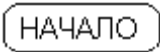
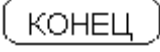
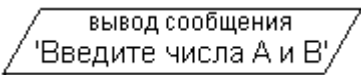
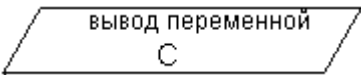
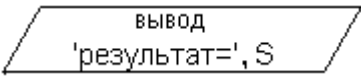



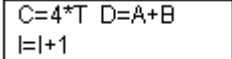
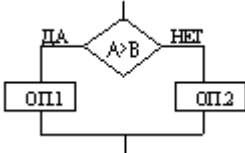
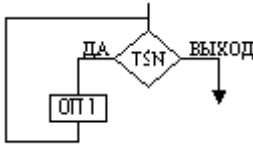
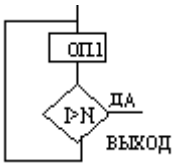
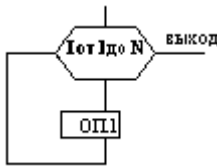
УВАЖАЕМЫЕ СТУДЕНТЫ!
ВАМ НЕОБХОДИМО ВЫПОЛНИТЬ СЛЕДУЮЩЕЕ:

1. Ознакомиться с теорией и законспектировать не менее трех страниц.
2. Запишите все таблицы примеры кодов определения и понятия из лекции.
3. Составить и ответить на вопросы.
4. Предоставит фото отчет в течении трех дней .
5. Отправить преподавателю на почту v.vika2014@mail.ru и указать свою Ф.И.О, группу, и название дисциплины тел 0721744922

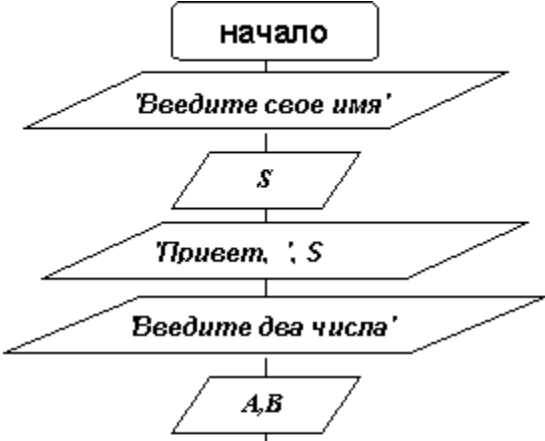
Тема: Реализация элементов блок – схемы алгоритма на языке Паскаль.

Реализация элементов блок – схемы алгоритма на языке Паскаль.

Элемент блок схемы	В программе	Действия
	BEGIN	<i>Начало работы программы (служебное слово)</i>
	END.	<i>Конец работы программы (служебное слово)</i>
	WRITE ('A,B)	<i>На экране появляется надпись: введите А, В (оператор вывода данных)</i>
	WRITE (C)	<i>На экране появляется значение переменной С. (оператор вывода данных)</i>
	WRITE ('результат=',S)	<i>На экране появляется текст результат= и значение переменной S. (оператор вывода данных)</i>
	READ (X,Y)	<i>Надо вводить два числа с клавиатуры (оператор ввода данных)</i>

	C:=4*T ; D:=A+B; I:=I+1;	После выполнения операторов, переменным присваиваются следующие значения: C=4T, D=A+B, I=I+1 (операторы присваивания)
	IF A>B THEN BEGIN ОП.1 END ELSE BEGIN ОП.2 END	Если условие A>B верно, то выполняется группа операторов ОП.1 , в противном случае – группа операторов ОП.2 (условный оператор)
	WHILE I<=N DO BEGIN ОП.1 END	Пока будет выполнено I<=N , выполняется группа операторов ОП.1 (оператор цикла с предусловием, ОП.1 – тело цикла)
	REPEAT ОП.1 UNTIL I>N	Выполняется группа операторов ОП.1 до тех пор, пока не будет выполнено условие I>N . (оператор цикла с постусловием, ОП.1 – тело цикла)
	FOR I:=1 TO N DO BEGIN ОП.1 END	Для каждого I от 1 до N выполняется группа операторов ОП.1 (оператор цикла с параметром, I – параметр цикла)

Линейный алгоритм. Простейшая программа (ввод/вывод данных, вычисление суммы, разности, произведения и частного двух чисел).

