

РЕКОМЕНДОВАНО  
Sun Microsystems

В. Монахов

# Язык программирования Java и среда NetBeans

2-е издание

Основные концепции и методы  
объектно-ориентированного  
программирования

Основные концепции UML-проектирования

Программирование мобильных устройств

Написание серверного программного обеспечения

Численные расчеты с использованием Java

Программы для систем  
с многоядерными процессорами

Desktop Application – новый вид программ  
с графическим интерфейсом



+ <sup>cd</sup>

Дистрибутивы  
JDK 6 и NetBeans 6.5  
для Windows и Linux

**В. Монахов**

**Язык  
программирования  
Java и среда  
NetBeans**

**2-е издание**

Санкт-Петербург

«БХВ-Петербург»

2009

УДК 681.3.068  
ББК 32.973.26-018.1  
М77

**Монахов В. В.**

М77 Язык программирования Java и среда NetBeans. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2009. — 720 с.: ил. + CD-ROM

ISBN 978-5-9775-0424-9

Книга написана на базе курса лекций, читаемых автором на кафедре вычислительной физики Санкт-Петербургского государственного университета. Второе издание ориентировано на использование среды NetBeans 6.5 и дополнено новыми материалами. Изложены основные синтаксические конструкции Java, принципы объектно-ориентированного программирования, особенности проведения численных расчетов. Приводятся сведения о разработке основных категорий программного обеспечения Java (ME, SE и EE). Рассказывается о создании сетевых приложений и приложений для мобильных устройств. Разбираются методики написания многопоточных приложений Java для систем с многоядерными процессорами. Материал сопровождается большим количеством примеров с подробным анализом их исходных кодов. На компакт-диске находятся дистрибутивы JDK 6u11 и NetBeans 6.5 для Windows и Linux, а также исходные тексты примеров.

*Для начинающих Java-программистов и студентов*

УДК 681.3.068  
ББК 32.973.26-018.1

**Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Леонид Кочин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 27.02.09.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 58,05.

Тираж 2000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.003650.04.08 от 14.04.2008 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов  
в ГУП "Типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0424-9

© Монахов В. В., 2009  
© Оформление, издательство "БХВ-Петербург", 2009

# Оглавление

- Введение ..... 11**
- Глава 1. Общие представления о языке Java..... 15**
  - 1.1. Java и другие языки программирования. Системное и прикладное программирование..... 15
  - 1.2. Виртуальная Java-машина, байт-код, JIT-компиляция. Категории программ, написанных на языке Java.....20
  - 1.3. Алфавит языка Java. Десятичные и шестнадцатеричные цифры и целые числа. Резервированные слова .....26
    - Алфавит языка Java .....27
    - Десятичные и шестнадцатеричные цифры и целые числа.....27
    - Резервированные слова и литералы языка Java.....29
  - 1.4. Управляющие последовательности. Символы Unicode. Специальные символы .....30
    - Управляющие последовательности.....30
      - Управляющие последовательности — символы формирования текста .....30
      - Управляющие последовательности — символы Unicode .....30
      - Простые специальные символы .....31
      - Составные специальные символы.....32
  - 1.5. Идентификаторы. Переменные и типы. Примитивные и ссылочные типы.....33
  - 1.6. Процедурное и объектно-ориентированное программирование. Инкапсуляция .....37
  - 1.7. Работа со ссылочными переменными. Сборка мусора.....41
  - 1.8. Проекты. Пакеты. Уровни видимости классов. Импорт классов .....46
  - 1.9. Базовые пакеты и классы Java .....50
  - 1.10. Технологии Java, .NET, ASP, PHP .....53
  - 1.11. Среды разработки NetBeans, Eclipse, JDeveloper, JBuilder, IntelliJ IDEA .....58
    - NetBeans 6.5 .....59
    - Eclipse 3.4.1 (Ganymede) .....61
    - JDeveloper 11 .....63
    - JBuilder 2008 .....66
    - IntelliJ IDEA 7.0.4 .....68
    - Общий итог сравнения сред разработки.....68
  - Краткие итоги по главе.....69
  - Типичные ошибки.....72
  - Задания.....72

<b>Глава 2. Среда NetBeans 6.5 .....</b>	<b>73</b>
2.1. Установка JDK 6 .....	73
2.2. Установка среды NetBeans 6.5 .....	77
2.3. Создание в NetBeans простейшего приложения Java .....	85
2.4. Компиляция файлов проекта и запуск приложения. Jar-файлы .....	91
2.5. Структура проекта NetBeans .....	94
2.6. Создание в NetBeans приложения Java с графическим интерфейсом .....	98
2.7. Редактор экранных форм .....	103
2.8. Свойства компонентов .....	109
2.9. Внешний вид приложения — технология Look and Feel .....	115
2.10. Ведение проектов .....	118
2.11. Редактирование меню экранной формы .....	120
2.12. Создание нового класса .....	129
2.13. Документирование исходного кода в Java .....	130
2.14. Основные компоненты пакетов <i>swing</i> и <i>awt</i> .....	139
2.15. Менеджеры размещения ( <i>Layout</i> ) и якоря ( <i>Anchor</i> ) .....	144
2.16. Создание приложения Desktop Application .....	145
Краткие итоги по главе .....	147
Типичные ошибки .....	148
Задания .....	148
 <b>Глава 3. Примитивные типы данных и операторы для работы с ними ....</b>	 <b>151</b>
3.1. Булевый (логический) тип .....	151
3.2. Целые типы, переменные, константы .....	152
3.3. Основные операторы для работы с целочисленными величинами .....	156
3.4. Вещественные типы и класс <i>Math</i> .....	157
3.5. Правила явного и автоматического преобразования типа при работе с числовыми величинами .....	161
3.6. Оболочечные классы. Упаковка ( <i>boxing</i> ) и распаковка ( <i>unboxing</i> ) .....	163
3.7. Приоритет операторов .....	165
Краткие итоги по главе .....	166
Типичные ошибки .....	168
Задания .....	168
 <b>Глава 4. Работа с числами в языке Java .....</b>	 <b>171</b>
4.1. Двоичное представление целых чисел .....	171
Позиционные и непозиционные системы счисления .....	171
Двоичное представление положительных целых чисел .....	173
Двоичное представление отрицательных целых чисел. Дополнительный код .....	175
Сложение положительного и отрицательного чисел .....	175
Сложение отрицательных чисел .....	176
Проблемы целочисленной машинной арифметики.	
Целочисленное переполнение .....	176
Сложение положительных чисел .....	176
Умножение положительных чисел .....	177

