**Servlet API.**

**Задание-8.**

**Гистограмма.**

Текущее практическое задание никакого практического смысла не имеет. Ну, разве что, потребует от нас немного пораскинуть мозгами, чтобы придумать как это можно при помощи технологии, которая напрямую этого не поддерживает, создать компонент, который нам никогда не потребуется ☺. Ни то, чтобы этот компонент был настолько бесполезен – даже, наоборот, в современных приложениях его довольно часто используют. Но реализовывать Вы его будете точно не при помощи сервлетов. Хотя о чем здесь говорить – это уже восьмое задание, которое Вы делаете, и, уверен, у Вас в голове уже прочно обосновалась мысль, что ничего в этой жизни Вы не будете делать при помощи сервлетов ☺. Итак, не будем больше нагнетать неопределенность, наш секретный, но крайне полезный компонет – это гистограмма! А задание звучит следующим образом:

Средствами сервлетов создайте динамическую Web-страницу «Гистограмма», которая в параметрах запроса получает целые числа, разделенные пробелами (например, ***«10 23 4 25 6 14 9»***) и строит соответствующую гистограмму. Внешний вид страницы может быть таким, как показано на рис. 1.

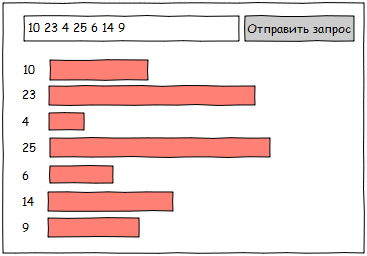


Рис. 1. Внешний вид приложения.

Давайте также, для разнообразия, клиентскую проверку корректности данных при помощи JavaScript-а в этом задании производить не будем, а вместо этого, заставим сервлет потреблять некорректные данные и в этих случаях, выводить вместо гистограммы соответствующие сообщения:

**«Вы не ввели исходные данные для построения диаграммы!»** – такое сообщение должно отобразиться, если пользователь не ввел ни одного числа в текстовое поле, либо ввел только пробельные символы.

**«Как минимум одно введенное значение некорректно!»** – такое сообщение должно отобразиться, если пользователь вместо хотя бы одного числа ввел некорректные символы, из которых число получить невозможно.

При этом обратите внимание, что если в текстовое поле были введены корректные целые числа, но между ними располагается больше одного пробельного символа, либо несколько пробельных символов расположены в начале или в конце строки, то диаграмма в этом случае **должна быть построена**.

Либо другими словами, более коротко – корректными данными для Вашей диаграммы должна стать последовательность из **одного и более целых чисел**, каждое из которых отделено пробельными символами. При этом количество и расположение пробельных символов не должно оказывать влияния на построение диаграммы.

И последний момент в текущей задаче. Давайте подробнее рассмотрим слова первого абзаца – *«…технологии, которая напрямую этого не поддерживает, создать компонент, который нам никогда не потребуется…»*. Уверен, что со вторым заявлением Вы уже готовы спорить.

Действительно, графики, гистограммы, и прочие графические элементы в приложениях способны нагляднее представить пользователям количественные данные любой системы. Более того, при таком способе подачи информации пользователям проще увидеть отношения между числовыми данными, или, например, отслеживать динамику их изменения. Соответственно, вывод один – гистограммам быть!

Остается только первая часть предложения – «*…технологии, которая напрямую этого не поддерживает…*». Да, действительно, это так. И здесь я даже не стану по обыкновению напоминать Вам, что сервлеты…

«…это стандарт, по которому можно мерить любой отстой!»

(О, спасибо тебе, Раджеш Кутрапалли, за эту, незаменимую во многих случаях, фразу)

Здесь, скорее, стоит говорить о том, что ***вся*** Java платформа не пригодна для подобного рода изобразительного искусства. Друзья, очень скоро Вы узнаете (хотя, скорее всего, Вы и сейчас догадываетесь), что все Web-приложения делятся на две большие части:

1. **Клиентская часть** **–** это внешний вид приложения.
2. И **Серверная часть** – бизнес-логика этого же приложения.

Так вот приложение будет хорошим тогда, когда каждая платформа будет заниматься своей работой. И поверьте, клиентская часть – не самая сильная сторона Java.