	Задание Написать программу, которая <ul style="list-style-type: none"> • знакомится с пользователем (на появившееся на экране сообщение пользователь вводит свое имя, и на экране появляется «приветствие» от компьютера с именем пользователя); • находит сумму (разность, произведение или частное) введенных с клавиатуры двух чисел); • выводит результат на экран. Примерный вид экрана при работе
---	---

		Введите свое имя Вася	
		Привет, Вася	
	Введите 2 числа 2 6		
	Сумма чисел равна 8		

Для выполнения задания можно использовать приведенный ниже текст программы или заранее подготовленный учителем файл Shablon1.pas с текстом программы, который находится в каталоге CLASS (там же находятся личные папки учащихся). Ученик проставляет вместо вопросительных знаков необходимые операторы и служебные слова. Комментарии в фигурных скобках поясняют, что необходимо сделать. Программа состоит из двух частей. В первой части программы демонстрируется использование операторов ввода и вывода, во второй, после комментария {ЗАДАНИЯ}, ученику надо самому записать необходимые операторы, используя приведенную выше блок-схему и комментарии в программе. Алгоритм создания программы по шаблону дан ниже.

Текст программы по линейному алгоритму

PROGRAM P1;

{Объявление переменной S для ввода имени, надо указать тип переменной - строковый}

VAR S: ????

{Объявление переменных A и B для ввода чисел, надо указать тип переменных - целые числа со знаком}

VAR A,B: ????

{Объявление переменной C для вывода результата, надо указать тип переменной - все действительные числа}

VAR C: ????

{Начало раздела инструкций}

BEGIN

{Оператор вывода на экран сообщения (просьба ввести имя)}

WRITE ('Введите свое имя');

{Оператор ввода данных (значение переменной S = имя пользователя)}

READLN (S);

{Вывод на экран сообщения (приглашения к работе) – слово "Привет"
и значение переменной S (введенное пользователем имя)}

WRITELN ('Привет, ', S);

{ЗАДАНИЯ:}

{1) Запишите оператор вывода на экран приглашения к вводу 2 чисел
(переменные A и B)}

???????

{2) Запишите оператор ввода для переменных A и B}

???????

{3) Запишите оператор присваивания для вычисления значения
переменной C (сумма, разность, произведение, частное двух чисел)}

C:=?????;

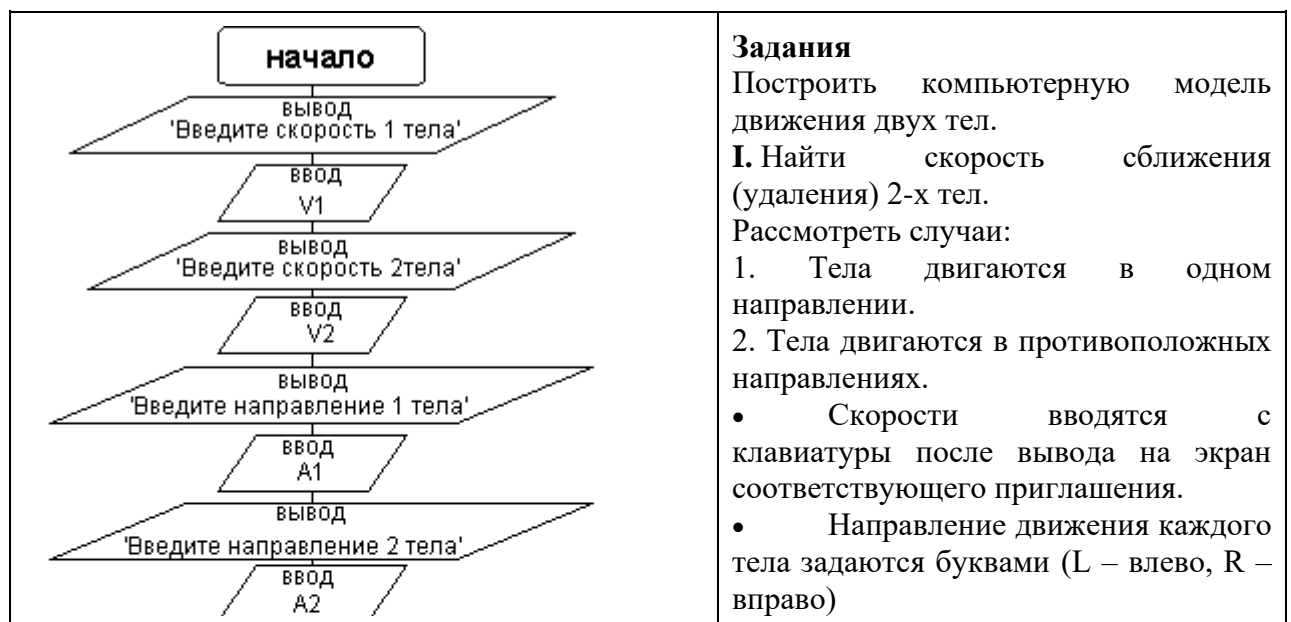
{4) Запишите оператор вывода на экран результата вычислений (сумма
(разность, произведение, частное) = <значение переменной>}

?????????

