



# БИБЛИОТЕКИ ЯЗЫКА ВЫСОКОГО УРОВНЯ PASCAL

*Сколько ни хорошо унаследовать библиотеку,  
еще лучше собрать ее самому.*

*О. Биррель*

# Содержание

2

- Понятие библиотеки
- Структура библиотеки
- Использование библиотек
- Разработка собственных библиотек

# Библиотека

3

- *Библиотека* — вспомогательная *неисполняемая* программная единица, содержащая определения подпрограмм и данных, используемая для коллективной разработки больших программных систем (размером от 1 тыс. строк).
- Библиотека может быть *подключена* к программам и другим библиотекам, в этом случае все ее программные объекты являются видимыми в этих программах и библиотеках.

# Пример: использование библиотеки

4

{ myprog2.pas, 26-ноя-12

**Иванов И.И.**

Пример программы,  
использующей библиотеку }

Program MyProg2;

**uses UCalc;**

procedure InpData (var A, B, C: Integer);

{ Осуществляет ввод исходных данных с клавиатуры }

begin

...

end;

procedure OutData (Result: Real);

{ Осуществляет вывод результатов вычислений на экран }

begin

...

end;

var

A, B, C: Integer; R: Real;

begin

InpData (A, B, C);

**Calculation (A, B, C, R);**

**OutData (Result);**

end.

{ ucalc.pas, 26-ноя-12

**Петров П.П.**

Пример библиотеки. }

unit Ucalc;

