

Занятие 1.

Введение в программирование
на языке Pascal

Язык программирования

Язык программирования – это формальная знаковая система записи компьютерных программ.

Уровни языков программирования:

- Язык машинных кодов
- Язык ассемблера
- Язык высокого уровня

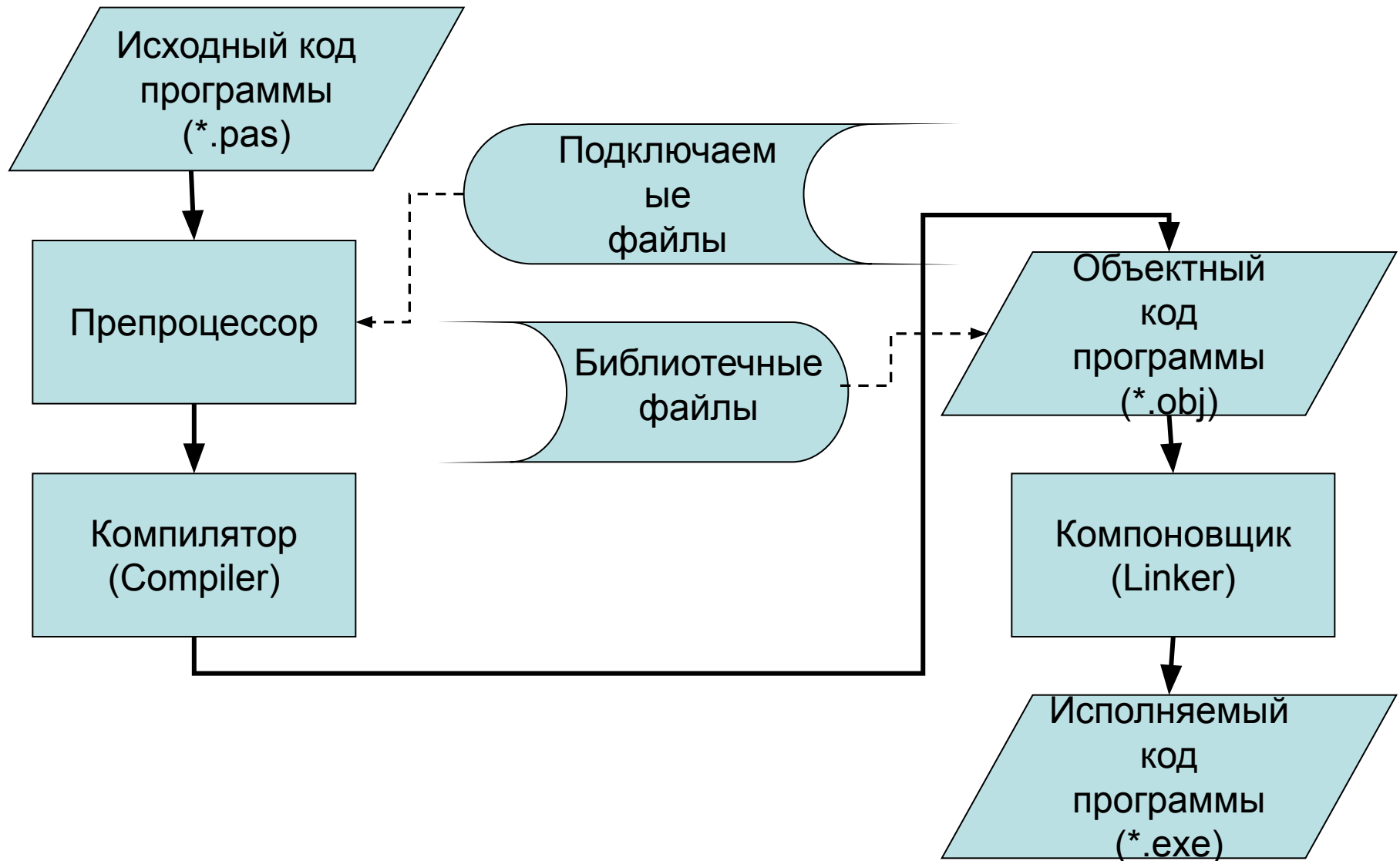
Языки программирования высокого уровня

- BASIC, Visual Basic
- C, C++, C#
- Pascal, Delphi
- Java
- PHP, Perl, Python

Философия программирования

1. Процедурное программирование
2. Структурное программирование
3. Объектно-ориентированное программирование
4. Обобщенное программирование

Процесс создания программ



Процесс создания программ

Исходный код – текст компьютерной программы на каком-либо языке программирования.