{Конец программы, конец раздела инструкций}

END.

**Ветвление. Моделирование равномерного прямолинейного
движения двух тел.**



	Примерный вид экрана при работе	
		Введите скорость 1 тела 10 Введите скорость 2 тела 5 Введите направление 1 тела L Введите направление 2 тела R Скорость равна 15
	<p>II. Добавить ввод начальных координат тел и определить сближаются или отдаляются тела.</p> <p>III. Определить расстояние между телами через время t (ввести с клавиатуры).</p> <p>IV. Выводить на экран текущие координаты тел.</p> <p>V. Выводить на экран картину движения тел.</p>	
<p>Примечания:</p> <ul style="list-style-type: none"> • блок-схема и заготовка для программы даны для I; • II, III – повышенного уровня; • IV, V – после изучения темы «Циклы». 		

Текст программы на ветвление

PROGRAM P2;

{Объявление переменных V1, V2 и V для значений скоростей, тип переменных - целые числа со знаком }

VAR V1, V2, V: ??? ;

{Объявление переменных A1 и A2 для значений направлений, значения переменных - символы}

VAR A1, A2: ??? ;

{Начало раздела инструкций}

BEGIN

{Оператор вывода на экран сообщения (просьба ввести скорость первого тела)}

WRITE ('Введите скорость 1 тела');

{Оператор ввода данных (значение переменной V1)}

READLN (V1);

{Тоже для второго тела}

????????????

??????????????

{Аналогично осуществить ввод направлений движения}

WRITE ('Введите направление 1 тела');

READLN (A1);

??????????????

??????????????

Условный оператор: проверка условия равенства значений переменных A1 и A2}

IF A1 = A2 **THEN** V := V1 - V2 **ELSE** V := V1 + V2;

{Определение модуля вектора ABS – функция вычисление абсолютной величины}

V:=ABS(V);

{Оператор вывода на экран результата вычислений }

????????

{Конец программы, конец раздела инструкций}

END.

Текст программы находится в файле Приложение3.pas (в кодировке MS DOS). Его надо переименовать в Shablon2.pas и можно использовать при создании программы по шаблону (см. алгоритм ниже).

Алгоритм создания программы по шаблону.

1. Войти в систему программирования Turbo Pascal 7.0.

2. Открыть файл *ShablonK.pas* (K - номер шаблона):

2.1. **File -> Open**

2.2. Перейти в каталог *CLASS* (в списке **Files** выбрать *..*)

2.3. Выбрать файл *ShablonK.pas* (K - номер шаблона)

2.4. Подтвердить выбор (**Open**)

3. Выполнить задание, заменяя **????**.

4. Сохранить файл в своем каталоге:

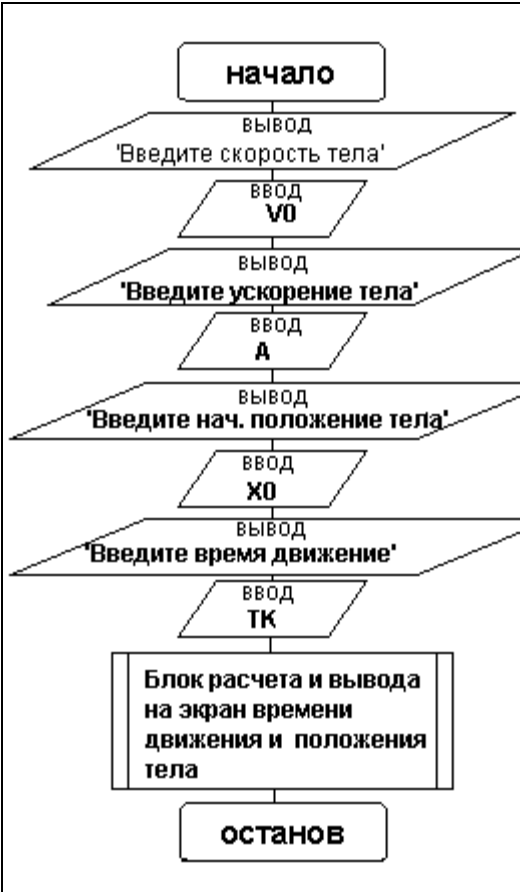
4.1. (**File -> Save as**)

