**Servlet API.**

**Задание-2.**

**Электронная анкета для оценки качества образования.**

Вторая задача по сервлетам практически полностью повторяет первую – по крайней мере, Вам не придется верстать ничего нового. Страницы должны быть те же самые, но теперь обе страницы

**Должны быть выполнены при помощи сервлетов.**

«Зачем повторять одно и тоже задание дважды?» – спросите Вы.

Ответ здесь очень прост. Основная цель, которая преследуется нами во второй задаче – показать разницу между **HTML-страницей**, как банальным текстовым документом, который все, что может – это вернуться от сервера и отобразиться в браузере и **сервлетом** – полноценным серверным компонентом бизнес-логики, который кроме отображения может еще принимать входящие параметры, работать с атрибутами, выполнять Java-код и еще много чего.

Вторая цель, которой хотелось бы достичь – показать распространенный подход, к программированию клиент-серверных приложений, при котором клиент, отправивший запрос (в нашем случае – это сервлет с формой), может получить этот запрос обратно. Например, тогда, когда сервер, принявший запрос (в нашем случае – это сервлет, отображающий результаты), проводя валидацию данных обнаружит ошибки, которые не позволяют ему работать дальше.

Именно это Вам и нужно реализовать. Итак:

1. Первый сервлет (форма) отправляет запрос на второй сервлет.
2. Второй сервлет принимает запрос и начинает валидировать данные.
3. Если второй сервлет обнаруживает ошибку (или ошибки) в данных, то он не отображает свой пользовательский интерфейс, а собирает все ошибки в виде карты атрибутов, и возвращает управление первому сервлету.
4. В этой ситуации первый сервлет:
   * Заново рисует форму.
   * Располагает все ранее введенные значения, в тех полях ввода, которые пользователь заполнил правильно.
   * А все компоненты, которые были заполнены неверно, оставляет пустыми и рядом с ними выводит сообщение об ошибке.
5. Если же второй сервлет не находит ни одной ошибки во входящих данных, то он просто рисует свою страницу так же, как и в задаче №1.

Для того, чтобы была возможность проверить этот функционал, пожалуйста, добавьте на форму checkbox «Отключить JavaScript валидацию», также как и в первой задаче.

Ну, и наконец, последняя, но почти самая главная цель этой работы – с первых же минут нашего знакомства с сервлетами привить Вам стойкую неприязнь к этой технологии, как к ***возможности генерирования пользовательского интерфейса для Web-страниц***.

***Примечание****: Обратите внимание на полужирный курсив выше – мы не ставим перед собой цель возненавидеть сами сервлеты. Не забывайте сервлеты – фундаментальный компонент технологии JEE, ее ядро. Без них невозможно функционирование Web-приложений на Java платформе. Но желание создавать GUI при помощи сервлетов у Вас должно пропасть навсегда. Я уверен, что если к практическим заданиям текущего занятия прибавить те альтернативные технологии, которые мы рассмотрим с Вами на будущих занятиях, мы сможем добиться поставленной цели по развитию контролируемой ненависти к отдельным техническим возможностям Servlet API.*