Да, в Java мире, например, существует такой фреймворк как GWT ([Google Web Toolkit](http://www.gwtproject.org/)), который производит и поддерживает одна маленькая, никому не известная [компания](http://www.google.com). При помощи этого фреймворка можно отрисовать вообще ***весь GUI вашего приложения*** на Java! И он скомпилируется в кроссбраузерный JavaScript! Но, пожалуйста, поверьте человеку, который составляет для Вас эту задачу как раз после того, как на основной работе «***отГэВэТэшил***» по-полной – как раз в тот момент, когда Вы получаете непередаваемое эстетическое удовольствие от отрисовки вообще ***всего GUI вашего приложения*** на Java… Вот именно в этот момент Вам почему-то больше всего хочется отрезать себе руки по самые ноги, чтобы всю жизнь получать пособие и не программировать больше никогда. Эта задача составлялась в 2014 году, и многие могут сказать, что к этому времени в GWT уже появился [UIBinder](http://www.gwtproject.org/doc/latest/DevGuideUiBinder.html), а, следовательно, автор чрезмерно драматизирует реальное положение вещей. Но, знаете, если в какой-то момент Вам хочется отрезать руки чуть короче, чем вчера – это все еще означает, что корень проблемы пока не решен ☺. Когда-то на одном из форумов, посвященных GWT, я увидел замечательную фразу. К сожалению, наизусть я ее не помню, а загуглить этот форум по обрывкам памяти – чего-то никак не получается. Но суть фразы в следующем:

«…GWT используют в тех Java командах, где жалеют денег на настоящих Web-девелоперов…»

Ну, согласитесь – шедеврально ☺. Однако мы отвлеклись, давайте сформулируем вывод:

Каждая технология хороша в своей стихии. И клиентская сторона Web-приложений – это не стихия для Java.

О, кстати, еще одна цель текущего задания – на собственном опыте убедиться, что графики Jav-ой лучше не рисовать. Так сказать, один раз сделаем, внимательно посмотрим, и пообещаем себе больше никогда не лепить из этого конфетку.

Но, блин, даже после вывода мы все равно не приблизились к завершению практического задания, так как все еще не озвучили тот самый *«последний момент»* в этой задаче. А он вот какой:

Друзья, скорее всего, во время выполнения этой задачи Вас заинтересует следующий вопрос – «А при помощи каких клиентских технологий можно порисовать диаграммы на Web-страницах?» И с этим вопросом Вы пойдете в Google, и увидите там просто сногсшибательные решения:

* И от [Jquery](http://2web-master.ru/10-besplatnyx-jquery-plaginov-dlya-sozdaniya-grafikov.html) – одного из популярнейших фреймворков по JavaScript-у
* И от [Bootstrap-а](http://getbootstrap.com/components/#progress-low-percentages) – На мой взгляд – лучшего CSS фреймворка.
* И от [Googl-а](https://google-developers.appspot.com/chart/interactive/docs/index) – Ну что Вы хотите, чтобы я здесь добавил? ☺
* И даже просто от [HTML-5](http://www.designonstop.com/tutorials/my-tuts/20-manualov-po-sozdaniyu-grafikov-i-diagramm-s-pomoshhyu-html5-i-css3.htm)!

После всего увиденного у Вас обязательно пропадет желание рисовать гистограмму средствами классического HTML и захочется использовать для этих целей одну из технологий, приведенных Выше. Ну или ту, которую сами нагуглите. И это очень хорошо – значит, Ваш мозг функционирует как нужно ☺.

Но беда заключается в том, что работа, в которой Вы нарисуете гистограмму любой из «продвинутых» Web-технологий ***приниматься не будет***. Почему? А как же в начале задачи – *«пораскинуть мозгами»*? Согласитесь, нарисовать гистограмму при помощи API-шки, которая для этого предназначена – дело пары пустяков. Соответственно, пропадает элемент творчества, этакого, бинарного вдохновения, если хотите. Ну а согласитесь, какое может быть обучение, если нет вдохновения. Вот я Вам и помогаю его найти ☺.

Но Ваше возражение так же вполне резонно. Скорее всего, оно будет таким – то, что мы сделаем самостоятельно, за пределами этого практического занятия не понадобится нам никогда. А изучить способы рисования с использованием популярных Web-технологий - это большой плюс в резюме!

Да, я с этим абсолютно согласен. И поэтому, на Вашем месте я бы выполнил текущую работу в компромиссном варианте. Добиться этого можно следующим образом – после того, как Вы нарисуете гистограмму самостоятельно, погуляйте по ссылкам выше, посмотрите, какие возможности предлагают Вам Web фреймворки, да и, черт возьми, добавьте в сервлет еще 2-3 гистограммы ниже своей! Только, если будете так делать, пожалуйста, обязательно подпишите на странице, при помощи какой технологии нарисована каждая гистограмма, а то, по правде сказать, Ваш преподаватель по части Web-технологий – большая лошара ☺

Итак, предпоследнее требование о том, что гистограмма должна быть реализована с использованием классических Web-технологий – обязательно. Предпоследнее требование, о том, что Вы так же можете попробовать рисование при помощи специализированных API – по желанию.

Вот вроде и все. Дамы и господа, успехов Вам и вдохновения во время выполнения этой работы!