4.2. Убедиться, что находитесь в своем каталоге (нижняя строчка)

- 4.3. Ввести имя файла
- 4.4. Подтвердить сохранение (*Ok*)
5. Запустить программу (*Run -> Run* или *Ctrl+F9*)
6. При наличии ошибок, внести изменения в программу и повторить пункт 5.
7. Просмотреть результат выполнения программы (*Debug User Screen* или *Alt+F5*)
8. Сохранить файл (*File -> Save* или *F2*)
9. Выйти из системы программирования (*File -> Exit* или *Alt+X*)

Для циклического алгоритма уже текст программы не дается. Учащиеся должны сами составить программу по блок – схеме.

Циклы. Моделирование равноускоренного движения.

	<p>Задания</p> <p>Построить модель равноускоренного движения тела ($X=X_0+V_0t+At^2/2$).</p> <p>I. Тело движется по прямой. Выводить на экран координату тела через каждые 10 секунд движения. Исходные данные (задаются с клавиатуры):</p> <ol style="list-style-type: none"> 1. Начальная скорость тела (V_0, м/с). 2. Ускорение тела со знаком (A, м/с²). 3. Начальное положение тела (X_0, м). 4. Время движения (TK, с). <p>Расчетные данные (выводятся на экран):</p> <ol style="list-style-type: none"> 1. Текущее положение тела (X, м). 2. Текущее время движения (T, с). <p>Примерный вид экрана при работе</p>
	<div style="background-color: black; color: white; padding: 10px;"> <p>Введите скорость тела 10</p> <p>Введите ускорение тела 2</p> <p>Введите нач. положение тела 0</p> <p>Введите время движения тела 200</p> <p>$T = 0 \quad X = 0$</p> <p>$T = 10 \quad X = 200$</p> <p>$T = 20 \quad X = 600$</p> <p>.....</p> <p>$T = 200 \quad X = 42000$</p> </div>

II. Рассмотреть случай, когда известно конечное положение тела, но неизвестно время движения.

III. Организовать ввод/вывод данных в других единицах (км, км/час, час) с пересчетом в программе.

Примечание: блок-схема и фрагменты программы даны для задания I.

Реализация блока расчета и вывода на экран времени движения и положения тела

Цикл с предусловием	Цикл с постусловием	Цикл с параметром
<pre> graph TD Init["X:=X0; T:=0;"] --> Cond{"T <= TK"} Cond --> Calc["X:=X0+V0*T+A*T^2/2"] Calc --> Out["ВЫВОД: T= ', T, ', X= ', X"] Out --> Inc["T:=T+10"] Inc --> Cond Cond --> Exit[" "] </pre>	<pre> graph TD Init["X:=X0; T:=0;"] --> Calc["X:=X0+V0*T+A*T^2/2"] Calc --> Out["ВЫВОД: T= ', T, ', X= ', X"] Out --> Inc["T:=T+10"] Inc --> Cond{"T > TK"} Cond --> Exit[" "] Cond --> Calc </pre>	<pre> graph TD Init["X:=X0; T:=0; N:=Trunc(TK/10)"] --> Cond{"i от 0 до N"} Cond --> Calc["X:=X0+V0*T+A*T^2/2"] Calc --> Out["ВЫВОД: T= ', T, ', X= ', X"] Out --> Inc["T:=T+10"] Inc --> Cond Cond --> Exit[" "] </pre>
<pre> x:=x0; t:=0; While T <= TK do begin X:=X0+V0*T+A*T*T/2; Writeln ('T = ',T,' X = ',X); T:=T+10; end; </pre>	<pre> X:=X0; T:=0; Repeat X:=X0+V0*T+A*T*T/2; Writeln ('T = ',T,' X = ',X); T:=T+10; Until T>=TK; </pre>	<pre> X:=X0; T:=0; N:=Trunc(TK/10); For i:=0 to N do begin T:=i*10; X:=X0+V0*T+A*T*T/2; Writeln ('T = ',T,' X = ',X); end; </pre>