

# GUI-приложения

Экранной формой называется область, которая видна на экране в виде окна с различными элементами — кнопками, текстом, выпадающими списками и т.п. А сами эти элементы называются компонентами.

Среды, позволяющие в процессе разработки приложения в интерактивном режиме размещать на формы компоненты и задавать их параметры, называются RAD-средами. RAD

расшифровывается как Rapid Application Development — быстрая разработка приложений.

В NetBeans и других современных средах разработки такой процесс основан на объектной модели компонентов, поэтому он называется Объектно-Ориентированным Дизайном (OOD — Object-Oriented Design).

NetBeans является RAD-средой и позволяет быстро и удобно создавать приложения с развитым графическим пользовательским интерфейсом (GUI — Graphic User Interface).

Чтобы создать приложение с графическим интерфейсом в NetBeans достаточно выбрать заготовку Java Desktop Application при создании проекта.

В проекте созданном на основе шаблона уже будет присутствовать экранная форма и среда предоставит два режима работы с ней: режим работы с исходным кодом (тот же, что используется для создания классов и целых консольных приложений) и режим дизайнера (также этот режим часто называется конструктором форм в других средах разработки).

Удобство режима дизайнера заключается в том, что используя набор predefined элементов пользовательского графического интерфейса разработчик может наполнять форму необходимыми элементами просто перетаскивая их на форму из библиотеки, размещая их нужным образом, задавая их основные свойства. Это не позволяет создавать полноценных приложений, но освобождает от написания рутинного кода, необходимого для создания стандартных элементов интерфейса. При желании можно переключиться в режим исходного кода и описать каждый компонент формы вручную.

Типичный набор компонентов формы:

- Button — кнопка;
- Radiobutton — группа переключателей, из которых может быть выбран только один;
- Checkbox — «флажок», активирующий или деактивирующий какую-то опцию.
- List — выпадающий список;
- Label — текстовая надпись;
- Text Field — поле для ввода текста;
- Textarea — многострочное поле для ввода текста.

Существуют и другие компоненты форм, их конкретный набор зависит от используемой библиотеки компонентов.

Современные комплектации Java предоставляют как минимум два набора компонентов AWT и Swing (хотя в современных приложениях имеется возможность совместно использовать эти наборы).

Библиотеку назвали AWT — Abstract Window Toolkit. Слово «абстрактный» в названии указывает на то, что все стандартные компоненты не являются самостоятельными, а работают в связке с соответствующими элементами операционной системы.

Поскольку Java-приложения предназначены для работы на разнообразных платформах, реализация графического пользовательского интерфейса (GUI) должна быть либо одинаковой для любой платформы, либо, напротив, программа должна иметь вид, типичный для данной операционной системы. В силу ряда причин, для основной библиотеки AWT по созданию GUI был выбран второй подход. Во-первых, это лишней раз показывало гибкость Java — действительно, пользователи разных платформ могли работать с одним и тем же Java-приложением, не меняя своих привычек. Во-вторых, такая реализация обеспечивала большую производительность, поскольку была основана на возможностях операционной системы. В частности, это означало и более компактный, простой, а значит, и более надежный код.

Любое приложение с GUI является расширением (наследником) определенного компонента из библиотеки, как правило это заготовка для формы, на которую в дальнейшем попадут другие

компоненты интерфейса (надписи, поля, кнопки и пр.).

У каждого созданного компонента имеется predetermined набор свойств. Например, для самой простой кнопки это координаты на форме, текст на кнопке, цвет фона кнопки, цвет текста и ещё десятки свойств. Все эти свойства могут задаваться также в режиме дизайна, но могут быть описаны и программным кодом. Вообще все параметры в программе, разумеется, хранятся в виде программного кода, но вот быстро создавать и редактировать этот код можно с помощью режима дизайна.

Кроме свойств у каждого компонента графического интерфейса есть и набор связанных с ним действий. Например, при нажатии кнопки возникает событие, к которому мы можем привязать какое-то действие (например, при нажатии кнопки может изменяться текст написанный на ней или вычисляться сумма чисел, введенных в два текстовых поля, и выводится в третье текстовое поле — при этом изменить текст, содержащийся в текстовом поле очень просто, надо просто присвоить новое значение нужному свойству компонента).

Для каждого компонента существует свой набор событий: выделение, нажатие, перетаскивание, редактирование, активация и пр. В ответ на каждое из этих событий может быть произведено какое-то действие, например, вызван какой-то метод. Какие именно действия будут происходить в ответ на определенные события можно описать уже только программным кодом (режим дизайна тут бессилён).

## Java-апплеты

**Апплет** (англ. applet от application — приложение и -let — уменьшительный суффикс) — это компонент программного обеспечения, работающий в контексте другого многофункционального приложения, предназначенный для одной узкой задачи и обычно не имеющий ценности в отрыве от базового приложения.

Примеры:

- Апплет, выводящий текущий курс норвежской-кроны, который используется внутри программы-путеводителя по городу Осло;
- Апплет, отображающий среднее время погрузки контейнера внутри автоматизированной системы для складского учёта.
- Апплет, показывающий текущее время в углу рабочего стола или на панели задач.

