**Лабораторная работа №1**

**ВВЕДЕНИЕ В СРЕДУ РАЗРАБОТКИ NETBEANS**

Цель: изучить среду разработки NetBeans, освоить простейшие операции и особенности языка.

NetBeans – среда разработки приложений с 1996 года. Является одной из немногих бесплатных платформ, обладающей практически аналогичным функционалом, как у некоторых платных аналогов. Интегрирована со всеми известными операционными системами. Изначально NetBeans был средой разработки Java приложений, но впоследствии, языковая база расширилась и теперь есть возможность программировать еще на С/С++ и PHP.

**Установка NetBeans**

Данную платформу можно загрузить из официального сайта, пройдя по ссылке [*https://netbeans.org/downloads/*](https://netbeans.org/downloads/), но перед установкой самой среды разработки необходимо, чтобы на компьютере присутствовал JDK (Java Development Kit) пакет, который можно найти на сайте Oracle (ссылка: [*http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html*](http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html)). Конечно, установка компонентов требует определенное количество времени и возникают проблемы с подключением данного пакета. По этой причине мы будем рассматривать альтернативный вариант установки.

Немного предыстории: команда разработчиков NetBeans не представляла собой серьезную компанию или группу программистов, которым поставили задачу по разработке данной платформы. Все начиналось со студентов-энтузиастов, которые практически с нуля разработали и ввели в эксплуатацию проект. Вплоть до нашего времени никаких контрастных изменений не произошло. Единственное существенное изменение, правда, в том, что теперь проект, осуществляемый NetBeans Community, спонсируется компанией Oracle, которая, в свою очередь, занимается разработкой JDK пакетов. Между прочим, данный пакет довольно часто обновляется в связи с активной разработкой технологий (в дальнейшем они будут называться классами), обеспечивая разработчиков новыми возможностями в процессе конструирования различного вида приложений.

В связи с этим на сайте Oracle есть веб-страница, где есть возможность загрузить самораспаковывающийся архив (для Windows) или репозиторий для Linux, который легко устанавливается через Yast приложение. С их помощью вы можете установить сразу и NetBeans, и JDK. Перейдем, собственно, к установке на операционную систему Windows:

1. Проходим по ссылке и загружаем в зависимости от операционной системы и ее разрядности один из файлов (ссылка: [*http://www.oracle.com/technetwork/articles/javase/jdk-netbeans-jsp-142931.html*](http://www.oracle.com/technetwork/articles/javase/jdk-netbeans-jsp-142931.html)).

После загрузки открываем файл и наблюдаем следующее (Рис.1.1)

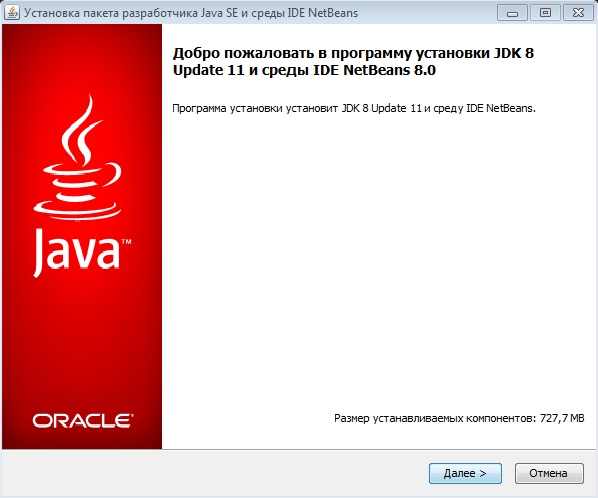


Рис.1.1 Окно установки NetBeans

1. Соглашаемся с установкой JUnit – модуль в языке Java, разработанный для тестирования созданных приложений (Рис.1.2).