Препроцессор – компьютерная программа, подготавливающая исходный код к трансляции.

Трансляция – преобразование программы с одного языка программирования на другой.

Транслятор – программа, выполняющая трансляцию.

Процесс создания программ

Компиляция – преобразование программы с языка высокого уровня в язык машинных кодов.

Компилятор – транслятор, выполняющий компиляцию.

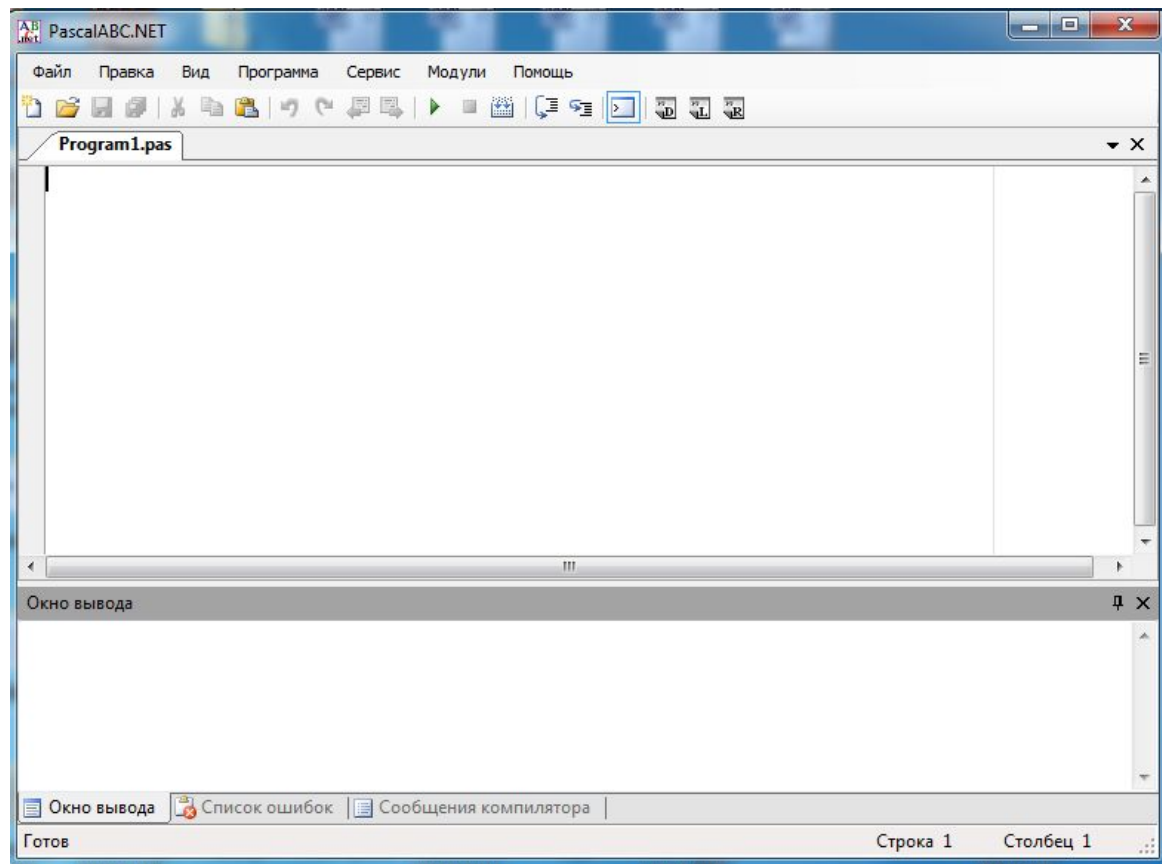
Объектный код – промежуточное представление отдельного модуля программы в виде машинных кодов с неопределенными адресами ссылок на другие модули.

Компоновщик – программа, устанавливающая связи между объектными модулями.

