и заголовка. Тело цикла — это набор повторяющихся операторов, а условие — это логическое выражение, в зависимости от результата которого и происходит повторное выполнение цикла.

While, или цикл с предусловием (пока истинно условие цикла, повторяется выполнение тела цикла)

Как вы, наверное, уже поняли из названия, **while** — это цикл, в котором условие стоит перед телом. Причем тело цикла выполняется тогда и только тогда, когда условие **истинно**; как только условие становится **неправдой**, выполнение цикла прекращается.

While имеет формат: **while** < условие> **do** <тело цикла>; {Пока .. делай ..}

Данный цикл подходит только для одного оператора, если же вы хотите использовать несколько операторов в своем коде, вам следует заключить их в операторные скобки — **begin** и **end;** .

Repeat, или цикл с постусловием

Repeat - полная противоположность **while**. **Repeat** - это цикл, в котором условие стоит после тела Причем оно выполняется тогда и только тогда, когда результат условия **неправда**; как только логическое выражение становится , выполнение цикла прекращается.

Repeat имеет формат:

repeat {повторяй…}

<тело цикла>

until {до…} <условие>

Begin и **end** не требуются.

For, или цикл с параметром

For — это цикл, в котором тело выполняется заданное количество раз. При выполнении оператора for сначала вычисляется значение выражения начальное_значение, затем вычисляется значение выражения конечное_значение, далее управляющая переменная цикла последовательно пробегает все значения от начального до конечного. В том случае, когда начальное значение оказывается больше конечного значения, тело цикла не будет выполняться вовсе. Начальное и конечное значения остаются неизменными в ходе выполнения всего цикла for. Параметр цикла i, если он целого типа, пробегает все значения с приращением 1, и его текущее значение не должно изменяться операторами внутри цикла. Такое изменение не запрещено правилами языка, но его последствия будут непредсказуемы. После завершения цикла параметр i считается неопределенным.

Существует две формы записи этого цикла:

Первая форма

for <счетчик1> := <значение1> **to** <конечное_значение> **do** <тело цикла>;

После каждой итерации значение <счетчик1> будет увеличиваться на 1.

<значение1> — это начальное значение счетчика. Это может быть переменная или число.
< конечное_значение> : как только значение <счетчик1> станет больше <конечное_значение>, выполнение цикла прекратится.

Если требуется написать несколько операторов в теле цикла, используем **begin** и **end**.

И <счетчик1>, и <конечное_значение>, и <значение1> — переменные **целого** типа.

Чаще всего в качестве счетчика используется переменная i.

Вторая форма

for <счетчик2> := <значение2> downto <конечное_значение> do <оператор1>;

После каждой итерации значение <счетчик2> будет уменьшатся на 1.

<значение2> - это начальное значение счетчика.
< конечное_значение> : как только значение <счетчик2> станет меньше <конечное_значение>, выполнение цикла прекратится.

Два важных примечания:

Цикл повторяется, пока значение счетчика лежит в отрезке [значение ; конечное_значение].

Изменять значение счетчика внутри тела **нельзя**!

В стандартном и расширенном Паскале есть такие простые типы: числа с плавающей запятой (real), целые (integer), символьный (char), логический (boolean) и перечисления (конструктор нового типа, введённый в Pascal).

Turbo Pascal дополнил язык вариациями этих типов: например, shortint будет короче integer, а longint — длиннее.

Современные диалекты Pascal, такие, как FPC или Delphi, считают, что integer — это наиболее подходящий для данной машины целый, применяемый, например, для индексов массива, а shortint, longint и другие — целые определённой длины.

Система типов данных Паскаля

Название	Обозначение	Допустимые значения	Область памяти
Целочисленный	integer	- 32 768... 32 768	2 байта со знаком
Вещественный	real	$\pm(2.9 * 10^{-39} ...1.7*10^{+38})$	6 байтов
Символьный	char	Произвольный символ алфавита	1 байт
Строковый	string	Последовательность символов длиной меньше 255	1 байт на символ
логический	boolean	True и False	1 байт

В Паскале подпрограммы делятся на процедуры и функции. При этом, функции явно возвращают значение (результат) определённого типа, а процедуры явно ничего не возвращают.

Синтаксически описание процедуры или функции состоит из заголовка, содержащего ключевое слово procedure или function, имени, за которым может следовать описание передаваемых (формальных) параметров в скобках. Для функции через символ «двоеточие» : указывается тип возвращаемого значения. Заголовок заканчивается символом «точка с запятой» ;. После заголовка следует тело, (возможно) содержащее секции описания локальных констант, типов, переменных, процедур, функций и (обязательно) содержащее блок операторов, после которого ставится символ «точка с запятой» ;.

Функция – подпрограмма, имеющая единственный результат, записываемый в ячейку памяти, имя которой совпадает с именем функции. Поэтому в блоке функций обязательно должен присутствовать оператор < имя_функции>:= <результат>

Процедура – подпрограмма, имеющая произвольное количество входных и выходных данных.

Описание процедуры имеет вид:

Procedure <имя_ процедуры> (<описание параметров значений>;

var: < описание параметров-переменных>);

Begin

<операторы>

End;

1тур: «Разминка» – выполни цепочку присваиваний, чему равно a и b

1 команда	2 команда
c:=5 d:=-3,5 a:=5*d b:=c a:=d-b	a:=1 c:=18 d:=a*c/9 b:=a-d a:=c

3тур: «На ошибках учатся»

Найдите ошибки в программе, постройте по данной программе блок-схему (Найдите периметр пятиугольника)

```
program  ком1;
var a, b, c, d, e, P: integer;
begin
writeln (‘введите стороны пятиугольника’);
readln (a,b,c,d,e);
P:=a+b+c+d+e;
writeln (‘периметр пятиугольника =‘, P);
end.
```

4тур: «Английский кроссворд»

							8.			
		2.					S		10.	
1.	I	F	6.		4.		Q		V	
		O	B		R	7.	R	E	A	L
	3.	R	E	P	E	A	T		R	
			G		A			12.		
			I	5.	D	O		W		
	9.	I	N	T	E	G	E	R		
								I		
								T		
							11.	E	N	D

Рефлексия и подведение итогов: