

Элементы языка Turbo Pascal

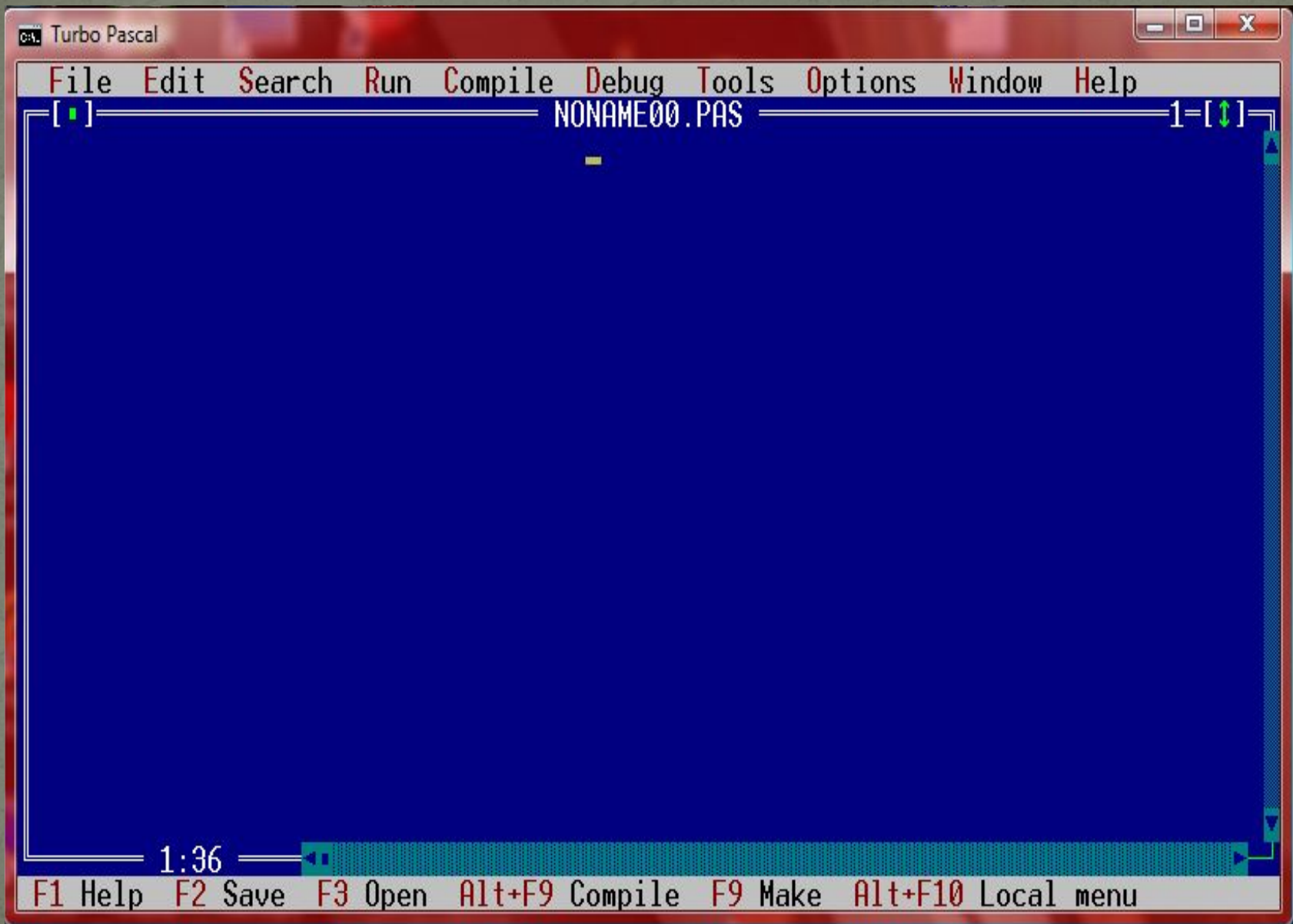
Лекция 1

История

- Язык программирования Турбо Паскаль, названный в честь французского математика и философа Блеза Паскаля (1623-1662), разработан в 1968 -1671 гг. Николаусом Виртом, профессором, директором Института информатики Швейцаркой высшей Политехнической школы. Язык Паскаль, созданный первоначально для обучения программированию как систематической дисциплине, вскоре стал широко использоваться для разработки средств в профессиональном программировании.

- Систему программирования Турбо Паскаль называют **интегрированной** (integration – объединение отдельных элементов в единое целое) средой программирования, так как она объединяет в себе возможности ранее разрозненных средств, используемых при разработке программ: редактора текстов, компилятора, компоновщика, отладчика, обеспечивая программисту следующие сервисные возможности:
 - Множества накладываемых окон;
 - Поддержка мыши, меню, диалоговых окон;
 - Многофайловый редактор;
 - Расширенные возможности отладки;
 - Полное сохранение и восстановление среды разработки.

- **редактор текстов**- используется для ввода и редактирования текста
- **компилятор**- для трансляции программы
- **компоновщик** –для построения используемого компьютером программного модуля с объединением разрозненных откомпилированных модулей и библиотекой стандартных процедур Турбо Паскаль
- **отладчик** - для отладки программ с анализом ее поведения, поиском ошибок, просмотром и изменением содержимого ячеек памяти компьютера



Алфавит

- Языком называется совокупность символов, соглашений и правил, используемых для общения. При записи алгоритма на языке программирования необходимо четко знать правила написания и использования элементарных информационных и языковых единиц. Основой Паскаля, как любого языка, является алфавит – конечный набор знаков, состоящий из букв, десятичных и шестнадцатеричных цифр, специальных символов (русские буквы в программе должны заключаться в апострофы).

Слова

- Неделимые последовательности знаков алфавита, отделенные друг от друга разделителями и несущие определенный смысл в программе. Разделителем могут служить пробел, символ конца строки, комментарий.

- Зарезервированные слова является составной частью языка, имеют фиксированное начертание и раз и навсегда определенный смысл.
- Идентификаторы – имена, используемые для обозначения программ, а в программе переменных и постоянных величин, реализуемых процедур, функций.

Константы и переменные

- *Константами* - называются элементы данных, значения которых установлены в описательной части программы и в процессе выполнения программы не изменяются.
- *Переменными* называются величины, которые могут менять свои значения в процессе выполнения программы. Каждая переменная и константа принадлежат к определенному типу данных.
- Кроме констант и переменных существуют так называемые *типизированные константы*, которые являются промежуточным звеном между переменными и константами.

Структура программы

- Заголовок программы
- Раздел описаний
 - Список имен подключаемых библиотечных модулей
 - Описание меток
 - Описание констант
 - Описание переменных
 - Описание процедур и функций
- Тело программы

Раздел подключаемых библиотечных модулей

- Этот раздел состоит из зарезервированного слова **uses** и списка имен подключаемых стандартных и пользовательских библиотечных модулей.
- Формат:

uses < имя1>,<имя2>, ...;

Например:

Uses Crt, Graph, MyLib;

Раздел описания меток.

- Метка состоит из имени и следующего за ним двоеточием. Именем может служить идентификатор или число. Раздел описания меток начинается зарезервированным словом `label`, за которым следуют имена меток, разделенными запятыми. В конце последнего имени ставиться точка с запятой.
- Формат:
`Label < имя, ..>`
- `ok10;`

- После записи метки в разделе операторов, следует двоеточие, показывающее компилятору, что идентификатор используется как метка:

Label

M1, M2; { описание метки }

begin

...

M1:< оператор > {использование M1 в разделе операторов}

...

M2:< оператор > {использование M2 в разделе операторов}

end

Раздел описания констант

- В разделе описания констант производится идентификация констант постоянных значений. Раздел начинается зарезервированным словом `const`, за которым следует ряд выражений, присваивающих идентификаторам постоянные числовые или строковые значения. Выражения присвоения отделяются друг от друга точкой с запятой.
- Формат: `const <идентификатор> = <значение>`

- Например:

Const

MaxInd:= word=100; {типизированная
константа}

Name ='Петя '; {строковая константа}

Code = \$124; {константа –
шестнадцатеричное значение}

N =10;

Pi= 3.14159265;

Раздел описания типов данных

- Тип данных может быть описан либо непосредственно в разделе описания переменных, либо определяться идентификатором типа. Стандартные типы не требуют описания.
- Раздел описания типов данных начинается зарезервированным словом **type**, за которым следуют одно или несколько определений типов (`integer`, `real`, `char`, `boolean` и т.д.), разделенных точкой с запятой.
- Формат:
type < имя типа >=<значение типа>

- Например:

type

```
LatLetter=( ' A ' .. ' z ' );
```

```
Days = 1 .. 31;
```

```
Mart = array [1 .. 10] of integer;
```

- Каждое описание задает множество значений и связывает с ЭТИМ множеством некоторое имя типа.

Раздел описания переменных

- Каждая встречающаяся в программе переменная должна быть описана. Описание обязательно предшествует использованию переменной. Раздел описания переменной начинается зарезервированным словом `var`, затем через запятую перечисляются имена переменных и через двоеточие следует их тип и точка с запятой.

- Формат :

`Var`

`<идентификатор,..> : <тип>;`

- Например:

`var`

раздела переменных}

`A, B, Proizved: integer;`

`Proizved – целые}`

`X, h, sum: real;`

`D, l, r: string;`

{описание

{переменные A, B,

Раздел описания процедур и функций

- В этом разделе размещаются тела подпрограмм. *Подпрограммой* называется программная единица, имеющая имя, по которому она может быть вызвана из других частей программы. В языке Паскаль роль подпрограмм выполняют процедуры и функции.
- Для описания подпрограмм используются зарезервированные слова **procedure** и **function**, которые записываются в начале подпрограмм.

Раздел операторов

- Это основной раздел программы. Раздел операторов начинается словом **begin**, далее следует оператор языка. Завершает раздел зарезервированное слово **end**.
- Операторы выполняются строго последовательно в том порядке, в котором они записаны в тексте программы в соответствии с синтаксисом и правилам пунктуации.

● Например:

```
begin                                {начало программы}
  Write('введите значение целого числа A>');
  {ввод запроса на экран}
  Readln(A);
  Write('введите значение целого числа B>');
  Readln(B);
  P:=A*B;                            {вычисление
  переменной P}
  Write('произведение чисел 'A,' и 'B' = 'P);
  {вывод ответа}
end.                                  {конец
  программы}
```

КОММЕНТАРИЙ

- Не выполняются программой, а служат для пояснения отдельных ее частей.
- В текст программы комментарии могут быть включены в фигурных скобках {это комментарий} или в круглых скобках в сопровождении символа «*» - (*это тоже комментарий *)