

Літератур

а

Программирование на Java 2. / Глушаков С.В. 2-е изд. - Харьков: ФОЛИО, 2003. – 536 с.

Програмування в Java / А. Г. Горбань, 2008. – 310 с.

Ткаченко О.М., Каплун В.А. / Об'єктно-орієнтоване програмування мовою Java. Навчальний посібник. – Вінниця: ВНТУ, 2006. – 101 с.

Кен Арнольд, Джеймс Гослинг
“Язык программирования JAVA”

Монахов В. В.
«Язык программирования Java и среда NetBeans»

http://www.bhv.ru/books/book.php?id=188402

← → ↺ www.bhv.ru/books/book.php?id=188402 ☆

bhv БХВ-Петербург Сила знаний

На главную | Где купить | Авторам и заказчикам | Вакансии | Реклама | Издательство | Каталог | Статьи наших авторов | Контакты | Искать

ТЕМЫ **СЕРИИ**

- Информатика и вычислительная техника
- Естественные науки
- Технические науки, промышленность
- Экономика и управление
- Гуманитарные науки
- Публицистика
- Популярная медицина
- Популярная психология
- Популярная юридическая литература
- Дом, быт, досуг
- Детская литература
- Учебная литература для вузов
- Учебная литература для школ
- Путеводитель
- Дерзай! Наборы по электронике

Язык программирования Java и среда NetBeans. 3-е изд.

Автор	Монахов В.
Название	Язык программирования Java и среда NetBeans. 3-е изд.
Серия	Внесерийные книги
ISBN	978-5-9775-0671-7
Формат	70*100 1/16
Объем	704
Тематика	Программирование, языки, библиотеки -> Java
Приложение	К книге прилагается DVD
Оптовая цена	439 руб.

1977

В Мне нравится 1

 Купить цифровую книгу через Robokassa за 351.00 руб. **КУПИТЬ** Информация

 Полное содержание  Бесплатный фрагмент  **Файлы для загрузки**



Java та інші мови програмування.

1956 – **Fortran**, первый язык программирования *высокого уровня*

1972 - **C**, язык *процедурного программирования*, его базовыми конструкциями являются *подпрограммы*.
Создание системного программного обеспечения.

1967—1972 направление *объектного программирования*, основанного на концепциях работы с *классами и объектами*

1974 - **Pascal**, язык *структурного программирования*, компиляция программ за счет *виртуальной машины*.

Java та інші мови програмування.

1983 - расширение языка **C**, первый компилятор языка **C++**, универсальный язык системного программирования

1995 - **Java**, универсальный язык прикладного программирования,

JavaScript, узкоспециализированный язык программирования HTML-документов

2000 - платформа **.NET**, альтернатива платформе Java

В основу розробки мови Java було покладено такі основні принципи:

- переносимість;
- об'єктна орієнтація;
- багатопоточність;;
- розподіленість;
- безпека.

Переносимостью называется возможность перенесения программы на другие компьютерные платформы и операционные системы

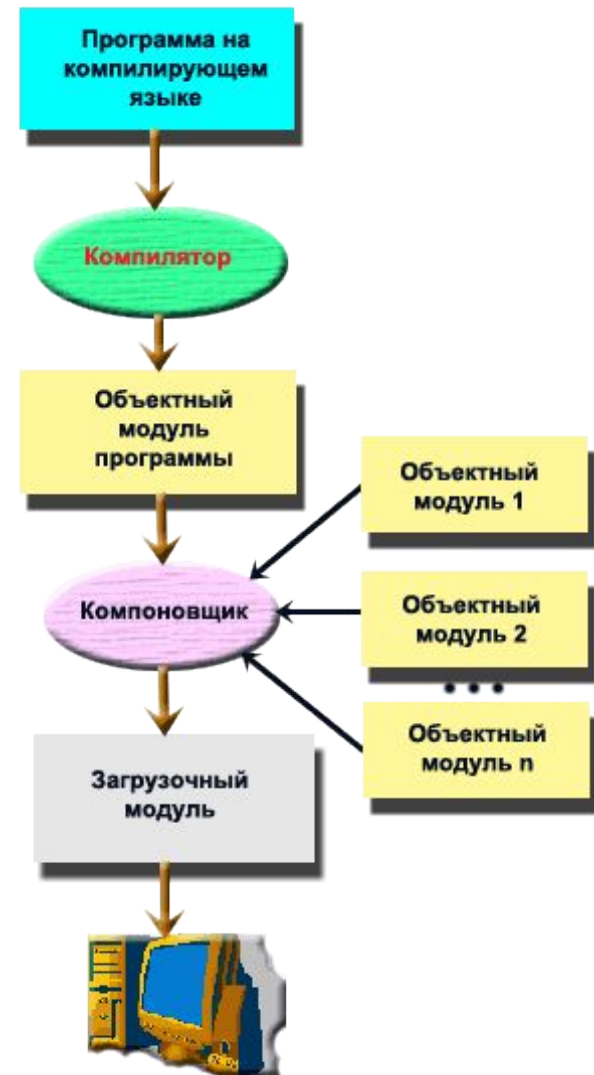
Преобразование программ в машинный код и выполнение программы может производиться одним из двух режимов: интерпретации и компиляции.

В режиме интерпретации выполняется независимое преобразование каждой команды программы в машинный код и эта команда или инструкция сразу же выполняется. Интерпретатор находится в оперативной памяти в течение всего времени выполнения программы пользователя (BASIC, JavaScript, VBScript).

При работе **компилятора** программа на исходном языке сначала преобразуется в эквивалентную программу на машинном языке – промежуточный **машинный код в объектном модуле**. Все объектные модули собираются в единый **загрузочный модуль** с помощью специальной программы – **компоновщика**. Такой модуль может быть загружен ОС в оперативную память и выполнен. (C, C++, Pascal).



**Выполнение программы
в режиме интерпретации**



**Выполнение программы
в режиме компиляции**

Исходные программы на языке Java компилируются, но не в машинные команды конкретной компьютерной платформы, а в команды **виртуальной машины Java** (JVM, Java Virtual Machine).

Виртуальная машина Java — это совокупность команд вместе с программной системой их выполнения. Команды JVM короткие, большинство из них имеет длину 1 байт, поэтому эти команды называют **байт-кодами** (byte-codes), хотя имеются команды длиной 2 и 3 байта. Байт-коды записываются в одном или нескольких файлах, могут храниться во внешней памяти или передаваться по сети.

Таким образом, **компиляция не зависит от компьютерной платформы** и в то же время при выполнении **интерпретируется не текст исходной программы, а не зависящие от компьютерной платформы байт-коды**.

Все стандартные функции Java, вызываемые в программе, подключаются к ней только на этапе выполнения, а не включаются в байт-коды (**динамическая компоновка**, dynamic binding) программы. Это сильно уменьшает объем откомпилированной программы.