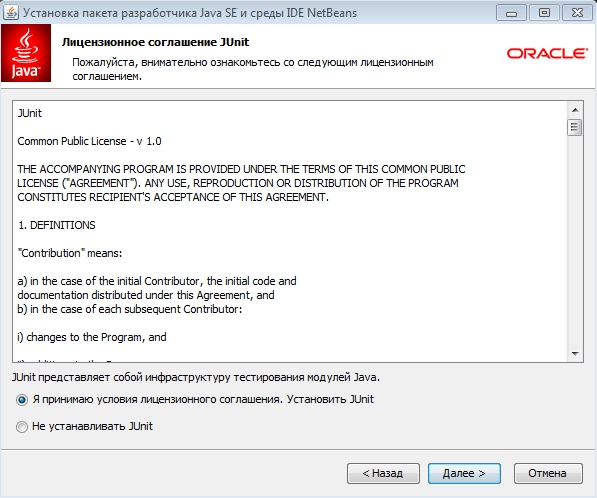


Рис.1.2 Окно установки NetBeans

1. Указываем директорию, где будет установлен JDK пакет. Рекомендуется не изменять ничего в данном окне, ввиду того, что из соображений производительности работы, JDK должен распологаться на вашем системном диске (Рис.1.3).

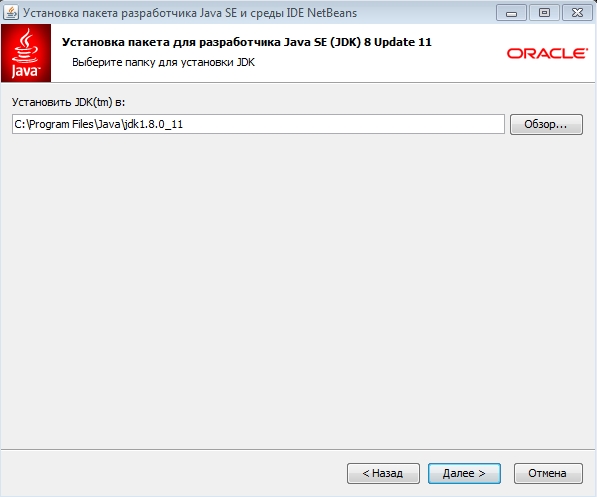


Рис.1.3 Окно установки NetBeans

1. Указываем директорию, где будет установлен NetBeans. Хотя вам предлагают указать путь к JDK, его изменять не требуется. Он по умолчанию фиксируется из предыдущего пункта (Рис.1.4).

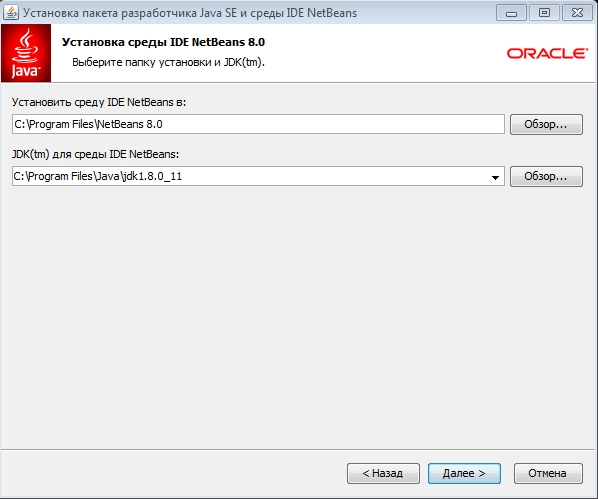


Рис.1.4 Окно установки NetBeans

1. Проверяем правильность установки согласно сводке. Подписку на обновление устанавливать на ваше усмотрение (Рис.1.5).

