Добро пожаловать в мир программирования на C++ для Windows. Сегодня начинается ваше путешествие в этот захватывающий мир. Большая часть сегодняшнего урока посвящена знакомству с Интегрированной средой разработки Borland C++ (IDE). Вы познакомитесь со следующими темами:

- Основы и история языка программирования C++
- Работа в интегрированной среде разработки Borland C++ (IDE)
- Что такое EasyWin-приложения
- Проекты и узлы
- Составление и запуск вашей первой программы на C++

Основы программирования на C++

Если у вас в прошлом нет опыта программирования, то это не страшно — этой книги будет достаточно, чтобы научить вас языку C++. Однако, если вы программировали прежде на каком-либо языке, то вам будет проще. Как и другие языки программирования, C++ состоит из операторов объявлений и исполняемых операторов,- которые определяют точную последовательность действий компьютера при выполнении программы.

C++ был разработан Бьерном Страуструпом в Bell Laboratories на основе популярного языка C как его возможный преемник. Главным нововведением C++ были объектно-ориентированные расширения языка.

Объектно-ориентированный язык программирования оперирует понятиями свойств объектов и действий над объектами. Такой подход к программированию лучше соответствует человеческому восприятию реального мира. И хотя это возможно — писать программы в объектно-ориентированном стиле на большинстве языков программирования — только объектно-ориентированный язык позволяет воплотить идеи объектно-ориентированного программирования (ООП) во всей полноте.

Кроме ООП, C++ предлагает вам ряд расширений C, которые не являются объектно-ориентированными. Кстати, изучение C++ дает вам возможность освоить и язык C. Однако, в отличие от C, который имеет стандарт, процесс стандартизации C++ комитетом ANSI/ISO еще продолжается.

До того момента, когда вы начнете программировать на C++, вам необходимо познакомиться со вспомогательными библиотеками, которые выполняют различные задачи типа ввода-вывода, работы со строками символов, математических операций, файлового ввода-вывода и т. д. В языках типа BASIC поддержка таких операций встроена в язык и не является проблемой. В результате многие программы являются независимыми от любых других программ и программных компонентов. В отличие от этих языков, программирование на C++ зависит от применяемых библиотек. Это дает вам преимущество, поскольку вы имеете возможность выбора различных библиотек, включая те, которые вы разрабатываете сами. Компиляторы C++, включая Borland C++, кроме файлов программ используют и файлы проекта. IDE Borland C++ применяет файлы проекта для управления процессом создания и модификации программ.

Файлы проекта содержат списки всех файлов, необходимых для создания одной или большего количества прикладных программ, и связи, определяющие зависимости между файлами и обеспечивающие правильную последовательность компиляции файлов. Файлы программ содержат исходные тексты, необходимые для создания прикладной программы.

Запуск Borland C++ IDE

Borland C++ IDE — интегрированная среда разработки C++, включающая в себя транслятор, компоновщик, отладчик и другие инструментальные средства, которые используются для создания, отладки и сопровождения программ C++. Вы можете загрузить IDE, выбрав пиктограмму Borland C++ в меню Start или дважды щелкнув кнопкой мыши на программе BCW.EXE в каталоге \BC5\BIN.

Краткий обзор Borland C++ IDE

IDE Borland C++ — прикладная программа класса MDI (Multiple Document Interface), имеющая следующие основные компоненты:

- Основное окно с меню и кнопками размера. Вы можете изменять размеры, перемещать, сворачивать и увеличивать до размера экрана главное окно IDE Borland C++.
- Панель инструментов, которая содержит специальные кнопки-пиктограммы, позволяет быстро выполнять определенные команды. IDE дает возможность вам установить нужный набор кнопок на панели инструментов. Кроме того, эти кнопки чувствительны к контексту. Их количество и тип может изменяться в зависимости от выполняющейся задачи или типа активного окна. IDE поддерживает подсказки, указывающие вам, что данная кнопка делает, когда вы помещаете курсор мыши на этой кнопке (текст появляется в строке состояния внизу окна и в небольшом блоке, который мгновенно открывается поблизости от самой кнопки).
- Область клиента, которая содержит различные окна, такие, как окно редактирования, окно сообщений, окно наблюдения, окно проекта и другие окна.
- Строка состояния размещается в нижней части окна IDE. Эта строка отображает краткую интерактивную справку, когда вы перемещаете курсор мыши через кнопки в панели инструментов, предлагает краткое объяснение различных пунктов меню, отображает координаты курсора, показывает состояние режима вставки/замены символов и даже текущее время.

На рисунке 1 показан типичный вид окна IDE Borland C++.

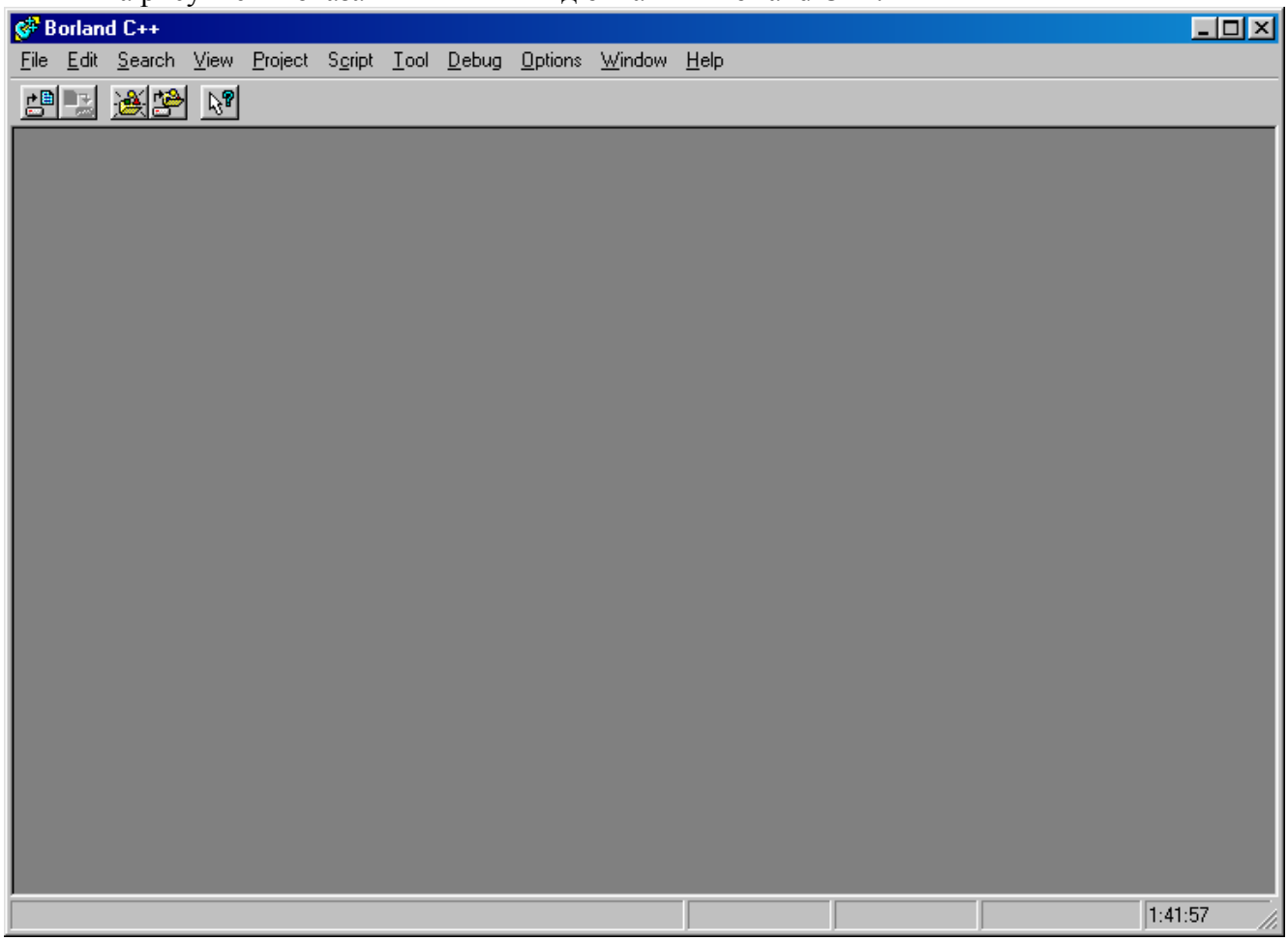

Рисунок 1. IDE Borland C++

Поскольку IDE в основном предназначена для разработчиков программного обеспечения, многие из ее возможностей покажутся вам слишком “утонченными”, если вы программист-новичок. Однако для начала вам достаточно будет познакомиться только с командами меню IDE и связанными с ними терминами. Когда вы станете более опытным, эти

команды и термины станут частью ваших знаний как программиста Borland C++.

Меню File

Меню File (Файл) содержит команды управления файлами, печати текстов и выхода из IDE. Все они перечислены в таблице 1. Меню File также содержит список последних открывавшихся файлов.

Таблица 1. Список команд меню File

Команда	Клавиши быстрого выбора команд	Выполняемые действия
New		Вызывает подменю, которое предлагает вам создать новое окно редактирования, файл проекта, окно AppExpert или ресурс.
Open...		Загружает существующий файл в новое окно редактирования или загружает проект.
Close		Закрывает текущее окно редактирования.
Save	Ctrl + K Ctrl + S	Сохраняет содержимое активного окна редактирования.
Save as...		Сохраняет содержимое активного окна редактирования в файле с новым именем.
Save all		Сохраняет все открытые окна, содержащие тексты программ, в соответствующих файлах.
Print...		Печатает содержимое окна с исходным текстом.
Print setup...		Устанавливает параметры печати.
Send...		Отсылает по электронной почте текущий редактируемый файл.
Exit		Выход из IDE.

Команда New

Команда New (Новый) предлагает вам на выбор создание нового файла одного из следующих типов:

- Text Edit открывает новое окно редактирования (называемое также окном исходного текста) и присваивает ему имя файла по умолчанию. Задаваемое по умолчанию имя файла первого нового окна, которое вы открываете — NONAME00.CPP. Аналогично, заданное по умолчанию имя файла второго по счету нового окна — NONAME01.CPP, и т. д. Новое открытое окно изначально пустое.
- Project ... создает новый проект. Эта команда сначала останавливается в диалоговом окне TargetExpert (показанном на рисунке 2), который дает вам возможность установить различные опции для нового проекта, такие как модель памяти и версия Windows.
- AppExpert ... вызывает утилиту AppExpert, которая является ценным и сложным инструментом для быстрой разработки программы. В день 20-й будет обсуждаться использование AppExpert.
- Resource Project... запускает интегрированный редактор ресурсов, в котором вы можете создать диалоги, битовые матрицы и пиктограммы. Редактор ресурсов обсуждается в дополнительном Дне 4.

