1. Что такое сервлет ? 2. Для чего нужен web-сервер? 3. Расскажите о принципах построения и работы сервлета? Объясните общее взаимодействие клиентской и серверной сторон. 4. Чем отличаются методы doGet и doPost друг от друга? В чем разница между методами GET и POST? 5. Назовите методы жизненного цикла сервлета. В каком порядке они вызываются? 6. Как определить name и URL сервлета? 7. Как получить значение параметра формы в сервлете? 8. Как сервлет возвращает результат своей работы на сторону клиента? 9. Как происходит развертывание сервлета? 10.Для чего нужен интерфейс ServletContext? 11. Что такое атрибут контекста? 4 12. Какую информацию можно получить из контекста сервлета? 13.Опишите интерфейс ServletConfig? 14. Какую информацию содержит интерфейс HttpServletRequest? 15. Как можно использовать интерфейс HttpServletResponse? 16. Как выполняется переадресация запросов и как перенаправление откликов? В чем принципиальная разница? 17.. Как можно сохранить данные приложения между запросами? 18.Как получить объект HttpSession и что она содержит? 19. Что такое файлы Cookie? Как его создать и поместить информацию информацию? 20. С какой целью используют интерфейс Filter? 21. Перечислите методы жизненного цикла фильтра

1. Сервлет - это компонент программного обеспечения, написанный на языке Java, который запускается на сервере и обрабатывает HTTP-запросы от клиентов.
2. Веб-сервер нужен для обслуживания запросов от клиентов, отправки ответов на эти запросы и управления статическими и динамическими ресурсами сайта.
3. Сервлеты строятся на основе Java-классов, реализующих интерфейс Servlet. Сервлеты принимают HTTP-запросы от клиентов, обрабатывают их и генерируют HTTP-ответы. Клиент и сервер взаимодействуют с помощью протокола HTTP.
4. Методы doGet и doPost - это методы HTTP, используемые для отправки данных на сервер. Основное отличие между ними заключается в том, что метод doGet отправляет данные через URL, тогда как метод doPost отправляет данные в теле запроса.
5. Методы жизненного цикла сервлета включают init(), service() и destroy(). Они вызываются в следующем порядке: init() при создании сервлета, service() для обработки запросов клиентов и destroy() при закрытии сервлета.
6. Имя сервлета и его URL определяются в файле web.xml или аннотацией @WebServlet.
7. Значение параметра формы можно получить с помощью метода getParameter() объекта HttpServletRequest.
8. Результат работы сервлета может быть возвращен клиенту с помощью объекта HttpServletResponse, который используется для генерации HTTP-ответа.
9. Развертывание сервлета происходит путем упаковки Java-класса в файл war или jar и размещения этого файла в директории webapps веб-контейнера.
10. Интерфейс ServletContext используется для обмена данными между сервлетами, настройки параметров веб-приложения и управления ресурсами.
11. Атрибут контекста - это объект, хранящий информацию, общую для всех сервлетов в контексте приложения. Он доступен через интерфейс ServletContext и может быть установлен, изменен и удален во время выполнения веб-приложения.
12. Из контекста сервлета можно получить информацию о параметрах и атрибутах контекста, доступных ресурсах, настройках и конфигурации веб-приложения.
13. Интерфейс ServletConfig используется для получения информации о конфигурации сервлета, включая его имя, описание, параметры и контекст.
14. Интерфейс HttpServletRequest содержит информацию о запросе, включая параметры, заголовки, метод, URL, куки, сессию и тело запроса.
15. Интерфейс HttpServletResponse используется для генерации HTTP-ответов на запросы клиентов, включая установку статуса ответа, заголовков и тела ответа.
16. Переадресация запросов выполняется с помощью метода sendRedirect() объекта HttpServletResponse. При перенаправлении откликов используется метод forward() объекта RequestDispatcher. Основное отличие заключается в том, что при переадресации запроса клиенту отправляется новый запрос с новым URL, тогда как при перенаправлении отклика отправляется тот же запрос с измененным телом и заголовками.
17. Данные приложения могут быть сохранены между запросами с помощью объекта HttpSession, куки или параметров запроса.
18. Объект HttpSession может быть получен с помощью метода getSession() объекта HttpServletRequest и содержит информацию о сессии клиента, включая атрибуты и параметры.
19. Файлы Cookie - это небольшие файлы, хранящие информацию на стороне клиента, которые используются для сохранения информации между запросами. Чтобы создать и установить куки, можно использовать метод addCookie() объекта HttpServletResponse.
20. Интерфейс Filter используется для обработки запросов до и после выполнения сервлетов, а также для манипуляции с данными запроса и ответа.
21. Методы жизненного цикла фильтра включают init(), doFilter() и destroy(). Они вызываются в следующем порядке: init() при создании фильтра, doFilter() для обработки запросов клиентов и destroy() при закрытии фильтра.