Кафедра «Автоматизированные станочные системы»

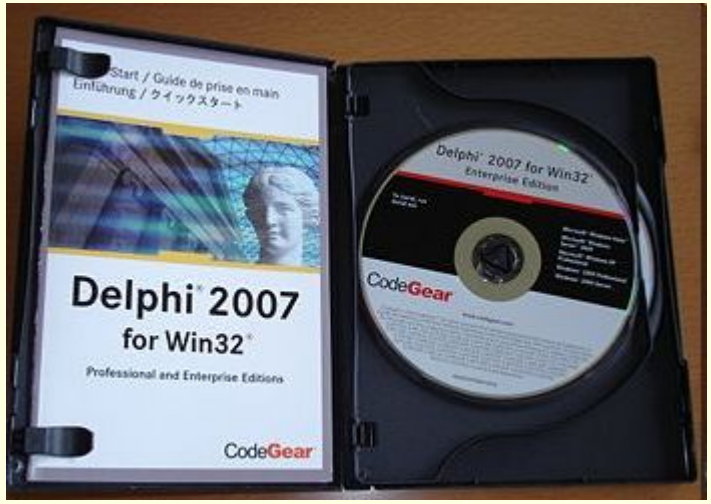
Dept. of Automated Manufacturing Systems

Основы языка Object Pascal/Delphi

Delphi – интегрированная среда разработки программ
Object Pascal – язык программирования

Разработчик: Borland – Imprise – Borland - CodeGear

Borland – в честь Фрэнка Бормана, командира экипажа Apollo 8, совершившего первый пилотируемый полет в дальний космос (1968)



Почему Delphi? Уже первая версия обеспечивала работу с сервером баз данных Oracle, что вызывало ассоциации с дельфийским оракулом