Окно среды разработчика

PascalABC.NET — это:

- **современный язык программирования**, основанный на Delphi (Object Pascal) и сочетающий простоту языка [Паскаль](#) и огромные возможности платформы .NET
- бесплатная, простая и мощная среда разработки, ориентированная на обучение программированию.
- уникальная [Web-среда](#), позволяющая разрабатывать и запускать программы на языке Паскаль из окна браузера, а также иметь личный каталог программ на сервере.



Панель управления

На панель управления вынесены иконки, с помощью которых выполняются наиболее часто используемые операции:



– **НОВЫЙ**



– **ОТКРЫТЬ**



– **СОХРАНИТЬ**



– **СОХРАНИТЬ ВСЕ**



– **ВЫРЕЗАТЬ**



– **КОПИРОВАТЬ**



– **ВСТАВИТЬ**



– **ОТМЕНИТЬ**



– **ВОССТАНОВИТЬ**



– **ПЕРЕЙТИ НАЗАД**



– **ПЕРЕЙТИ ВПЕРЕД**



– **ВЫПОЛНИТЬ**



– **ЗАВЕРШИТЬ**



– **КОМПИЛИРОВАТЬ**



– **ШАГ БЕЗ ВХОДА В ПОДПРОГРАММУ**



– **ШАГ С ВХОДОМ В ПОДПРОГРАММУ**



– **ОКНО ВЫВОДА**



– **ПРОСМОТРЕТЬ ЗАДАНИЯ**



– **СОЗДАТЬ ШАБЛОН ПРОГРАММЫ**



– **ПРОСМОТРЕТЬ РЕЗУЛЬТАТЫ**

Алфавит языка Pascal

Алфавит языка – набор символов,
допустимых в этом языке.

1. Латинские буквы – A, B, ..., Z; a, b, ..., z
*Язык Pascal не чувствителен к регистру (AB, Ab)
2. Арабские цифры – 0, 1, 2, ..., 9
3. Специальные знаки –
“ { } , | [] () + - / % \ ; ‘ : ? < = > ! _ & # ~ ^ . * ”
4. Неизображаемые символы – пробел,
табуляция, начало новой строки.

Лексемы языка Pascal

Лексема (token) – наименьшая, неделимая последовательность символов в тексте программы.

Типы лексем:

- идентификаторы (identifier);
- служебные слова (keyword);
- литералы (literal);
- операторы (operator);
- разделители (знаки пунктуации – punctuator).

Идентификаторы языка Pascal

- **Имена (идентификаторы)** называют элементы языка - константы, метки, типы, переменные, процедуры, функции, модули, объекты. Идентификатор в Турбо Паскале может включать в себя:
 - буквы латинского алфавита,
 - цифры
 - символ подчеркивания.
- **Идентификатор** может быть произвольной длины

Идентификаторы языка Pascal

Пример допустимых идентификаторов:

`x, x1, x_1, _abc, ABC123z, sum, RATE, count`

Пример НЕ допустимых идентификаторов:

`12, 3X, %change, data-1, my.c`

Зарезервированные слова языка Pascal

Ключевые слова – лексемы,
зарезервированные в языке и
библиотеках для специального
использования.

Ключевые слова не могут быть
использованы в качестве
идентификаторов.

Зарезервированные слова языка Pascal

Примеры:

program	else	uses
begin	false	or
case	and	mod
const	for	true
integer	if	div
double	end	while

Литералы языка Pascal

Литерал – лексема, представляющая собой непосредственно записанное в программе фиксированное значение.

Типы литералов:

целые	1 -92 01 -062 0xA -0xFF
вещественные	66. .0 .12 3.14159 1.12e-2 2E+6
логические	true, false
символьные	'z' '*' 'v' '\n'
строковые	'Hello, World!'

Оператор языка Pascal

Оператор – лексема выражающая некоторую команду языка программирования, выполняющую некоторую операцию.