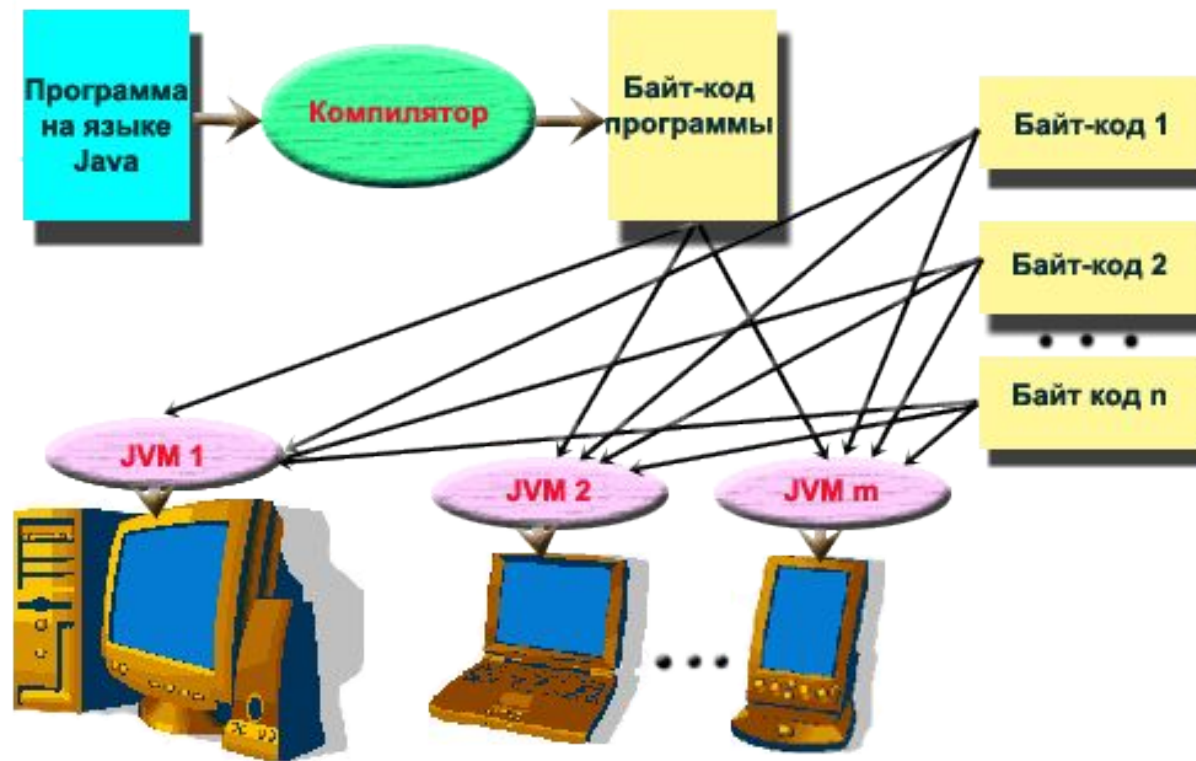


Схема выполнения программы на языке Java

Об'єктна орієнтація

Языки программирования по областям применения можно разделить на две группы: **языки общего назначения** и **специализированные языки**.

В языках общего назначения программа традиционно являлась последовательностью операций (процедур) над данными различных типов, которые реализуют алгоритм решения задачи. Поэтому языки общего назначения назывались также **процедурно-ориентированными языками**.

На процедурно-ориентированном языке можно программировать задачи любых предметных областей, поэтому языки программирования общего назначения иногда называют **универсальными языками программирования**.

Специализированные языки программирования ориентируются на конкретную предметную область (задачу), поэтому их называют также **проблемно-ориентированными языками программирования**.

Появившаяся в начале 80-х годов и впервые реализованная в языке C++ технология **объектно-ориентированного программирования** заключается в том, что программа адаптирует себя к языку описания задачи. При объектно-ориентированном подходе программы разрабатываются с точки зрения вовлеченных в нее объектов программируемой предметной области, их свойств и поведения. Объектно-ориентированные языки объединяют в себе преимущества специализированных языков программирования и универсальность языков общего назначения. Преимущества объектно-ориентированного подхода особенно ярко проявляются при написании больших и сложных программ.

Язык Java, так же как и языки C++, Object Pascal, Perl, Python и C#, является **объектно-ориентированным языком**.

Багатопоточність

В современных ОС может одновременно выполняться несколько независимых задач (**многозадачные операционные системы**). С каждой задачей связан один или несколько **процессов**, которые могут выполняться параллельно.

ОС обеспечивает защиту каждого выполняемого процесса от воздействия других процессов. Высокая степень изоляции процессов друг от друга позволяет увеличить устойчивость работы компьютера, но при этом

возрастают расходы системных ресурсов на запуск и поддержание отдельных процессов.

Была предложена концепция **потоков команд** (**threads** – дословно «нити», либо **lightweight processes** – упрощенные процессы). Потоки команд не защищены друг от друга средствами ОС. Их главное преимущество - очень быстрый запуск.

Java является одним из немногих языков программирования, поддерживающих режим **работы с несколькими потоками**.

Розподіленіс

ть

Распределенное программирование - отдельные компоненты приложения выполняются в разных узлах сети.

В состав Java включены высокоуровневые средства доступа к данным по протоколу HTTP, используемому для передачи Web-страниц, а также по транспортным протоколам сети Internet.

Удаленный вызов процедур в языке Java реализует технология вызова удаленных методов **RMI (Remote Method Invocation)**.

Служба **JNDI (Java Naming and Directory Interface** – интерфейс имен и каталогов Java) позволяет обращаться к объектам сети из программ, используя при этом символьные имена объектов.

Java содержит службу сообщений – **JMS (Java Message Service)**, для программирования обмена сообщениями в сети, средства для создания клиентских приложений электронной почты, а также средства для создания сетевых приложений на базе архитектуры **CORBA (Common Object Request Broker Architecture** – архитектура универсального посредника запросов объекта).

Безпек

Программы Java могут содержать потенциально опасный код, поэтому в языке Java предусмотрены средства, обеспечивающие **безопасность выполнения** программ.

Первоначальная модель безопасности, обеспечиваемая Java, известная как **модель «песочницы»**, обеспечивало очень жесткие правила выполнения программ. В соответствии с этими правилами локальная программа имела доступ к ресурсам компьютера, а программы, полученные из сети могли получить только ограниченный доступ к ресурсам внутри «песочницы».

В дальнейшем было введено понятие «сертифицированного апплета» (signed applet), который может иметь доступ к локальным ресурсам, а затем было введено понятие политики безопасности, реализованной в Java, начиная с JDK 1.2.

Политика безопасности на компьютере определяет набор разрешений доступных программам от различных источников и может быть сконфигурирован пользователем или системным администратором.

Категорії програм Java

- Приложение (*application*) — аналог обычной прикладной программы.
- Апплет (*applet*) — специализированная программа с ограниченными возможностями, работающая в окне WWW-документа под управлением браузера.
- Мидлет (*midlet*) — специализированная программа с ограниченными возможностями, работающая на мобильном устройстве.
- Сервлет (*servlet*) — специализированная программа с ограниченными возможностями, работающая в WWW на стороне сервера.
- Серверное приложение (*Enterprise application*) — предназначено для многократного выполнения на стороне сервера.
- Библиотека (*Java Class Library* — библиотека классов, либо *NetBeans Module* — модуль платформы NetBeans) — предназначена для многократного использования программами Java

Способи установки Java-машини на клієнтський комп'ютер:

- JRE из поставки **Java Development Kit** (JDK) — комплект разработки программного обеспечения на языке Java. Как синоним может употребляться термин SDK — Software Development Kit (комплект разработки программного обеспечения);

