

Оглавление

1 Цель лабораторной работы.....	3
2 Запуск визуальной среды разработки приложений Visual Prolog.....	3
3 Создание проекта.....	4
4 Запуск и тестирование программы.....	7
5 Комментарии к свойствам утилиты Test Goal	8
5.1 Замечание.....	8
6 Обработка ошибок.....	9
7 Команды построения	9
7.1 Команда Project / Compile Module.....	9
7.2 Команда Project / Build.....	9
7.3 Команда Project / Rebuild All.....	10
7.4 Команда Project I Stop Building.....	10
7.5 Команда Project I Run.....	10
7.6 Команда Project I Link Only.....	10
7.7 Команда Project / Test Goal.....	10
7.8 Команда Resource I Build Resource Only.....	11
7.8.1 Пример	11
8 Команды отладки.....	11
8.1 Команда Project Debug.....	11
9 Задания к лабораторной работе.....	12

1 Цель лабораторной работы

Изучить среду визуальной разработки Visual Prolog. Создать проект и запустить его на выполнение.

2 Запуск визуальной среды разработки приложений Visual Prolog.

Для того, чтобы запустить Visual Prolog, необходимо выполнить следующие действия: Пуск | Программы | Visual Prolog 5.2 | Visual Prolog 32.

При этом открывается основное окно, которое называется окном **Task** (рис. 1).

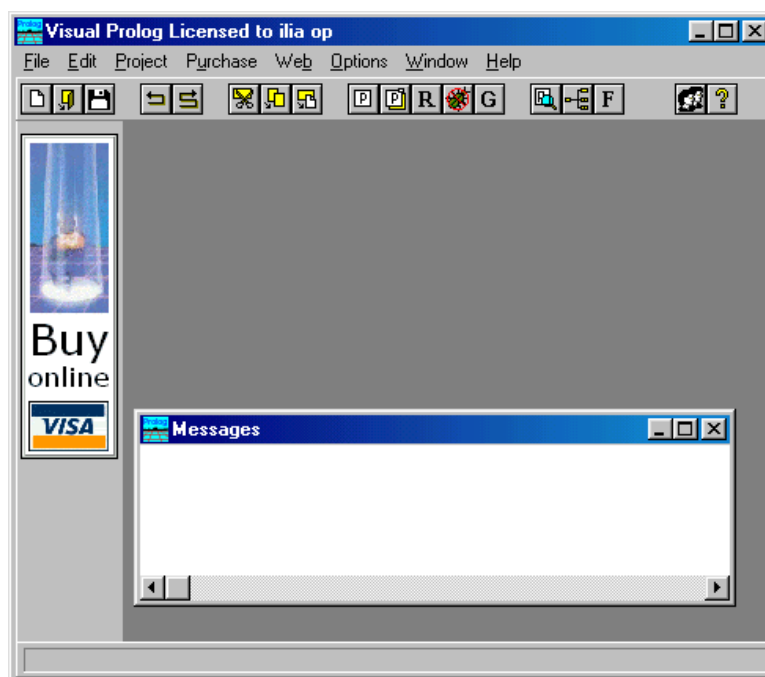


Рис. 1 Окно Task

Обычно в нем доступны меню **File**, **Edit**, **Project**, **Options**, **Help** и **Window**, но при активизации некоторых других окон в меню могут появиться дополнительные пункты.











Часто используемые команды меню могут быть выполнены и при помощи кнопок на панели инструментов (рис. 2).



Рис. 2. Панель инструментов

Каждая из пиктограмм (табл. 1) на панели инструментов выполняет ту же функцию, что и соответствующая команда меню.

Таблица 1. Команды меню и кнопки панели инструментов

Кнопка панели инструментов	Команда меню	Кнопка панели инструментов	Команда меню
	File New		Edit Copy
	File Open		Edit Paste
	File Save		Project (Compile file)
	Edit Undo		Project Build
	Edit Redo		Project Run
	Edit Cut		Project Debug
	Project Test Goal		Options Temporary Font
	Project Browse		Help Local Help
	Project Tree		

В нижней части окна **Task**, как уже говорилось, расположена строка подсказки. Она разделена на две части (рис.3).



Рис. 3. Строка подсказки

Левое поле используется для отображения контекстно-зависимой информации, например, подсказок для командных кнопок на панели инструментов или информации о текущем элементе управления в редакторе диалоговых окон и т. д.