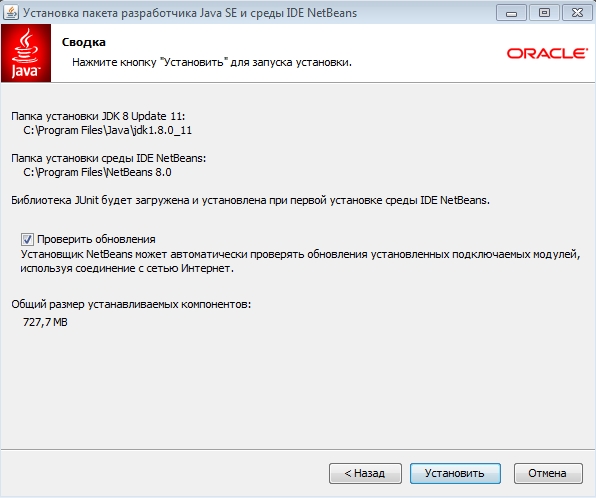


Рис.1.5 Окно установки NetBeans

1. Ожидаем завершения установки.

Таким образом, мы через распаковку лишь одного пакета имеем на компьютере JDK и NetBeans. Теперь рассмотрим интерфейс и основные операции для осуществления процесса разработки.

**Интерфейс**

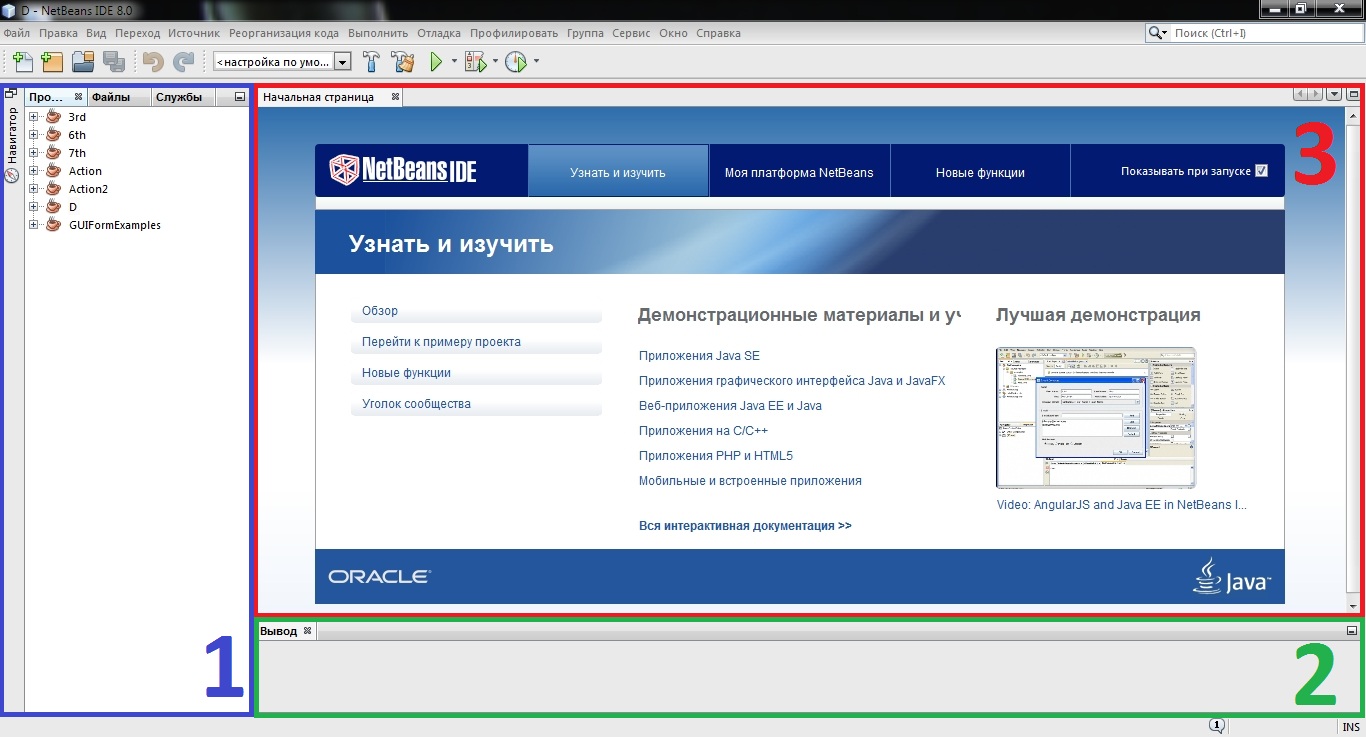


Рис.1.6 Интерфейс среды разработки

1. Менеджер проектов (создание, редактирование иерархии файловой системы проектов)
2. Консоль вывода (отображение рабочего варианта вашего приложения)
3. Рабочая среда (создание, редактирование компонентов вашего проекта)

