ТЕМА УРОКА: УСЛОВНЫЙ ОПЕРАТОР.

ЦЕЛЬ УРОКА

1. Познакомить учащихся с организацией программ разветвляющейся структуры.

2. Познакомить учащихся с условным оператором программы на языке Паскаль, его полной и неполной формой.

3. Научить детей составлять элементарную программу с использованием условного оператора.

4. Научить детей решать задачи с использованием условного оператора**.**

**ОБОРУДОВАНИЕ**

1. Раздаточный теоретический материал.

2. Таблицы, иллюстрации, рисунки.

3. Программное обеспечение Турбо Paskal.

4. Тесты.

**ПЛАН УРОКА**

1. Организационный момент.

2. Проверка домашнего задания.

3. Повторение ранее изученного материала.

4. Знакомство программами разветвляющейся структуры.

5. Знакомство с условным оператором.

6. Решение задач по новой теме.

7. Тестирование.

8. Выполнение самостоятельной работы.

9. Домашнее задание.

Итог урока.

**ХОД УРОКА**

**1. Организационный момент.**

Здравствуйте, ребята! Сегодня на уроке мы изучим условный оператор на языке Паскаль. Научимся составлять программы разветвляющейся структуры, научимся решать задачи с использованием условного оператора.

**2. Проверка домашнего задания**.

Дома вы должны были написать программу, которая вычислит среднюю дневную температуру за неделю.

Какие возникли вопросы при написании программы? Какую программу вы использовали в качестве основы?

**3. Повторение ранее изученного материала.**

Ответьте на следующие вопросы:

1) Как записывается оператор, называемый оператором присваивания?

2) Какие действия выполняет оператор присваивания?

3) Назовите известные вам типы переменных, используемых в паскале.

4) Какие математические действия можно выполнять с переменными типа real?

5) Какие математические действия можно выполнять с переменными типа word?

6) Как записываются в языке Паскаль комментарии и для чего они предназначены?

7) Какими символами нельзя пользоваться при написании программы на языке Паскаль?

**4. Знакомство программами разветвляющейся структуры.**

В жизни часто приходится принимать решения в зависимости от сложившейся ситуации. Если идет дождь, мы надеваем плащ, если жарко и солнечно, идем в легкой одежде. Встречаются и более сложные положения, когда нужно сделать тот или иной выбор в зависимости от того, выполняется или не выполняется некоторое условие.

Мы знаем, что разветвляющиеся алгоритмы выглядят следующим образом:



Форма организации действий, при которой в зависимости от выполнения некоторого условия совершается одна или другая последовательность действий, называется ветвлением.

5**. Знакомство с условным оператором.**

Для реализации на Турбо-Паскале алгоритмов, содержащих структуру ветвления, используется команда **If**, которая выглядит следующим образом:

 **If** условие **Then**

     оператор 1

 **Else**

     оператор 2 ;

По этой команде сначала проверяется выполнение условия. Простые условия записываются в виде равенств или неравенств. Сложные условия составляют из простых с помощью логических операций. Если условие выполнено, то выполняется оператор 1, стоящий за служебным словом **Then**. После выполнения оператора 1 выполнение команды **If** завершается (оператор 2 не выполняется). Если условие не выполнено, то выполняется оператор 2, стоящий после слова **Else**.

Следует обратить внимание, что в каждой из ветвей условного оператора может стоять всего лишь один оператор языка Турбо-Паскаль. Если в ветви требуется выполнить несколько действий, используется составной оператор **Begin … end**;

Условный оператор предписывает выполнять некоторое действие только в том случае, когда выполняется заданное условие. Это условие записывается в виде логического выражения, а действие, которое нужно выполнить, задается в виде последовательности операторов.

Заметим, что перед ключевым словом **Else** недопустимо использование точки с запятой, означающее окончание оператора, так как это ключевое слово, так же как и **If** и **Then**, входит в его состав. Точка с запятой указывается только после полного завершения оператора, то есть после действия 2. Написание условного оператора **If** в одну или в несколько сторок зависит от вкуса программиста и длины самого оператора – в Турбо-Паскале строки с командами не должны превышать 127 символов, включая и используемые в командах пробелы.

Откройте программу usl1.pas в интегрированной среде Турбо Паскаль. Как это сделать? (войдите в среду Турбо Паскаль дважды щелкнув по ярлыку Турбо Паскаль на рабочем столе. File → Open → usl1.pas) Перед вами откроется следующая программа:

Эта программа по введенной температуре тела даёт рекомендации. Если температура высокая, то на экран выводится текст «ты болен, останься дома и лечись!», иначе на экран выводится текст «ты здоров, собирайся в школу!»



Найдите в программе условный оператор. Какое условие должно проверяться?

Запустите программу на выполнение. Как это сделать? (Run → Run) Выполните требования программы. Просмотрите результат выполнения программы. (Debug → User screen)

Мы рассмотрели программу, в которой используется полная структура ветвления. Кроме неё существует неполная структура, графическая схема которой выглядит следующим образом:



Вторая серия команд в условном операторе может отсутствовать. При этом признак ее начала – служебное слово **Else** – опускается.