Крайнее правое поле используется построителем программ (make facility) для отображения состояний генерации/компиляции/компоновки текущего ресурса.

3 Создание проекта.

Для создания проекта требуется определить некоторые (не предопределенные) опции компилятора Visual Prolog. Для этого выполните следующие действия:

1. Запустите среду визуальной разработки Visual Prolog.

При первом запуске VDE () проект не будет загружен, и вы увидите окно, показанное на рис. 4. Также вас проинформируют, что по умолчанию создан инициализационный файл для Visual Prolog VDE.

2. Создайте новый проект.

Выберите команду **Project | New Project**, активизируется диалоговое окно **Application Expert**.

3. Определите базовый каталог и имя проекта.

Имя в поле **Project Name** следует определить как "Test". Щелкните мышью внутри поля **Name of .VPR File**

Также установите флажок **Multiprogrammer Mode** и щелкните мышью внутри поля **Name of .PRJ File**. Вы увидите, что появится имя файла проекта **Test.prj** (рис. 4).

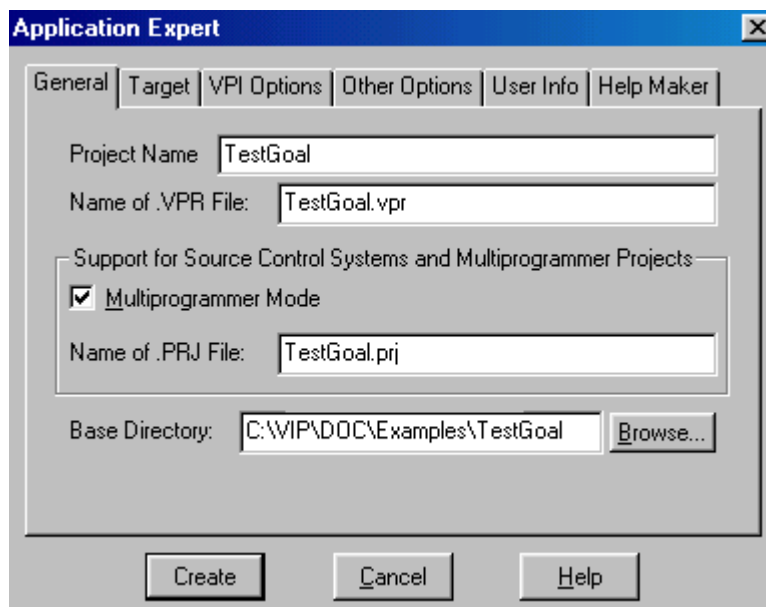


Рис. 4. Общие установки диалогового окна **Application Expert**

Определите цель проекта.

На вкладке **Target** мы рекомендуем выбрать параметры, отмеченные на рис. 5.

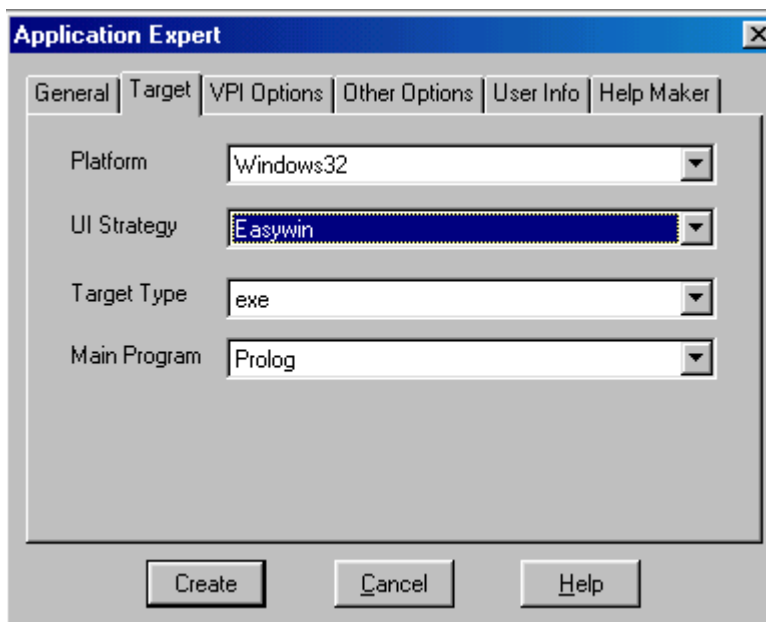


Рис. 5. Установки на вкладке **Target** диалогового окна **Application Expert**

Теперь нажмите кнопку **Create** для того, чтобы создать файлы проекта по умолчанию.