<http://www.oracle.com/technetwork/java/>

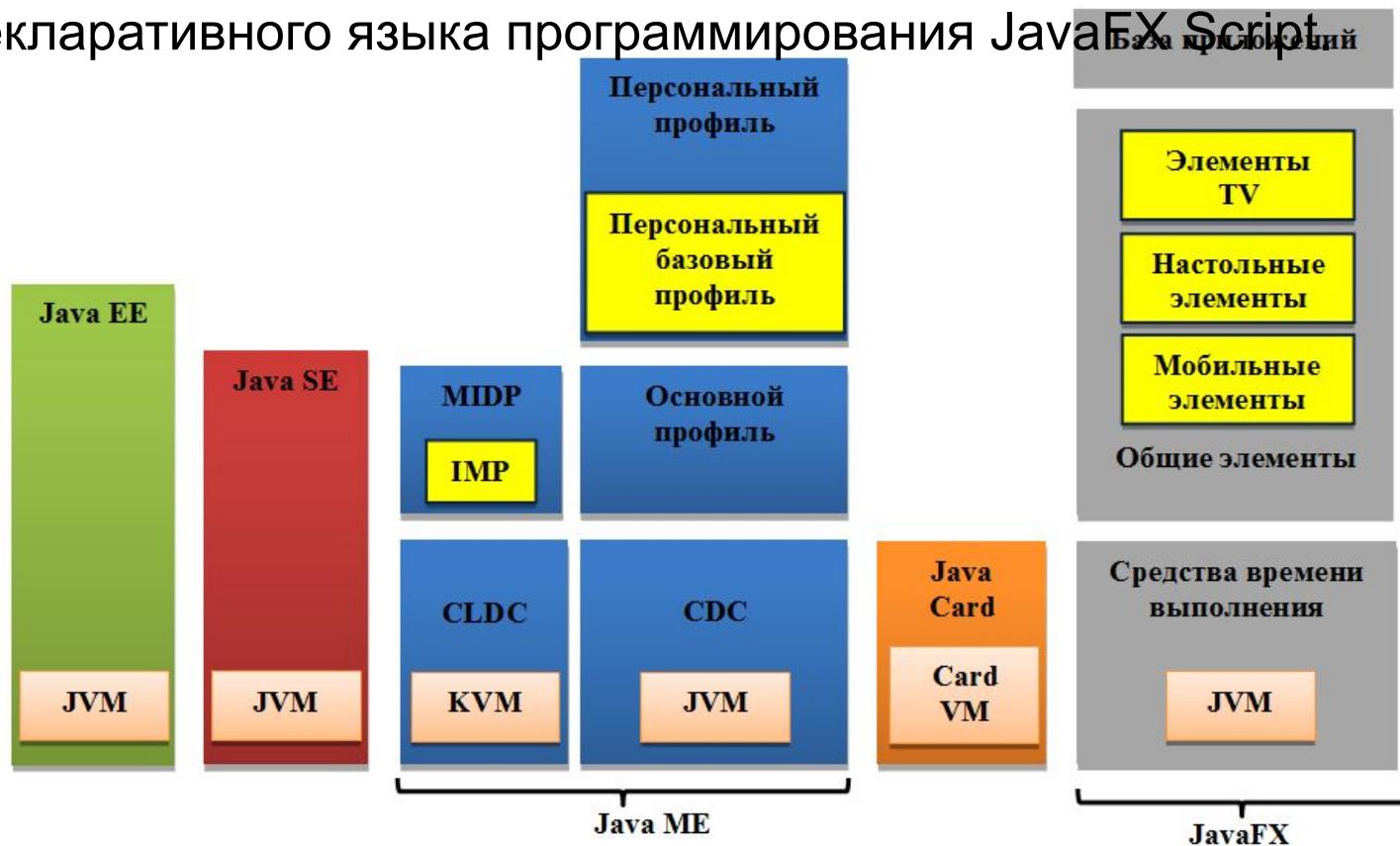
- Специализированный вариант JRE в составе интернет-браузера, называющийся **Java plugin**.

<http://www.java.com/>.

По застосовуваних засобів розробки і умов виконання додатків Java розрізняють платформи:

- Микроиздание — Java Platform, Micro Edition (**Java ME**) для программирования «тонких аппаратных клиентов». Для разработки приложений применяется не JDK, а специальный набор инструментов (Sun Java Wireless Toolkit for CLDC, Sun Java Toolkit for CDC, NetBeans 6.5, и т.д.).
- Стандартное издание — Java Platform, Standard Edition (**Java SE**) для программирования «толстых клиентов». Основой средств разработки служит JDK.
- Корпоративное издание — Java Platform, Enterprise Edition (**Java EE**) для написания серверного программного обеспечения. Является надстройкой над платформой Java SE. В SDK данной платформы входит мощный программный сервер Sun Java System Application Server (сервер приложений Java), а также большое число других исполняющих сред, основанных на технологиях Java.

- **Java Card** – платформа, используемая для создания прикладных программ, работающих на смарт-картах и других устройствах с очень ограниченным объемом памяти и возможностями обработки;
- **JavaFX** – платформа создания прикладных программ с графическим интерфейсом для настольных компьютеров, мобильных и телевизионных устройств с помощью декларативного языка программирования **JavaFX Script**



Інтегровані середовища розробки для Java

NetBeans Бесплатная среда с открытым исходным кодом.

JDeveloper Бесплатная среда, ориентированная на работу с базами данных Oracle на основе технологий Java EE, но может использоваться как универсальная среда разработки для Java.

Eclipse Бесплатная среда с открытым исходным кодом. Является основой (платформой) для многих коммерческих сред разработки

JBuilder 2008 Имеется три редакции — бесплатная (JBuilder 2008 Turbo) и две коммерческие (JBuilder 2008 Professional и JBuilder 2008 Enterprise).

IntelliJ IDEA Коммерческая среда. Доступна в двух версиях: бесплатная *Community Edition* и коммерческая *Ultimate Edition*. Очень популярна в западных странах.

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

www.oracle.com/technetwork/java/index.html

ORACLE

[Sign In/Register](#) [Help](#) [Country](#) [Communities](#) [I am a...](#) [I want to...](#)

Search



[Products](#) [Solutions](#) [Downloads](#) [Store](#) [Support](#) [Training](#) [Partners](#) [About](#)

OTN

Oracle Technology Network > **Java**



Virtual Technology Summit Replay

Learn about Java 8, Docker, 3D Printing, IoT and Enterprise Applications.

Posted 12/07/15 // Tags: [java](#), [JDK](#), [Java SE](#) // [Headlines Archive](#)

Software Downloads

[View All Downloads](#)

Top Downloads

[Java SE](#)

[Java EE and GlassFish](#)

[JavaFX](#)

[Java ME](#)

[JDeveloper 11g and ADF](#)

[Enterprise Pack for Eclipse](#)

[NetBeans IDE](#)

[Pre-Built VM for Java Devs](#)



Downloads



New Downloads

Java SE 8 Update 71 and 8 Update 72

Released 2016/01/19

Java SE Embedded 8 Update 65

Released 2015/10/20

***Java CPU and PSU Releases Explained**

Oracle Java ME Embedded 8.2

Released 2015/09/17

What's New



Java in the Cloud: Rapidly develop and deploy Java business applications in the cloud. Try it FREE for 30 days.

[Java SE](#)[Java EE](#)[Java ME](#)[Java SE Support](#)[Java SE Advanced & Suite](#)[Java Embedded](#)[Java DB](#)[Web Tier](#)[Java Card](#)[Java TV](#)[New to Java](#)[Community](#)[Java Magazine](#)[Overview](#)[Downloads](#)[Documentation](#)[Community](#)[Technologies](#)[Training](#)

Java SE Downloads



Java Platform (JDK) 8u71 / 8u72



NetBeans with JDK 8

Java Platform, Standard Edition

Java SE 8u71 / 8u72

Java SE 8u71 includes important security fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release. Java SE 8u72 is a patch-set update, including all of 8u71 plus additional features (described in the release notes).

