1. Что такое сервлет? (Ответ) Сервлет - это класс в языке Java, который расширяет возможности веб-сервера для обработки запросов от клиента и возвращения ответа на эти запросы.
2. Для чего нужен web-сервер? (Ответ) Web-сервер - это программное обеспечение, которое отвечает за обработку и передачу запросов от клиентов (обычно браузеров) к серверу, на котором хранится веб-сайт.
3. Расскажите о принципах построения и работы сервлета? Объясните общее взаимодействие клиентской и серверной сторон. (Ответ) Сервлеты построены на основе API Servlet, который обеспечивает обработку запросов HTTP и обновление ответов. Когда клиент (например, браузер) отправляет запрос на сервер, web-сервер обрабатывает этот запрос и передает его на соответствующий сервлет. Сервлет обрабатывает запрос и создает ответ, который возвращается клиенту.
4. Чем отличаются методы doGet и doPost друг от друга? В чем разница между методами GET и POST? (Ответ) Методы doGet и doPost - это методы, используемые в сервлетах для обработки HTTP-запросов. Разница между ними заключается в том, что метод doGet используется для запросов GET, тогда как метод doPost используется для запросов POST. Метод GET передает параметры в URL строке, тогда как метод POST передает параметры в теле запроса.
5. Назовите методы жизненного цикла сервлета. В каком порядке они вызываются? (Ответ) Методы жизненного цикла сервлета:

* init()
* service()
* destroy() Они вызываются в следующем порядке: init() - > service() - > destroy().

1. Как определить name и URL сервлета? (Ответ) Name сервлета определяется в дескрипторе развертывания сервлета (web.xml). URL сервлета состоит из имени сервера, порта и пути до сервлета.
2. Как получить значение параметра формы в сервлете? (Ответ) Значение параметра формы можно получить, используя методы объекта HttpServletRequest. Например, req.getParameter("parameterName") возвращает значение параметра с именем "parameterName".
3. Как сервлет возвращает результат своей работы на сторону клиента? (Ответ) Сервлет может возвращать результат своей работы на сторону клиента с помощью объекта типа HttpServletResponse, который предоставляет методы для установки заголовков ответа и отправки содержимого ответа. Например, можно установить тип контента, отправить данные в поток вывода и выполнить перенаправление на другую страницу.
4. Как происходит развертывание сервлета? (Ответ) Развертывание сервлета происходит путем добавления файлов сервлета в каталог веб-приложения на веб-сервере, после чего веб-сервер сканирует файлы и инициализирует сервлеты. Затем веб-сервер автоматически создает экземпляры сервлетов и обрабатывает запросы от клиентов, направляя их на соответствующий сервлет.
5. Для чего нужен интерфейс ServletContext? (Ответ) Интерфейс ServletContext предоставляет доступ к общим ресурсам, связанным с веб-приложением, таким как параметры контекста, файлы JSP и HTML, объекты, которые могут использоваться в нескольких сервлетах и другие данные, необходимые для работы веб-приложения.
6. Что такое атрибут контекста? (Ответ) Атрибут контекста - это объект, который хранится в ServletContext и может быть использован для обмена данными между сервлетами или JSP-страницами. Атрибут контекста может быть установлен, получен или удален из ServletContext.
7. Какую информацию можно получить из контекста сервлета? (Ответ) Информацию, которую можно получить из контекста сервлета, включает параметры контекста, объекты, которые могут использоваться в нескольких сервлетах, файлы JSP и HTML, логгеры и другие данные, необходимые для работы веб-приложения.
8. Опишите интерфейс ServletConfig? (Ответ) Интерфейс ServletConfig используется для получения информации о конфигурации сервлета из его описания в deployment-дескрипторе веб-приложения (web.xml). ServletConfig содержит информацию о параметрах инициализации сервлета, таких как URL-адрес, настройки контекста и другие свойства, которые могут использоваться для настройки работы сервлета.
9. Какую информацию содержит интерфейс HttpServletRequest? (Ответ) Интерфейс HttpServletRequest предоставляет методы для получения информации о запросе, такие как параметры запроса, заголовки, метод HTTP, путь к ресурсу и другие сведения о запросе.
10. Как можно использовать интерфейс HttpServletResponse? (Ответ) Интерфейс HttpServletResponse предоставляет методы для управления откликом сервера, такие как установка статуса ответа, отправка заголовков ответа и отправка содержимого ответа. Эти методы могут использоваться для формирования и отправки ответа на клиентский запрос.
11. Как выполняется переадресация запросов и как перенаправление откликов? В чем принципиальная разница? (Ответ) Переадресация запросов и перенаправление откликов - это два разных процесса. Переадресация запроса происходит на стороне сервера и происходит путем отправки нового запроса клиенту с новым URL, который указывает на другой ресурс. При этом клиент переходит по новому URL и делает новый запрос. Перенаправление отклика происходит на стороне сервера и происходит путем изменения статуса ответа и отправки нового содержимого в ответ на клиентский запрос.
12. Как можно сохранить данные приложения между запросами? (Ответ) Данные приложения можно сохранять между запросами, используя различные механизмы, такие как атрибуты контекста, сессии и файлы cookie. Например, данные можно сохранять в атрибутах контекста сервлета, которые могут быть доступны для всех сервлетов, использующих данный контекст.
13. Как получить объект HttpSession и что она содержит? (Ответ) Объект HttpSession представляет сессию, связанную с текущим клиентом. Его можно получить из объекта HttpServletRequest, вызвав метод getSession(). HttpSession содержит данные, связанные с текущей сессией, которые могут использоваться между запросами, например, для сохранения состояния приложения или для аутентификации пользователя.
14. Что такое файлы Cookie? Как его создать и поместить информацию? (Ответ) Файлы Cookie представляют собой механизм хранения информации на стороне клиента в виде пары ключ-значение. Куки можно создать и отправить на клиентскую сторону, используя объект HttpServletResponse и метод addCookie(). Например, чтобы создать файл Cookie с именем "username" и значением "John", можно использовать следующий код:

Cookie cookie = new Cookie("username", "John"); response.addCookie(cookie);

1. С какой целью используют интерфейс Filter? (Ответ) Интерфейс Filter используется для обработки запросов и/или ответов сервера до их достижения сервлета или после его завершения. Он позволяет изменять параметры запроса или ответа, фильтровать их, обрабатывать ошибки и т.д. Фильтры могут использоваться для обеспечения безопасности, сжатия, кэширования, логирования и других целей.
2. Перечислите методы жизненного цикла фильтра? (Ответ) Методы жизненного цикла фильтра:

* init(FilterConfig filterConfig): вызывается при инициализации фильтра для передачи ему конфигурационной информации;
* doFilter(ServletRequest request, ServletResponse response, FilterChain chain): вызывается для обработки каждого запроса;
* destroy(): вызывается при завершении работы фильтра для освобождения ресурсов.