Прошу меня сердечно извинить за возможно

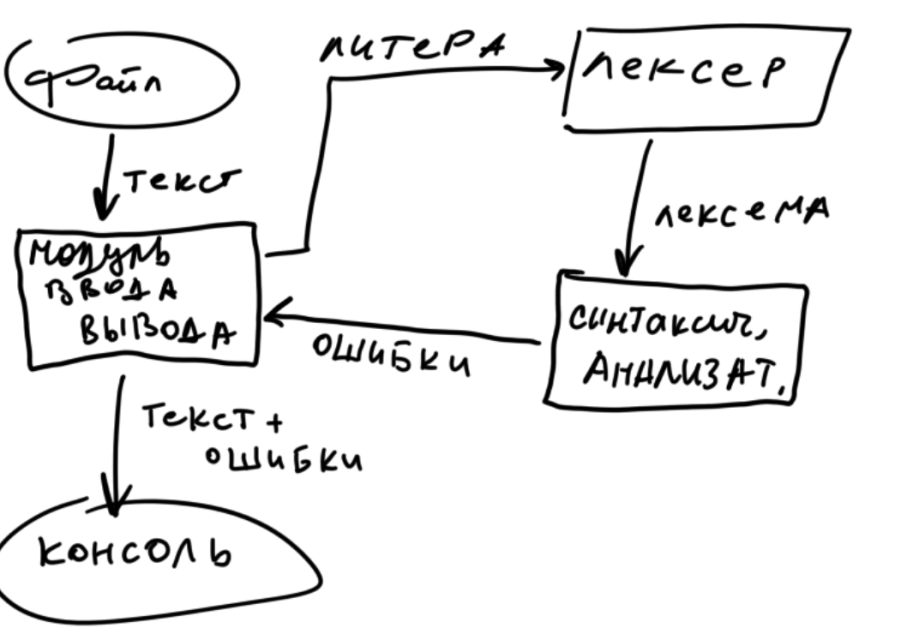
излишнее юзание транслита

**ОБЩИЕ ПОЛОЖЕНИЯ**

В целом компилятор состоит из 3х классов == 3х модулей (Ввод – вывод, Лексер, Синтаксический анализатор), а так же файлика со структурами и перечислимыми типами данных (так называемые Енамы), в том числе литеры, лексемы, иденификаторы, коды токенов, их типы и др. Программа считывается с файла и листинг выводится в консоль.

Проверка на семантические правила вшита в синтаксический анализатор.

Для большего понимания я нарисовал схемку:

****

**МОДУЛЬ ВВОДА-ВЫВОДА**

Имеет метод GetNext, который возвращает литеру.

1. Проверяет место чтения – если конец файла, то работа прекращается, если конец строки – то считывает новую строку, в других вариантах – считывает следующий символ
2. После считывания одного символа происходит сопоставление с образцами и поиск типа литеры
3. В случае ненахода модуль возвращает то значение, которое пришло, например модуль не может сопоставить символ $ и передаст его лексеру, где тот, не найдя подходящий тип токена распознает его несуществующий символ.
4. В литере хранятся данные (местоположение(строка и место в ней) и её значение)

**ЛЕКСЕР**

Получает из модуля ввода литеры и кастует их в лексемы, выдавая им тип токена и код токена (перечислимые типы данных). В случае, когда приходит несуществующий символ, создается лексема с соответствующими значениями тип = None, код = None.

**СИНТАКСИЧЕСКИЙ АНАЛИЗАТОР**

Получает из лексера лексемы и проверяет их порядок при помощи метода accept() и разного рода БНФ’ок, попутно проверяя семантику:

1. Проверяет на наличие слова program и идентификатора для программы, после чего заносит его в таблицу идентификаторов (таблица идентификаторов реализована с помощью словаря (ключ = имя идентификатора, значение = структура)).
2. Проверяет, есть ли слово var, в случае его наличия происходит запуск БНФ’ку, в которой в таблицу идентификаторов записывает все декларируемые идентификаторы, попутно генерируя исключения по надобности.
3. Составной оператор (BEGIN-END), проверяет наличие ключевый слов, и в цикле прогоняет БНФ’ки по присваиванию значения.
4. accept(точка).

**ВОЗМОЖНЫЕ ОШИБКИ:**

|  |  |
| --- | --- |
| СУТЬ ОШИБКИ | СИТУАЦИЯ ВОЗНИКНОВЕНИЯ |
| Получение неожидаемого кода токена | В параметр accept() передается код, не соответствующий коду текущей лексемы, например, в случае, когда в самом начале отсутствует слово program и accept сравнивает KWProgram и Ident, или когда отсутствует “;”. |
| Попытка создания нескольких идентификаторов с одинаковыми именами | При описании переменных в блоке VAR при повторении имени одного и того же идентификатора или при повторении имени программы |
| Попытка создания переменной с неверным типом данных | При ошибке в описании типа в блоке VAR, например bool вместо boolean |
| Попытка обратиться к несуществующему идентификатору | При обращении к переменной не описанной в блоке VAR |
| Попытка обратиться к пустому идентификатору | При обращении к переменной не содержащей значение |
| Попытка присвоить идентификатору данные неподходящего типа | При присваивании выражения, содержащего типы данных не возможные по описанию в блоке VAR, например, при попытке записать в целочисленную переменную 0.5 или при попытке присвоить новое значение идентификатору программы |
| Неверно описанное выражение | При вызове выражения, которое не может существовать, например, неравное количество открывающихся и закрывающихся скобок, арифметические знаки, после которых не следует выражение, попытка использования нечисловых данных как числовых. |

**ПРИМЕР:**

исходный код на паскале

|  |
| --- |
| program baklush:  var  x, obb, t: integer;  a, a : string;  y: boolean;  z: bool;  begin  y := True;  x := 5;  x := 5 + 0.5 ;  a := "uwu";  y := (1);  t := - +3;  y := ((((((((((-1))))))))));  x := - - t + 5 \* (37 - z) \* (42 + -3);  x := 5+ 3 - ;  z := 1888;  x := a;  baklush := 15;  x := p;  x := obb;  obb := 3;  x := 5 + obb;  end. |

листинг

|  |
| --- |
| program baklush :  ^ Ожидался токен: OpSemicolon  var  x , obb , t : integer ;  a , a : string ;  ^ An item with the same key has already been added. Key: a  y : boolean ;  z : bool ;  ^ Ожидался существующий тип данных  begin  y := True ;  x := 5 ;  x := 5 + 0.5 ;  ^ Несовместимость типов integer и real  a := "uwu" ;  y := ( 1 ) ;  ^ Ожидалось логическое выражение  t := - + 3 ;  y := ( ( ( ( ( ( ( ( ( ( -1 ) ) ) ) ) ) ) ) ) ) ;  ^ Ожидалось логическое выражение  x := - - t + 5 \* ( 37 - z ) \* ( 42 + -3 ) ;  ^ The given key 'z' was not present in the dictionary.  x := 5 + 3 - ;  ^ Ожидалось численное выражение  z := 1888 ;  ^ The given key 'z' was not present in the dictionary.  x := a ;  ^ Несовместимость числового и нечислового типов  baklush := 15 ;  ^ Невозможно присвоить значение идентификатору программы  x := p ;  ^ The given key 'p' was not present in the dictionary.  x := obb ;  ^ Идентификатор не содержит значений  obb := 3 ;  x := 5 + obb ;  end . |