Шестнадцатеричное представление целых чисел и перевод из одной системы счисления в другую .....	179
Преобразование чисел из системы с меньшим основанием в систему с большим основанием .....	180
Преобразование чисел из системы с большим основанием в систему с меньшим основанием .....	181
Преобразование чисел в системах счисления с кратными основаниями .....	181
4.2. Побитовые маски и сдвиги .....	182
4.3. Двоичное представление вещественных чисел .....	185
Двоичные дроби .....	185
Мантисса и порядок числа .....	185
Научная нотация записи вещественных чисел .....	187
Стандарт IEEE 754 представления чисел в формате с плавающей точкой .....	188
Краткие итоги по главе .....	193
Типичные ошибки .....	194
Задания .....	194
<b>Глава 5. Управляющие конструкции .....</b>	<b>197</b>
5.1. Составной оператор .....	197
5.2. Условный оператор <i>if</i> .....	198
5.3. Оператор выбора <i>switch</i> .....	202
5.4. Условное выражение <i>...?... :</i> .....	203
5.5. Операторы инкремента <i>++</i> и декремента <i>--</i> .....	204
5.6. Оператор цикла <i>for</i> .....	204
5.7. Ошибки при использовании вещественного счетчика цикла .....	206
5.8. Эффективная организация циклов при вычислениях в формате с плавающей точкой .....	209
5.9. Особенности целочисленных вычислений — организация циклов, приоритет операторов и арифметическое переполнение .....	214
5.10. Оператор цикла <i>while</i> — цикл с предусловием .....	219
5.11. Оператор цикла <i>do...while</i> — цикл с постусловием .....	221
5.12. Операторы прерывания <i>continue</i> , <i>break</i> , <i>return</i> , <i>System.exit</i> .....	221
Краткие итоги по главе .....	225
Типичные ошибки .....	226
Задания .....	226
<b>Глава 6. Начальные сведения об объектном программировании .....</b>	<b>227</b>
6.1. Наследование и полиморфизм .....	227
6.2. Функции. Модификаторы. Передача примитивных типов в функции .....	236
6.3. Локальные и глобальные переменные. Модификаторы доступа и правила видимости. Ссылка <i>this</i> .....	239
6.4. Передача ссылочных типов в функции. Проблема изменения ссылки внутри подпрограммы .....	242
6.5. Наследование. Суперклассы и подклассы. Переопределение методов .....	249
6.6. Наследование и правила видимости. Зарезервированное слово <i>super</i> .....	255

6.7. Статическое и динамическое связывание методов. Полиморфизм .....	257
6.8. Базовый класс <i>Object</i> .....	259
6.9. Конструкторы. Резервированные слова <i>super</i> и <i>this</i> . Блоки инициализации.....	261
6.10. Удаление неиспользуемых объектов и метод <i>finalize</i> . Проблема деструкторов для сложно устроенных объектов.....	264
6.11. Перегрузка методов .....	265
6.12. Правила совместимости ссылочных типов как основа использования полиморфного кода. Приведение и проверка типов .....	268
Краткие итоги по главе.....	271
Типичные ошибки.....	273
Задания.....	274

## **Глава 7. UML-диаграммы. Прямое и обратное проектирование.**

<b>Рефакторинг.....</b>	<b>275</b>
7.1. UML-диаграммы .....	275
7.2. Структура окон проектов с UML-диаграммами.....	284
7.3. Панель инструментов проектов с UML-диаграммами.....	287
7.4. Прямое проектирование (Forward engineering) — построение кода классов по UML-диаграммам .....	291
7.5. Обратное проектирование (Reverse engineering) — построение UML-диаграмм по разработанным классам .....	298
7.6. Рефакторинг .....	304
Первый случай — переименование элементов программы .....	306
Второй случай — перемещение класса .....	308
Третий случай — копирование класса для его дальнейшей модификации .....	308
Четвертый случай — безопасное удаление элемента программы.....	309
Пятый случай — изменение списка параметров метода.....	309
Шестой случай — инкапсуляция полей данных .....	310
Краткие итоги по главе.....	311
Типичные ошибки.....	311
Задания.....	312

## **Глава 8. Важнейшие объектные типы..... 313**

8.1. Массивы.....	313
8.2. Коллекции, списки, итераторы .....	319
8.3. Перебор в цикле элементов коллекций. Оператор цикла <i>for-each</i> .....	323
8.4. Работа со строками в Java. Строки как объекты. Классы <i>String</i> , <i>StringBuffer</i> и <i>StringBuilder</i> .....	324
8.5. Типы-перечисления (enum).....	338
8.6. Работа с датами и временем. Класс <i>GregorianCalendar</i> .....	340
8.7. Работа с графикой и графическим пользовательским интерфейсом.....	343
Код визуализации и код бизнес-логики приложения.....	343
Графические примитивы.....	344
Пример метода, работающего с графикой .....	347
8.8. Исключительные ситуации .....	350
Обработка исключительных ситуаций.....	350

Иерархия исключительных ситуаций .....	353
Объявление типа исключительной ситуации и оператор <i>throw</i> .....	356
Объявление метода, который может возбуждать исключительную ситуацию. Зарезервированное слово <i>throws</i> .....	358
8.9. Работа с файлами и папками .....	361
Работа с файлами и папками с помощью объектов типа <i>File</i> .....	361
Выбор файлов и папок с помощью файлового диалога .....	365
Работа с потоками ввода/вывода .....	369
Краткие итоги по главе .....	377
Типичные ошибки .....	379
Задания .....	379
<b>Глава 9. Интерфейсы и композиция .....</b>	<b>381</b>
9.1. Проблемы множественного наследования классов. Интерфейсы .....	381
9.2. Отличия интерфейсов от классов. Проблемы наследования интерфейсов .....	384
9.3. Пример работы с интерфейсами .....	386
9.4. Композиция — еще одна альтернатива множественному наследованию .....	389
Краткие итоги по главе .....	391
Типичные ошибки .....	391
Задания .....	391
<b>Глава 10. Многопоточное программирование и многоядерные системы ....</b>	<b>393</b>
10.1. Потоки выполнения (threads) и синхронизация .....	393
10.2. Преимущества и проблемы при работе с потоками выполнения .....	394
10.3. Синхронизация по ресурсам и событиям .....	395
Синхронизация по ресурсам .....	395
Синхронизация по событиям .....	397
10.4. Класс <i>Thread</i> и интерфейсы <i>Runnable</i> и <i>Callable</i> . Создание и запуск потока выполнения .....	398
10.5. Поля и методы, заданные в классе <i>Thread</i> .....	400
10.6. Работа многопоточных приложений в многопроцессорных и многоядерных системах .....	402
Причины перехода к многопроцессорным и многоядерным системам .....	402
Пример многопоточной программы .....	404
Работа многопоточного приложения на многоядерном компьютере .....	412
10.7. Синхронизация на основе интерфейсов <i>Lock</i> и <i>Condition</i> .....	414
Интерфейс <i>Lock</i> .....	414
Интерфейс <i>Condition</i> .....	418
Краткие итоги по главе .....	420
Типичные ошибки .....	421
Задания .....	422
<b>Глава 11. Введение в сетевое программирование .....</b>	<b>423</b>
11.1. Краткая справка по языку HTML .....	423
Система WWW и язык HTML .....	423



Теги и их атрибуты .....	425
Математические и специальные символы, греческие буквы .....	431
11.2. Апплеты .....	433
Структура апплета .....	433
Примеры апплетов. Аннотация <i>@Override</i> .....	436
Создание проекта с апплетами .....	440
Редактор HTML-документов, стилей CSS и кода JavaScript.....	443
11.3. Сервлеты .....	448
Создание сервлета и конфигурирование сервера приложений.....	450
Локализация сервлетов — работа с языковыми форматами.....	457
11.4. Технология JSP — Java Server Pages.....	462
11.5. Технология JSTL — JSP Standard Tag Library.....	467
Краткие итоги по главе.....	469
Типичные ошибки.....	470
Задания.....	470

## **Глава 12. Вложенные классы .....** **471**

12.1. Виды вложенных классов.....	471
12.2. Статические (static) вложенные классы и интерфейсы.....	472
12.3. Внутренние (inner) классы .....	474
12.4. Локальные (local) классы .....	477
12.5. Анонимные (anonymous) классы и обработчики событий .....	477
12.6. Анонимные (anonymous) классы и слушатели событий (listeners) .....	478
Краткие итоги по главе.....	481
Типичные ошибки.....	483
Задания.....	483