interface

**procedure Calculation**

**(A, B, C: Integer; var Result: Real);**

**{ Выполняет некоторые вычисления с  
исходными данными }**

implementation

**procedure Calculation**

**(A, B, C: Integer; var Result: Real);**

**{ Выполняет некоторые вычисления с  
исходными данными }**

**begin**

**...**

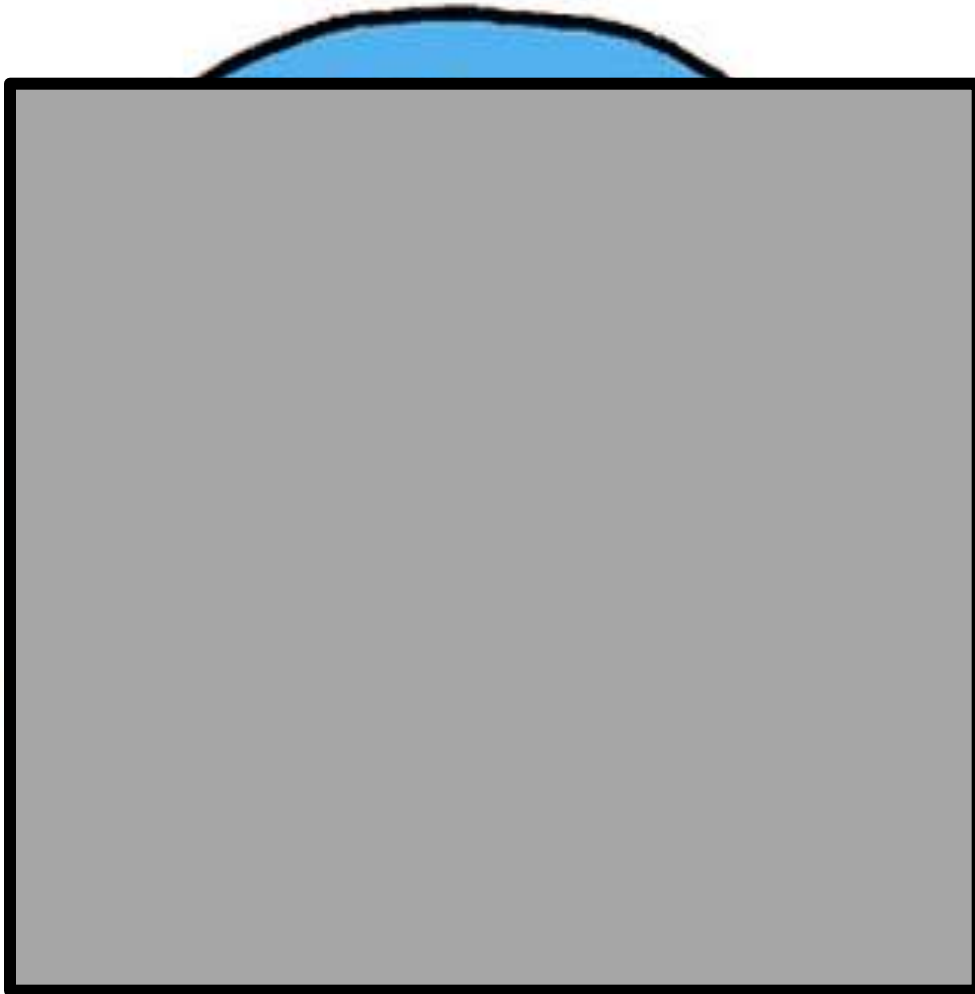
**end;**

**...**

**end.**

# Структура библиотеки

5



- *Интерфейс* – экспортируемые объекты (видимая часть).
- *Реализация* – скрытая часть.

# Библиотеки в языке Pascal/Delphi

6

```
unit <Имя библиотеки>;                                { Заголовок }

interface                                             { Интерфейс }
  [ uses <Список используемых библиотек>; ]
  <Описания видимых программных объектов>
  <Заголовки экспортируемых подпрограмм>

implementation                                       { Реализация }
  [ uses <Список используемых библиотек>; ]
  <Описания скрытых программных объектов>
  <Реализации экспортируемых подпрограмм>

  [ begin                                             { Инициализация }
    <Операторы инициализации> ]
end.
```

# Секция `interface`

7

- *Интерфейсная секция* содержит объявления программных объектов, *видимых* в любой программе/библиотеке, которая подключает данную библиотеку.
- Интерфейсная секция может содержать подключения других библиотек, а также объявления констант, типов, переменных, подпрограмм.
- При этом для подпрограмм описываются *только заголовки*. Полные описания подпрограмм помещаются в секцию реализации.

# Секция `implementation`

8

- Секция реализации содержит полные описания подпрограмм из интерфейсной секции.
- Реализация модуля может содержать также объявления внутренних программных объектов, *не видимых* в других программах и библиотеках.
- Все программные объекты, объявленные в интерфейсной секции, *видимы* в секции реализации.



# Секция инициализации

- Библиотека может содержать *секцию инициализации*, которая обычно содержит операторы инициализации данных этой библиотеки.
- При запуске программы, использующей библиотеки, перед выполнением ее тела выполняются операторы из секций инициализации всех используемых ею библиотек.

# Создание библиотек

10

- Библиотеки компилируются *независимо* друг от друга и от программы и сохраняются как машинные коды в файлах с расширением **.dcu**.
- Интерфейсная часть также сохраняется в файле **.dcu** в виде *таблицы символов*, используемой для связывания данной библиотеки с другими библиотеками.
- Во время компоновки к машинному коду программы добавляются машинные коды всех модулей, прямо или косвенно используемых в данной программе.

# Пример: использование библиотеки

11

{ myprog2.pas, 26-ноя-12

**Иванов И.И.**

Пример программы,  
использующей библиотеку }

Program MyProg2;

**uses UCalc;**

procedure InpData (var A, B, C: Integer);

{ Осуществляет ввод исходных данных с клавиатуры }

begin

...

end;

procedure OutData (Result: Real);

{ Осуществляет вывод результатов вычислений на экран }

begin

...

end;

var

A, B, C: Integer; R: Real;

begin

InpData (A, B, C);

**Calculation (A, B, C, R);**

**OutData (Result);**

end.