4. Установите требуемые опции компилятора для созданного проекта.

Для активизации диалогового окна **Compiler Options** выберите команду **Options | Project | Compiler Options**. Откройте вкладку **Warnings**. Выполните следующие действия:

- установите переключатель **Nondeterm**. Это нужно для того, чтобы компилятор Visual Prolog принимал по умолчанию, что все определенные пользователем предикаты — недетерминированные (могут породить более одного решения);
- снимите флажки **Not Quoted Symbols**, **Strong Type Conversion Check** и **Check Type of Predicates**. Это будет подавлять некоторые возможные предупреждения компилятора;
- нажмите кнопку **ОК**, чтобы сохранить установки опций компилятора.

В результате этих действий диалоговое окно **Compiler Options** будет выглядеть, как показано на рис. 6.

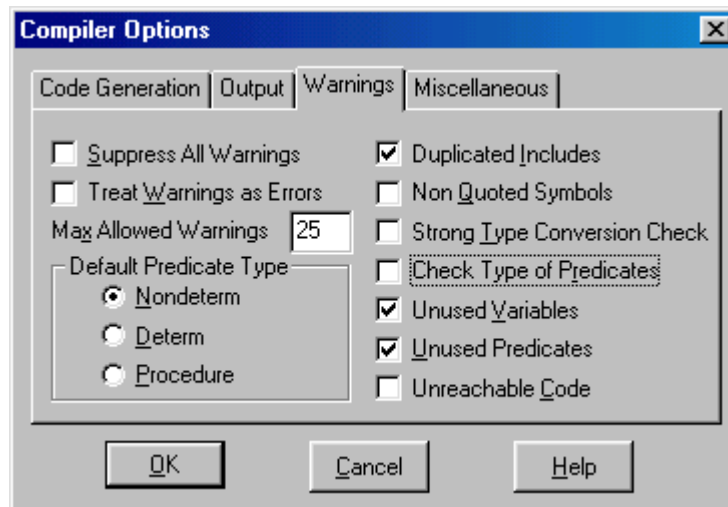


Рис. 6. Установки опций компилятора

4 Запуск и тестирование программы

Для проверки того, что ваша система настроена должным образом, следует выполнить следующие действия:

1. в окне проекта открыть файл **test.pro** (рис. 7)

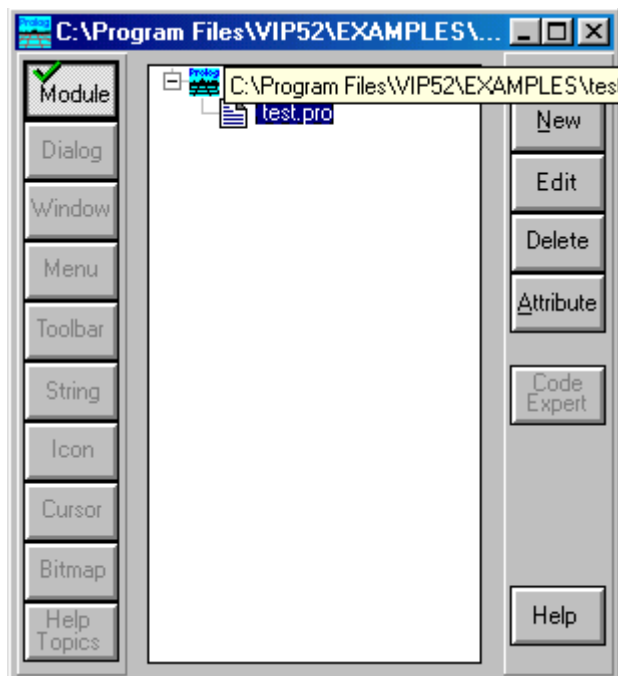


Рис. 7 Окно проекта.

2. В разделе GOAL наберите с клавиатуры write ("Hello world"), nl .