## **Глава 13. Приложение с графическим интерфейсом —**

### ***DesktopApplication* .....** **485**

13.1. Структура простой заготовки <i>DesktopApplication</i> .....	485
13.2. Исходный код класса <i>DesktopApplication</i> .....	487
13.3. Исходный код класса <i>DesktopApplicationView</i> .....	489
13.4. Конструктор главной формы приложения. Ресурсы приложения и концепция управляющих синглетонов.....	493
13.5. Аннотация <i>@Action</i> и показ справки.....	498
13.6. Концепция управляющего объекта-модели. Модель <i>TaskMonitor</i> , класс <i>Task</i> и обработчик <i>propertyChange</i> .....	500
13.7. Создание задачи типа <i>Task</i> . Назначение иконок и добавление изображений.....	507
13.8. Локализация приложения.....	512
13.9. Измерение времени и досрочное прекращение выполнения задания .....	518
13.10. Работа с межпрограммным буфером обмена — Clipboard .....	519
Копирование строк и изображений из буфера обмена в программу.....	519
Копирование строк из программы в буфер обмена. Класс <i>StringSelection</i> .....	521
Копирование изображений из программы в буфер обмена. Создание класса <i>ImageSelection</i> .....	523
13.11. Многопоточная система вычислений и индикации .....	527

Краткие итоги по главе.....	531
Типичные ошибки.....	532
Задания.....	532

## **Глава 14. Программирование мобильных телефонов..... 533**

14.1. Спецификация Java Micro Edition. Конфигурации и профили. Мидлеты .....	533
14.2. Создание приложений для мобильных устройств.....	534
14.3. Дизайнер пользовательского интерфейса мидлета .....	540
14.4. Исходный код мидлета <i>Hello, World!</i> .....	546
14.5. Заготовка мидлета "Крестики-нолики" .....	555
14.6. Код бизнес-логики мидлета .....	564
14.7. Создание собственного компонента, являющегося наследником <i>TableItem</i> , и мидлета на его основе. Создание библиотеки .....	571
14.8. Запуск мидлета "Крестики-нолики" на реальном телефоне.....	579
14.9. Пример мидлета с переключением экрана и списком выбора.....	581
14.10. Примеры приложений для мобильных телефонов, поставляемые со средой NetBeans .....	582
Краткие итоги по главе.....	586
Типичные ошибки.....	587
Задания.....	587

## **Глава 15. Компонентное программирование..... 589**

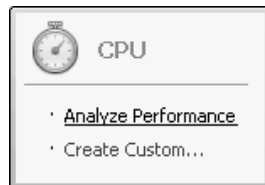
15.1. Компонентная архитектура JavaBeans .....	589
15.2. Создание компонента в NetBeans 6.5 .....	590
15.3. Пример создания компонента в NetBeans 6.5 — панель с заголовком .....	594
15.4. Добавление в компонент новых свойств .....	598
15.5. Добавление в компонент новых событий .....	600
Краткие итоги по главе.....	603
Типичные ошибки.....	605
Задания.....	605

## **Глава 16. Платформо-зависимые методы и модуль C/C++ Pack..... 607**

16.1. Использование динамических библиотек. Объявление платформо-зависимых (native) методов.....	607
16.2. Интерфейс JNI (Java Native Interface) взаимодействия Java с C/C++. Соответствие типов Java и C++ .....	608
16.3. Пакет C/C++ Pack. Подключение компиляторов C++, C и библиотек.....	614
16.4. Создание приложений C++ и C .....	624
16.5. Примеры приложений с native-методами .....	628
Первый этап. Создание исходного кода Java для native-методов.....	628
Второй этап. Генерация h-файла заголовка C++, содержащего декларацию методов .....	629
Третий этап. Написание на C++ реализации методов, заголовки которых заданы в сгенерированном файле <code>jnipackage_JniMultiply.h</code> .....	630
Четвертый этап. Компиляция кода C++ в динамическую библиотеку (DLL) .....	631

Пятый этап. Запуск приложения вместе с DLL в NetBeans .....	632
Шестой этап. Распространение приложения вместе с DLL .....	632
Краткие итоги по главе.....	637
Типичные ошибки.....	637
Задания.....	638
<b>Глава 17. Отладка, тестирование и профилирование программ .....</b>	<b>639</b>
17.1. Отладка приложений в среде NetBeans.....	639
17.2. Команды режима пошагового выполнения .....	643
17.3. Пример отладки неправильно работающего приложения.....	650
17.4. Принцип "презумпции виновности" и тестирование классов .....	654
17.5. Профилирование приложений .....	668
17.6. Профилирование отдельных методов .....	677
17.7. Настройка профилировщика "на лету". Точки профилирования .....	684
17.8. Нахождение проблем с использованием памяти — телеметрия.....	692
17.9. Профилирование использования памяти классами.....	695
Краткие итоги по главе.....	701
Типичные ошибки.....	702
Задания.....	702
<b>Заключение .....</b>	<b>703</b>
<b>Описание прилагаемого компакт-диска.....</b>	<b>705</b>
<b>Литература .....</b>	<b>711</b>
<b>Предметный указатель .....</b>	<b>713</b>

## ГЛАВА 2



# Среда NetBeans 6.5

NetBeans — это интегрированная среда разработки (Integrated Development Environment, IDE), т. е. такая программа (или набор программ в одной оболочке), которая позволяет пользователю решать практически все задачи в его проблемной области без выхода из этой программы. Все современные средства разработки программного обеспечения представляют собой интегрированные среды.

До появления интегрированных сред разработки приходилось набирать программы в текстовом редакторе, запускать компилятор командной строки и читать его сообщение о месте предполагаемой ошибки компиляции, после чего в текстовом редакторе искать нужное место исходного кода, исправлять ошибку, читать следующее сообщение об ошибке и т. д.

Интегрированные среды позволяют сразу перейти в место с предполагаемой ошибкой, а некоторые наиболее продвинутые (NetBeans относится к ним) подчеркивают ошибки красной волнистой линией непосредственно в процессе редактирования исходного кода программы.

## 2.1. Установка JDK 6

Комплект программного обеспечения, находящийся на прилагаемом к книге компакт-диске, в первую очередь рассчитан на пользователей Windows, однако работа с NetBeans возможна и из других сред. В частности, для пользователей Linux предназначены файлы, в названии которых присутствует слово *linux* (см. описание компакт-диска).

Требования к аппаратуре у новых версий программного обеспечения, как обычно, повышенные. Среда NetBeans 6.5 с открытым простейшим приложением занимает в памяти примерно 353 Мбайт. Из них на ноутбуке автора (с Windows XP SP3 и 1,5 Гбайт оперативной памяти) примерно 176 Мбайт располагаются в оперативной памяти, а 177 Мбайт операционная система хранит в виртуальной памяти (в файле подкачки на жестком диске). При этом файлы самой среды занимают в памяти всего около 7 Мбайт, а остальную память потребляет виртуальная Java-машина (процесс *java.exe*).

Работа с виртуальной памятью примерно в 1000 раз медленнее, чем с оперативной, и ее использование заметно замедляет процесс в тех случаях, когда операционная система вынуждена к ней обращаться. Исходя из личного опыта, автор делает вывод, что

512 Мбайт оперативной памяти можно рассматривать как нижний предел, с которым еще можно работать в среде NetBeans 6.5. На компьютерах, где памяти заметно меньше, реально работать нельзя.

Требования к тактовой частоте процессора в данном случае не столь существенны — процессоры Pentium 4 1,6 ГГц и Celeron 2,4 ГГц вполне справлялись с работой.

На прилагаемом к книге компакт-диске содержится дистрибутив среды NetBeans 6.5. Для его установки в папке NetBeans 6.5 в операционной системе Windows следует запустить файл `netbeans-6.5-ml-windows.exe`.