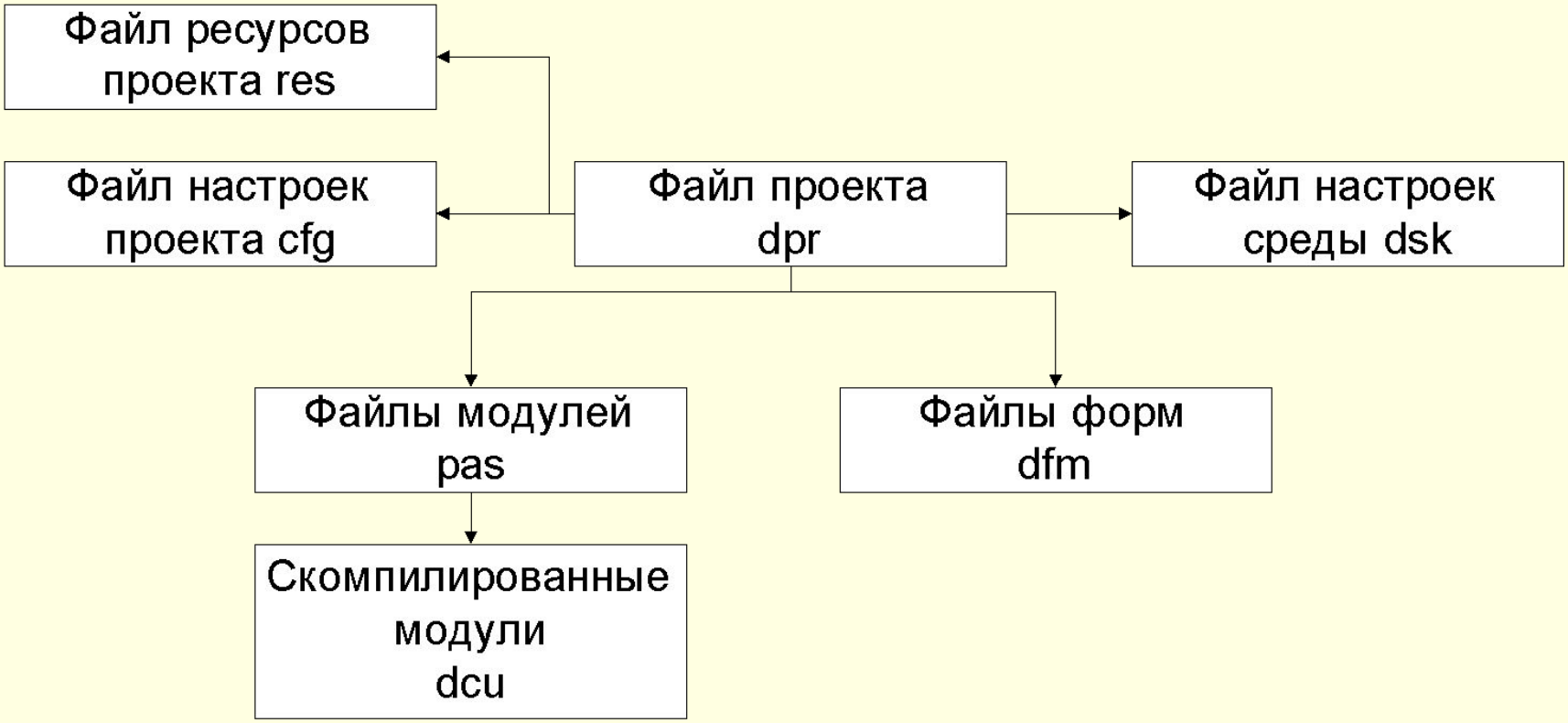
Что написано на Delphi

Собственно Borland Delphi, Borland C++ Builder, Borland JBuilder versions 1 & 2
Игры Astral Masters, Astral Tournament, Smugglers series, Soldat.
СУБД MySQL Tools
Skype, The Bat!
Web приложения: Macromedia HomeSite (HTML editor), TopStyle Pro (CSS editor), Macromedia Captivate (screencast), Ad-Aware (anti-spyware),
Total Commander
САПР ТП КОМПАС-Автопроект, ВЕРТИКАЛЬ

И тысячи других программ

Преимущества: ясность и понятность синтаксиса, встроенная поддержка баз данных, расширяемый набор компонентов, высокая скорость компиляции и работы приложений.

Структура проекта на Delphi



Каждый проект – в отдельном каталоге!

Пример файла проекта

```
program Project1;
uses
  Forms,
  Unit1 in 'Unit1.pas' {Form1};
{$R *.res}
begin
  Application.Initialize;
  Application.CreateForm(TForm1,
Form1);
  Application.Run;
end.
```

Файл проекта (.dpr)

- Слово **Program** со следующим за ним именем программы и точкой с запятой образуют заголовок программы.
- За заголовком следует раздел описаний, в котором программист или сам Delphi описывает используемые в программе идентификаторы.
- Собственно тело программы начинается со слова **begin** и ограничивается словом **end** с точкой. Тело состоит из операторов языка Object Pascal. В каждом операторе реализуется некоторое действие - изменение значения переменной, анализ результата вычисления, обращение к подпрограмме и т. п.

В теле простой программы - три исполняемых оператора:

```
Application.Initialize;
Application.CreateForm(TForm1, Form1);
Application.Run;
```

Как это работает

В объекте Application собраны данные и подпрограммы, необходимые для нормального функционирования Windows-программы в целом. Delphi автоматически создает объект-программу Application для каждого нового проекта.

Строка

Application.Initialize;

означает обращение к методу Initialize объекта Application. Прочитав эту строку, компилятор создаст код, который заставит процессор перейти к выполнению некоторого фрагмента программы, написанного для нас разработчиками Delphi. После выполнения этого фрагмента управление процессором перейдет к следующей строке программы, в которой вызывается метод CreateForm и т. д.

Модули

Модули - это программные единицы, предназначенные для размещений фрагментов программ. С помощью содержащегося в них программного кода реализуется вся функциональность программы.

Любой модуль имеет следующую структуру:

- **заголовок,**
- **секция интерфейсных объявлений,**
- **секция реализации,**
- **завершение.**

- ✔Заголовок открывается зарезервированным словом Unit, за которым следует имя модуля и точка с запятой.
- ✔Секция интерфейсных объявлений открывается зарезервированным словом Interface.
- ✔Секция реализации - словом implementation.
- ✔Окончанием является end с точкой.

Следующий фрагмент программы является синтаксически правильным вариантом модуля:

```
unit Unit1;
interface
// Секция интерфейсных объявлений
implementation
// Секция реализации
end.
```

- В секции интерфейсных объявлений описываются программные элементы (типы, классы, процедуры и функции), которые будут «видны» другим программным модулям
- В секции реализации раскрывается механизм работы этих элементов.