Команда Open...

Команда Open... (Открыть...) позволяет вам загрузить содержимое существующего файла в новое окно редактирования. В IDE вы можете открыть несколько файлов. Команда вызывает диалоговое окно Open a File, показанное на рисунке 3. Диалоговое окно содержит несколько списков и комбинированных списков, которые дают вам возможность найти и выбрать нужный файл. Эти средства управления помогут вам выбрать дисковод, каталог и задать шаблон имени файла, что упростит для вас поиск файла.

Команда Close

Команда Close (Закрыть) закрывает текущий файл. Команда дает тот же результат, что и нажатие на маленькую кнопку x в правом верхнем углу окна в Windows 95 или двойной щелчок мыши на левом верхнем углу окна.

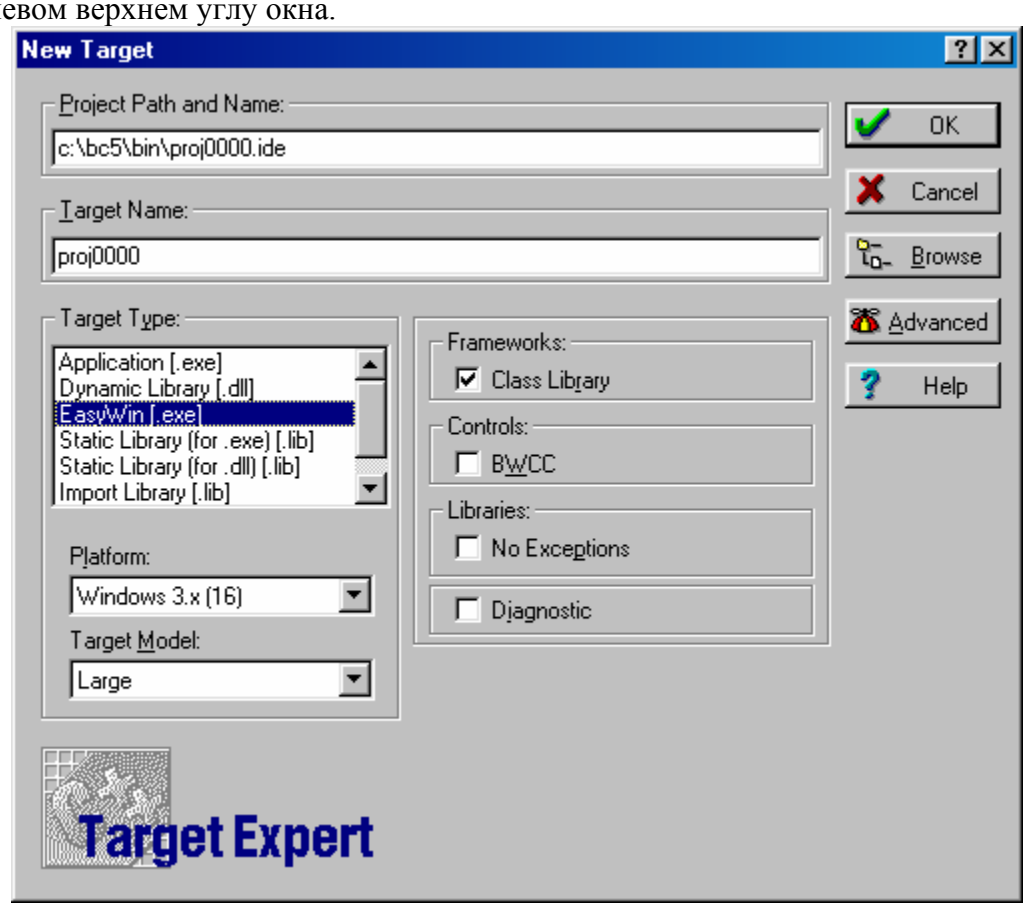

Рисунок 2. Типовое диалоговое окно New Target.

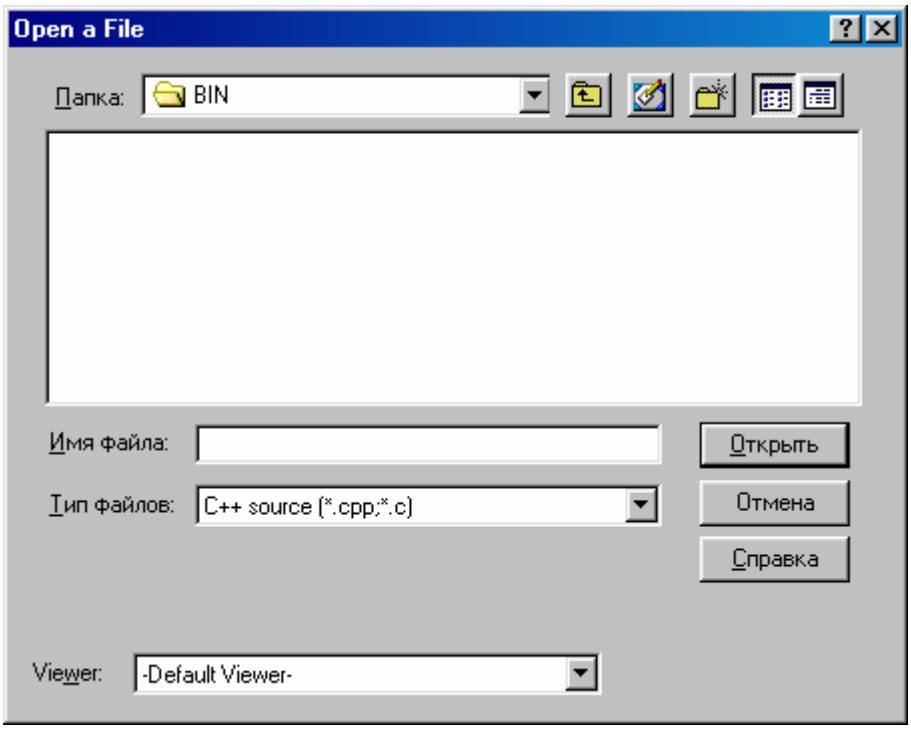

Рисунок 3. Диалоговое окно Open File.

Команда Save

Команда Save (Сохранить) позволяет вам сохранить содержимое активного окна редактирования в связанном с ним файле. Если вы выполняете эту команду для нового окна редактирования, команда Save вызовет диалоговое окно Save File As, показанное на рисунке 4. В этом диалоговом окне вы можете задать имя файла, отличное от имени по умолчанию, а также изменить дисковод и каталог. Быстрые клавиши вызова команды Save — Ctrl+K, Ctrl+S. Обратите внимание, на то, что они могут быть изменены, если вы измените назначения клавиш в меню Options | Environment.

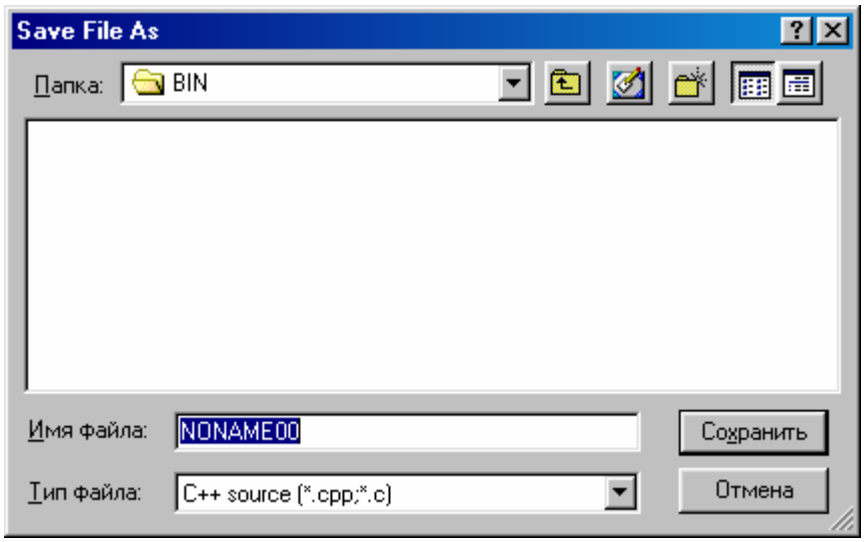

Рисунок 4. Диалоговое окно Save File As.

Команда Save As ...

Команда Save As ... (Сохранить как…) дает вам возможность сохранить содержимое активного окна редактирования в файле, отличном от файла, связанного в настоящее время с окном редактирования. Новое имя файла становится новым связанным с окном именем файла. Команда Save As ... вызывает диалоговое окно Save File As, показанное на рисунке 4. Если вы выбираете существующий файл, команда отображает диалоговое окно сообщения, которое спрашивает вас, действительно ли вы хотите переписать содержимое существующего файла данными из активного окна редактирования, или не хотите.

Команда Save All

Команда Save All (Сохранить все) сохранит содержимое всех измененных окон редактирования в связанных с ними файлах. Если в IDE были открыты новые окна редактирования, эта команда вызовет диалоговое окно Save File As, чтобы сохранить содержимое этих новых окон.

Команда Print ...

Команда Print ... (Печать...) распечатывает содержимое активного окна редактирования. Команда отображает диалоговое окно Print Options, показанное на рисунке 5. Это диалоговое окно содержит кнопки установки следующих параметров печати:

- Печать заголовка и номеров страниц
- Печать номеров строк
- Выделение при печати ключевых слов жирными символами
- Использовать цвет (если ваш принтер поддерживает цвета)
- Перенос строк при печати
- Установка левого поля

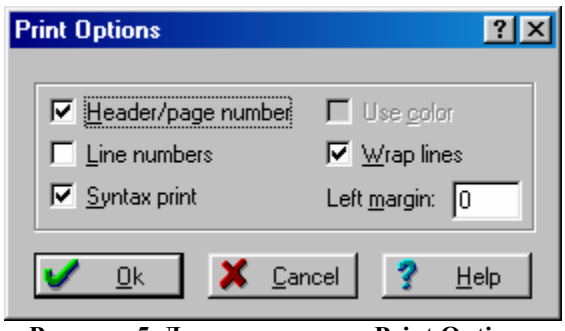

Рисунок 5. Диалоговое окно Print Options.

Команда Print Setup ...

Команда Print Setup ... (Параметры распечатки...) дает вам возможность установить параметры принтера, не печатая файла. Команда подготовки принтера вызовет окно диалога Setup. Это — то же самое диалоговое окно, которое появляется всякий раз, когда вы выбираете команду Print Setup ... из любой программы.