Но предварительно требуется установить JDK. При отсутствии на компьютере необходимой версии JDK (5 или 6) выдается сообщение, показанное на рис. 2.1.



**Рис. 2.1.** Сообщение, выдаваемое в случае, когда необходимая версия JDK на компьютере не установлена

Следует нажать кнопку **Exit Installer** (Выйти из программы установки) и запустить с диска файл `jdk-6u11-windows-i586-p.exe`.

По функциональности JDK 5 и JDK 6 примерно одинаковы, при установке полных объемов дистрибутивов заметных изменений в версии 6 по сравнению с 5-й нет. Но предпочтительнее ставить JDK 6 даже в том случае, если на компьютере уже имеется JDK 5, поскольку версия 6 обеспечивает более высокую скорость работы приложений, лучшую многопоточность (в частности, на многоядерных процессорах), меньшее время старта Java-машины, дополнительные возможности управления режимами Java-машины и обладает большей устойчивостью. Кроме того, начиная с JDK 6 update 10 имеется возможность при появлении очередных обновленных версий JDK не переставлять весь дистрибутив, а устанавливать только обновления. При этом нет необходимости удалять предыдущие версии JDK и JRE — они могут работать на компьютере одновременно. Хотя при запуске приложений Java возможны проблемы из-за наличия пути к неправильной версии исполняющей среды.

JDK 6 рассчитана на достаточно новые версии операционных систем. Например, для MS Windows поддерживаются Windows 2000 SP4 (пакет обновлений версии 4) и выше, XP SP2, Vista. При попытке установить JDK для более старой версии (например, XP SP1) выдается предупреждение о том, что требуется более новая версия операционной системы, но можно на свой страх и риск продолжить установку и практически все будет работать правильно (но без гарантий Sun). Если же вы захотите работать не только с NetBeans, но и с JDeveloper 11, необходима XP SP3 и достаточно новая версия JDK 6

(update 7 или более поздний). Для установки JDK зайдите в папку jdk1.6 и запустите файл jdk-6u11-windows-i586-p.exe.

На короткое время появится окно с предложением подождать, пока идет подготовка к процессу установки, после чего появится окно с лицензией, предлагаемой Sun (рис. 2.2).



Рис. 2.2. Окно с лицензией

### Замечание

Указанные на рисунках цифры соответствуют JDK 1.6u11 (JDK 6 обновление 11). На компакт-диске может оказаться более новая версия JDK, поэтому внешний вид изображений на экране может несколько отличаться от приводимых в книге. Указанная далее информация по объему дискового пространства, требуемого для модулей NetBeans 6.5, также приближительна.

Нажатие на кнопку **Accept >** (Принять) означает ваше согласие с данной лицензией и начало установки JDK. Нажатие на кнопку **Decline** (Отклонить) — отказ от продолжения установки. В лицензии говорится о том, что Sun дает право на бесплатное использование программ JDK в целях запуска Java-программ при условии, что не будет проводиться модификация кодов программ и текстов документации JDK, а также о том, что программное обеспечение поставляется в соответствии с принципом "AS IS" (как есть), без какой-либо ответственности Sun за применение этого программного обеспечения пользователем.

После нажатия кнопки **Accept >** появляется диалог выбора устанавливаемых элементов JDK (рис. 2.3). Рекомендуются установить их все. Это займет примерно  $300 + 46 + 49 + 173 + 25 = 593$  Мбайт дискового пространства. В крайнем случае, если дискового пространства не хватит, можно установить только Public JRE (исполняющую среду Java для приложений — 173 Мбайт). Но лучше все-таки установить JDK на другой диск (кнопка **Change...** — изменить место установки) или просто удалить ненужные программы с рабочего диска.

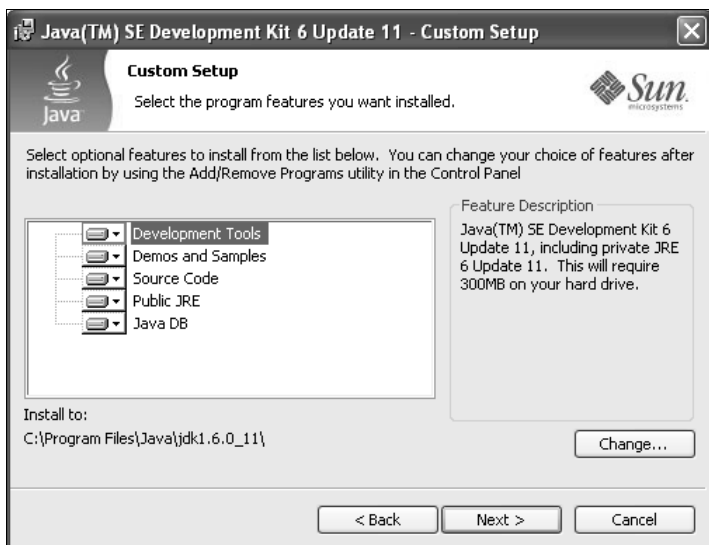


Рис. 2.3. Диалог выбора устанавливаемых элементов JDK

Нажатие на кнопку **Next >** (Далее, следующий) продолжает установку. По очереди появляются сообщения о выполняемых при установке действиях (это занимает 2—3 минуты), после чего открывается окно с предложением выбрать папку (рис. 2.4), где будет установлена исполняющая среда JRE 6. Напомним, что это комплект JRE, предназначенный для обычного выполнения, а не тот вариант JRE, который входит в состав JDK и служит для отладки при разработке приложений.

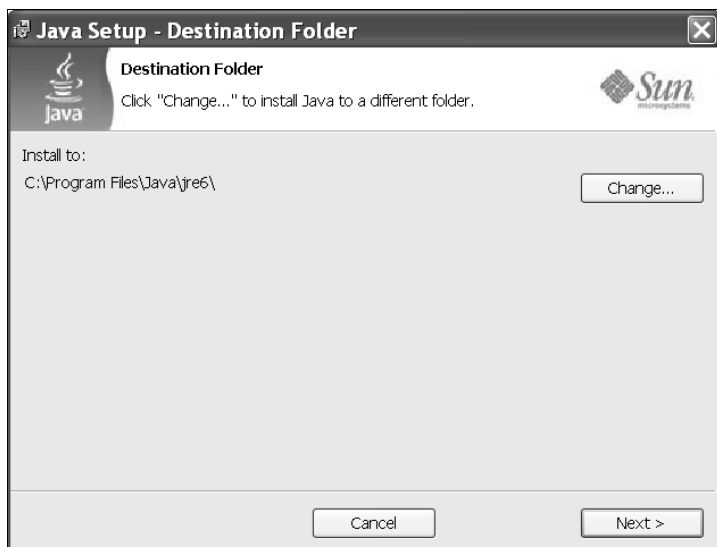


Рис. 2.4. Окно с предложением о смене папки, в которую устанавливается JRE

Нажимаем кнопку **Next >** и продолжаем установку. Наконец, через 3—4 минуты появляется сообщение, показанное на рис. 2.5, означающее успешное завершение установки.

Для завершения установки следует нажать кнопку **Finish** (Завершить).