Неполная команда ветвления выглядит так:

 **If** условие **Then** действие;

Здесь при справедливости условия выполняется действие, а если условие нарушено, то сразу переходим к оператору, который следует за условным оператором.

Чтобы после ключевых слов **Then** или **Else** могла выполняться последовательность операторов, последние должны быть заключены в операторные скобки – ключевые слова **Begin** и **End**. Операторные скобки охватывают фрагмент исходного текста программы, который необходимо считать одним оператором. В качестве операторов, используемых в составном операторе, могут использоваться любые операторы, допустимые в Турбо-Паскале, в том числе условные операторы, а также составные операторы.

**6. Решение задач.**

1. Составьте программу, удваивающую значение целой переменной а, если а>5.

**Решение**

Если а > 5, то значение а надо заменить на 2а. В противном случае (а <= 5) никаких действий производить не нужно. Используем неполную команду ветвления.

 **Program Pr2;**

 **Var** а : Integer;

 **Begin**

 **WriteLn** (‘Введите число’);

 **ReadLn** (а);

 **If** а >5 **Then** а := а\*2;

 **WriteLn** (‘а =', а);

 **End.**

2. Используя оператор if…then…else составьте программу, которая бы в ответ на введённую оценку по информатике выводила на экран следующий текст:

если оценка «5», то «молодец, я тобой горжусь!»(1)

если оценка «4», то «я рад, надеюсь, будет «5»» (2)

если оценка «3», то «не ленись и всё получится» (3)

иначе «ты, наверное, не ходишь на уроки» (4)

Решение.

Алгоритм решения задачи будет следующим.

**ввод** оценки

**если** оценка 5 **то** вывод на экран текста (1)

**иначе**

**если** оценка 4 **то** вывод на экран текста (2)

**иначе**

**если** оценка 3 **то** вывод на экран текста (3)

**иначе** вывод на экран текста (4)

**конец**

Здесь мы используем вложение одного условия в другое несколько раз. Программа на паскале будет иметь следующий вид:

**program** usl\_2;

**var** x:byte;

**begin** writeln ('какие оценки по информатике ты получаешь чаще всего?');

 readln(x);

 **if** x=5 **then** writeln('молодец! Я тобой горжусь!')

 **else**

 **if** x=4 **then** writeln(‘я рад! Надеюсь будет "5"')

 **else** **if** x=3 **then** writeln('не ленись и всё получится! ')

 **else** writeln('ты, наверное не ходишь на уроки')

 end .

Откройте программу usl\_2.pas и просмотрите её исполнение.



Какие возникли вопросы? Покажите начало и конец первого условного оператора; последнего условного оператора; сколько здесь всего условных операторов?

3. Коля, Петя и Вася коллекционируют марки. По введенному с клавиатуры количеству марок каждого мальчика определить у кого марок больше всех.

Решение задачи сводится к нахождению большего из трёх введённых чисел.

Блок-схема этой задачи выглядит следующим образом:

Здесь х, у, z - введённые с клавиатуры числа, наименьшее из которых нам нужно найти, p – переменная, которой мы присваиваем минимальное значение.

Вначале переменной р мы присваиваем первое число х, затем проверяем условие p<y; если условие выполнено, то р присваиваем у, иначе проверяем другое условие: p<z; если оно выполняется, то р присваиваем z.

**7. Самостоятельная работа.**

Напишите программу для решения задачи №3, отредактируйте и отправьте на выполнение.

**8.Тестирование.**

Какие условные операторы записаны без ошибок:

1) If a>=b then writeln(a) else writeln(b);

2) If a=b else writeln(b);

3) If a>b then writeln(a);

4) If a<=b else writeln(b) then writeln(a);

5) If a<b then writeln(a); else writeln(b);

Ответы. Правильно записаны операторы под номерами 1, 3.

В операторе под номером 2 пропущено ключевое слово **Then**. В операторе под номером 4 ключевые слова **Then** и **Else** поменялись местами. В операторе под номером 5 ошибочно поставлена точка с запятой перед **Else**.

**9.Домашнее задание.**

1. Прочтите теоретический раздаточный материал

2. Ответьте на следующие вопросы:

 1)Что означает фраза: «задачи с ветвлением»?

 2)Как записывается полный условный оператор?

 3)Как записывается неполный условный оператор?

 4)Какие математические действия можно выполнять в задачах с ветвлением?

 5)Какими символами в языке Паскаль можно пользоваться только при написании комментариев?

3. Составьте программу, удваивающую значение целой переменной а, если а<10 и утраивающую значение переменной, если a>=10.

**10. Итог урока.**

Сегодня на уроке мы познакомились с программами разветвляющейся структуры.Изучили условный оператор, его полную и неполную структуру. Научились использовать условный оператор при решении задач. Повторили применение комментариев, символов; правила пунктуации в Паскале. Научились решать задачи, записывать разветвляющуюся программу на Паскале.