

**Решение задач с
использованием
оператора `select case`**

Актуализация знаний

по теме: «Решение задач с использованием условного оператора»

- 1) Какой оператор языка Паскаль позволяет реализовать в программе алгоритмическую конструкцию «ветвление»? Каков его общий вид?
- 2) Поясните, как выполняется оператор `if` в ходе работы программы.
- 3) Сколько вариантов действий можно предусмотреть, используя один оператор `if` с простым условием?

Оператор выбора Case

Оператор выбора **Case** является обобщением оператора `if` – он дает возможность выполнить одно из нескольких действий в зависимости от назначения переключателя.

Логика работы оператора выбора можно описать так: вычисленное значение **Переключателя** определяет, какой из **Операторов** должен быть выполнен. Если **Переключатель** принимает одно из значений <списка значений 1>, то выполняется <оператор 1>, а остальные операторы пропускаются. Если **Переключатель** принимает одно из значений <списка значений 2>, то выполняется <оператор 2>. аналогично будут выполняться <оператор 3> и до <оператора n>. <Оператор m> будет выполняться в том случае, если значение **Переключателя** не совпало ни с одним из значений списков значений от 1-го до n-го.

Форма оператора Case

В общем виде оператор выбора записывается так:

```
Case <переключатель> of  
<список значений 1> : <оператор 1>;  
<список значений 2> : <оператор 2>;  
...  
<список значений n> : <оператор n>;  
Else <оператор m>;  
end;
```

Правила выполнения оператора

В качестве переключателя в операторе Case можно использовать переменную типа integer либо char. Переменную типа real использовать нельзя.

Значения в списках можно группировать, а именно:

записывать перечисление значений: 2,4,6

записывать диапазон значений: 10..15

использовать смешанную запись: 21,22,25..30

Одно и то же значение переключателя не должно повторяться в разных списках.

Если в каком-либо варианте значений переключателя должно выполняться несколько действий, то их записывают в виде составного оператора: `begin <действие 1>;... <действие k>; end;`

Поэтапное решение задачи

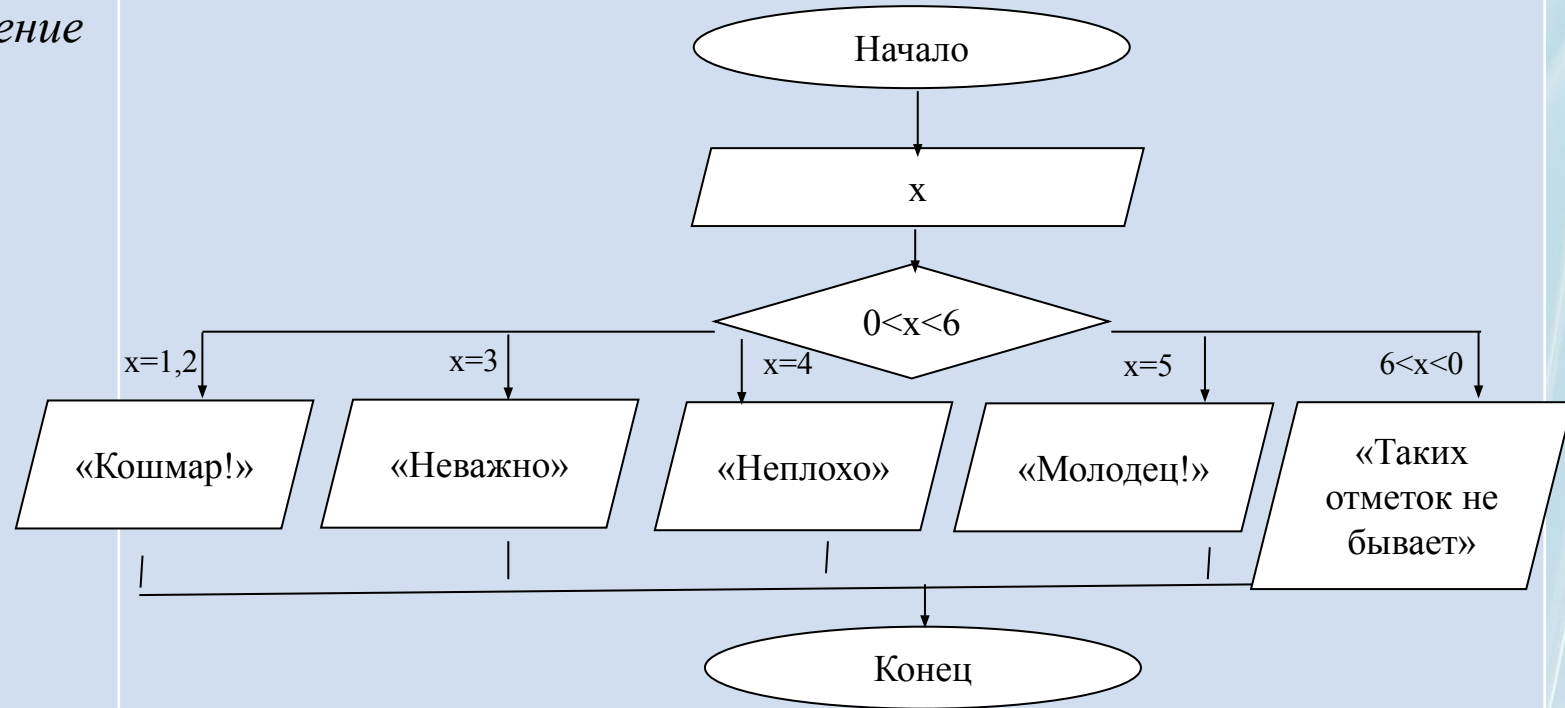
Задание 1. Дана отметка по информатике (оценка от 1 до 5). Необходимо в зависимости от отметки вывести соответствующее сообщение: 1-2 – Кошмар!; 3 – Неважно; 4 – Неполохо; 5 – Молодец!; в остальных случаях – Таких отметок не бывает