Изначально в рабочей среде присутствует вкладка «Начальная страница». В дальнейшем, там будут все ваши открытые компоненты проекта. Создадим программу, которая осуществит печать строки «Hello World» на каждой нечетной итерации (круге) цикла.

**Создание проекта**

Начнем с того, что в Java проект состоит из классов, интерфейсов, графических конструкторов и т.д. Для того, чтобы NetBeans осуществил сборку проекта, необходимо наличие главного класса. Главный класс объединяет все компоненты проекта и строит из методов (в СИ методы = функция) реализацию в методе main(). Чтобы создать проект, нажимаем ***CTRL+SHIFT+N*** и получаем (Рис.1.7):

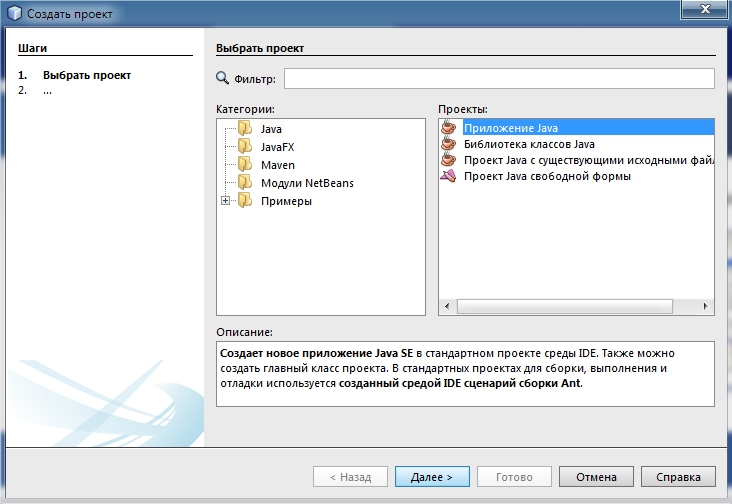


Рис.1.7 Меню создания проекта

Здесь нужно выбрать только «Приложение Java» и нажать «Далее». Следующее окно будет более интересным (Рис.1.8):

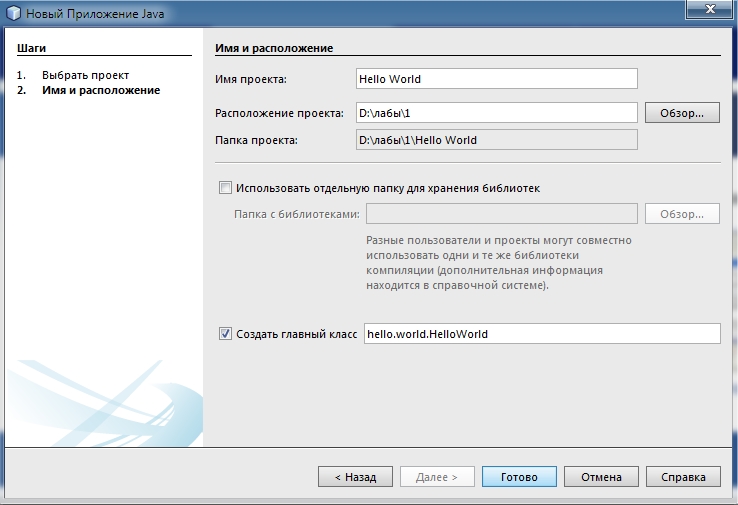


Рис.1.8 Меню настройки проекта

Здесь важно имя проекта, его расположение и то самое имя нашего главного класса. На скриншоте «hello.world» есть имя пакета, где хранится класс, «HelloWorld» - это его имя. Нажимаем «Готово» и с этого момента проект считается созданным.