{ ucalc.pas, 26-ноя-12

**Петров П.П.**

Пример библиотеки. }

unit Ucalc;

interface

**procedure Calculation**

**(A, B, C: Integer; var Result: Real);**

**{ Выполняет некоторые вычисления с  
исходными данными }**

implementation

**procedure Calculation**

**(A, B, C: Integer; var Result: Real);**

**{ Выполняет некоторые вычисления с  
исходными данными }**

**begin**

**...**

**end;**

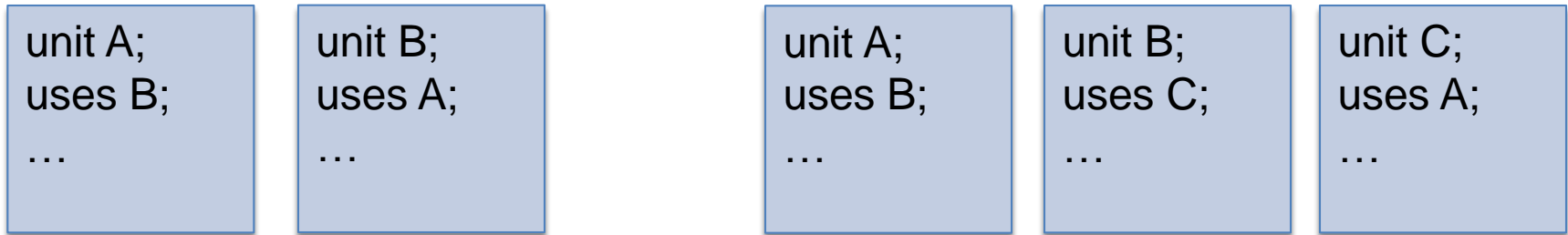
**...**

**end.**

# Предотвращение конфликтов

12

- Недопустимо использование библиотек с *циклическими ссылками*. Например:



- При совпадении идентификаторов в различных библиотеках, используемых в одной программе, необходимо использовать *квалификацию идентификаторов*. Например:  
`Res := UnitA.Limit + UnitB.Limit(L, 5);`

# Преимущества библиотек

13

- *Независимая компиляция и кодирование*
  - Коллективная разработка подсистем большой программной системы.
- *Инкапсуляция (скрытие) деталей реализации.*
  - Возможно использование библиотеки как .dcu файла и текстового файла с интерфейсной секцией.
    - `Unit MyProg; ≠ {$i myprog.pas}`
- *Повторное использование*
  - Разработал библиотеку один раз, uses библиотеку много раз.
  - Изменение реализации библиотеки требует лишь перекомпиляции программы.

# Стандартные библиотеки

- Библиотеки **System**, **WinCrt**, **Printer** и др. составляют *библиотеку времени выполнения*.
- Библиотека **System** подключается к программе автоматически, остальные нужно указывать в предложении **uses**.
- Назначение стандартных библиотек
  - **System** – подпрограммы стандартного Pascal и встроенные подпрограммы Object Pascal, не вошедшие в др. библиотеки.
  - **WinCrt** – подпрограммы управления текстовым режимом работы экрана, работы с расширенными кодами клавиатуры, цветовой палитрой, окнами и звуками.
  - **Printer** – обеспечивает вывод текстов на принтер.