Команда Send ...

Команда Send ... (Послать) посылает содержимое текущего окна редактирования по электронной почте. Это — та же самая команда, которую вы найдете в любой поддерживающей электронную почту прикладной программе.

Команда Exit

Команда Exit (Выход) позволяет вам выйти из IDE Borland C++. Если в окнах редактирования остался измененный, но не сохраненный текст, IDE предупредит вас об этом.

Меню Edit

Меню Edit (Редактирование) содержит команды редактирования текста в окнах редактирования. В таблице 2 перечислены команды меню Edit.

Таблица 2. Обзор команд меню Edit

Команда	Клавиши быстрого выбора команд	Выполняемые действия
Undo	Ctrl+Z	Отменяет последнюю операцию редактирования.
Redo	Shift+Ctrl+Z	Отменяет действие последней команды Undo.
Cut	Ctrl+X	Удаляет отмеченный текст и копирует его в Буфер обмена. Предыдущее содержимое Буфера обмена теряется.
Copy	Ctrl+C	Копирует отмеченный текст в Буфер обмена. Предыдущее содержимое Буфера обмена теряется.
Paste	Ctrl+V	Вставляет содержимое Буфера обмена в текущем положении курсора.
Clear	Ctrl+Delete	Удаляет отмеченный текст, но не записывает его в Буфер обмена.
Select all		Отмечает весь текст в активном окне редактирования.
Buffer list...		Выводит на экран диалоговое окно Buffer List.

Команда Undo

Команда Undo (Отменить) отменяет действие последней операции редактирования и восстанавливает содержимое активного окна редактирования в соответствующем виде. Клавиши быстрого выбора этой команды — Ctrl+Z. Эта команда позволяет вам быстро и эффективно исправлять ошибки редактирования.

Команда Redo

Команда Redo (Возвратить) отменяет действие последней команды Undo. Клавиши быстрого выбора команды Redo — Shift+Ctrl+Z. Команда Redo дает вам возможность переключаться между двумя версиями редактируемого исходного текста.

Команда Cut

Команда Cut (Вырезать) удаляет отмеченный текст и помещает его в Буфер обмена. Предыдущее содержимое Буфера обмена теряется. Клавиши быстрого выбора команды Cut — Ctrl+X.

Команда Copy

Команда Copy (Копировать) копирует отмеченный текст в Буфер обмена. Предыдущее содержимое Буфера обмена теряется. Клавиши быстрого выбора команды Copy — Ctrl+C.

Команда Paste

Команда Paste (Вставить) вставляет содержимое Буфера обмена в текущей точке вставки. Содержание Буфера обмена остается неизменным. Таким образом, вы можете использовать команды Cut и Paste для перемещения текста в том же самом окне редактирования или между различными окнами. Вы также можете использовать команды Cut и Paste для дублирования фрагментов текста в пределах одного окна редактирования или в различных окнах. Клавиши быстрого выбора команды Paste — Ctrl+V.

Команда Clear

Команда Clear (Очистить) удаляет отмеченный текст, не копируя его в Буфер обмена. Это не означает, что удаленный текст безнадежно потерян — вы можете использовать команду Undo, чтобы восстановить его. Клавиши быстрого выбора команды Clear — Ctrl+Delete.

Команда Select All

Команда Select All отмечает весь текст в активном окне редактирования. Вы можете копировать этот текст в Буфер обмена, используя команду Copy. Вы можете затем вписать содержимое Буфера обмена в другое окно редактирования, используя команду Paste.

Команда Buffer List...

Команда Buffer List... (Список буферов...) позволяет вам просмотреть список буферов, используемых с различными окнами редактирования. Эта команда выводит окно диалога Buffer List, в котором вы можете выбрать и загрузить буфер в окно редактирования. Диалоговое окно содержит список буферов, причем те из них, которые были изменены после своей загрузки, помечены словом modified (изменен), помещенным в круглые скобки.

Диалоговое окно списка буферов — особенность, заимствованная из редактора BRIEF. В этом редакторе имеется список буферов и одно окно редактирования; вы переключаетесь между различными файлами, заменяя просматриваемый в настоящее время буфер в окне редактирования. Диалоговое окно Buffer List интегрированной среды позволяет вам заменять содержимое текущего окна редактирования, не закрывая файлы. Если замененный файл не загружен в другое окно редактирования, он становится скрытым. Позже вы можете использовать список буферов, чтобы загрузить скрытый буфер в окно редактирования.

Вы можете использовать кнопку Save диалогового окна Buffer List, чтобы обновить файл, связанный с выбранным буфером. После этого приписка modified у выбранного буфера исчезнет. Вы также можете использовать кнопку Delete для удаления выбранного буфера из памяти, если этот буфер не загружен в окно редактирования.

Меню Search

Меню Search (Поиск) содержит команды поиска информации различных типов, таких, как фрагменты текста, определения символов, объявления функций и ошибки при построении программы. В таблице 3 перечислены команды меню Search.

Таблица 3. Обзор команд Search-меню.

Команда	Клавиши быстрого выбора команд	Выполняемые действия
Find ...	Ctrl+F	Поиск текста в активном окне редактирования.
Replace ...	Ctrl+Q Ctrl+A	Замена текста в активном окне редактирования.
Search again	F3	Повторяет последнюю команду Find или Replace.
Browse symbol ...	Ctrl+O Ctrl+B	Просматривает символ в любом исходном коде, который является частью текущего проекта.
Locate symbol ...		Поиск символа.
Previous message	Alt+F7	Выбирает предыдущее сообщение, полученное при построении программы, и помещает курсор в строку, вызвавшую это сообщение.
Next message	Alt+F8	Выбирает следующее сообщение, полученное при построении программы, и помещает курсор в строку, вызвавшую это сообщение.

Команда Find...

Команда Find... (Найти) организует поиск текста в активном окне редактирования. Эта команда, имеющая клавиши быстрого выбора Ctrl+F, отображает диалоговое окно Find Text, показанное на рисунке 6. Это диалоговое окно имеет следующие средства управления:

- Поле ввода со списком Text to find, которое позволяет вам ввести текст для поиска или повторить фрагменты текста, которые уже участвовали в поиске.
- Кнопки параметров, которые включают в себя
- Переключатель Case sensitive, задающий учитывающий регистр или не учитывающий регистр режимы текстового поиска.
- Переключатель Whole words only, который дает вам возможность выбрать между поиском слов целиком или последовательности символов.
- Переключатель Regular expression, который включает или отключает возможность задания в строке поиска символов шаблона редактора BRIEF.
- Радиокнопки Direction — задают поиск в прямом (Forward) и обратном (Backward) направлениях.
- Радиокнопки Scope, которые дают возможность вам выбрать между поиском по всему тексту (Global) и поиском в выбранном фрагменте текста (Selected text).
- Радиокнопки Origin, которые дают вам выбор между поиском во всей области поиска (Entire scope) и поиском от текущей позиции курсора (From cursor).
- Кнопки OK, Cancel и Help.

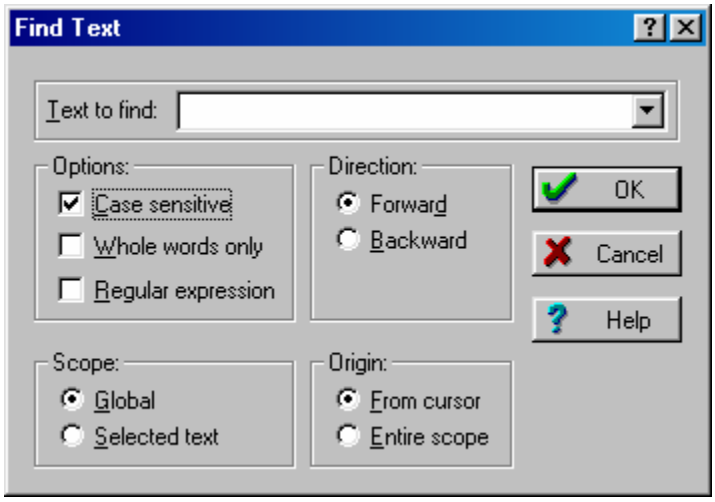

Рисунок 6. Диалоговое окно Find Text.

Команда Replace...

Команда Replace... (Заменить) поддерживает замену текста в активном окне редактирования и имеет клавиши быстрого выбора команды Ctrl+Q Ctrl+A. Эта команда выводит диалоговое окно Replace Text, которое выглядит точно также, как и окно диалога Find..., если не считать нескольких добавлений. Появляется новое поле ввода со списком, в которое вы вводите текст, используемый для замены найденного текста; имеется новый параметр, устанавливающий, спрашивать вас или нет относительно каждой замены, и имеется кнопка Change All, которая будет делать замены сразу во всей области поиска и замены.

Команда Search Again

Команда Search Again (Повторный поиск) дает вам возможность повторить последнюю команду Find... или Replace... . Клавиша быстрого выбора для этой команды — функциональная клавиша F3.

Команда Browse Symbol

Команда Browse Symbol (Просмотр символов) позволяет вам просматривать символы (идентификаторы), включая классы, функции и переменные. Эти символы не обязательно должны быть определены в активном окне редактирования, они могут быть определены в одном из файлов, входящем в проект (ваши программные файлы или файлы библиотек); информацию о символах предоставляет компилятор, если установлена опция включения в программный код информации для отладчика. На рисунке 7 показано типовое диалоговое окно просмотра символа.

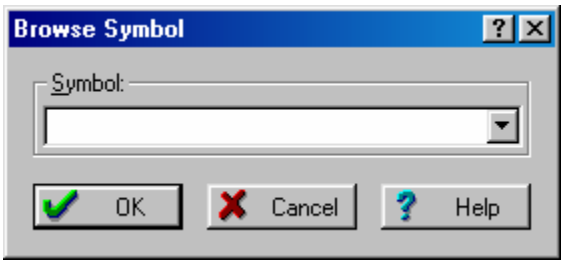

Рисунок 7. Типовое диалоговое окно просмотра.

Команда Locate Symbol

Команда Locate Symbol (Найти символ) находит определение символа. Эта команда выводит диалоговое окно Locate Symbol, которое просит вас ввести имя символа для поиска. IDE показывает вам определение символа в существующем окне редактирования, если оно находится там, либо открывает новое окно.

Команда Previous Message

Команда Previous Message (Предыдущее сообщение) дает вам возможность увидеть строку исходного текста, которая связана с предыдущим сообщением из окна Message. IDE показывает эту строку в окне редактирования. Клавиши быстрого выбора этой команды — Alt+F7. Эта команда полезна, когда трансляция завершилась с сообщениями об ошибках и вам нужно найти строки исходного текста, связанные с этими ошибками.