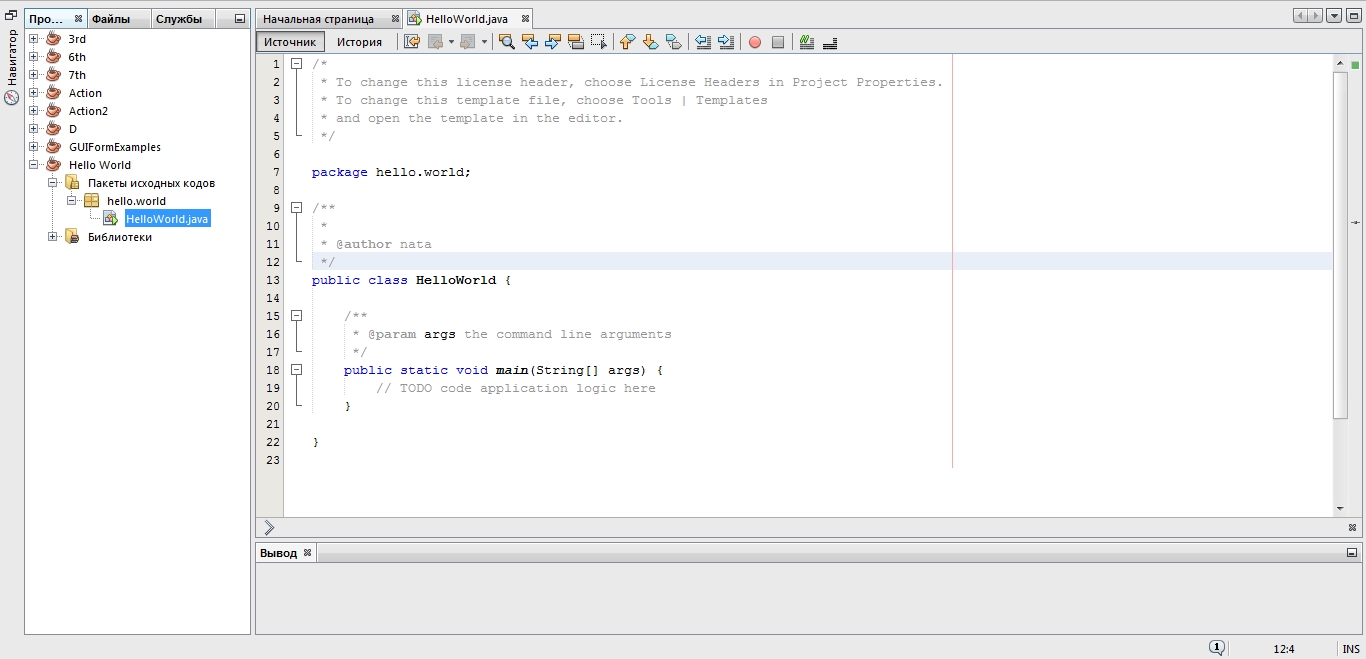


Рис.1.9 Пример созданного проекта

Обратите внимание на секцию менеджера проектов: создан сам проект, в котором содержится «Пакет исходных кодов» и «Библиотеки». В последнем лежат необходимые компоненты из JDK пакета. В первом – ваши созданные компоненты, которые можно распределить по пакетам. В нашем примере имеем только один пакет, в котором храниться один компонент – класс HelloWorld.

Рассмотрим рабочую среду. Здесь наблюдаем созданный класс. У нас:

package Hello.World – означает принадлежность класса к пакету Hello.World

public class HelloWorld(){} – имя класса и его содержимое, заключенное в фигурные скобки

public static void main(String[] args){} – метод, где будет расположена реализация проекта.