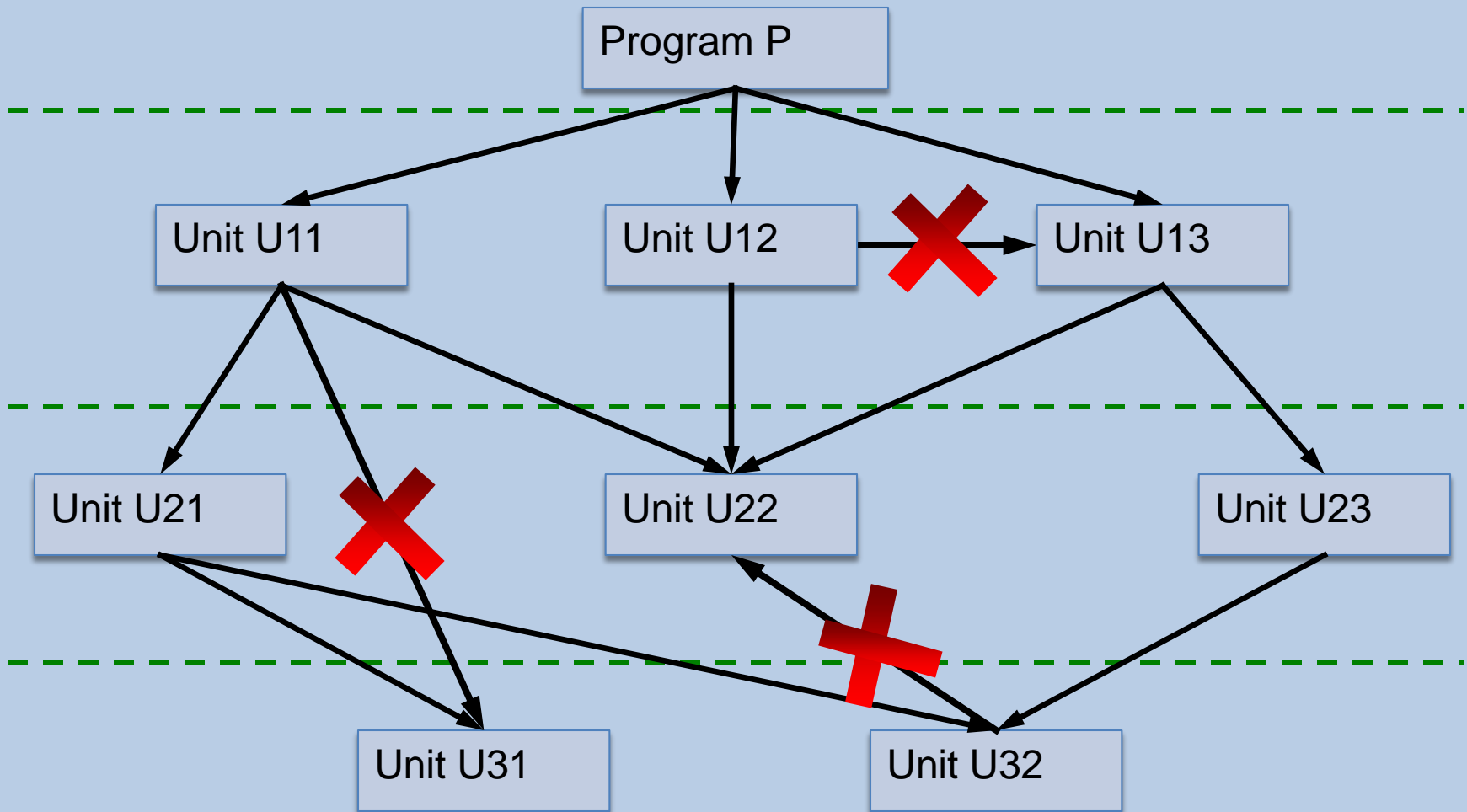
# Разработка собственных библиотек

15

- Библиотека оформляется в виде отдельного текстового **.pas** файла. Имя файла должно совпадать с именем библиотеки. В одном файле может быть только одна библиотека.
- Способы компиляции:
  - **Compile** – создание **.dcb** библиотеки с текущими версиями **.dcb** включаемых библиотек
  - **Make** – создание **.dcb** библиотеки с перестроением **.dcb** измененных включаемых библиотек
  - **Build** – создание **.dcb** библиотеки с перестроением **.dcb** всех включаемых библиотек .
- Большую программную систему необходимо строить как *иерархию модулей*.

# Иерархия библиотек

16





# Недостатки модулей unit

17

- Интерфейс и реализация могли бы быть независимыми программными объектами.
- Как тестировать подпрограммы, скрытые в секции реализации?

# Заключение

- Библиотека – неисполняемая, отдельно хранимая и независимо компилируемая программная единица, содержащая определения подпрограмм и других программных объектов.
- Библиотека состоит из следующих секций: `interface` (описание), `implementation` (реализация) и секция инициализации.
- При включении библиотеки в предложение `uses` программы (другой библиотеки) все программные объекты, объявленные в интерфейсе, становятся доступными в данной программе (данной библиотеке).
- Большую программную систему необходимо строить как иерархию библиотек.