[Learn more](#)

- [Installation Instructions](#)
- [Release Notes](#)
- [Oracle License](#)
- [Java SE Products](#)
- [Third Party Licenses](#)

JDK

DOWNLOAD

Server JRE

DOWNLOAD

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Java.net](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)



<https://netbeans.org/downloads/index.html>

 **NetBeans**

NetBeans IDE | NetBeans Platform | Plugins | Docs & Support | Community | Partners

Search 

HOME / Download

Загрузка среды NetBeans 8.2

8.1 | 8.2 | Разработка | Архив

Электронная почта (необязательно):

Подписаться на новости:

☐ Ежемесячные ☐ Еженедельные

☐ NetBeans может использовать данный адрес для связи со мной

Язык IDE: Русский Платформа: Windows

Внимание: Технологии, отмеченные серым цветом, недоступны для данной платформы.

Сборки интегрированной среды NetBeans

Поддерживаемые технологии *	Java SE	Java EE	HTML5/JavaScript	PHP	C/C++	Все
④ Пакет SDK платформы NetBeans	•	•				•
④ Java SE	•	•				•
④ Java FX	•	•				•
④ Java EE		•				•
④ Java ME						•
④ HTML5/JavaScript		•	•	•		•
④ PHP			•	•		•
④ C/C++					•	•
④ Groovy						•
④ Java Card(tm) 3 Connected						•
Поставляемые серверы						
④ GlassFish Server Open Source Edition 4.1.1		•				•
④ Apache Tomcat 8.0.27		•				•

Загрузить

Загрузить

Загрузить x86

Загрузить x86

Загрузить x86

Загрузить

Загрузить x64

Загрузить x64

Загрузить x64

Бесплатно, 95 MB

Бесплатно, 197 MB

Бесплатно, 108 - 112 MB

Бесплатно, 108 - 112 MB

Бесплатно, 107 - 110 MB

Бесплатно, 221 MB

Вміст папки

JDK

- **bin** — содержатся файлы инструментов разработки;
- **demo** — файлы примеров с исходными кодами;
- **include** — заголовки файлов C для доступа к ряду библиотек Java и отладчику виртуальной Java-машины на платформо-зависимом уровне — на основе интерфейсов JNI (Java Native Interface) и JVMDI (Java Virtual Machine Debugging Interface) соответственно;
- **jre** — файлы, необходимые для работы с виртуальной Java-машиной;
- **lib** — ряд библиотек и сопроводительных файлов для работы инструментов из папки bin;
- **sample** — примеры с исходными кодами.
- **src.zip** файлы с исходными кодами стандартных классов Java.

Средства разработки приложений

Утилита	Назначение
javac	Компилятор в режиме командной строки для программ, написанных на языке Java
java	Утилита для запуска в режиме командной строки откомпилированных программ-приложений
appletviewer	Утилита для запуска на исполнение и отладку апплетов без браузера. При этом не гарантируется работоспособность отлаженного апплета в браузере
jdb	Отладчик программ, написанных на языке Java
javadoc	Генератор документации по классам на основе комментариев, начинающихся с /**
jar	Создание и управление Java-архивами jar
javah	Генератор заголовочных файлов C/C++ для подключения к программам Java внешних библиотек C/C++ на основе интерфейса JNI
javap	Дизассемблер классов
extcheck	Утилита для обнаружения конфликтов между файлами архивов jar
native2ascii	Утилита для конвертации в режиме командной строки параметра, передаваемого в виде текста на национальном алфавите, в последовательность символов Unicode

Средства разработки приложений

rmic, rmiregistry, rmid - средства поддержки работы в WWW и корпоративных сетях (интранет) с интерфейсом RMI — интерфейсом удаленного вызова методов (Remote Methods Invocation).

keytool, jarsigner, policytool средства поддержки информационной безопасности и ряд других категорий утилит.

Оформлення програми на мові Java

Программа на языке Java записывается в текстовом виде и состоит из вводимых элементов. Программа разбивается на строки.

Вводимые элементы могут быть следующих типов: пробельные символы, комментарии и элементы программы (tokens).

Элементами программы являются: ключевые слова, идентификаторы, константы (литералы), разделители и операции.

Символы верхнего и нижнего регистра в ключевых словах и идентификаторах считаются разными символами.

Алфавіт мови Java

Латинские буквы ASCII:

ABCD...XYZ — заглавные;

abcd...xyz — строчные.

Дополнительные буквенные символы ASCII:

_ — знак подчеркивания;

\$ — знак доллара.

Национальные буквы на примере русского алфавита:

АБВГ...ЭЮЯ — заглавные (прописные);

абвг...эюя — строчные.

Десятичные цифры:

0 1 2 3 4 5 6 7 8 9

Разделители:

() { } [] ; , .

Вісімкові і шістнадцяткові цифри

Шестнадцатеричное значення задається з допомогою символів "0x" или "0X", за которым идет значение числа

0 1 2 3 4 5 6 7 8 9 A B C D E F

$$0xF = F_{<16>} = 15;$$

$$0x10 = 10_{<16>} = 16;$$

$$0x100 = 100_{<16>} = 16 \cdot 16 = 256;$$

$$0x1000 = 1000_{<16>} = (16)^3 = 4096;$$

$$0x1F = 1F_{<16>} = 1 \cdot 16 + 15 = 31;$$

$$0x21 = 21_{<16>} = 2 \cdot 16 + 1 = 33;$$

$$0xFF = FF_{<16>} = 15 \cdot 16 + 15 = 255;$$

Число в **восьмеричной** записи должно начинаться с нуля, за которым следует одна или несколько восьмеричных цифр, например **077777**.

Ключові слова

<code>abstract</code>	<code>assert</code>	<code>boolean</code>	<code>break</code>	<code>byte</code>
<code>case</code>	<code>catch</code>	<code>char</code>	<code>class</code>	<code>const</code>
<code>continue</code>	<code>default</code>	<code>do</code>	<code>double</code>	<code>else</code>
<code>enum</code>	<code>extends</code>	<code>final</code>	<code>finally</code>	<code>float</code>
<code>for</code>	<code>goto</code>	<code>if</code>	<code>implements</code>	<code>import</code>
<code>instanceof</code>	<code>int</code>	<code>interface</code>	<code>long</code>	<code>native</code>
<code>new</code>	<code>package</code>	<code>private</code>	<code>protected</code>	<code>public</code>
<code>return</code>	<code>short</code>	<code>static</code>	<code>strictfp</code>	<code>super</code>
<code>switch</code>	<code>synchronized</code>	<code>this</code>	<code>throw</code>	<code>throws</code>
<code>transient</code>	<code>try</code>	<code>void</code>	<code>volatile</code>	<code>while</code>

Литералы **false, null, true**

Ключевые слова **const** и **goto** не используются в Java. Они введены для того, чтобы сгенерировать сообщение об ошибке компиляции при появлении этих ключевых слов C++ в программе.

Если в тексте программы требуются символы, которые обычным образом в текст ввести нельзя, то прибегают к управляющей последовательности.