+	-	*	/	&&		>	<
<=	>=	==					

Пример программы

```
Program lab1_1; {название программы}
```

```
Var a: Integer;
```

```
b: String;
```

```
Begin {начало программы}
```

```
Writeln ('Введите имя ');
```

```
Readln (b);
```

```
Writeln ('Ваш возраст? ');
```

```
Readln (a);
```

```
Writeln (b,', вам',a,'лет?');
```

```
End..{конец программы}
```

Пример программы

```
Program lab1_3; {заголовок программы}  
Var a,b,c: integer; {раздел описания  
переменных}  
BEGIN {начало раздела операторов}  
c:=5;  
b:=4; {занесение в ячейки c и b начальных  
значений}  
a:=c*b; {вычисление значения переменной a}  
writeln(a); {вынесение на экран значения a}  
END. {конец программы}.
```

Задание

Составить программу, которая

- выводит на экран имена, фамилии, даты рождения её авторов (введенные с клавиатуры)
- вычисляет сумму 4-х чисел (числа даны в тексте программы)

Переменные

Переменная – область памяти, которой назначено некоторое имя. С помощью имени осуществляется доступ к данным, хранящимся в этой области.

Переменная – это пара «ИМЯ - ЗНАЧЕНИЕ».

ИМЯ – адрес участка памяти;

ЗНАЧЕНИЕ – содержимое этого участка памяти

Атрибуты переменных

1. Имя

Имя должно являться допустимым идентификатором языка

2. Значение

Содержимое соответствующего имени участка памяти

3. Тип переменной

Определяет допустимые значения переменной.

Определяет объем участка памяти, зарезервированного за переменной

Типы переменных

Идентификатор	Длина (байт)	Диапазон значений	Операции
Целые типы			
integer	2	-32768..32767	+, -, /, *, Div, Mod, >=, <=, =, <>, <, >
byte	1	0..255	+, -, /, *, Div, Mod, >=, <=, =, <>, <, >
word	2	0..65535	+, -, /, *, Div, Mod, >=, <=, =, <>, <, >
shortint	1	-128..127	+, -, /, *, Div, Mod, >=, <=, =, <>, <, >
longint	4	-2147483648.. 2147483647	+, -, /, *, Div, Mod, >=, <=, =, <>, <, >

Типы переменных

Идентификатор	Длина (байт)	Диапазон значений	Операции
Вещественные типы			
real	6	$2,9 \times 10^{-39} - 1,7 \times 10^{38}$	+ , - , / , * , >= , <= , = , <> , < , >
single	4	$1,5 \times 10^{-45} - 3,4 \times 10^{38}$	+ , - , / , * , >= , <= , = , <> , < , >
double	8	$5 \times 10^{-324} - 1,7 \times 10^{308}$	+ , - , / , * , >= , <= , = , <> , < , >
extended	10	$3,4 \times 10^{-4932} -$ $1,1 \times 10^{4932}$	+ , - , / , * , >= , <= , = , <> , < , >

Типы переменных

Идентификатор	Длина (байт)	Диапазон значений	Операции
Логический тип			
boolean	1	true, false	Not, And, Or, Xor, >=, <=, =, <>, <, >
Символьный тип			
char	1	все символы кода ASCII	+, >=, <=, =, <>, <, >

Порядок работы с переменными

1. Объявление

- выделение области памяти и закрепление за ним определенного имени

2. Инициализация

- присвоение переменной начального значения

3. Использование

Объявление переменных

Все переменные должны быть объявлены до использования – в противном случае следует ошибка компилятора.

Синтаксис:

```
var имя_пер_1, имя_пер_2, ... : имя_типа;
```

Пример:

```
var count, numberOfItems : integer;  
distance : double;
```

Инициализация переменных

Все переменные рекомендуется
инициализировать до использования – в
противном случае возможны логические
ошибки.

Синтаксис:

```
имя_пер_2 := выр_2;
```

Пример:

```
count := 10; numberOfItems := 999;  
istance := 780.8;
```

Оператор присваивания

Оператор присваивания позволяет изменить значение переменной, стоящей слева на значение выражения, стоящего справа.

Компиляция ведется справа налево: сначала вычисляется выражение, затем результат присваивается переменной.

Синтаксис:

`имя_переменной := выражение ;`

Пример:

```
speedOfCar := 65; timeOfTrip := 3.5;
```

```
distanceOfTrip := speedOfCar * timeOfTrip;
```

Именованные константы

Именованная константа – это переменная, инициализированная некоторым значением, которое нельзя изменить в процессе выполнения программы.

Синтаксис:

```
const имя_переменной = литерал;
```

Пример:

```
const PI = 3.14;
```

```
const E = 2.71;
```

```
const TAB = '\t';
```

```
const BANK_RATE = 10;
```

Определение объема использованной памяти

В Pascal существует оператор **sizeof**, позволяющий определить размер в байтах участка памяти, выделяемой для хранения стоящего справа операнда.