Команда Next Message

Команда Next Message (Следующее сообщение) дает вам возможность увидеть строку исходного текста, которая связана со следующим сообщением в окне Message. IDE показывает эту строку в окне редактирования. Клавиши быстрого выбора этой команды — Alt+F8.

Меню View

Меню View (Просмотр) содержит команды просмотра и обзора разнообразной информации. Эта информация много шире объявлений в файлах исходного текста вашего проекта. В таблице 4 перечислены команды меню View.

Таблица 4. Список команд меню View.

Команда	Выполняемые действия
ClassExpert	Вызывает утилиту ClassExpert, которая работает с файлами проекта, сгенерированными AppExpert.
Project	Открывает окно Project.
Message	Открывает окно Message.
Classes	Просмотр классов.
Globals	Просмотр глобальных данных, констант и переменных.
CPU	Открывает окно CPU.
Process	Открывает окно Process.
Watch	Выбирает или открывает окно Watch.
Breakpoint	Выбирает или открывает окно Breakpoints.
CallStack	Выбирает или открывает окно Call Stack.

Команда ClassExpert

Команда ClassExpert (Эксперт классов) вызывает утилиту ClassExpert, которая работает только с файлами проектов, созданными AppExpert (обе эти утилиты будут обсуждаться в День 20). Эта команда вызывает окно ClassExpert со следующими панелями:

- Панель Classes (Классы) содержит список классов, включенных в проект утилитой AppExpert. Информация в других двух панелях относится к выбранному в данной панели классу.
- Панель Events (События) выводит список команд, элементов управления, виртуальных функций, сообщений Windows и других событий, связанных с классом, выбранным в панели Classes.
- Панель исходного текста, в котором определен выбранный класс.

Команда Project

Команда Project (Проект) выбирает или открывает окно Project, в котором представлены компоненты — цели и узлы — текущего IDE-файла. Окно Project отображает файлы (узлы) цели в виде дерева. Узлы этого дерева вы можете раскрыть или свернуть, в том случае, если они имеют узлы, от которых зависят. Такие узлы имеют слева от себя знак « + » или «-». Если это знак «+» , то узел имеет подузлы, от которых он зависит и которые в настоящее время скрыты. Если вы щелкнете на знаке «плюс», то вы увидите дерево узлов, от которых этот узел зависит, при этом IDE заменит знак « + » на знак «-». Узлы, не имеющие знаков «+» или «-», не зависят

от других узлов.

Если вы щелкнете правой кнопкой мыши на узле в окне Project, IDE выведет всплывающее меню, в котором вы можете просмотреть различные компоненты проекта, управлять узлами и редактировать связанные с проектом компоненты.

Окно проекта будет обсуждаться более подробно чуть позже.

Команда Message

Команда Message (Сообщения) выводит, выбирает или открывает окно Message, которое содержит различные сообщения, приходящие из различных частей IDE. Закладка Buildtime открывает сообщения, связанные со сборкой проекта, закладка Runtime открывает сообщения, полученные со время выполнения приложений, а закладка Script открывает сообщения подсистемы сценариев.

Команда Classes

Команда Classes (Классы) выводит окно Browsing Objects, в котором показаны классы текущего проекта и то, как они связаны друг с другом. Обычно окно Browsing Objects имеет вертикальную и горизонтальную линейки прокрутки, чтобы дать вам возможность просмотреть все классы проекта. Эта команда будет обсуждаться немного позднее, когда у нас появятся классы, которые мы сможем просмотреть.

Команда Globals

Команда Globals (Глобальные объекты) выводит окно Browsing Globals, в котором отображаются глобальные типы данных — константы, переменные и функции. На рисунке 8 показано типовое окно Browsing Globals. Окно идентифицирует каждый объект, используя следующие значки:

- Знак F означает, что символ является функцией.
- Знак T указывает на то, что символ — тип данных.
- Знак V указывает на то, что символ — переменная.
- Знак C — указание на то, что символ — константа.

Окно Browsing Globals содержит переключатели, которые дают возможность отфильтровать для просмотра глобальные символы нужного типа. Окно также содержит поле ввода, в котором вы можете указать имя символа, который вы хотите найти. Список объектов фильтруется после каждого нажатия клавиши в поле ввода.

Рисунок 8. Типовое окно Browsing Globals.

Команда CPU

Команда CPU (Процессор) выводит окно, которое используется обычно при выполнении и отладке программы. В этом окне отображается машинный код программы, содержимое

регистров, состояние флагов и другие данные. Эта информация, безусловно, предназначена для опытного программиста, поскольку здесь приводится код ассемблера. Для опытного программиста — это неоценимый инструмент наблюдения за выполнением программы.

Команда Process

Команда Process (Процесс) выводит окно со списком различных процессов, в настоящее время выполняемых в IDE. Обратите внимание на то, что в списке перечислены только программы, которые запускаются через IDE, и не указаны остальные программы, выполняющиеся на компьютере.

Команда Watch

Команда Watch (Наблюдение) выбирает или открывает окно Watch. Это окно содержит список наблюдаемых во время отладки переменных вашей программы. Слева от каждой переменной находится переключатель. Переключатель, по умолчанию, отмечен для того, чтобы отображать и обновлять значение переменной. Вы можете снять отметку, чтобы временно отключить вывод значения переменной. Эта возможность удобна тогда, когда наблюдаемая переменная не определена в выполняемой функции. Окно Watch обсуждается в дополнительном Дне 1.

Команда Breakpoint

Команда Breakpoint (Контрольная точка) выводит окно Breakpoints, в котором перечислены точки останова (контрольные точки), их расположение и тип. Контрольная точка ставится на операторе программы, на котором программа приостанавливается, чтобы дать возможность вам просмотреть переменные. Окно Breakpoints отображает список со следующей информацией:

- Переключатель, обозначающий состояние контрольной точки (активна или неактивна)
- Имя файла, в котором установлена контрольная точка
- Расположение контрольной точки (это может быть номер строки, имя файла и т. д.)
- Текущее состояние контрольной точки (проверяемая, непроверяемая или недействительная)
- Число проходов (т. е. сколько раз должен выполнится оператор, на котором установлена контрольная точка, прежде чем будет сделан останов)

Если вы дважды щелкнете на любой контрольной точке в окне Breakpoints, IDE выведет диалоговое окно Breakpoints Properties. Это диалоговое окно позволяет редактировать данные контрольной точки. Это диалоговое окно обсуждается в дополнительном Дне 1.

Команда Call Stack

Команда Call Stack (Стек вызовов) выводит окно Call Stack, содержащее список незавершенных процедур или функций DLL, которые были вызваны (и еще не завершили работу), когда программа достигла текущей контрольной точки или перешла к следующему оператору в случае пошагового выполнения. Для функции DLL указывается имя DLL-библиотеки, за которым следует адрес функции. Эта команда используется при отладке программ и будет освещаться в дополнительном Дне 1.

Меню Project

Меню Project (Проект) предлагает команды управления проектом создания исполняемой программы или библиотеки. В таблице 5 перечислены команды меню Project.

Таблица 5. Список команд меню Project.

Команда	Клавиши быстрого выбора	Выполняемые действия

	команд	
Open project...		Открывает существующий проект и закрывает текущий проект.
Close project		Закрывает текущий проект.
New target...		Создает новую цель в текущем проекте.
Compile	Alt+F9	Компилирует файл в активном окне редактирования.
Make all	F9	Обновляет файлы проекта, компилируя и компонуя необходимые файлы исходного кода.
Build all		Безусловно компилирует и компонует все файлы источников проекта.
Generate makefile		Создает make-файл на основе текущего проекта.

Команда Open Project...

Команда Open Project... (Открыть проект...) открывает новый проект и автоматически закрывает текущий проект. Эта команда выводит диалоговое окно Open Project File, которое походит на диалоговое окно File Open. Диалоговое окно Project File дает возможность вам определить дисковод, каталог и шаблон имени файла при выборе .IDE- или .PRJ-файла проекта. Файлы проекта .PRJ из предыдущих версий Borland C++ могут загружаться благодаря автоматическому преобразованию их в .IDE-файлы.

Команда Close Project

Команда Close Project (Закрыть проект) закрывает текущий проект и его окна редактирования.

Команда New Target...

Команда New Target... (Новая цель...) позволяет добавить в проект новый целевой модуль. Команда сначала выводит диалоговое окно Add Target, в котором нужно ввести имя и тип цели. Цель может быть типа Standard или SourcePool. Если вы выбираете цель стандартного типа, то IDE вызывает диалоговое окно Add Project. Если вы выбираете SourcePool — IDE добавляет целевой узел пула источников. Окно Project отразит добавление новой цели и укажет ее тип.

Пул источников содержит набор узлов, которые не указываются в проекте по отдельности. Пул источников выполняет роль шаблона, на который делаются ссылки, что позволяет различным целям использовать общий исходный текст.

Команда Compile

Команда Compile (Компилировать) компилирует исходный текст программы в активном окне редактирования. Команда открывает диалоговое окно Compile Status, в котором вы можете увидеть результат компиляции файлов — количество обработанных строк, число предупреждений и число сообщений об ошибках. Когда процесс компиляции завершится, в окне Message появятся общие сообщения по этапам компиляции и предупреждения и сообщения об ошибках, сгенерированные компилятором, компоновщиком и другими инструментальными средствами. Клавиши быстрого выбора этой команды — Alt+F9.

Команда Make All

Команда Make All (Собрать) обновляет цель проекта, компилируя и компонуя заново только те файлы, которые были изменены после выполнения команд Make All или Build All в последний раз. Команда использует диалоговое окно Compile Status для отображения процесса компиляции и сборки. После того, как этот процесс завершится, можно просмотреть сообщения в окне Message по поводу компиляции и компоновки, и, кроме того, предупреждения и сообщения об ошибках. Клавиши быстрого выбора этой команды — F9.

Команда Build All

Команда Build All (Перестроить) подобна команде Make All, за исключением того, что она перетранслирует и связывает все файлы проекта.

Команда Generate Makefile

Команда Generate Makefile (Создать Make-файл) создает файл утилиты make для текущего проекта. Файл Make подобен файлу проекта, только он имеет текстовый формат и выполняется из командной строки.

Меню Script

Меню Script (Сценарий) содержит набор команд управления объектными сценариями, при помощи которых, кстати, можно контролировать большую часть IDE. В таблице 6 перечислены команды меню Script. Полное описание объектных сценариев и языка cScript можно найти в файлах интерактивной справки.

Таблица 6. Список команд меню cScript.