По сравнению с языком Си, то метод выглядит немного иначе за счет наличия переменной в скобках. На самом деле, в Си метод main() тоже может принимать аргументы (даже два). Что касается назначения аргумента String[] args, то рассмотрим следующее. Допустим, вы запускаете приложение, которая выполняет поиск в некоторой базе данных при помощи команды запуска приложения в командной строке (в Windows *Пуск- >Выполнить* и вводим в строку cmd, после чего открывается консоль). Сам запуск выполняется при помощи строки C:\docs\ java filename. Т.к. у вас прописан аргумент в методе main(), то теперь вы можете при открытии ввести ключ для поиска в базе, т.е. C:\docs\ java filename Lenny Fontana. Первая часть остается без изменений, а последние добавленные два слова – это аргументы, вносимые в массив строк args[]. Чтобы приложение осуществляло поиск по введенным ключам, нужно только в коде оперировать с элементами массива args[0], где лежит строка «Lenny» и args[1], где находится «Fontana».

Создадим реализацию и получаем следующее (Рис.1.10):

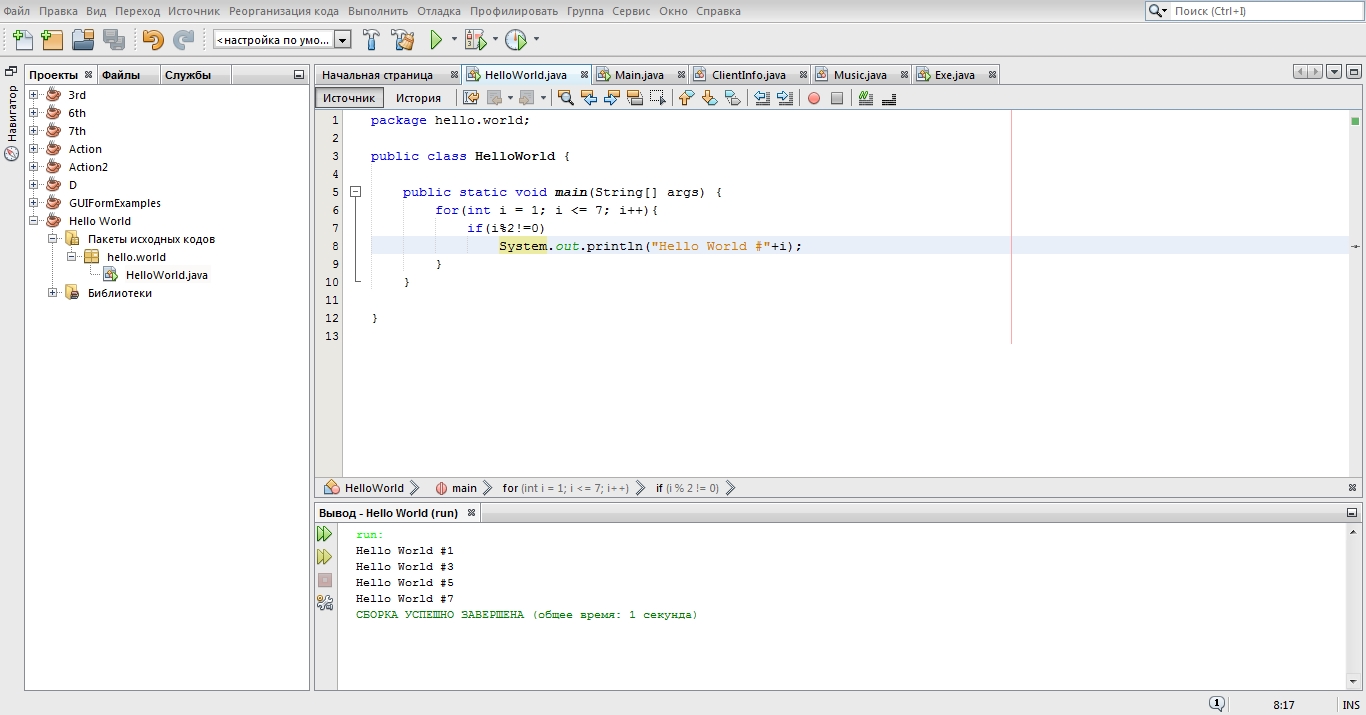


Рис.1.10 Код программы и его выполнение

Здесь в цикле осуществляется вывод строки «Hello World» каждую нечетную итерацию. Как видно, синтаксис цикла и оператора ветвления ничем не отличается от синтаксиса в языке Си. А вот метод печати заметно отличается. Если в Си нужно было вызвать функцию printf() при условии, если вы подключили библиотеку stdio.h, то в Java вы непосредственно работаете с пакетом JDK: оттуда обращаетесь к классу System, далее к его дочернему классу out, в котором содержится метод println(). Синтаксис метода следующий:

System.out.println(String str);

Обратите внимание на окно вывода. Печатается «Hello World #» и далее идет нечетный номер. Вернемся к методу печати и смотрим на вводимую строку. Чтобы совместить в одной записи строку и число, содержащейся в поле (в Си поле = переменная) i, совершаем контактенацию строки str с помощью символа ‘+’. Важно понимать, что при записи в нашем примере, метод System.out.println()не принимает два параметра. Это тот же самый str, который мы подвергли следующей операции:

String str = str+i; //в str у нас «Hello World #»

Чтобы выполнить проект и вывести его на экран вывода, нажмите ***F6***.

**Важно:** в Java проекты не компилируется. Процесс выполнения правильно называть интерпретацией. Разница в следующем: компиляция есть преобразование кода в набор нулей и единиц, который понятен так называемой «начинке» вашего компьютера, а интерпретация – это перенос вашего кода в виртуальную Java-машину, который, в свою очередь, функционирует в пределах вашей операционной системы. Пример Java-машины – JRE (Java Runtime Environment).

**Отладка проекта**

Что такое отладка? Часто в практике проектирования приложений сталкиваются с ошибками, которые связаны, например, с неправильной обработкой некоторого поля в некотором методе. Можно просто смотреть на код и внимательно проговаривать, что делает каждая строчка, а можно провести пошаговое выполнение программы, чтобы отследить значение поля при выполнении определенного метода.

Не случайно был выбран пример. Пользуясь им, рассмотрим функцию отладки проекта в среде разработки NetBeans. Чтобы пошагово отладить проект, нажмите ***F7*** и мы получаем (Рис.1.11):