Управляющие последовательности

Символ	Что означает
<code>\b</code>	Возврат на один символ назад
<code>\f</code>	Перевод на новую страницу
<code>\n</code>	Перевод на новую строку
<code>\r</code>	Возврат к началу строки
<code>\t</code>	Горизонтальная табуляция
<code>\'</code>	Кавычка
<code>\"</code>	Двойные кавычки
<code>\\</code>	Обратная косая черта
<code>\u</code>	Начало кодировки символа Unicode

Управляющие последовательности — символы Unicode

`\u0030` — `\u0039` — цифры ISO-LATIN от 0 до 9;

`\u0024` — знак доллара \$;

`\u0041` — `\u005a` — буквы от A до Z;

`\u0061` — `\u007a` — буквы от a до z.

Спеціальні символи

- + Оператор сложения
- Оператор вычитания
- * Оператор умножения
- / Оператор деления
- % Оператор остатка от целочисленного деления
- = Оператор присваивания
- ~ Оператор побитового дополнения (побитовое НЕ)
- ^ Оператор XOR (исключающее ИЛИ)
- & Оператор AND (И)
- | Оператор OR (ИЛИ)
- ! Оператор NOT (НЕ)
- > Больше
- < Меньше

Спеціальні символи

- ? Вопросительный знак — часть тернарного (состоящего из трех частей) условного оператора "? :"
- : Двоеточие — часть условного оператора "? :". Также используется для задания метки — ставится после имени метки
- { Левая фигурная скобка — разделитель, открывающий блок кода
- } Правая фигурная скобка — разделитель, закрывающий блок кода
- , Запятая — разделитель в списке параметров оператора; разделитель в составном операторе
- . Точка — десятичный разделитель в числовом литерном выражении; разделитель в составном имени для доступа к элементу пакета, класса, объекта, интерфейса
- (Левая круглая скобка — разделитель, открывающий список параметров в операторах и группируемую часть в выражениях
-) Правая круглая скобка — разделитель, закрывающий список параметров в операторах и группируемую часть в выражениях

Спеціальні символи

- [Левая квадратная скобка — разделитель, открывающий индекс массива
-] Правая квадратная скобка — разделитель, закрывающий индекс массива
- ; Точка с запятой — разделитель, обозначающий окончание оператора
- ' Апостроф (одиночная кавычка) — открытие и закрытие символа
- " Двойные кавычки — открытие и закрытие строки символов
- \ Обратная косая черта (backslash) — используется для задания управляющих последовательностей символов
- @ Коммерческое а ("эт") — знак начала аннотации
- Знак пробела
- Знак табуляции
- Символы # ∅ «» № § не имеют специального назначения

Складені спеціальні символи

++ Оператор инкремента (увеличения на 1) эквивалентно $x = x + 1$

-- Оператор декремента (уменьшения на 1) эквивалентно $x = x - 1$

&& Оператор AND (логическое И)

|| Оператор OR (логическое ИЛИ)

<< Оператор левого побитового сдвига

>>> Оператор беззнакового правого побитового сдвига

>> Оператор правого побитового сдвига с сохранением знака

== Равно

!= Не равно

+= $y += x$ эквивалентно $y = y + x$

-= $y -= x$ эквивалентно $y = y - x$

***=** $y *= x$ эквивалентно $y = y * x$

/= $y /= x$ эквивалентно $y = y / x$

Складені спеціальні символи

`%=` $y \% = x$ еквівалентно $y = y \% x$

`|=` $y |= x$ еквівалентно $y = y | x$

`^=` $y \wedge = x$ еквівалентно $y = y \wedge x$

`>>=` $y >> = x$ еквівалентно $y = y >> x$

`>>>=` $y >>> = x$ еквівалентно $y = y >>> x$

`<<=` $y << = x$ еквівалентно $y = y << x$

`/*` Начало многострочного комментария

`/**` Начало многострочного комментария, предназначенного для автоматического создания документации по классу

`*/` Конец многострочного комментария (открываемого как `/*` или `/**`)

`//` Однострочный комментарий

Идентификатор

Идентификаторами называют имена, присваиваемые различным элементам программы. Идентификаторы в языке Java строятся из символов стандарта Unicode.

Согласно спецификации Java, **идентификаторы** должны состоять из букв и цифр (без пробелов), причем первым символом не должна быть цифра. Идентификаторы не должны совпадать с ключевыми словами, булевскими константами **true** и **false**, а также переменной **null**.

Буква может быть любым буквенным символом Unicode (в том числе и буквой кириллицы – только для версии Multilanguage). Следует однако отметить, что буква **К** латинского алфавита и буква **К** кириллицы считаются разными буквами. Идентификатор, согласно спецификации, может иметь произвольную длину.

Кроме того, для совместимости с ранними версиями Java буквами считаются символ подчеркивания "_" и символ доллара "\$".

Змінні і типи

Переменная - это именованная ячейка памяти, которой сопоставлен **тип** и содержимое которой может изменяться.

Перед тем как задействовать какую-либо переменную, она должна быть задана в области программы, предшествующей тому месту, где эта переменная используется. При объявлении переменной сначала указывается ее тип, а затем идентификатор.

- **int** — целое число;
- **float** — вещественное число;
- **boolean** — логическое значение;
- **Object** — самый простой объектный тип (класс) Java;

Имеется возможность задавать собственные объектные типы (классы):

```
MyType1 a1,b1;
```

Объявление переменных может сопровождаться их **инициализацией** - присваиванием начальных значений.

```
int i1=5;  
int i2=-78;
```

либо

```
int i1=5, i2=-78;
```

Примитивными типами (primitive types) называются такие, для которых данные содержатся в одной ячейке памяти, не имеющей вложенных ячеек.

Ссылочными типами называются такие, для которых в ячейке памяти содержатся не сами данные, а только адреса этих данных, т. е. *ссылки* на данные. При присваивании в ссылочную переменную заносится новый адрес, а не сами данные. Но непосредственного доступа к адресу, хранящемуся в ссылочных переменных, нет. Это сделано для обеспечения безопасности работы с данными.

Константы представляют значения числовых типов и символьного типа, а также типа **String** (строковые константы).

Значение **null** также является константой.

Разделителями в Java являются следующие 9 символов:

- (– открывающая круглая скобка;
-) – закрывающая круглая скобка;
- { – открывающая фигурная скобка;
- } – закрывающая фигурная скобка;
- [– открывающая квадратная скобка;
-] – закрывающая квадратная скобка;
- ; – точка с запятой;
- , – запятая;
- . – точка.

Операции Java – это символы унарных и бинарных операций (арифметических, логических, операций отношения и присваивания), символы составных операций (типа "+="), а также символы "?" и ":" условного оператора.

Последовательность элементов программы образует **предложение** или **оператор** языка Java. В одной строке может размещаться несколько операторов. Оператор может занимать несколько строк, причем перенос на новую строку можно делать в том месте оператора, где находится пробельный символ или символ-разделитель (до или после этого символа). Признаком окончания оператора является символ ";".

Операторы Java, как и операторы языка C, могут образовывать **блоки**. Операторы блока заключаются в символы "{" и "}". Символ ";" после завершения блока (символа "}") ставить не надо.

Комментарии в языке Java можно оформлять одним из способов:

1. */* текст */* Весь текст между этими сочетаниями символов, будет проигнорирован. Такой комментарий может занимать несколько строк.
2. *// текст* Весь текст, следующий после символов "//" до конца строки, игнорируется.

В Java существует еще один тип комментариев, используемый для автоматического документирования программы – комментарии Javadoc.