3. Нажать на панели инструментов кнопку **G** (либо комбинацию клавиш <Ctrl>+<G>, либо активировать команду **Project | Test Goal**). В терминологии языка Пролог это называется *GOAL*, и этого достаточно для программы, чтобы она могла быть выполнена. Если ваша система установлена правильно, то экран монитора будет выглядеть, как показано на рисунке 8.

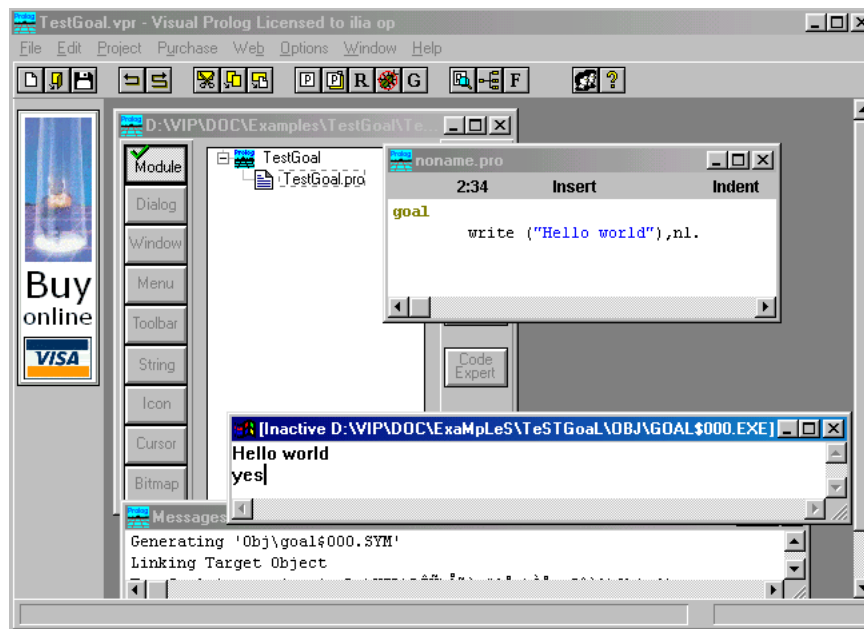


Рис. 8. Тестовая программа “Hello world”.

Результат выполнения программы будет расположен сверху в отдельном окне, которое необходимо закрыть перед тем, как тестировать другую GOAL.

5 Комментарии к свойствам утилиты Test Goal

Утилита среды визуальной разработки интерпретирует GOAL как специальную программу, которая компилируется, компоуется, генерируется в исполняемый файл и Test Goal запускает его на выполнение. Эта утилита внутренне расширяет заданный код GOAL, чтобы сгенерированная программа находила все возможные решения и показывала значения всех используемых переменных. Утилита Test Goal компилирует этот код с использованием опций компилятора, заданных для открытого проекта (рекомендуемые опции компилятора для TestGoal-проекта мы определили ранее).

5.1 Замечание

Утилита Test Goal компилирует только тот код, который определен в активном окне редактора (код в других открытых редакторах или модулях проектов, если они есть, игнорируется).

При компоновке исполняемого файла TestGoal использует стратегию EASYWIN. Нельзя определить какие-либо опции компоновки для TestGoal, т. к. игнорируются любые установки **Make Options**, заданные для открытого проекта. Поэтому TestGoal не может использовать никакие глобальные предикаты, определенные в других модулях. Утилита имеет ограничение на количество переменных, которые могут быть использованы в GOAL. На данный момент их 12 для 32-разрядной среды визуальной разработки, но это число может быть изменено без дополнительных уведомлений.

6 Обработка ошибок

Если вы допустили ошибки в программе и пытаетесь скомпилировать ее, то среда визуальной разработки отобразит окно Errors (Warnings), которое будет содержать список обнаруженных ошибок.

Дважды щелкнув на одной из этих ошибок, вы попадете на место ошибки в исходном тексте. Можно воспользоваться клавишей <F1> для вывода на экран интерактивной справочной системы Visual Prolog. Когда окно помощи откроется, щелкните по кнопке Search, наберите номер ошибки, и на экране появится соответствующее окно помощи с более полной информацией о ней.

Подробному рассмотрению основных функций интегрированной среды визуальной разработки VDE Visual Prolog посвящена следующая глава.

