

Переменные X являются идентификаторами. Приоритет операторов (от высшего к низшему) представлен в таблице. Круглые скобки могут использоваться для задания приоритета.

Высший приоритет			
**			
*	/	%	
+	-		
>	>=	<	<=
==	!=		
&&			
Низший приоритет			

- Простые операторы выглядят согласно абстрактному синтаксису.
 - skip
 - :=
- Чтение из входного потока и вывод в выходной требуют взятия аргументов в круглые скобки (без пробелов перед ними, как бы чтение и вывод были просто функциями)
 - write(1)
 - read(ident)
- Присутствует дополнительный оператор assign-and-write, записывающийся как >:= и раскрывающийся в сначала присваивание, а потом сразу же запись той же переменной.
- Присутствуют дополнительные операторы op= для арифметических и логических операций
 - a += b
 - u ||= v && False
- Условия для условного оператора и цикла берутся в круглые скобки. Тела могут быть многострочными, они в любом случае берутся в begin,end скобки;
 - if (true) then begin skip end else begin write(x); z := 13 end;
 - while (false) do begin skip ; write(x) ; z := 13 end;
- При определении функции аргументы заключаются в круглые скобки и разделяются запятыми. Тело заключается в begin,end скобки. В функции может присутствовать (а может и не присутствовать) возвращение значения через return
 - $square(A, LunarState) \leftarrow \text{begin if } (A \geq 0 \ \&\& \ LunarState) \text{ then } A * A \text{ else } (-A * -A); \text{ end}$
- При вызове функции аргументы заключаются в круглые скобки и разделяются запятыми.
 - fixEverything(A);
- Многострочные комментарии начинаются комбинацией символов (* и заканчиваются *). Вложенных комментариев нет: всё, что находится между этими символами, считается комментарием.

В конце программы, перед end и перед return точка с запятой не ставится.