**2слайд. Что такое Vaadin?**

**Vaadin** — открытый набор библиотек для создания так называемых RIA(Rich Internet application) веб-приложений. В отличие от библиотек на Javascript и специфических плагинов для браузеров, Vaadin предлагает сервер-ориентированную архитектуру, базирующуюся на Java Enterprise Edition.  Использование JEE позволяет выполнять основную часть логики приложения на стороне сервера, тогда как технология [AJAX](https://ru.wikipedia.org/wiki/AJAX), используемая на стороне браузера, позволяет [интерактивно](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D1%8C) взаимодействовать с пользователем, не отставая от аналогичных десктоп-приложений. Для отображения элементов [пользовательского интерфейса](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F) и взаимодействия с [сервером на стороне клиента](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B8%D0%B5%D0%BD%D1%82-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80) Vaadin использует [Google Web Toolkit](https://ru.wikipedia.org/wiki/Google_Web_Toolkit" \o "Google Web Toolkit).

**Rich Internet application** (RIA, «Насыщенное („богатое“) Интернет-приложение») — это веб-приложение, доступное через Интернет, насыщенное функциональностью традиционных настольных приложений, которое предоставляется либо уникальной спецификой браузера, либо через плагин, либо путём «песочницы» (виртуальной машины).

Тип – веб-фреймворк.

Разработчик – Vaadin Limited.

Написана на Java.

Первый выпуск состоялся 20мая 2009года.

Аппаратная платворма – Java

Сайт vaadin.com

(кстати, в переводе с финского это слово означает «олениха»)

**3слайд. Возможности**

* Использование Java как единственного языка программирования при создании веб-приложений и веб-контента — одна из наиболее значимых функций в Vaadin. Фреймворк использует событийную модель и определенные элементы пользовательского интерфейса, виджеты, что делает её очень близкой к модели разработки десктоп-приложений на Java с использованием HTML и Javascript
* Организация модели данных и виджетов позволяет отображать в браузере большие объёмы данных без значительной загрузки памяти и без дополнительных действий со стороны разработчика.
* Использование Google Web Toolkit для отображения страниц с результатами поиска и обработки действий пользователя (наподобие терминального клиента). Так как Google Web Toolkit функционирует только на стороне клиента, Vaadin добавляет дополнительную валидацию данных на стороне сервера: это решает проблемы безопасности, связанные с возможностью подмены данных или кода Javascript. Соответственно, при изменении и повреждении данных, поступающих от браузера, сервер, определив это, не пропускает запросы.
* Возможность расширения стандартного набора виджетов Vaadin за счет других виджетов, написанных для GWT, а также кастомизации его при помощи CSS. Однако стандартное приложение, создаваемое на Vaadin, не требует программирования именно на GWT и последующей компиляции GWT-компилятором, если только разработчик не добавляет в проект нестандартные виджеты.

**4слайд. Архитектура Vaadin.**

Структурно Vaadin состоит из серверного API, клиентского API, набора компонентов пользовательского интерфейса с обеих сторон, механизма тем для оформления интерфейса и модели данных, позволяющей связывать серверные компоненты непосредственно с данными. Можно применять две основные модели разработки: на стороне сервера и на стороне клиента (браузера)

Vaadin поддерживает все распространенные браузеры как обычных компьютеров, так и мобильных устройств и планшетов. Вся разработка ведется на Java, но Java-код выполняется только на сервере, на клиенте же выполняется чистый JavaScript.

**5слайд. Запуск.**

Первым шагом в NetBeans IDE будет установка плагина (Tools -> Plugins -> Available Plugins, ввести vaadin в поле Search, установить галочку у 'Vaadin Plug-in for NetBeans' и нажать Install, согласившись со всеми вопросами).   
  
Теперь при создании нового проекта (File -> New Project) стала доступна новая категория Vaadin. Выберем Vaadin Web Application Project, нажмем Next и укажем имя нового проекта, например vaadintest.

После нажатия Finish будет создана группа проектов приложения Vaadin по умолчанию. Основной файл с минимальным примером исходного кода приложения Vaadin расположен в проекте myvaadin-ui, файл /Source Packages/com.mycompany.myvaadin/MyUI.java.