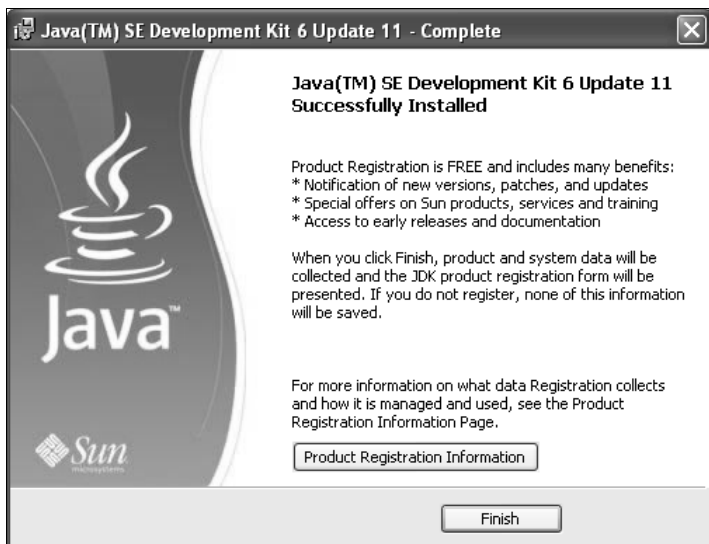


Рис. 2.5. Сообщение об успешном завершении установки JDK

## 2.2. Установка среды NetBeans 6.5

Как уже говорилось, для установки NetBeans IDE следует запустить файл `netbeans-6.5-ml-windows.exe`. Напомним, что IDE означает "интегрированная среда разработки".

Сначала появится окно, показывающее прогресс в конфигурации инсталлятора (рис. 2.6).

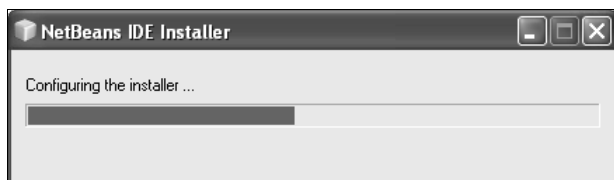


Рис. 2.6. Окно, показывающее прогресс в конфигурации инсталлятора

Это не очень быстрый процесс, придется подождать 2—3 минуты до тех пор, пока не появится следующее окно (рис. 2.7), в котором необходимо нажать кнопку **Customize...**, после чего откроется окно для выбора устанавливаемых модулей (рис. 2.8).

### Замечание

Даже если у вас большой жесткий диск, не следует устанавливать ненужные модули. Особенно это относится к исполняющим средам (runtime) — после установки они по



умолчанию будут автоматически запускаться каждый раз при включении компьютера, замедляя загрузку операционной системы и отнимая у компьютера много оперативной памяти.

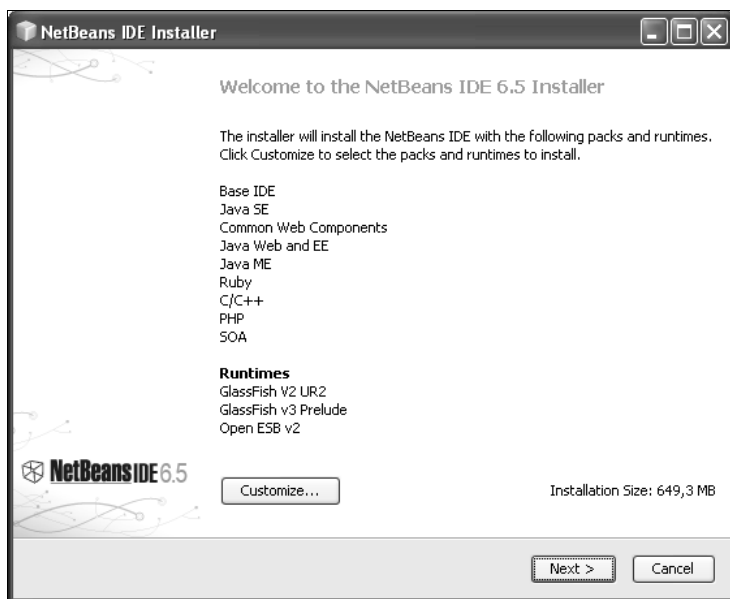


Рис. 2.7. Окно, в котором показываются устанавливаемые модули

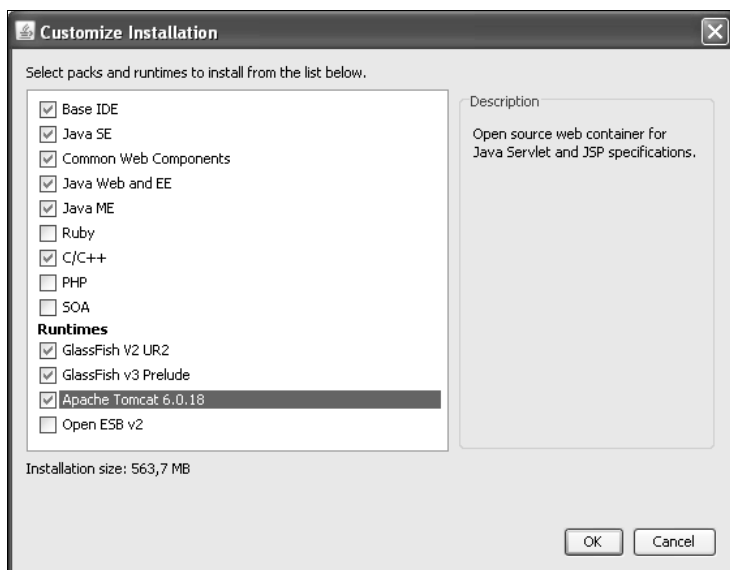


Рис. 2.8. Окно выбора устанавливаемых модулей

Перечислим пункты выбора в данном окне (в скобках приведен перевод английского текста, поясняющего соответствующую опцию установки, а затем даны комментарии по использованию модуля в рамках данного учебного курса).

- ◆ **Base IDE.** 70,8 Мбайт. (Основные компоненты интегрированной среды разработки.) Платформа NetBeans без модулей расширения.

Данный модуль следует оставить отмеченным как необходимый.

- ◆ **Java SE.** 63,9 Мбайт. (Основные инструменты для программирования на языке Java.) Интегрированная среда разработки Java. Включает редактор, отладчик, профилировщик, поддержку рефакторинга и инструменты проектирования для создания графического пользовательского интерфейса (ранее известные как Matisse).

Модуль расширения платформы NetBeans для использования языка Java. Следует оставить отмеченным как необходимый.

- ◆ **Common Web Components.** 5,2 Мбайт. (Компоненты, используемые всеми веб-ориентированными проектами для Java Web, PHP, Ruby on Rails, Grails. Поддержка работы с библиотеками JavaScript).

Данный модуль следует оставить отмеченным как необходимый.

- ◆ **Java Web and EE.** 110,1 Мбайт. (Инструменты для создания WWW-приложений Java, а также корпоративных (серверных) приложений, совместимых со спецификациями J2EE 1.4 и Java EE 5.) Включает поддержку сервлетов, приложений JSP (Java Server Pages), библиотек тегов, Spring (библиотека с набором "легких" переносимых оболочечных компонентов для всех основных технологий Java EE — JDBC, ORM, JTA, Servlets/Jsp, JMX и др.), Struts ("распорки" — технология для создания WWW-приложений), Java Persistence API (поддержка технологии Java Persistence — устойчивость Java), Enterprise Java Beans (серверные компоненты Java Beans), JAX-WS (веб-службы JAX. Аббревиатура JAX расшифровывается как Java API for XML-Based Remote Procedure Calls — программируемый пользовательский интерфейс для основанных на XML удаленных вызовах процедур), RESTful (веб-службы, использующие принцип определения ресурсов по универсальным идентификаторам ресурсов URI), Java Server Faces (лица сервера Java), языка Groovy, комплекса Grails. Обеспечивает окружение для визуального проектирования пользовательского интерфейса соответствующих приложений.