Команда	Клавиши быстрого выбора команд	Выполняемые действия
Run…		Выполняет заданную команду cScript.
Commands...		Выводит список доступных команд cScript.
Modules...		Выводит список доступных модулей cScript.
Compile File		Компилирует cScript-файл, загруженный в активном окне редактирования.
Run File		Выполняет cScript-файл, загруженный в активном окне редактирования.

Меню Tool

Меню Tool (Инструменты) открывает вам доступ к нескольким программным утилитам. Команда Tools... меню Options предоставляет вам возможность настроить список инструментальных средств, доступных в меню Tool. Таблица 7 перечисляет заданные по умолчанию команды меню Tool.

Таблица 7. Команды, заданные по умолчанию в меню Tool.

Команда выбора	Выполняемые действия
Turbo Debugger	Вызывает автономный отладчик.
Grep	Выполняет утилиту Grep на отмеченных узлах.
WinSight	Вызывает утилиту WinSight, контролирующую сообщения Windows.
CodeGuard Config	Конфигурирует параметры утилиты CodeGuard.

Меню Debug

Меню Debug (Отладка) содержит команды управления отладкой и выполнением ваших программ на C или C++. В таблице 8 собраны команды меню Debug. В дополнительном Дне 1 вам будет предложен краткий обучающий курс по некоторым простым методам отладки программ.

Таблица 8. Команды меню Debug.

Команда	Клавиши быстрого выбора команд	Выполняемые действия

Run	Ctrl+F9	Выполняет программный модуль текущей цели. В случае необходимости эта команда, также выполняет компилирование и компоновку файлов проекта.
Load...		Загружает программу для отладки.
Attach...		Подсоединяется к выполняющейся программе для отладки.
Run to...		Выполняет программу до указанного места.
Pause process		Приостанавливает выполнение программы и передает управление отладчику.
Reset this process		Сбрасывает программу, ~ при последующем ее запуске или пошаговом выполнении программа начинается с начала.
Terminate process	Ctrl+F2	Останавливает программу.
Source at execution point		Пробует найти исходный текст, соответствующий текущему исполняемому машинному коду.
Add breakpoint...		Открывает окно диалога Add breakpoint.
Breakpoints Options...		Устанавливает параметры контрольной точки.
Add Watch...	Ctrl+F5	Открывает диалоговое окно Watch Properties, чтобы добавить новую переменную для наблюдения.
Evaluate...	Ctrl+F7	Вычисляет значение выражения и изменяет значение переменной.
Inspect...	Alt+F5	Показывает значение переменной.

Меню Options

Меню Options (Параметры) дает возможность настроить работу компилятора, компоновщика, редактора и всех остальных компонентов IDE. В таблице 9 перечислены команды меню Options.

Таблица 9. Команды меню Options.

Команда	Выполняемые действия
Project...	Просмотр и редактирование параметров текущего проекта.
Environment...	Просмотр и редактирование установок IDE.
Tools...	Добавление, удаление или изменение инструментальных средств в меню Tool.
Style Sheets...	Редактирование параметров листов стиля.
Save...	Сохранение параметров конфигурации проекта, рабочего стола и среды.

Команда Project...

Команда Project... (Проект ...) выводит диалоговое окно с заголовком Project Options, показанное на рисунке 9. Диалоговое окно Project Option содержит список параметров в левой половине и набор средств управления в правой. Набор средств управления меняется соответственно разделу, выбранному в левом списке. Параметры проекта объединены в разделы следующим образом:

- В разделе Directories (Каталоги) определяются каталоги для include-фай-лов, библиотек и файлов исходных текстов, и, также, каталоги для промежуточных и конечных файлов.
- Раздел Compiler (Компилятор) дает вам возможность настроить компиляцию исходных модулей C и C++, определить директивы препроцессора, управлять включением в код отладочной информации и управлять прекомпиляцией файлов заголовков.
- Раздел 16-bit Compiler (16-битный компилятор) и раздел 32-bit Compiler (32-битный компилятор) управляет зависимыми от системы параметрами компиляторов.
- Раздел C++ определяет, как компилятор C++ интерпретирует ваш исходный текст и учитывает языковые особенности нового и старого C++.
- Раздел Optimizations (Оптимизация) позволит вам так настроить компилятор, чтобы он

генерировал более эффективный код.

- Раздел Messages (Сообщения) дает вам возможность указать тип сообщений, генерируемых при создании программы. Эти параметры позволяют выбрать между очень строгим и очень слабым контролем программы и, соответственно, количеством предупреждений и сообщений об ошибках.
- Раздел Linker (Компоновщик) управляет процессом сборки файлов .OBJ и .LIB в исполняемый .EXE-файл.
- Раздел Librarian (Библиотекарь) дает возможность вам объединить набор файлов .OBJ в файл .LIB библиотеки и управлять этим процессом.
- В разделе Resources (Ресурсы) вы можете определить целевую версию Windows и создать нужный для нее тип файла ресурсов .RES.
- Раздел Make (Сборка) содержит общие опции процесса построения программы.

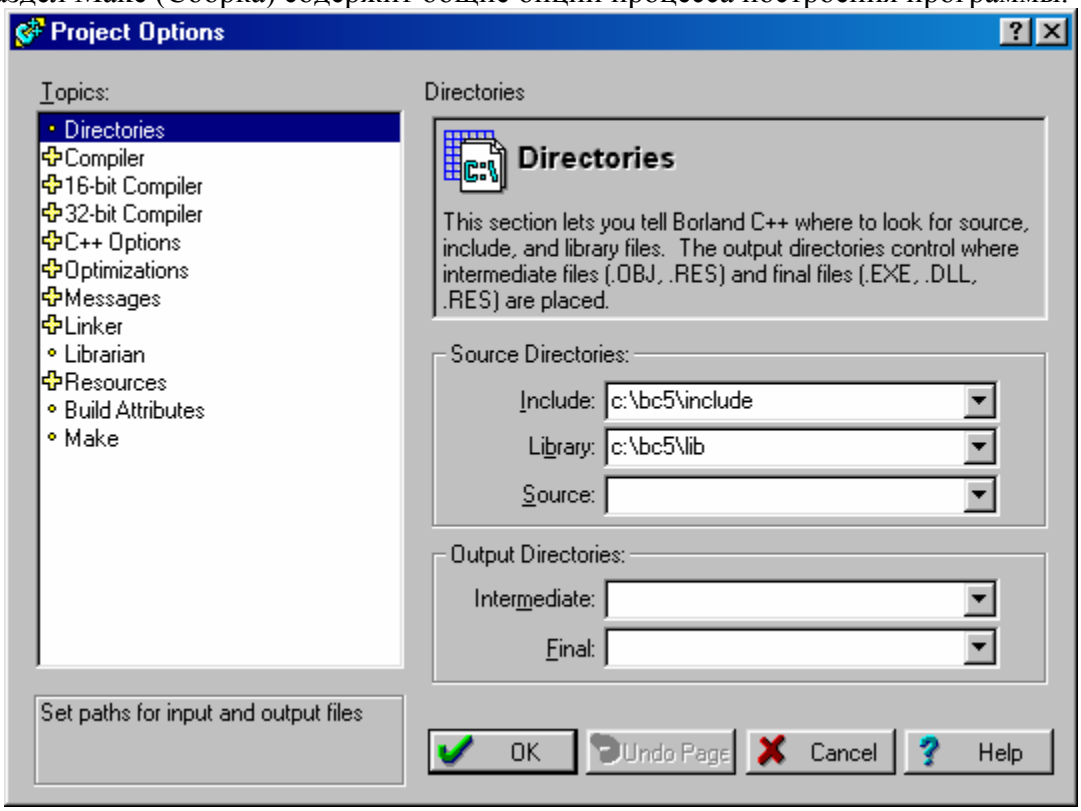

Рисунок 9. Образец диалогового окна Project Options.

Команда Environment

Команда Environment (Среда...) отображает диалоговое окно Environment Options, показанное на рисунке 10, в котором вы можете настроить различные параметры IDE. Эти настройки собраны в следующие разделы:

- Раздел Browser (Окна просмотра) определяет заданные по умолчанию фильтры Browser. Кроме того, вы можете установить режим создания новых окон при просмотре иерархии классов.
- Раздел Editor (Редактор) задает настройки редактора текста IDE. Вы можете выбрать текстовый редактор по умолчанию (похожий на редактор WordStar), классический текстовый редактор IDE, эмулятор BRIEF-редактора, эмулятор Epsilon-редактора или настроить различные параметры текущего текстового редактора.
- Раздел Syntax Highlighting (Выделение синтаксиса) определяет цвет и стиль, используемый редактором, при отображении исходного текста. По умолчанию задан некий предопределенный наборов цветов и стилей.
- Раздел Speedbar (Оперативная панель) дает вам возможность настроить расположение и состав кнопок оперативной панели.
- Экран Scripting (Сценарии) определяет настройки объектных сценариев — задание

сценария, выполняющегося при загрузке IDE, и каталог файлов сценария.

- Раздел Process Control (Управление процессом) дает возможность вам управлять процессом построения программы. Вы можете определить количество отображаемой информации и приоритеты IDE.
- Раздел Preferences (Предпочтения) устанавливает настройки, связанные с автоматическим сохранением различных компонентов IDE, таких, как редактируемые файлы, параметры среды, рабочий стол и проект. В разделе можно определить части рабочего стола, которые должны сохраняться.
- Раздел Fonts (Шрифты) позволяет вам определить отображение текста в различных окнах IDE.
- Раздел Project View (Просмотр проекта) определяет тип информации, включаемой в окно проекта — размер кода, размер данных, расположение, имя, число строк, тип узла и т. д.
- Раздел Debugger (Отладчик) позволяет установить параметры и исходный каталог для отлаживаемой программы, а также установить параметры отладчика и указать место для вывода информационных сообщений и сообщений об ошибках.
- Раздел Resource Editors (Редакторы ресурсов) устанавливает различные настройки интегрированного редактора ресурсов. Здесь вы можете установить библиотеки управляющих элементов VBX.