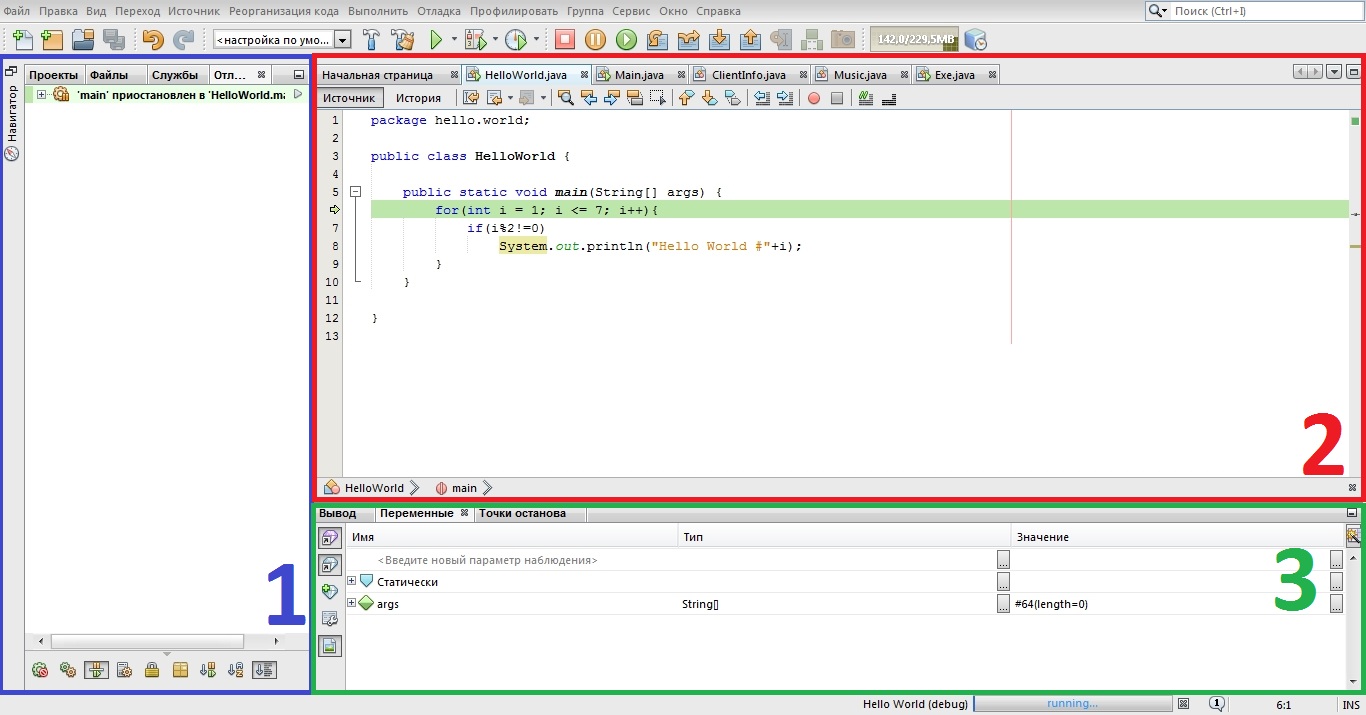


Рис.1.11 Вид отладки

1. Менеджер процессов
2. Среда разработки
3. Окно контроля переменных

У нас напротив метода System.out.println()стоит стрелка. Это позиция, где в определенный момент работает программа. Нажимая ***F7*** несколько раз, стрелка будет двигаться от метода печати до тела цикла до тех пор, пока не выполнится условие его завершения. В свою очередь, значение поля i можно контролировать в окне контроля переменных.

Правда, данный метод предполагает полную отладку, т.е. вы будете прослеживать каждый шаг, включая вхождения в созданные вами методами. Обычно начинают с более краткого метода обхода программы, используя перешагивание. Это значит, что стрелка будет смещаться нажатием клавиши ***F8***, но вход в функции будет пропущен, т.к. они выполнятся автономно (без вмешательства пользователя).

Допустим, вы хотите проследить ход программы где-то в середине кода, а до нее ***F7*** нажимать надо долго. Для этой ситуации придумана точка останова. Слева от области, где расположен код находится нумерация строк. Двойной щелчок по номеру строки создаст точку останова. Далее нажимаем ***CTRL+F5****,* и программа остановится на той строке, где вы поставили точку. Если вы установили две или более точки, то можно дойти до первой точки, отследить значение переменной и чтобы продолжить работу проекта до следующей точки, нажимаем ***F5*** и так далее.

Существует альтернативный вариант. Можно поставить курсор на строку, с которой вы хотите начать отладку и нажать ***F4***. В этом случае программа выполнится до курсора, а далее работа программы осуществляется пошагово. Но если точек останов можно сделать несколько, то курсор у вас всего один, так что стоит иметь в виду данную деталь.

**Задание к лабораторной работе №1**

Реализовать цикл с двадцатью итерациями. С помощью операторов ветвления распечатать в столбик четные числа, начиная со второго десятка.

**Контрольные вопросы**

1. Какой оператор осуществляет вывод содержимого на экран?
2. Как называется процесс выполнения проекта?
3. Что такое отладка проекта?
4. Как выполняется запуск приложения с командной строки?