Модуль следует оставить отмеченным как необходимый. Он нужен для работы с серверными технологиями Java (см. главу 11).

- ◆ **Java ME.** 100,0 Мбайт. (Всеобъемлющий набор инструментов для создания приложений Java Micro Edition для сотовых телефонов и других портативных устройств.) Включает визуальный дизайнер, мастер веб-соединений и инструменты для решения проблем фрагментации памяти устройств.

Модуль следует оставить отмеченным как необходимый. Он нужен для работы с технологиями Java для мобильных телефонов и других тонких аппаратных клиентов (см. главу 14).

- ◆ **Ruby.** 38,3 Мбайт. (Инструментальные средства для программирования серверных приложений на языке Ruby (Рубин) и Ruby on Rails.) Включает интерпретатор JRuby,

исполняющую среду (виртуальную машину) Rails, мощный редактор, отладчик, "менеджер драгоценного камня", интерактивную оболочку Ruby.

В рамках данного учебного курса не требуется.

- ♦ **C/C++**. 13,4 Мбайт. (Инструменты для создания приложений C и C++.) Включает заготовки проектов, поддержку существующих проектов, развитый редактор, отладчик, мастер управления конфигурациями. Модуль необходим для работы с платформо-зависимыми (native) методами — *глава 16*.

- ♦ **PHP**. 4,7 Мбайт. (Инструменты для создания в среде NetBeans приложений PHP.)

В рамках данного учебного курса не требуется.

- ♦ **SOA**. 23,7 Мбайт. (Средства для создания составных приложений в рамках SOA (Service Oriented Architecture) — архитектуры, основанной на использовании сервисов.) Включает графический пакет BPEL, предназначенный для создания и визуального проектирования приложений, ориентированных на использование сервисов, в том числе графический редактор, отладчик, поддержка рефакторинга.

Требуется, чтобы были установлены исполняющие среды GlassFish и Open ESB 2.0.

В рамках данного учебного курса не нужен, следует снять флажок.

#### **Runtimes (исполняющие среды):**

- ♦ **GlassFish v2 UR2**. 84,8 Мбайт. (Бесплатный высокопроизводительный сервер приложений Java.) Базовая реализация спецификации Java EE 5. Проект с открытым исходным кодом.

Предназначен для разработки и использования совместимых с платформой Java EE 5 приложений и веб-серверов в крупномасштабных промышленных средах. Основан на исходных кодах серверов приложений Sun Java System Application Server PE 9 фирмы Sun и TopLink persistence фирмы Oracle.

Модуль необходим для проверки работы с серверными приложениями (*см. разд. 11.3 и 11.4*). В рамках данного курса не требуется подключения к Интернету, сервер позволяет отлаживать приложения на локальном компьютере.

- ♦ **GlassFish v3 Prelude**. 36,8 Мбайт. Предварительный вариант разрабатываемой новой версии сервера приложений GlassFish. Отличается от версии 2 модульностью, небольшим количеством потребляемых ресурсов, расширяемостью архитектуры. Компонент не обязателен. Но желательно подключить его, чтобы при работе с серверными приложениями можно было осуществлять выбор между разными программными серверами.

- ♦ **Apache Tomcat 6.0.18**. 10,4 Мбайт. (Бесплатный веб-сервер с открытым исходным кодом.) Базовый пример реализации сервера для технологий сервлетов и JSP (JavaServer Pages).

Компонент не обязателен. Но желательно подключить его, чтобы при работе с серверными приложениями можно было сравнивать этот легкий (потребляющий мало ресурсов) сервер с GlassFish, а также осуществлять выбор между разными программными серверами.

- ♦ **Open ESB v2.** 12,2 Мбайт. (Открытая реализация исполняющей среды Enterprise Service Bus (ESB — корпоративная сервисная шина), использующая как основу Java Business Integration (JBI — Java-интеграция бизнеса).) Open ESB реализует открытые стандарты, имеет открытый исходный код и является платформой для построения решений, основанных на SOA.

В рамках данного учебного курса не требуется.

После выбранных опций требуемое на диске место уменьшится до 564 Мбайт. Нажатие на кнопку **ОК** возвращает нас в начальный диалог установки NetBeans. И теперь уже можно нажать кнопку **Next >**.

Появится окно с лицензией. В нижней части окна следует отметить флажком пункт выбора **I accept the terms in the license agreement** (Я принимаю условия лицензионного соглашения). Лицензия дает право на бесплатное использование среды NetBeans и предоставляемых с ней модулей.

Для продолжения установки следует нажать кнопку **Next >**.

Появится следующая форма, в которой можно выбрать папку, куда устанавливается NetBeans и модули к ней, а также папку с версией JDK, под управлением которой будет работать среда NetBeans (рис. 2.9).

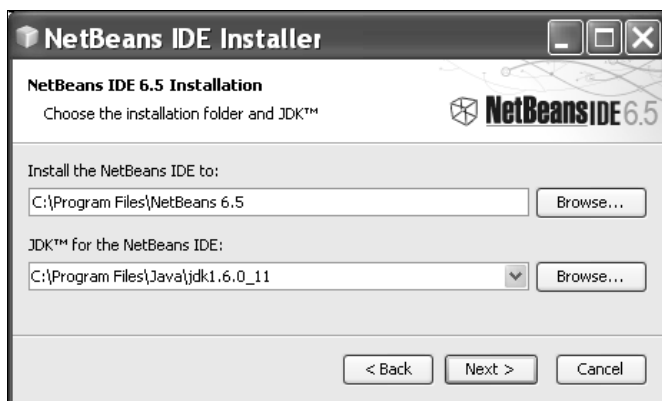


Рис. 2.9. Форма выбора папки, в которую устанавливается среда NetBeans

После нажатия на кнопку **Next >** появляется диалоговое окно с параметрами администрирования сервера **GlassFish** (рис. 2.10).

Пароль администратора, конечно, желательно изменить, чтобы никто не смог зайти на сервер приложений в качестве администратора по стандартному паролю. Остальные параметры лучше не менять.

После нажатия кнопки **Next >** появится форма с возможностью выбора папки для установки сервера **Glassfish-v3-prelude**. Ничего не меняем и нажимаем кнопку **Next >**. Будет выведена форма выбора папки для установки сервера **Apache Tomcat 6.0**. Ничего не меняем и опять нажимаем кнопку **Next >**.

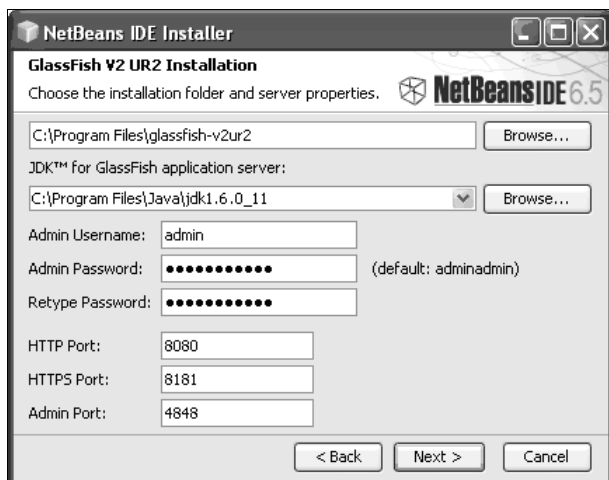


Рис. 2.10. Диалоговое окно с параметрами администрирования сервера **GlassFish**

Появится форма с информацией о выбранных параметрах. Нажатием на кнопку **Install** (Установить) запустим установку. Откроется диалоговое окно, в котором несколько минут будет выводиться текущая информация о ходе установки, после чего отобразится сообщение **Setup complete** (установка завершена) — рис. 2.11.