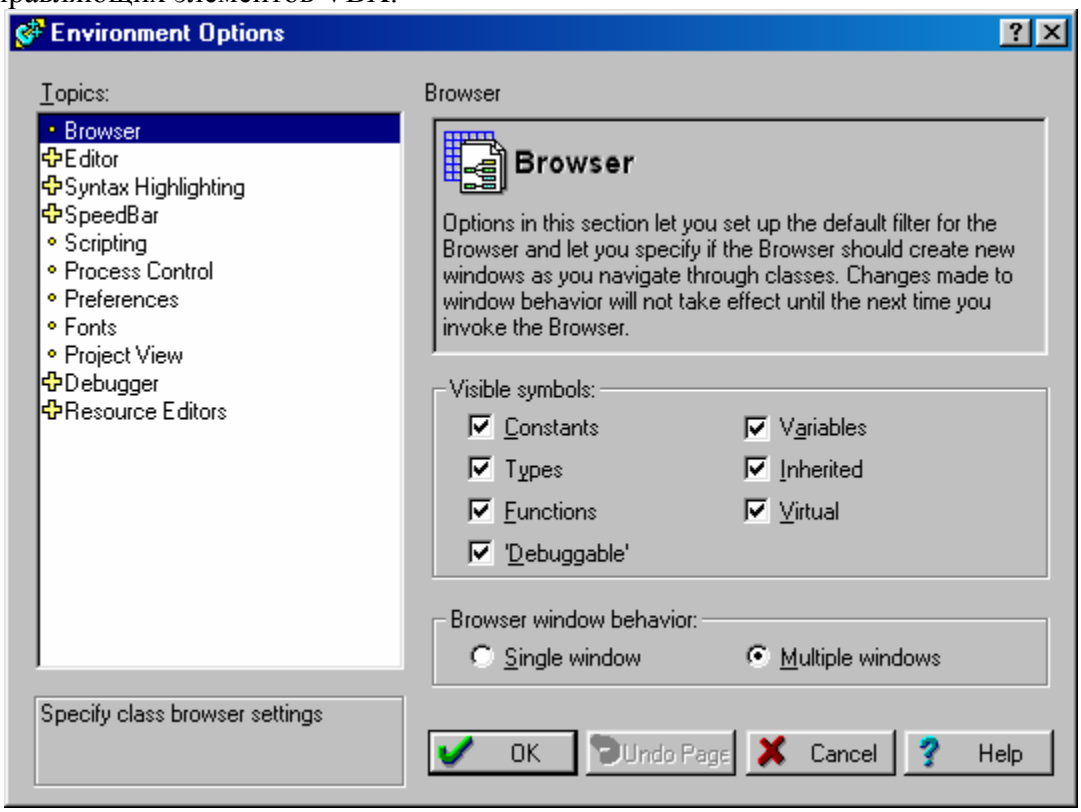

Рисунок 10. Диалоговое окно Enviroment Options.

Команда Tools...

Команда Tools... (Инструменты...) дает возможность добавлять и удалять команды меню Tool. На рисунке 11 показано диалоговое окно Tools, которое содержит список доступных инструментальных средств. Если вы щелкнете на кнопке Edit, появится окно диалога Tool Options, показанное на рисунке 12. В этом диалоговом окне вы можете определить имя инструмента, его каталог, командную строку, текст пункта меню и подсказку (которая появляется в строке состояния).

Обратите внимание на то, что большинство инструментальных средств, отображаемых в списке, фактически являются внутренними инструментальными средствами и не появляются в меню Tool. Некоторые из перечисленных инструментов могут быть добавлены в меню, если вы пожелаете, однако многие не имеет смысла туда переносить.

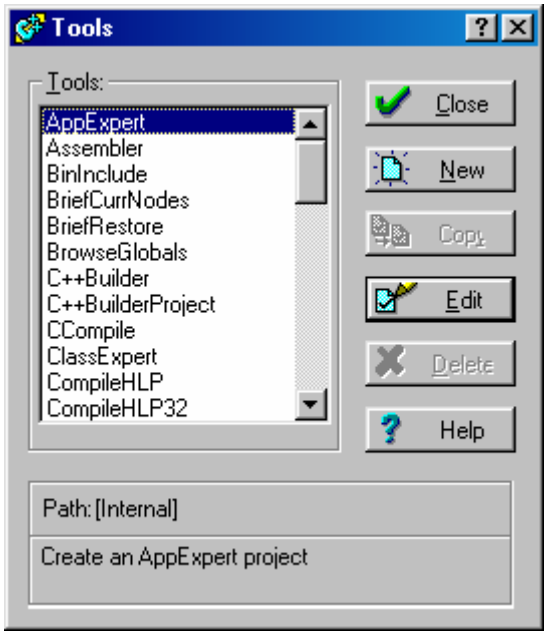

Рисунок 11. Диалоговое окно Tools.

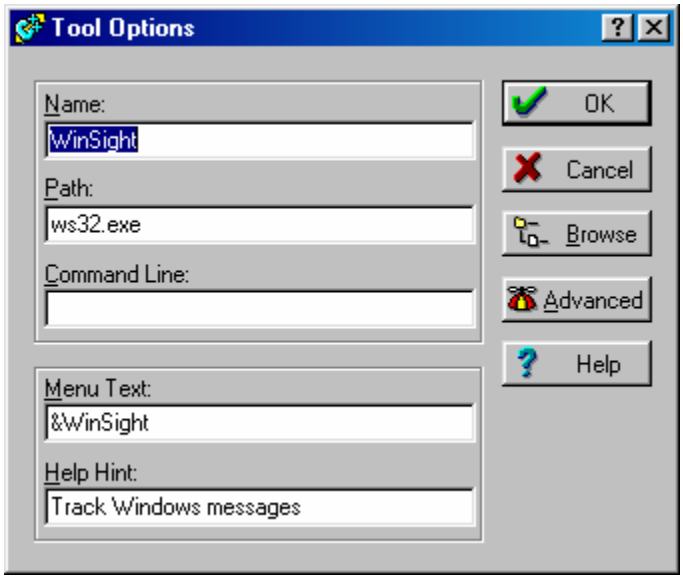

Рисунок 12. Диалоговое окно Tool Options.

Команда Style Sheets-Style Sheets...

Команда Style Sheets-Style Sheets... (Листы стиля ...) команда отображает диалоговое окно Style Sheets, показанное на рисунке 13, где вы можете выбрать конфигурацию компилятора и исполняющей системы для проекта. Каждый лист стиля содержит предопределенную совокупность установок, которые можно присоединить к узлу проекта.

Команда Save...

Команда Save... (Сохранить ...) сохраняет текущие установки рабочего стола, среды и файл проекта. Эта команда вызывает диалоговое окно Save Options с кнопками-переключателями, определяющими сохраняемые компоненты.

Меню Window

Меню Window (Окно) содержит команды управления окнами в области клиента IDE. Эти команды, которые перечислены в таблице 10, дают возможность вам упорядочивать, закрывать, минимизировать и восстанавливать окна по отдельности или все сразу. В дополнение к командам, меню Window содержит список открытых в данный момент окон.

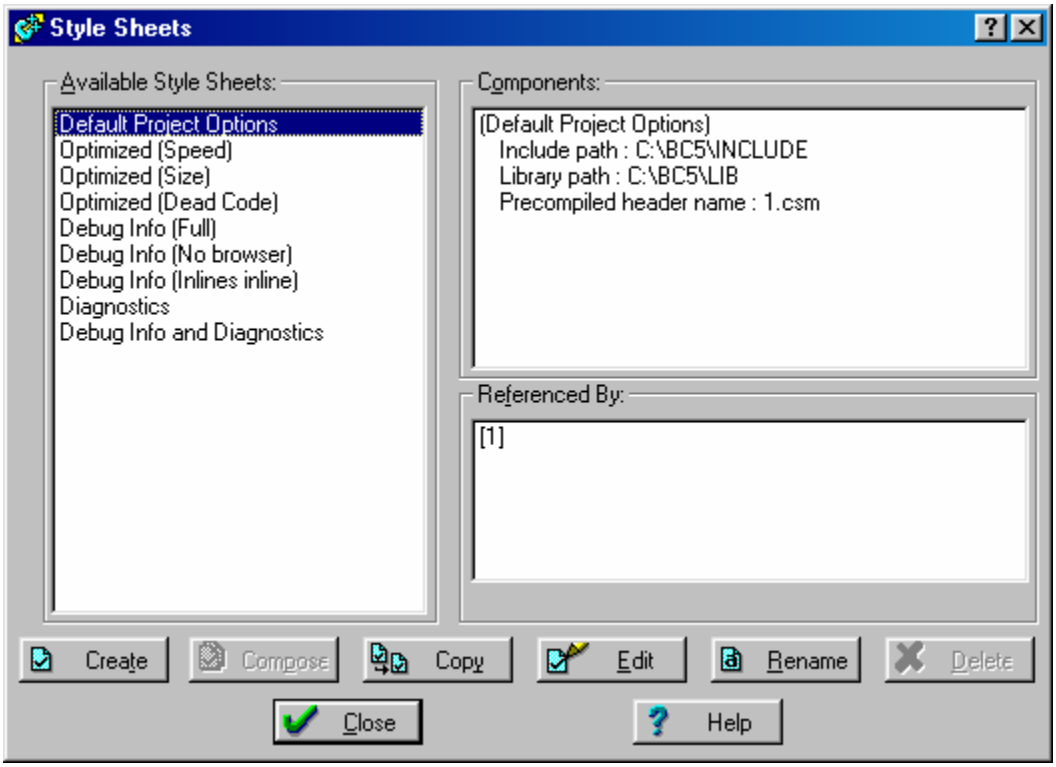

Рисунок 13. Диалоговое окно Style Sheets.

Таблица 10. Команды меню Window.

Команда	Клавиши быстрого вызова	Выполняемые действия
Cascade	Shift+F5	Располагает окна ниспадающим каскадом в области клиента IDE.
Tile horizontal	Shift+F4	Располагает окна мозаикой – горизонтально в области клиента IDE.
Tile vertical		Располагает окна мозаикой – вертикально в области клиента IDE.
Arrange icons		Упорядочивает пиктограммы в области клиента IDE.
Close all		Закрывает все окна – окна отладчика, окна просмотра или окна редактора.
Minimize all		Минимизирует все окна – окна отладчика, окна просмотра или окна редактора.
Restore all		Восстанавливает все окна – окна отладчика, окна просмотра или окна редактора.

Меню Help

Меню Help (Справка) содержит команды интерактивной справки. В таблице 11 перечислены команды меню Help.

Таблица 11. Список команд Help меню.

Команда	Выполняемые действия
Contents	Выводит оглавление системы интерактивной справки.
Keyword search	Выдает справку о ключевом слове, на котором расположен курсор.
Keyboard	Выдает справку по клавиатуре.
Using help	Выводит справку о том, как использовать систему интерактивной справки.
Windows API	Вызывает справочник Windows API.
OWL API	Открывает справочник по OWL.
About...	Выводит сообщение о версии программного обеспечения и авторских правах.