В этом простейшем проекте объявляется класс MyUI, являющийся наследником класса UI. В нем переопределяется метод init(). Внутри него создается вертикальная компоновка VerticalLayout, у нее включается отступ (margin), создается новая кнопка с обработчиком нажатия, который добавляет компонент типа Label с текстовой строкой. Затем кнопка добавляется к компоновке вызовом метода addComponent(). Директива [Theme](http://habrahabr.ru/users/theme/)(«mytheme») задает используемую тему оформления (о них чуть ниже).  
  
Перед первым запуском пересоберем все проекты (правый клик на 'myvaadin — myvaadin-parent' -> Build)  
  
Для запуска в отладочном режиме можно использовать плагин Jetty или интегрированный в NetBeans сервер GlassFish Server  
Щелкнем правой кнопкой на проекте -> Debug –> в окне Select Deployment Server из выпадающего списка выберем GlassFish Server.  
  
После того как будут установлены все зависимости и перекомпилированы компоненты приложения, запустится сервлет и автоматически откроется окно браузера.

**6слайд. Темы и стили.**

Все стили берутся из файла styles.css. Этот файл находится в разделе /Web Pages/VAADIN/themes/mytheme/ проекта myvaadin-ui и генерируется из SASS-файлов styles.scss, mytheme.scss и addons.scss, размещенных в том же каталоге. В них за основу берется базовый стиль, называющийся Valo (его предыдущая версия называлась Reindeer и местами по-прежнему упоминается в документации).   
  
Основные параметры темы вынесены в переменные, и для полной смены внешнего вида приложения достаточно изменения считаного числа параметров. Например, цвет шрифта определяется автоматически на основании цвета фона. Сам же цвет фона задается переменной $v-background-color. Чтобы изменить его, добавим в начало файла mytheme.scss следующую строку:

$v-background-color: #000;

Затем нужно щелкнуть правой кнопкой мышки на проекте myvaadin-ui и выбрать Vaadin -> Compile Widgetset and Theme или Compile Theme, после чего обновить в браузере страницу.   
При этом фоновый цвет большинства элементов проекта изменится на черный, цвет шрифта изменится автоматически.  
Благодаря такому подходу для того, чтобы полностью переоформить приложение, скажем, под плоский стиль Metro, достаточно двух десятков строк, переопределяющих значения переменных, без изменения собственно стилей.   
Можно переопределять стиль и напрямую. К примеру, изменить цвет надписи на кнопке можно, добавив в файл mytheme.scss следующие строки (ниже строки «// Insert your own theme rules here»):

$textcolor: red;

.v-button-caption {

color: $textcolor;

}

Затем перекомпилировать темы и обновить страницу браузера.  
Вместо того чтобы создавать свою тему, можно воспользоваться и одной из готовых, изменив название темы с mytheme на одно из следующих: valo, runo, reindeer, chameleon, liferay.

Различные темы <https://vaadin.com/wiki/-/wiki/Main/Valo+Examples>.

**7слайд. Lazy loading.**

Vaadin поддерживает технологию, под названием Lazy load, т.е. Ленивая или отложенная подгрузка - это технология позволяющая загружать только те данные или изображения на странице, которые в данный момент нужны пользователю. Результат внедрения — ускорение загрузки страницы, довольный посетитель и увеличение конверсии.

Представьте, что у вас на странице около 200 изображений (это может быть страница категорий в интернет магазине или раздел фотогалереи или даже запись в блоге). Пусть каждое изображение размером 250 на 250 пикселей и весит 50 Кб. Итого получаем почти 10 мб изображений (200 шт \* 50 кб = 10’000 кб) и 200 запросов к серверу.

Зачем нагружать пользователя таким количеством информации сразу? Давай покажем ему только первые 8 изображений, а остальные по мере того как они ему понадобятся. Загрузка страницы в этом случае пройдёт в разы быстрее. Кроме того, большинство пользователей вообще не будут скролить страницу до конца. Им будет достаточно этих 8 уже загружённых изображений.

**8слайд. Сплит-панели.**

Vaadin очень прост в использовании. Написав 8 строк кода, мы получили готовые сплит панели. В JS такие панели писались бы в десятках строк кода.