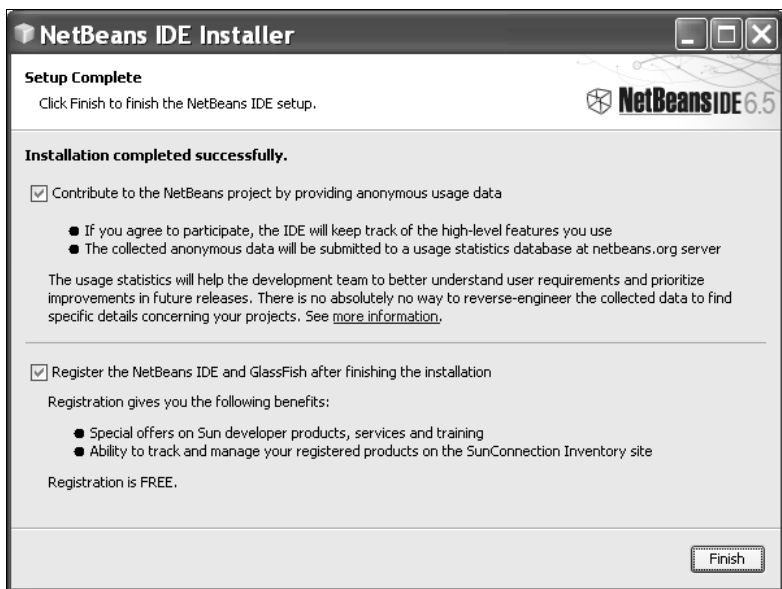


Рис. 2.11. Окно окончания установки NetBeans 6.5

Если ваш компьютер подключен к Интернету и вы согласны ответить на ряд вопросов по среде NetBeans, чтобы помочь ее разработчикам усовершенствовать продукт, или если

вы хотите зарегистрироваться как пользователь NetBeans и GlassFish, сразу нажмите кнопку **Finish** (Завершить), и ответьте на вопросы на английском языке в появляющихся после этого формах. Остальным пользователям следует снять галочки в обоих пунктах выбора (квадратиках) и только после этого нажать кнопку **Finish**, что и завершит установку. На рабочем столе компьютера появится ярлык NetBeans, а в меню **Пуск** — рабочая группа **Программы/ NetBeans 6.5/NetBeans IDE**.

Для того чтобы в дальнейшем добавить еще не установленные модули и исполняющие среды, нужно еще раз запустить инсталлятор среды NetBeans, нажать кнопку **Customize...** и отметить необходимые пункты. Не следует опасаться того, что при повторном запуске инсталлятора затрется предыдущая установка или потеряются какие-то параметры или файлы.

Изменить состав установленных модулей и исполняющих сред также можно с помощью пункта меню среды NetBeans **Tools/Plugins** (Инструменты/Присоединяемые модули).

Удалить среду NetBeans и/или исполняющие среды (серверы приложений или Open ESB, если она была установлена) можно либо из панели управления в пункте **Установка и удаление программ**, либо с помощью запуска программы Uninstall в папке **Program Files\ NetBeans 6.5**.

Отметим, что при наличии на компьютере антивирусных программ (например, Kaspersky Antivirus с включенной проактивной защитой) после окончания работы инсталлятора они могут выдавать предупреждение о потенциально опасных действиях элемента инсталлятора (программы nbi-cleaner, запускаемой из временной папки). Это вовсе не означает, что дистрибутив заражен вирусом или является "троянской" программой. Данной программе следует разрешить выполнять необходимые действия — она удаляет ненужные файлы и записывает в реестр компьютера информацию о конфигурации прошедшей инсталляции (по-видимому, именно это вызывает тревогу).

После запуска NetBeans 6.5 появляется страница приветствия (welcome page, start page) этой среды (рис. 2.12). Если не хочется, чтобы данное окно появлялось каждый раз после старта среды, снимите галочку в нижней части окна в пункте **Show On Startup** (Показывать при запуске).

На этом наша работа по установке программ еще не закончена. Читателям, знакомым с английским языком (а для профессионального программирования знать технический английский абсолютно необходимо), полезно установить документацию к JDK, находящуюся в файле с архивом jdk-6-doc.zip на прилагаемом к книге компакт-диске. Архив следует целиком скопировать в папку на жестком диске (например, в папку C:\Program Files\Java), после чего запустить интегрированную среду разработки (IDE) NetBeans и выбрать в главном меню пункт **Tools/Java Platforms**.

Появится диалоговое окно (рис. 2.13), в котором следует переключиться на вкладку **Javadoc**, нажать кнопку **Add ZIP/Folder...** и выбрать путь к архиву (при попадании файлового диалога в папку с архивом придется подождать — среда долго читает заголовки архивных файлов, это не "зависание" программы).

После подтверждения выбора архива (установки на него курсора и нажатия кнопки **Add ZIP/Folder...**) документация к JDK установлена. Можно осуществлять быстрый доступ