Операнд – переменная, именованная константа, литеральное значение.

Пример использования оператора `sizeof`

Операнд – переменная:

```
short sh;  
Writeln(sizeof (sh) );
```

Операнд – именованная константа:

```
const c_sh = 0;  
Writeln(sizeof (c_sh) );
```

Операнд – литеральное значение:

```
Writeln(sizeof (12) );
```


Задание

1. Вывести на экран объем памяти в битах, необходимый для хранения переменных каждого из простейших типов.
2. Вывести объем памяти в битах, необходимый для хранения именованных констант.
3. Вывести на экран 2 строки, содержащие информацию о предельных значениях целочисленных типов **short** и **int**.

Арифметические операторы и выражения

Символ операции	Название операции	Пример
*	умножение	2*3 (результат: 6)
/	деление	30/2 (результат: 1.5E+01)
+	сложение	2+3 (результат: 5)
-	вычитание	5-3 (результат: 2)
div	целочисленное деление	5 div 2 (результат: 2)
mod	остаток от деления	5 mod 2 (результат: 1)

Операнды – целые и вещественные числовые типы.

Тип результата зависит от типа операндов.

Если оба операнда имеют тип целочисленный тип – результат тоже целочисленный

Если хотя бы один операнд имеет тип вещественный тип, то и результат будет вещественным

Арифметические операторы и выражения

Порядок выполнения операторов в выражениях определяется правилами старшинства операторов.

Совет: указывайте порядок выполнения операторов в выражениях с помощью круглых скобок. Чем подробнее, тем лучше.

Форматирование выводимых данных

Чтобы программа выводила данные в требуемой форме у объекта `write/writeln` есть ряд функций, позволяющих управлять отображением данных на экране.

Общий формат:

```
write (имя_переменной: количество_цифр:  
      количество_знаков_после_запятой) ;
```

Форматирование выводимых данных

Например, имеем переменную "a" типа real. Ранее мы присвоили ей значение 12.5435.

И теперь нам необходимо её вывести на экран.

```
write(a:7:4);
```

Так число выведется в первоначальном виде.

Т. е. всего 7 знаков (с учетом запятой) и 4 после запятой.

Соответственно 2 знака под целую часть.

Также необходимо знать, что для вещественных переменных, если не задавать форматированный вывод, они выводятся в экспоненциальной форме.

Например, наша переменная, будет выглядеть примерно так:

```
1.2543000000e+01
```

Ввод данных

При вводе исходных данных происходит преобразование из внешней формы представления во внутреннюю, определяемую типом переменных. Переменные, образующие список ввода, могут принадлежать либо к целому, либо к действительному, либо к символьному типам. Чтение исходных данных логического типа в языке Паскаль недопустимо.

Значения исходных данных могут отделяться друг от друга пробелами и нажатием клавиш табуляции и Enter.

Не допускается разделение вводимых чисел запятыми!

Ввод данных

Синтаксис:

`Readln(перемен_1, перемен_2 , ...)`

Пример:

`ReadLn(A1,A2,...AK);`

Процедура производит чтение K значений исходных данных и присваивает эти значения переменным $A1, A2, \dots, AK$.

Задание

Составить программу запрашивающую текущую температуру в градусах Цельсия и выводящую на экран значение этой температуры, по шкале Фаренгейта.

Для перевода температуры из шкалы Цельсия в шкалу Фаренгейта нужно умножить исходное число на коэффициента $k = 9/5$ и прибавить 32.

Коэффициент задать с помощью именованной константы.

Задание

Составить программу, которая запрашивает ввод расхода топлива (в литрах на сто километров) и преобразует его в американский стиль (мили на галлон).

1 американский галлон = 3.78541178 литра

1 миля = 1.609344 километра

Результирующее значение расхода вывести:

1. в формате с фиксированной точкой и тремя знаками после запятой,
2. в научной форме.

Задание

Составить программу, которая запрашивает ввод длину провода (в дюймах) и преобразует его в привычный стиль (метры, сантиметры, миллиметры).

1 дюйм = 2.54 сантиметра

Результирующее значение длины провода вывести в формате:

метры, сантиметры, миллиметры

Например: 8м 10см 2мм