Этап решения	Описание
1. Постановка задачи – <i>определение исходных данных и искомых результатов</i>	Дано x – вводимое число в диапазоне 1-5 (тип данных integer) – это отметка по информатике. Вывести соответствующее сообщение в зависимости от вводимого числа (отметки).
2. Формализация – <i>переход к задаче обработки некоторой знаковой системы (математ. задаче)</i>	$0 < x < 6$

Этап решения

Описание

3. Построение алгоритма – *определение структуры алгоритма, последовательность команд*



Этап решения	Описание
<p>4. Составление программы – запись и отладка программы на языке программирования</p>	<p>Сначала перефразируем оператор CASE на естественный языке:</p> <p>В СЛУЧАЕ ЕСЛИ отметка РАВНА</p> <ul style="list-style-type: none">1 или 2 печатай 'Кошмар!'3 печатай 'Неважно'4 печатай 'Неплохо'5 печатай 'Молодец!' <p>ИНАЧЕ печатай 'Таких отметок не бывает'</p> <p>КОНЕЦ оператора case</p>
<p>5. Тестирование – экспериментальное доказательство правильности алгоритма и работоспособности программы</p>	

Этап решения	Описание
<p>4. Составление программы – запись и отладка программы на языке программирования</p>	<pre>Program Otmetka VAR x: Integer; BEGIN WriteLn ('Какую отметку ты получил по информатике?'); ReadLn (x); CASE x OF {Перевод: В СЛУЧАЕ ЕСЛИ отметка РАВНА...} 1,2: WriteLn ('Кошмар!'); 3: WriteLn ('Неважно'); 4: WriteLn ('Неплохо'); 5: WriteLn ('Молодец!'); ELSE WriteLn ('Таких отметок не бывает') END END.</pre>
<p>5. Тестирование – экспериментальное доказательство правильности алгоритма и работоспособности программы</p>	<p>Протестируйте программу на Турбо Паскале 7.0</p>

Домашнее задание

Семакин И. Г. Информатика. Базовый уровень: учебник для 10 класса. С. 116-131.

Выполнить самостоятельную работу студента №3-4.