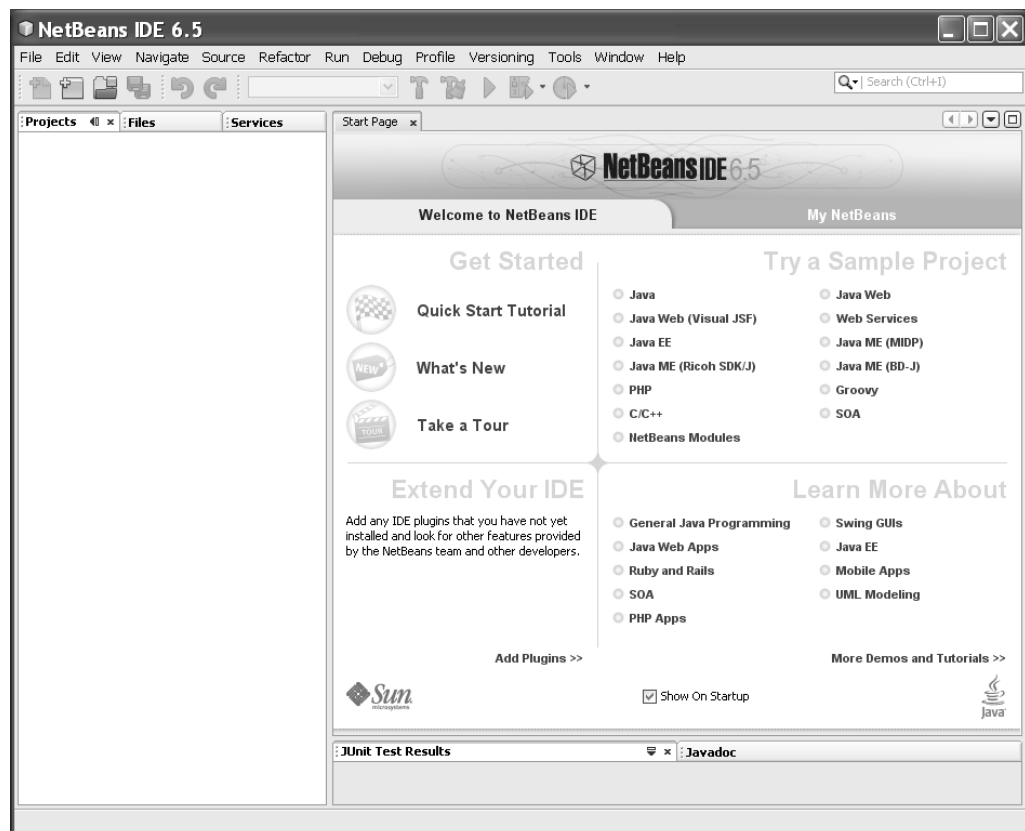


Рис. 2.12. Окно приветствия среды NetBeans 6.5

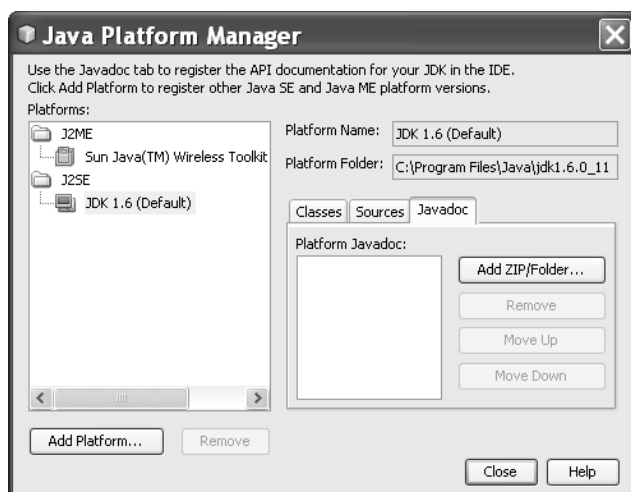


Рис. 2.13. Установка документации к JDK

к документации из среды NetBeans: в меню **Help** (справка) открывать появляющийся после этого пункт **Javadoc References/Java Platform SE 6**, а также выполнять поиск **Help/Javadoc Index Search** (<Shift>+<F1>) (рис. 2.14).

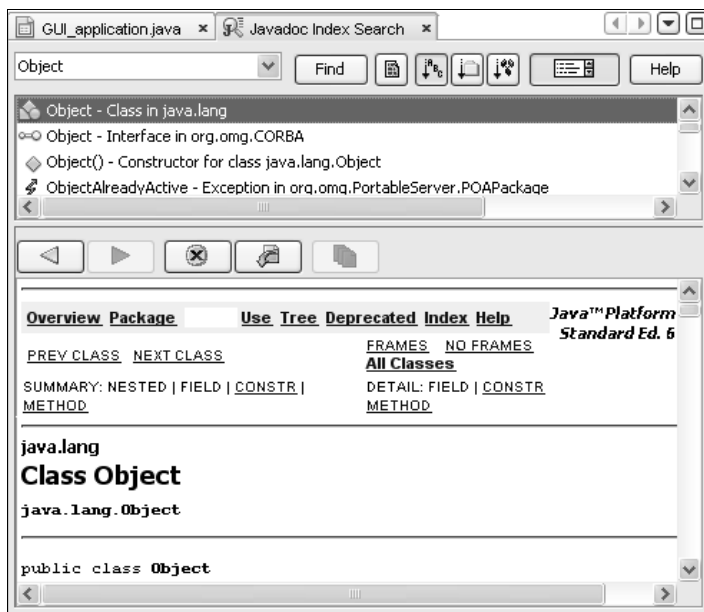


Рис. 2.14. Справка Javadoc Index Search

Если в исходном коде курсор стоит на каком-либо имени, то появляющаяся при нажатии комбинации клавиш <Shift>+<F1> справка оказывается контекстной. Однократный щелчок по имени в списке найденных вхождений приводит к показу в нижнем окне информации, соответствующей этому вхождению, двойной щелчок — к отображению той же информации (HTML-документа) в независимом окне браузера, которое можно развернуть на весь экран.

## 2.3. Создание в NetBeans простейшего приложения Java

Создадим с помощью среды NetBeans приложение Java. Для этого запустим интегрированную среду разработки (IDE) NetBeans, в главном меню **File/New Project...**, в открывшемся диалоговом окне (рис. 2.15) в столбце **Categories** (Категории) выберем пункт **Java**, после чего в столбце **Projects** (Проекты) укажем вид проекта **Java Application** и нажмем кнопку **Next >**.

Появится форма (рис. 2.16), в которой можно изменить некоторые параметры создаваемого проекта.



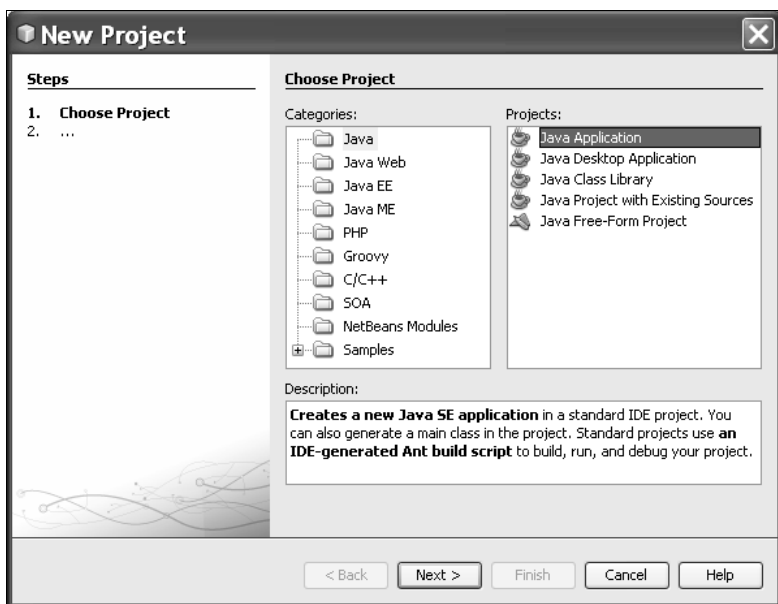


Рис. 2.15. Создание нового проекта. Шаг 1

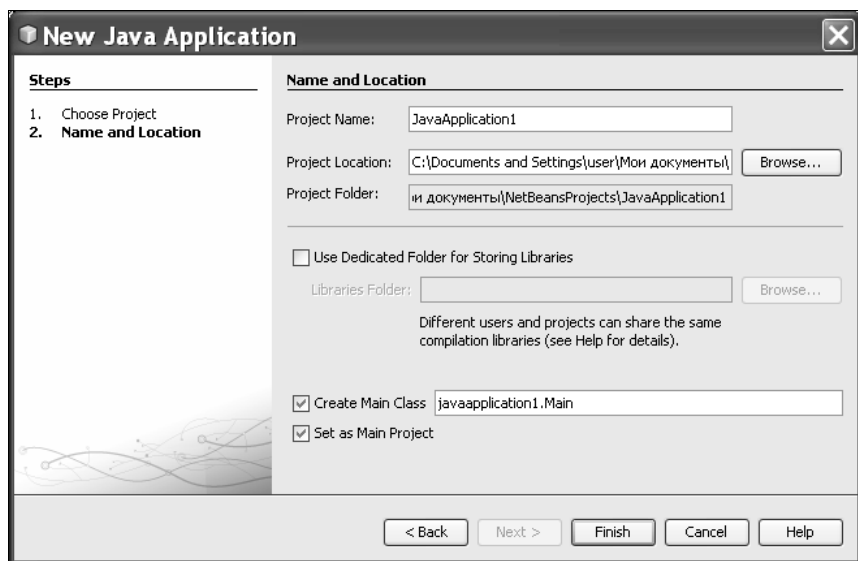


Рис. 2.16. Создание нового проекта. Шаг 2

Нажмем кнопку **Finish**. Значения по умолчанию для начала менять не стоит — это можно будет сделать потом, когда вы освоитесь со средой разработки.

На рис. 2.17 показано, как выглядит редактирование исходного кода приложения в среде NetBeans.