Алгоритм

Сервлет Java – клас, призначений для обробки викликів HTTP і генерації вмісту веб-сторінки. Сервлети є спеціалізованим механізмом Java для створення Web ресурсів. Для розробки сервлетів використовується пакет javax.servlet.

Всі призначені для користувача сервлети підтримують інтерфейс javax.servlet.Servlet. Даний інтерфейс визначає методи, які керують сервлетом і його спілкуванням з клієнтом. При написанні нового сервлету, я реалізувував потрібні методи цього інтерфейсу. Всього інтерфейс містить п’ять методів:

1. public void init(ServletConfig config) throws ServletException – даний метод викликався в момент першого завантаження сервлету. Метод дозволяє здійснити ініціалізацію ресурсів (відкриття фалів, установка з’єднань з віддаленими серверами, і т.п.). Гарантується, що метод init() закінчиться перед будь-яким іншим зверненням до сервлету. Параметр config містить відомості про конфігурацію сервлету.
2. public ServletConfig getServletConfig () – використовувався для отримання системної інформації про конфігурацію сервлету.
3. public String getServletInfo () – використовувався для повернення рядка, що містить загальні відомості про сервлет.
4. public void service (ServletRequest request, ServletResponse response) throws ServletException, java.io.IOException – метод service() був серцем сервлету. Кожен запит від клієнта призводить до одного виклику методу service(). Цей метод читав запит і формував відповідь за допомогою своїх двох аргументів ServletRequest і ServletResponse:

* Об’єкт ServletRequest містив параметри HTTP запиту (дані, отримані від клієнта).
* Об'єкт ServletResponse містив відповідь сервлету клієнта.

Робота методу service() по суті проста – він створював відповідь на кожен клієнтський запит, переданий йому з сервера. Однак необхідно пам’ятати, що існували кілька паралельних запитів, що оброблялися в один і той же час. Якщо метод service() вимагав яких-небудь зовнішніх ресурсів, таких як файли, бази даних, то необхідно було гарантувати, щоб доступ до ресурсів був потокозахищеним.

1. public void destroy () – метод destroy() викликався для звільнення всіх ресурсів (наприклад, відкриті файли і з’єднання з базою даних) перед вивантаженням сервлету. Цей метод міг бути порожнім, якщо не було необхідності виконання будь-яких завершальних операцій.

Пакет javax.servlet містить два абстрактних класу, які реалізують інтерфейс Servlet: клас GenericServlet (з пакета javax.servlet) і клас HttpServlet (з пакета javax.servlet.http). Ці класи надають реалізацію за замовчуванням для всіх методів інтерфейсу Servlet. Сервлети, що працюють з клієнтами через Web, в даній лабораторній розширюють клас HttpServlet. Метод service перевизначався, щоб була можливість розрізняти стандартні методи запитів, одержувані від Web-браузера клієнта. Двома найбільш поширеними типами запитів HTTP (їх також називають методами запитів) є get і post. Запит get отримує (або витягує) інформацію. Запит post поміщає (або відправляє) дані на сервер. Типове застосування методу post – відправка на сервер інформації для аутентифікації, або даних з форми, в яку користувач ввів інформацію. У класі HttpServlet зокрема були визначені методи doGet і doPost для реакції на запити типу get і post клієнта. Ці методи викликалися методом service класу HttpServlet, який, в свою чергу, викликався при надходженні запиту на сервер. Метод service спочатку визначав тип запиту, а потім викликав відповідний метод. Методи doGet і doPost приймали в якості параметрів об’єкти HttpServletRequest і HttpServletResponse, які дають можливість здійснювати взаємодію між клієнтом і сервером. Методи інтерфейсу HttpServletRequest полегшують доступ до даних запиту. Методи інтерфейсу HttpServletResponse полегшують повернення результатів Web-клієнта у вигляді HTML.

У проекті було створено два нових файлу: файл з вихідним кодом класу нового сервлету, успадкованого від класу HttpServlet, і дескриптор розгортання web.xml. Дескриптор розгортання описував класи, ресурси і конфігурацію застосунку. Даний дескриптор використовувався веб-сервером, зокрема, для зіставлення зазначеного в запиті шляху URL з певним сервлетом.