Вы можете также использовать клавишу F1 для получения контекстно-чувствительной

справки относительно активного в настоящее время компонента IDE (окна редактирования, окна проекта или диалогового окна). Эта контекстно-зависимая справка, возможно, является наиболее важным инструментом в IDE, поскольку позволяет вам получить помощь, как только она вам понадобиться.

Приложения EasyWin

Microsoft Windows — сложная среда программирования. Прикладные программы для Windows должны уметь создавать окна, обеспечивать поддержку мыши и делать много других вещей. До того, как появились оконные операционные среды, программы поддерживали простой консольный ввод/вывод, как, например, в MS-DOS. В процессе выполнения такая программа получает входные данные с клавиатуры и выводит посимвольно сообщения на экран. Хотя такой интерфейс не очень удобен, он вполне достаточен для создания простых программ, и вам не нужно заботиться относительно соблюдения тех многих условностей, которые требуются для составления полной Windows-программы.

IDE Borland C++ дает возможность вам создавать специальный вид программ, называемых Easy Win-программами, которые выполняются в простом окне, напоминающем интерфейс старого стиля. Все программы в главах с 1-й по 11-ю будут прикладными программами EasyWin — это даст вам возможность сосредоточиться на изучении C++ как такового, используя консольный интерфейс и соответствующие процедуры ввода/вывода. EasyWin окно имеет простое меню с небольшим набором команд.

Нужно отметить, что EasyWin-программы являются 16-битными. Чтобы иметь простой ввод и вывод в 32-битной программе, вы должны выбрать в TargetExpert в качестве цели консольную прикладную программу Win32. Этот тип приложения задается по умолчанию, когда вы используете инструментальные средства командной строки. Проблема с приложением такого рода, однако, состоит в том, что когда вы запускаете его из Windows, по окончании работы оно сразу закрывает окно. Это значит, что весь вывод, который вас, возможно, интересует, исчезнет с экрана прежде, чем вы сможете его увидеть. Но у вас есть выход — запускать такие программы на выполнение из окна MS-DOS.

Чтобы создать прикладную программу EasyWin, выполните следующие шаги:

1. Загрузите IDE Borland C++.
2. Выберите в меню команду File.
3. Выберите в выпадающем меню команду New, затем Project..., вызывающую диалоговое окно New Target.
4. Введите маршрут и имя файла проекта .IDE в поле ввода в верхней части окна New Target. Вводимое вами имя проекта будет повторено в поле ввода имени цели, тем самым имя программы будет соответствовать имени проекта. Вам следует отредактировать имя цели, если оно не соответствует имени файла проекта .IDE. Вы можете использовать кнопку Browse для выбора каталога, содержащего файлы проекта.
5. Выберите EasyWin [.exe] пункт в списке Target type.
6. Щелкните на кнопке Advanced, чтобы вызвать диалоговое окно Advanced Options. Выберите в нем переключатель, помеченный как .cpp Node. Этот выбор заставит IDE вставлять .CPP-узлы для исходных файлов EasyWin. Это диалоговое окно также дает вам возможность добавлять или удалять модули.RC или .DEF. Так как реально они нужны только при создании прикладных программ Windows, а не для EasyWin-приложений, удостоверьтесь, что эти две кнопки не выбраны. Закройте диалоговое окно Advanced Options, нажав на кнопку OK.
7. Нажмите кнопку OK, чтобы создать новый файл проекта.
8. IDE выведет окно Project, в котором перечислены узлы различных программ. Когда вы создаете новый файл проекта, окно Project будет содержать только один узел.
9. Щелкните на главном узле, чтобы увидеть файлы, содержащиеся в этом узле. Узлы программ EasyWin содержат только один файл — .CPP-файл, содержащий исходный текст. Дважды щелкните на файле .CPP, для того чтобы начать редактирование этого

файла. Первоначально исходное окно файла .CPP пусто.

10. Введите исходный текст для программы EasyWin (вроде показанного в листинге 1).
11. Нажмите клавиши Ctrl+F9 , чтобы откомпилировать, собрать и выполнить программу EasyWin. Вы можете сделать это, нажав кнопку с изображением молнии. Компилятор, в случае обнаружения ошибок, внесет их в окно Message. Если в программе нет ошибок, IDE выполнит ее.

Введение в проекты и узлы

Когда вы создаете новый проект или загружаете готовый файл проекта с диска, IDE Borland C++ открывает окно Project. Это окно содержит все узлы и подузлы, необходимые для создания приложения. Термин *узел* применяется для обозначения различных объектов, находящихся в окне проекта. Эти узлы имеют специфические правила, связанные с ними, которые IDE использует при построении прикладной программы. Кроме того, каждый узел может, в свою очередь, зависеть от одного или большего количества узлов. Это означает, что до обработки такого узла те узлы, от которых он зависит, должны быть предварительно успешно обработаны. Зависимости представляются линиями; подчиненные узлы изображаются со сдвигом.

Вы можете внести больше, чем одну программу в файл .IDE. Я предполагаю, что вы будете группировать отдельные программы каждого из первых двенадцати дней в .IDE-файлах, названных по имени соответствующей главы, типа DAY1.IDE, DAY2.IDE и т. д. Группировка в одном проекте файлов, связанных по смыслу, сохранит вам дисковое пространство, потому что IDE-файлы не маленькие. Меньше IDE файлов — больше дискового пространства. Чтобы добавить еще одну цель в существующий .IDE-файл, выполните следующие шаги:

1. Загрузите IDE Borland C++.
2. Выберите пункт меню Project.
3. Выберите команду New Target, чтобы вызвать диалоговое окно New Target.
4. Введите имя новой цели и нажмите кнопку OK .
5. IDE выведет диалоговое окно New Project. Нажмите кнопку OK, чтобы добавить новый целевой узел в окно Project. Все изменения, которые вы сделали ранее, включая установки окна диалога Advanced, учитываются.
6. IDE покажет окно проекта, в котором новые узлы цели внесены в список.

Этот метод только добавляет в проект новую цель верхнего уровня вместе с одиночным .CPP-файлом для основного исходного текста цели. Вы можете добавить большее количество узлов зависимости к любой цели, выбирая нужный узел и нажимая правую кнопку мыши, и выбирая команду Add Node из выпадающего меню. В появившемся диалоговом окне Add to Project List вы можете указать имя файла нового узла.

Вы должны обратить внимание на то, что некоторые узлы имеют связанный с ними знак «плюс» или «минус». Они означают то же самое, что и папки в File Manager. Знак плюс означает, что узел имеет зависимости, которые в данный момент скрыты, чтобы сэкономить место на экране монитора. Щелкнув на знаке «плюс», вы измените его на знак «минус» и раскроете зависимости этого узла. Нажатие на знак минус узла снова свернет зависимости и изменит минус на плюс.

Бывает чрезвычайно полезно иметь несколько исходных файлов для одной прикладной программы. В конечном счете, когда вы начнете писать полноценные прикладные программы Windows в День 12, вы увидите довольно большие программы. Поместив весь исходный код в одиночном файле, вы создадите громоздкую программу, тогда как разбиение исходного текста на несколько файлов может сделать жизнь программиста намного проще. Мы больше поговорим на эту тему в День 12, и затем в День 20, когда будем обсуждать AppExpert.

Ваша первая программа C++

Первая программа C++, предлагаемая в этой книге, выводит строку приветствия. Эта

простая программа даст вам возможность познакомиться с основными компонентами программы C++.

Листинг 1 содержит исходный текст для программы HELLO.CPP с пронумерованными строками. Не вводите номера строк, когда вы будете набирать текст программы. Эти номера служат только для ссылки на строки программы в тексте книги. Эта простая программа выводит строку «Hello Programmer!» Выполните следующие действия, чтобы создать и выполнить эту первую программу C++:

1. Загрузите IDE Borland C++, если она еще не загружена.
2. Выберите пункт меню File.
3. Выберите меню New и в нем команду Project, которая вызовет диалоговое окно New Project.
4. Наберите c:\bc5\bc21day\dayl\dayl.ide в поле ввода, в котором вводится каталог и имя проекта. (Обратите внимание — здесь предполагается, что вы установили Borland C++ 5 в каталоге, предлагаемом при установке по умолчанию, — c:\bc5; измените, если нужно, имя каталога на соответствующее вашей установке). Теперь вы можете увидеть имя day1 в поле ввода Target, определяющем имя цели. Вы можете изменить это имя на имя Hello.
5. Выберите опцию EasyWin [.exe] в списке Target type.
6. Щелкните на кнопке Advanced, чтобы вызвать диалоговое окно Advanced Options. Выберите кнопку-переключатель, помечающую узел как узел .cpp и удостоверьтесь в том, что опции RC и DEF выключены. Закройте диалоговое окно Advanced Options.
7. Нажмите кнопку OK, чтобы создать новый файл проекта. IDE откроет окно Проекта, в котором содержится узел программы Hello.
8. Щелкните на значке «плюс» слева от узла hello.exe, чтобы посмотреть на узел hello.cpp. Дважды щелкните на узле hello.cpp, чтобы вызвать редактор IDE.
9. Наберите в открывшемся новом окне программу, показанную в листинге 1.1.
10. Выберите команду Save в меню File.
11. Нажмите Ctrl + F9, чтобы откомпилировать, собрать и выполнить программу HELLO.EXE.

С небольшими изменениями, такими/ как имя IDE-файла и имя программы, эти инструкции могут использоваться для всех примеров-программ.

Когда программа EasyWin завершается, исполняющая система изменяет заголовок окна программы, включая в него слово Inactive. Чтобы закрыть окно, выберите команду Close из системного меню окна или нажмите Alt + F4.

Листинг 1. Исходный текст программы HELLO.CPP.

```
1:    // Тривиальная программа C++, которая выводит строку приветствия
2:
3:    #include <iostream.h>
4:
5:    int main ()
6:    {
7:          cout << "Hello Programmer!";
8:          return 0;
9:    }
```

На рисунке 14 приведен вывод программы. Обратите внимание на то, что текст заголовка окна начинается со слова Inactive, указывающего на то, что программа завершилась.