7 Команды построения

7.1 Команда *Project / Compile Module*

Эта команда (ей соответствует комбинация клавиш <Ctrl>+<F9>) делает попытку компилировать модуль, содержащий редактируемый в данный момент файл. Выполнение команды зависит от следующих свойств файла:

- если файл имеет расширение pro и является модулем текущего проекта, то VDE пытается компилировать этот файл;
- если файл не является модулем текущего проекта и его расширение — pro, pre, inc, con или dom, то VDE пытается найти модуль проекта, который включает этот файл, и откомпилировать первый найденный модуль;
- во всех остальных случаях VDE пытается компилировать модуль, выбранный в окне проекта.

VDE не может компилировать файл, который не является частью открытого проекта. Вместо этого файла VDE будет компилировать модуль, выбранный в окне проекта.

Если в VDE не открыт ни один проект, то никакие файлы компилироваться не будут. Команда меню Project | Compile Module заблокирована; комбинация клавиш <Ctrl>+<F9> не работает. Единственно возможное действие — это запустить утилиту Test Goal.

7.2 Команда *Project / Build*

Если со времени последнего построения проекта были изменены какие-либо ресурсы, то эксперты кода могут обновить некоторые секции в исходных файлах перед построением.

Эта команда (ей соответствует комбинация клавиш <Alt>+<F9>) строит проект, проверяя метки времени всех исходных файлов в проекте, поэтому если исходные файлы (или файлы, которые в них включены) являются более новыми, чем зависимые **OBJ**-файлы, то соответствующие модули проекта будут перекомпилированы.

Команда **Build** также строит файлы ресурсов и файл интерактивной справки (если необходимо).

Затем проект компоуется для генерации целевого модуля (исполняемая программа или DLL).

7.3 Команда *Project / Rebuild All*

Эта команда (ей соответствует комбинация клавиш <Ctrl>+<Alt>+<F9>) выполняет то же действие, что и **Project | Build**, причем все файлы будут повторно сгенерированы или откомпилированы и скомпонованы независимо от их меток времени.

7.4 Команда *Project | Stop Building*

Эта команда (ей соответствует комбинация клавиш <Alt>+<F10>) используется для остановки компиляции/компоновки.

7.5 Команда *Project | Run*

Если необходимо, то эта команда (ей соответствует клавиша <F9>) выполнит действие **Project | Build** и затем запустит сгенерированный исполняемый файл.

7.6 Команда *Project | Link Only*

Эта команда (ей соответствует комбинация клавиш <Shift>+<F9>) используется для выполнения компоновки. В этом случае построитель программ вызывает компоновщика и не проверяет, нужно ли повторно компилировать какие-либо модули проекта (или даже впервые компилировать).

7.7 Команда *Project / Test Goal*

Эта команда (ей соответствует комбинация клавиш <Ctrl>+<G>) используется для тестирования простых целей (Goals). Программа компилируется и компоуется в специальном режиме, и затем запускается соответствующий исполняемый файл. Утилита Test Goal ищет все решения для определенной в программе цели. Для каждого решения Test Goal отображает значения всех переменных из секции GOAL и число решений. Эта особенность — удобный способ проверить локальные предикаты в модуле.

Например, следующая цель:

GOAL X = 2; X = 1, Y = X + 1

приведет к такому результату (рис. 9):



Рис. 9 Вывод режима тестирования цели.

7.8 Команда *Resource I Build Resource Only*

Когда окно проекта активизировано, в меню Project появляется команда Resource. При выборе этого пункта (или нажатии комбинации клавиш <Alt>+<F8>) генерируются выбранные файлы с расширениями rc и res и необходимые файлы констант.

7.8.1 Пример

Загрузите программу ch02e01.pro в среду визуальной разработки Visual Prolog и запустите ее утилитой Test Goal (см. разд. "Тестирование примеров данного руководства" гл. 8).

```
predicates
likes(symbol,symbol)
clauses
likes(ellen,tennis).
likes(john,football).
likes(tom,baseball).
likes(eric,swimming).
likes(mark,tennis).
likes(bill,Activity):-likes (tom, Activity) .
goal
likes(bill, baseball).
```

Утилита Test Goal ответит в окне приложения: yes (да)

Попробуйте также следующий запрос в GOAL-разделе: likes(bill, tennis) .

Утилита Test Goal ответит: no (нет).