**Java-апплет** — программа или компонент программы в двоичном коде виртуальной машины Java, исполняется в окне браузера.

Иными словами, Java-апплет — это программа на Java, встроенная в веб-страницу и отображаемая в браузере. Исполняется программа Java-машиной, но результаты возвращаются браузеру и уже он демонстрирует их пользователю, также из браузера могут собираться различные входные данные (текст введённый в поля форм, перемещения и клики мыши и пр.).

Java-машина взаимодействует с браузером через отдельную программу, которая обычно называется **плагином** (plugin). У плагина очевидная задача: передавать входные данные из браузера в Java-машину, ждать результатов работы программы, передавать результаты обратно в браузер.

Чтобы апплет работал, нужно чтобы на компьютере была установлена Java-машина и плагин для браузера. Тогда страницу с апплетом можно будет просмотреть в браузере и увидеть сам апплет.

Java-апплеты создавались именно для использования в роли интерактивных элементов веб-страниц, в связи с этим к ним предъявляются повышенные требования в плане безопасности. Для защиты от потенциальных проблем апплеты лишены многих возможностей, например они не могут обращаться к файлам на компьютере пользователя, в отличие от обычных программ на Java.

Примеры:

- <http://java.sun.com/applets/jdk/1.4/demo/applets/SortDemo/example1.html> — апплет наглядно показывающий процесс сортировки массива методом пузырька, двунаправленным методом пузырька и методом быстрой сортировки (для запуска анимации надо кликнуть по нужной картинке).
- <http://java.sun.com/applets/jdk/1.4/demo/applets/Clock/example1.html> — апплет отображающий стрелочные часы.

Первый апплет реагирует на действия пользователя: анимации запускается после клика мышью.

Второй апплет получает ряд настроек из содержания той веб-страницы, на которой он отображается.

Апплет встраивается в веб-страницу с помощью таких инструкций (в языке разметки веб-страниц они называются тегами):

```
<applet code="Clock.class" width=170 height=150>
  <param name=bgcolor value="000000">
  <param name=fgcolor1 value="ff0000">
  <param name=fgcolor2 value="ff00ff">
</applet>
```

Первая строка содержит инструкции, указывающие класс с апплетом, а также размеры региона (ширину и высоту в пикселях) в котором апплет будет отображаться.

Три следующих строки передают параметры апплету. Для каждого параметра указывается имя (bgcolor, fgcolor1, fgcolor2) и значения (000000, ff0000, ff00ff). В данном случае, каждый из параметров задаёт цвет (фона, стрелок и текста в апплете). Цвет кодируется в виде трёх шестнадцатичных чисел (от 00 до FF — от 0 до 255 в десятичной системе счисления), каждое из которых отражает уровень красного, синего и зелёного в цвете. Например: ff0000 — даст красный цвет, ffff00 — жёлтый (как смешение красного и зелёного), 888888 — серый цвет (всех цветов поровну), 000000 — чёрный.

Т.е. не изменяя сам апплет через параметры можно каким-то образом передать в апплет некие константы (в данном случае это будут цвета, позволяющие встроить апплет в дизайн страницы, на которой он станет отображаться).

Пример простейшего апплета:

```
import java.awt.*;
import java.applet.*;
public class Main extends Applet{
    public void paint(Graphics g){
        g.drawString("Привет, мир!", 20, 20);
    }
}
```

Этот апплет просто напечатает «Привет, мир!» на экране.

Нужно откомпилировать этот апплет в байт-код, и в папке с полученным классом создать веб-страницу.

Например, такую (это текстовый файл, который нужно будет сохранить с расширением html, чтобы его открыл веб-браузер):

```
<html>
  <head><title>My First Applet</title></head>
  <body>
    <applet code="HelloWorld.class" width="200" height="100">
    </applet>
  </body>
</html>
```

При этом ваш класс должен называться HelloWorld.class, либо вы должны указать другое название класса в имени файла с апплетом, когда будете указывать его в содержании страницы.

Из кода нашей программы видно, что создаваемый апплет является расширением (наследником) класса Applet. Этот класс содержится в пакете java.applet.\*, который мы импортировали в нашей программе.

Т.к. апплет является приложением с графическим интерфейсом, то необходимо было также импортировать пакет java.awt.\* для построения графического интерфейса.

Запустить апплет напрямую с помощью Java-машины нельзя, но можно воспользоваться утилитой appletviewer, которая позволяет просматривать апплет без запуска какого-то конкретного браузера в самостоятельном окне, словно это обычное GUI-приложение.

В среде NetBeans для просмотра апплета через appletviewer можно воспользоваться комбинацией Shift+F6, запускающей не целый проект, а текущий файл.

Итак:

- Апплеты — это Java-программы с графическим интерфейсом, встраиваемые в веб-страницы и исполняемые через браузер;
- Каждый апплет выводится в ограниченной области веб-страницы. Сама веб-страницы содержит имя класса с байт-кодом апплета, а также информацию о размерах региона, в котором апплет будет отображаться;
- Апплет может реагировать на различные действия пользователя;
- Апплет может читать параметры, заданные в содержании веб-страницы;
- Апплет исполняется в защищённой среде, где отсутствует доступ к локальным файлам и прочие возможности, потенциально способные нанести вред пользовательской машине.