Разделение модуля на две секции обеспечивает удобный механизм обмена алгоритмами между отдельными частями одной программы. Он также реализует средство обмена программными разработками между отдельными программистами. Получив откомпилированный «посторонний» модуль, программист получает доступ только к его интерфейсной части, в которой содержатся объявления элементов. Детали реализации объявленных процедур, функций, классов скрыты в секции реализации и недоступны другим модулям.

Элементы программы

Элементы программы – это минимальные неделимые ее части, несущие в себе определенный смысл для компилятора.

К элементам относятся:

- зарезервированные слова;
- идентификаторы;
- типы;
- константы;
- переменные;
- подпрограммы;
- комментарии.

Алфавит

Алфавит языка Object Pascal включает буквы, цифры, шестнадцатеричные цифры, специальные символы, пробелы и зарезервированные слова.

Буквы - это буквы латинского алфавита от a до z и от A до Z , а также знак подчеркивания "_". В языке нет различия между заглавными и строчными буквами алфавита.

Цифры - арабские цифры от 0 до 9.

Специальные символы Object Pascal:

+ - * / = , ' . : ; < > ( ) { } " @ $ # []

К специальным символам относятся также следующие пары символов:

< > , < = , > = , : = , (* , *) , (. , .) , / /.

В комментариях и текстовых константах могут встречаться любые символы

Зарезервированные слова

Зарезервированные слова - это английские слова, указывающие компилятору на необходимость выполнения определенных действий. Зарезервированные слова не могут использоваться в программе ни для каких иных целей кроме тех, для которых они предназначены.

Примеры зарезервированных слов:

and	exports	mod	Shr
array	file	nil	String
as	finalization	not	then
asm	finally	object	var
begin	for	of	to
case	function	or	try

Идентификаторы

Идентификаторы в Object Pascal - это придумываемые программистом названия констант, переменных, меток, типов, объектов, классов, свойств, процедур, функций, модулей, программ и полей в записях. Идентификаторы могут иметь длину до 40 символов.

Идентификатор всегда начинается с буквы, за которой могут следовать буквы и цифры. Буквой считается также символ подчеркивания. Пробелы и специальные символы алфавита не могут входить в идентификатор.

Примеры правильных идентификаторов:

E_, A1, ALPHA

Примеры неправильных идентификаторов:

1Program // начинается цифрой

block#l // содержит специальный символ

В названии идентификатора для Delphi никакого смысла не содержится. Без разницы, назвать переменную A или TheBestVariableInMyLife

Константы

Константы определяют области памяти, которые не могут изменять своего значения в ходе работы программы. Как и любые другие элементы программы, константы имеют свои собственные имена.

Константы задаются в разделе описаний.

Объявлению имен констант должно предшествовать зарезервированное слово const. Например, мы можем определить константы

```
const
        Kbyte = 1024;
        Mbyte = Kbyte*Kbyte;
        Gbyte = 1024*Mbyte;
```

Константа PI=3.14159… встроена в язык

- Целые числа записываются со знаком или без него по обычным правилам и могут иметь значение в диапазоне от
- -2^{63} до $+2^{63}$
- Вещественные числа записываются со знаком или без него с использованием десятичной точки и/или экспоненциальной части.
- Шестнадцатеричное число состоит из шестнадцатеричных цифр, которым предшествует знак доллара $. Диапазон шестнадцатеричных чисел - от $ffffffffffffffff до $7FFFFFFFFFFFFFFF.
- Логическая константа - это либо слово false (ложь), либо слово true (истина).
- Символьная константа - это любой символ, заключенный в апострофы.
- Строковая константа - любая последовательность символов, заключенная в апострофы.

Выражения

Основными элементами, из которых конструируется исполняемая часть программы, являются константы, переменные и обращения к функциям. Каждый из этих элементов характеризуется своим значением и принадлежит к какому-либо типу данных. С помощью знаков операций и скобок из них можно составлять выражения, которые фактически представляют собой правила получения новых значений.

Примеры выражений:

```
y
21 - (a + b) * c
sin(t)
a > 2
not Flag and (a = b)
NIL
[1, 3..7]
```