8 Команды отладки

8.1 Команда *Project | Debug*

Запускает процесс отладки. Отладчик также можно запустить сочетанием клавиш CTRL+SHIFT+F9.

При помощи диалога View можно открывать дополнительные информационные окна, которые отображают различные состояния среды и переменных в режиме отладки:

View → Call Stack (Открывает информационное окно стека вызова)

View → Local Variables (Открывает информационное окно локальных переменных)

Для выполнения шагов отладки используются следующие команды:

Run → Trace Intro [F7]

Run → Step Over [F8]

Run → Run to Cursor [F4]

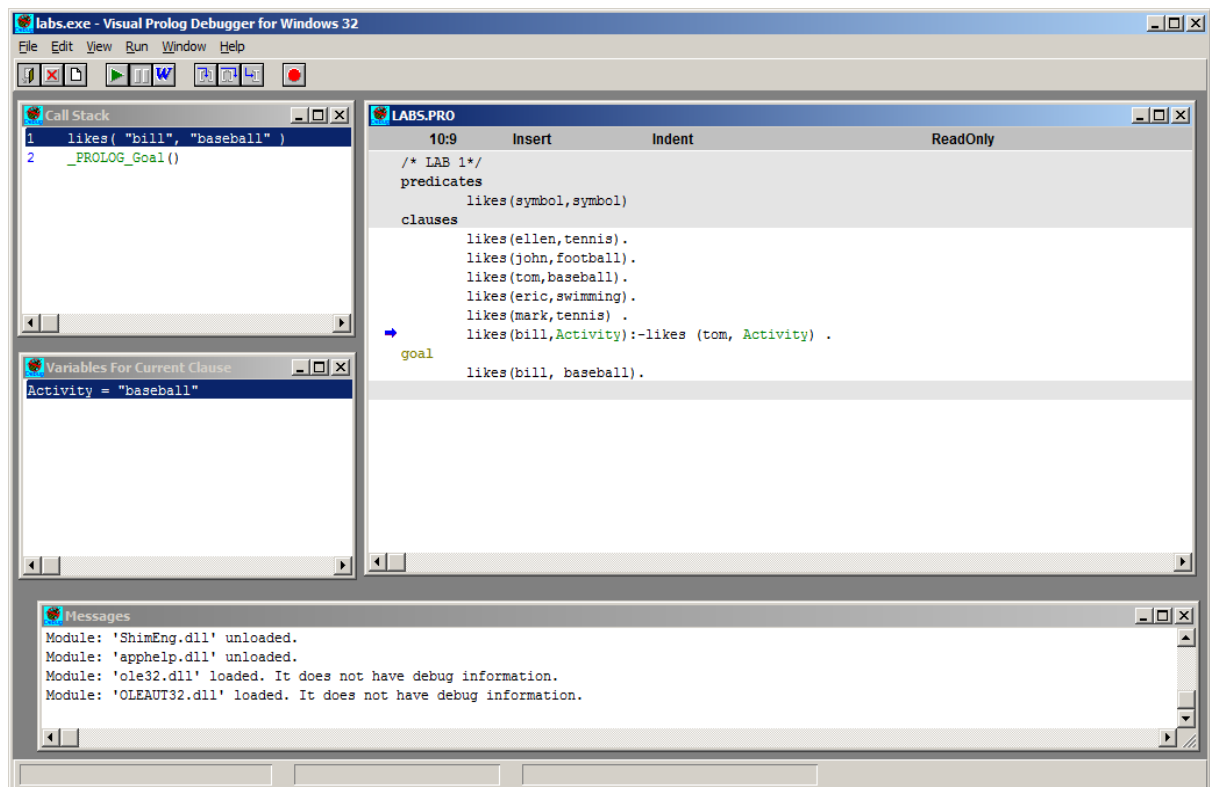


Рис. 10. Окно отладчика с открытым листингом ch02e01.pro

9 Задания к лабораторной работе

Запустить листинг в среде Visual Prolog на выполнение. Запустить листинг в среде Visual Prolog в режиме отладки.

```

predicates
    likes(symbol,symbol)
clauses
    likes(ellen,tennis).
    likes(john,football).
    likes(tom,baseball).
    likes(eric,swimming).
    likes(mark,tennis) .
    likes(bill,Activity):-likes (tom, Activity) .
goal
    likes(bill, baseball).

```