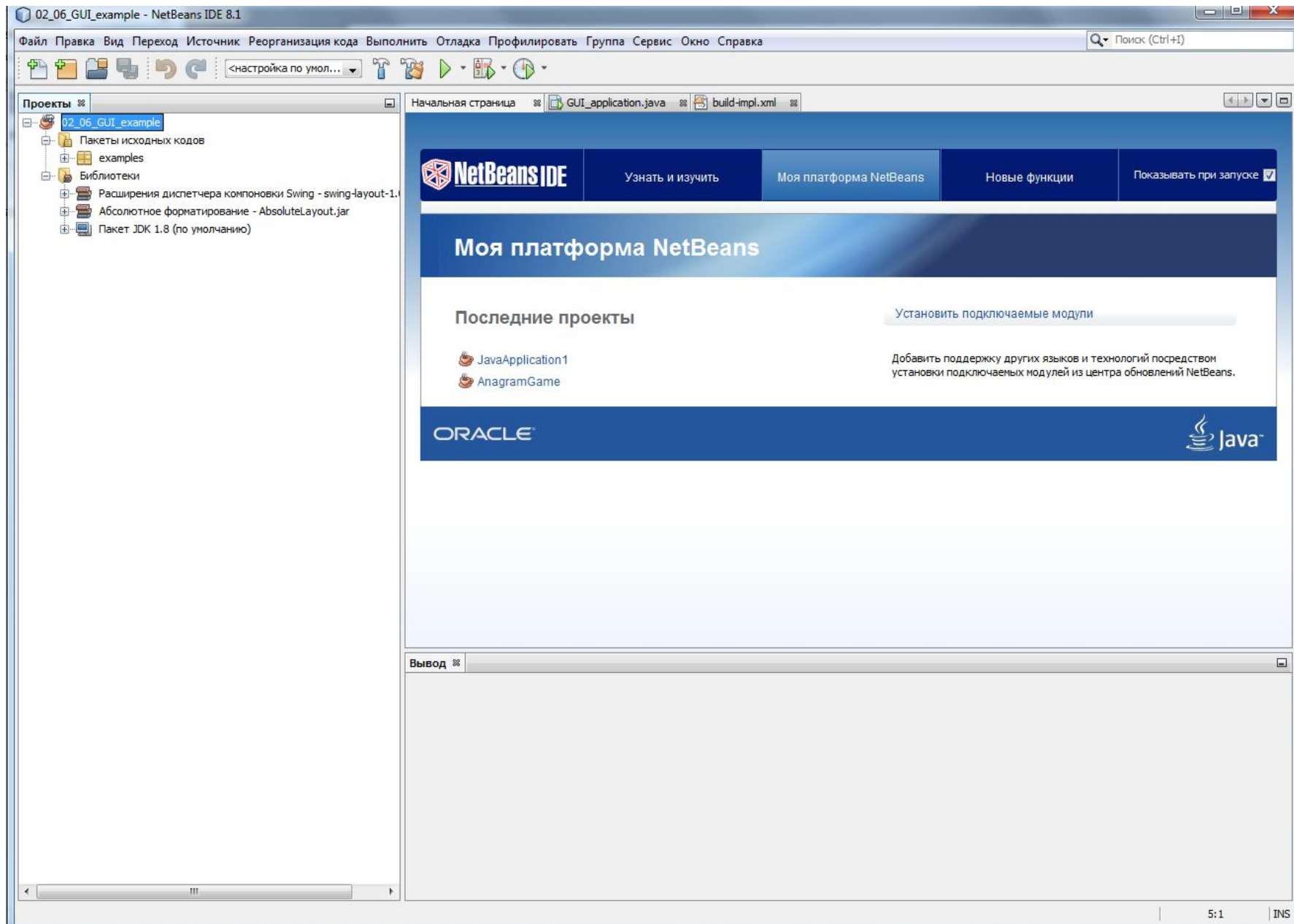
В Java діють наступні угоди про регістрі букв в ідентифікаторах:

- імена примитивних типів следует писать в нижнем регистре, например: **int, float, boolean** и т. д.;
- імена ссылочных типов следует начинать с заглавной буквы, а далее для имен, состоящих из одного слова, писать все остальные буквы в нижнем регистре (например, **Object, Float, Boolean, Collection, Runnable**). Но если имя составное, то новую часть имени начинают с заглавной буквы (например, **JButton, JTextField, JFormattedTextField, MyType** и т. д.).
- для переменных и методов імена, состоящие из одного слова, следует писать в нижнем регистре (например, **i, j, object1**). Если имя составное, новую часть имени начинают с заглавной буквы: **myVariable, jButton2, jTextField2.getText()** и т. д.;
- імена пакетов следует писать полностью в нижнем регистре даже в том случае, когда имя составное: **mypackage, desktopapplication1** и т. д.;

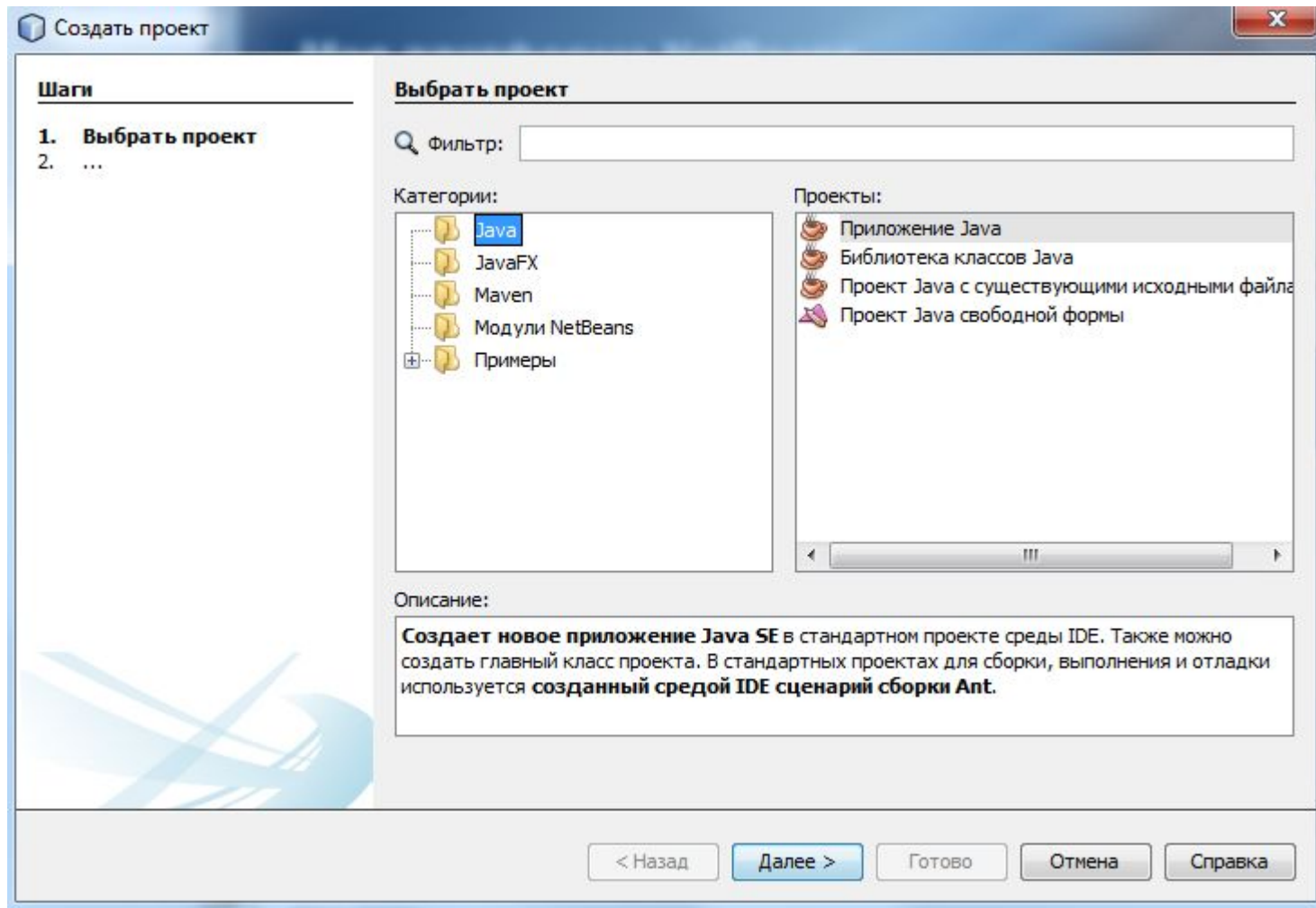
В Java діють наступні угоди про регістрі букв в ідентифікаторах:

- имена констант следует писать в верхнем регистре, разделяя входящие в имя составные части символом подчеркивания "_", например: *Double.MIN_VALUE, Double.MAX_VALUE, JOptionPane.INFORMATION_MESSAGE, MY_CHARS_COUNT* и т. п.;
- символ подчеркивания "_" рекомендуется для разделения составных частей имени только в именах констант и пакетов.

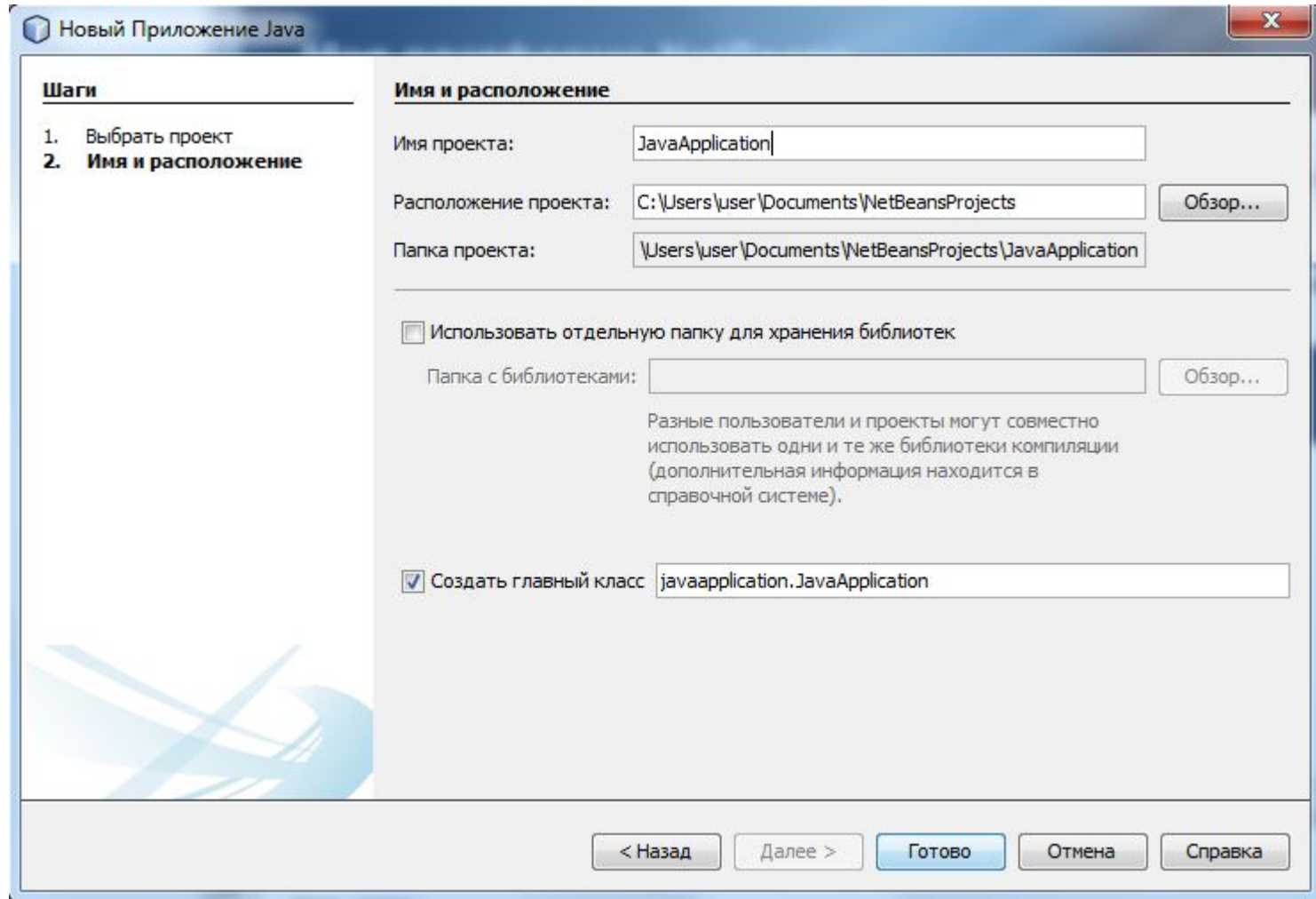
Вікно вітання середовища NetBeans



Створення нового проекту



Створення нового проекту



Новый Приложение Java

Шаги

1. Выбрать проект
2. **Имя и расположение**

Имя и расположение

Имя проекта:

Расположение проекта: Обзор...

Папка проекта:

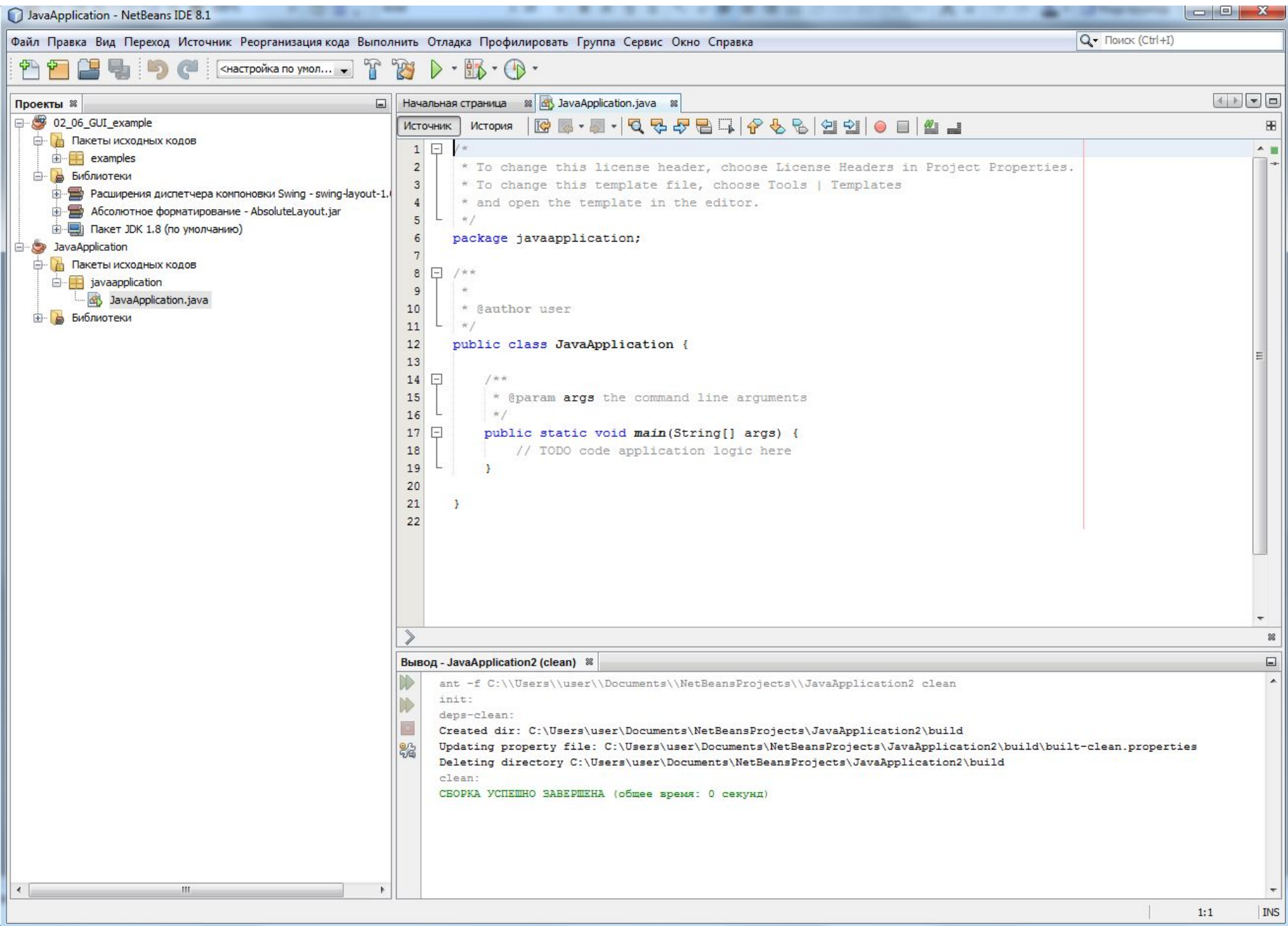
☐ Использовать отдельную папку для хранения библиотек

