

Когда клиент переходит по адресу URL, который обрабатывается сервлетом, контейнер сервлета перехватывает запрос и вызывает метод **doGet()** или **doPost()**. Эти методы вызываются после конфигурации объектов, наследующих интерфейсы **HttpServletRequest**, **HttpServletResponse**. Задача методов **doGet()** и **doPost()** – взаимодействие с HTTP-запросом клиента и создание HTTP-ответа, основанного на данных запроса. Метод **getWriter()** объекта-ответа возвращает поток **PrintWriter**, который используется для записи символьных данных ответа.

### Интерфейсы **ServletResponse** и **HttpServletResponse**

Генерируемые сервлетами данные пересылаются серверу-контейнеру с помощью объектов, реализующих интерфейс **ServletResponse**, а сервер, в свою очередь, пересылает ответ клиенту, инициировавшему запрос.

Можно получить ссылки на потоки вывода одним из двух методов:

**ServletOutputStream** **getOutputStream()** – извлечение ссылки на поток **ServletOutputStream** для передачи бинарной информации;

**PrintWriter** **getWriter()** – извлечение ссылки на поток типа **PrintWriter** для передачи символьной информации;

Если метод **getOutputStream()** уже был вызван для этого ответа, то генерируется **IllegalStateException**. Обратное также верно.

В интерфейсе **HttpServletResponse**, наследующем интерфейс **ServletResponse**, есть еще несколько полезных методов:

**void** **setContentType(String type)** – установка MIME-типа генерируемых документов;

**void** **addCookie(Cookie c)** – добавление cookie к объекту ответа для последующей пересылки на клиентский компьютер;

**void** **sendError(int sc, String msg)** – сообщение о возникших ошибках, где **sc** – код ошибки, **msg** – текстовое сообщение;

**void** **setDateHeader(String name, long date)** – добавление даты в заголовок ответа;

**void** **setHeader(String name, String value)** – добавление параметров в заголовок ответа. Если параметр с таким именем уже существует, то он будет заменен.

### Обработка запроса

Распределенное приложение может быть эффективным только в случае, если оно способно принимать информацию от физически удаленных клиентов. В следующем примере сервлет извлекает данные пользовательской формы, переданные вместе с запросом по методу **GET**.

Приведенная на рисунке 18.1 форма является результатом отображения JSP-страницы **index.jsp**, находящейся в папке **/webapps/FirstProject3**.

В форме имеется текстовое поле с именем **name** и значением по умолчанию «**Название проекта**». Значение поля можно изменить непосредственно на странице.