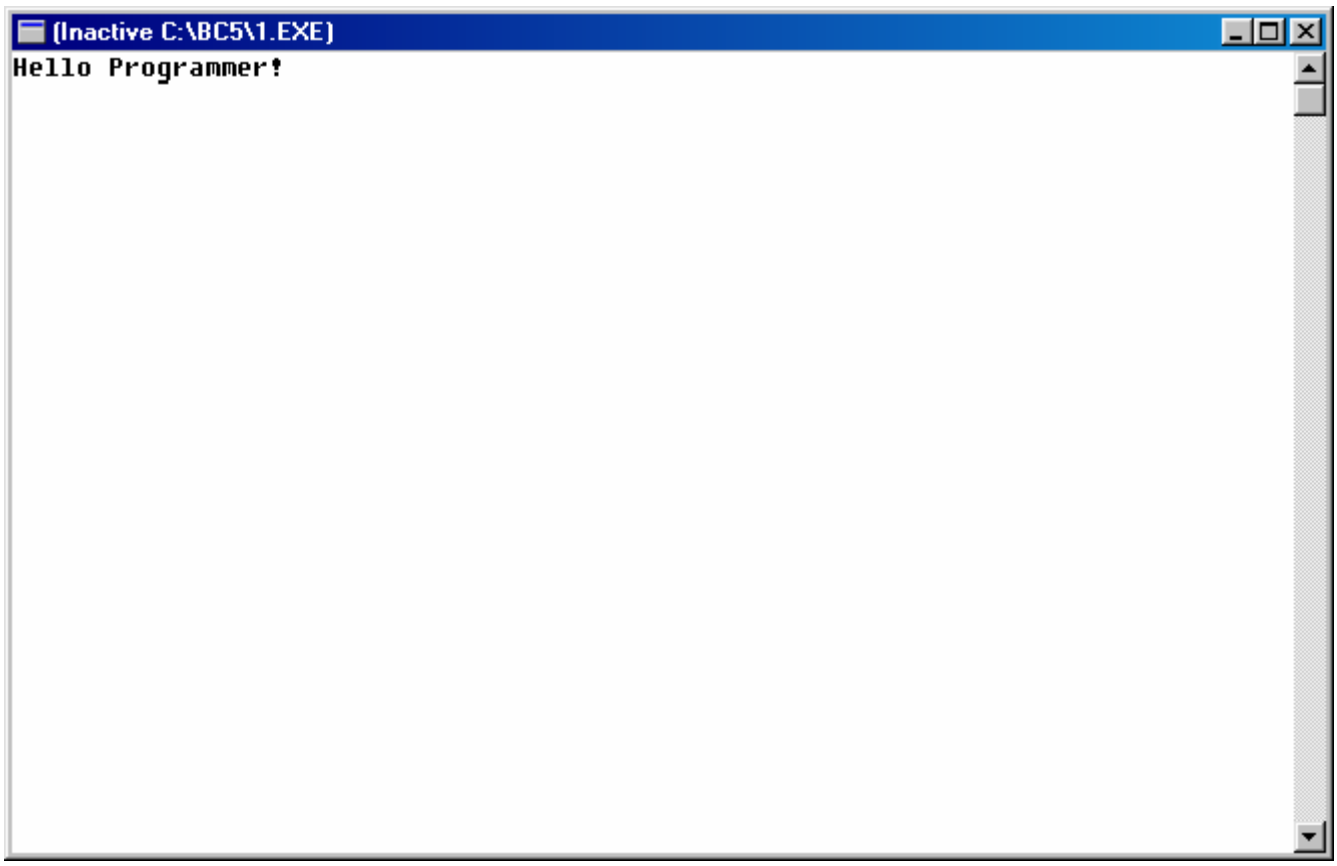

Рисунок 14. Вывод программы HELLO.EXE.

Рассмотрим короткий код программы, и обратим внимание на следующие правила:

- C++ использует символы // для комментария, который продолжается до конца строки. C++ поддерживает комментарии языка C, которые начинаются с символов /* и заканчиваются символами */. Строка 1 содержит комментарий, который кратко описывает программу.

Комментариями называются пояснения, помещаемые в тексте программы для того, чтобы объяснить или описать некоторые ее части. Транслятор игнорирует комментарии, но программист использует их, чтобы знать, что делает программа, особенно если она не использовалась длительное время, и о ее особенностях забыли.

- Программа C++ не имеет никаких зарезервированных ключевых слов, которые обозначают ее конец. C++ использует довольно простую схему организации программы. Эта схема поддерживает два уровня кода: глобальный и уровень функций. Кроме того, функция main, определяемая со строки 5, играет очень специфическую роль, потому что выполнение программы C++ всегда начинается с этой функции. Следовательно, в программе может быть только одна функция main. Вы можете располагать функцию main в любом месте программы.
- Строки и символы C++ заключаются соответственно в двойные и одиночные кавычки. Таким образом, 'A' является одиночным символом, в то время как "A" — строка, состоящая всего из одного символа. Смешивание в C++ односимвольных строк и символов запрещено.

Строки могут содержать любое число символов, в том числе ни одного. Строка, не имеющая символов, называется пустой строкой.

- C++ определяет операторные блоки, ограниченные символами { и }. См. строки с 6-ой по 9-ю соответственно.
- Каждый оператор в программе C++ должен заканчиваться точкой с запятой (;). При просмотре программ может показаться, что имеются исключения из этого правила, но это не так. Мы обсудим это позже.
- Программы на C++ содержат директиву препроцессора #include. Пример этому

можно найти в строке 3, в которой компилятору Borland C++ дается указание включить файл заголовка IOSTREAM.H в текст программы. В C++ расширено понятие потоков, которые уже существовали в языке C. IOSTREAM.H обеспечивает операции, которые поддерживают базовый потоковый ввод и вывод. C++ не имеет встроенных операций ввода / вывода. Вместо этого язык полагается на библиотеки, специализирующиеся в различных типах ввода / вывода.

Директива компилятора — одна из специальных команд компилятора, которые будут описаны в разделе «Препроцессор».

Файл заголовка содержит объявления констант, типов данных, переменных и прототипов функций.

Поток — последовательность данных, передающихся из одной части компьютера в другую.

- Программа выводит строку "Hello Programmer!" в стандартный поток вывода cout, который является окном EasyWin. При этом программа использует операцию вывода «, направляющую выводимую строку в выходной поток.
- Функция main должна возвращать значение, которое отражает состояние программы C++. Возвращаемое значения 0 сообщает операционной системе о том, что программа завершилась без ошибок.

Выход из IDE

Чтобы выйти из IDE, достаточно выбрать команду Exit из меню File.

Заключение

За сегодняшний урок вы познакомились с огромным количеством материала, касающегося IDE Borland C++ и составили первую программу на C++. Вы должны были усвоить следующие основные пункты:

- Программы C++ разбиты на модули и используют стандартные и заказные библиотеки.
- Справочная информация доступна через систему справки, которая активируется при помощи меню IDE, при помощи всплывающего меню, когда нажимается правая кнопка мыши, или нажатием F1 либо Ctrl + F1.
- Имеются два способа загрузки IDE Borland C++. Вы можете выбрать пиктограмму Borland C++ из меню Start Windows 95 или дважды щелкнуть на файле BCW.EXE.
- IDE Borland C++ — универсальная среда для разработки, поддержки и отладки программ на C и C++ и библиотек для приложений Windows.
- Меню File управляет созданием новых файлов, открытием файлов, сохранением файлов, печатью и выходом из IDE.
- Меню Edit предлагает команды выполнения обычных операций редактирования, таких как отмена, вырезка, копирование и вставка.
- Меню Search дает вам возможность найти и заменить фрагменты текста, просматривать символы, находить функции и строки исходного текста, вызвавшие сообщения об ошибке.
- В меню View можно просмотреть большое количество разнообразной информации. В том числе узлы проекта, сообщения транслятора и компоновщика, иерархию классов проекта, глобальные символы, контролируемые отладчиком переменные, стек вызовов функций и регистры процессора.
- Меню Project содержит команды создания, открытия, закрытия и управления проектом. Эти команды дают вам возможность компилировать и связывать файлы исходных текстов.
- В Меню Debug собраны команды отладки, пошагового выполнения программы и наблюдения значений переменных в окне Watch.
- Меню Tool открывает вам быстрый доступ к ряду инструментальных средств Windows, таких, как Turbo Debugger, утилита трассировки сообщений WinSight и добавленных

вами инструментов.

- Меню Options настраивает различные параметры вашего проекта — среду, инструментальные средства и листы стиля проекта.
- Меню Window управляет, перераспределяет, закрывает и восстанавливает окна на рабочем столе IDE.
- Меню Help — меню интерактивной справки.
- Прикладные программы Easy Win — это прикладные программы Windows, выполняющиеся в специальном окне, которое имитирует стандартные устройства ввода/вывода. EasyWin-приложения дают вам возможность писать программы в стиле DOS.
- Новые проекты легко создавать, устанавливая параметры проекта в простом диалоговом окне.
- Файлы проекта Borland C++ позволяют создавать несколько целей, которые могут содержать по нескольку файлов.
- Ваша первая программа C++ — простая программа приветствия — иллюстрирует применение основных компонентов программы C++. Эти компоненты включают комментарии, директиву #include и функцию main.
- Для выхода из IDE нужно выбрать команду Exit из меню File.

Вопросы и ответы

Нумеруются ли строки в программе на C++?

Нет. Я использую номера строк в листингах программ в этой книге только для того, чтобы можно было ссылаться на строки программы в тексте. Если вы набираете код программ вручную, то для контроля имеет смысл сравнивать номер текущей строки, который вам сообщает редактор, с номером строки в листинге программы.

Контролирует ли редактор IDE вводимый текст?

Да, контролирует. Когда вы вводите ключевое слово C++, IDE выделяет это ключевое слово цветом или жирным шрифтом.

Что произойдет, если я забуду напечатать вторую двойную кавычку в нашей первой программе?

Компилятор сообщит вам об ошибке в программе. Вы должны добавить вторую двойную кавычку и собрать проект.

Как удалить все вхождения некоторого текста в окне редактирования?

Используйте команду Replace в меню Edit и ничего не определяйте для строки замены, или используйте команды Cut и Clear в том же меню Edit.

Практикум

Наш практикум предлагает вам проверочные вопросы для закрепления материала и практического использования пройденного. Не переходите к новой главе книги, не разобравшись с вопросами практикума и ответами на них. Ответы можно найти в приложении A.

Контрольные вопросы

1. Что выведет на экран следующая программа?

```
1:    // Вопрос #1
2:
3:    #include <iostream.h>
4:
5:    int main()
6:    {
7:          cout << "C++ in 21 days?";
```

```
8:          return 0;
9:      }
```

2. Что выведет на экран следующая программа?

```
1:      // Вопрос #2
2:
3:      #include <iostream.h>
4:
5:       int main()
6:      {
7:          // cout << "C++ in 21 days?";
8:          return 0;
9:      }
```

3. Найдите ошибку в следующей программе.

```
1:      // Вопрос #3
2:
3:      #include <iostream.h>
4:
5:      int main()
6:      {
7:          cout << "C++ in 21 days?"
8:          return 0;
9:      }
```

Упражнение

Составьте программу, выводящую на экран сообщение “I am a C++ Programmer”.