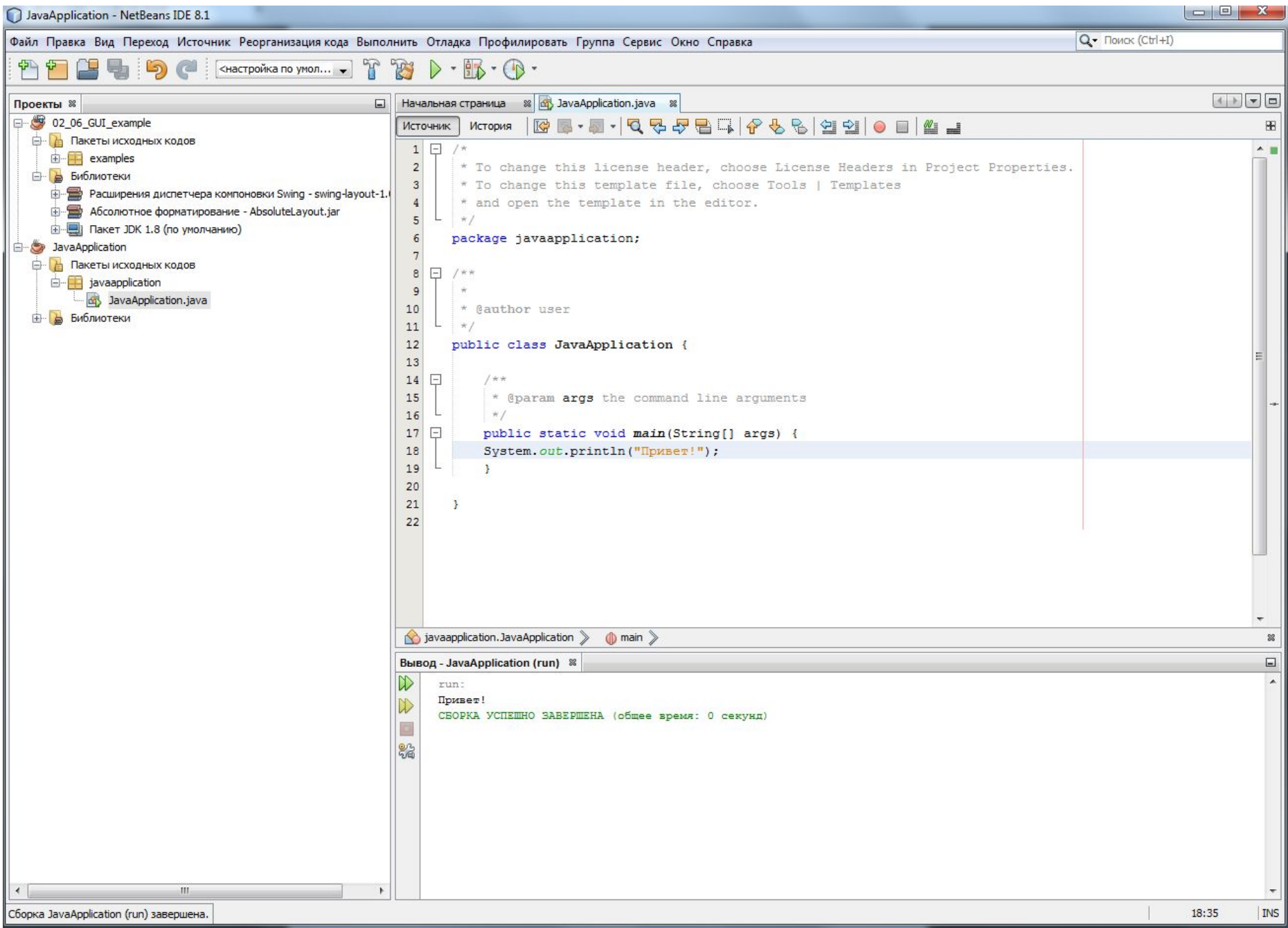
Папка с библиотеками: Обзор...

Разные пользователи и проекты могут совместно использовать одни и те же библиотеки компиляции (дополнительная информация находится в справочной системе).

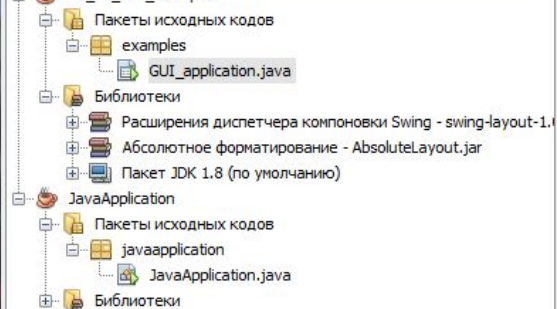
☒ Создать главный класс

< Назад Далее > Готово Отмена Справка





Проекты



Начальная страница JavaApplication.java GUI_application.java

Источник

Проект

История

```
1  /*
2  * GUI_application.java
3  *
4  */
5
6  package examples;
7
8  /**
9  *
10 * @author Б.Б.Монахов
11 */
12 public class GUI_application extends javax.swing.JFrame {
13
14     /** Creates new form Chapter2 */
15     public GUI_application() {
16         initComponents();
17     }
18
19
20     /** This method is called from within the constructor to
21      * initialize the form.
22      * WARNING: Do NOT modify this code. The content of this method is
23      * always regenerated by the Form Editor.
24      */
25     Generated Code
125
126     private void initComponentsPerformed(java.awt.event.ActionEvent evt) {
```

Вывод - JavaApplication (run)

```
run:
Привет!
СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)
```

