**Задание 3. Authorization Filter.**

Реализуйте класс фильтра, который будет отрабатывать при запросах к определенным страницам Вашего приложения, ***доступ к которым возможен только после того, как пользователь пройдет процедуру авторизации***.

Пусть класс называется **AuthorizationFilter**. Его задача – при поступлении к нему запроса, обратиться к сессии пользователя и проверить, присутствует ли в сессии атрибут, подтверждающий, что ранее авторизация была выполнена пользователем.

Если авторизация ранее проводилась, и пользователь имеет право доступа к защищенной странице, фильтр должен просто передать управление далее по цепочке. В противном случае, фильтр должен перенаправить пользователя на страницу авторизации, используя механизм **forward** объекта **RequestDispatcher**.

Добейтесь того, чтобы ваш Фильтр был универсальным, т. е. мог бы использоваться в других проектах. Для этого организуйте возможность передачи в фильтр дополнительной информации при помощи init-параметров. В качестве такой информации должны выступить следующие параметры:

1. Имя атрибута сессии, который может подтвердить, что авторизация выполнялась ранее. Имя для этого параметра может быть, например, **authArrributeName**. В качестве значения этого параметра (строка в тэге <param-value>) разработчик сможет указать имя атрибута, который необходимо искать в сессии для подтверждения факта авторизации.
2. URL Web-компонента (сервлет, JSP страница, и т. д.), который реализует страницу авторизации, и на который фильтр должен передать управление в случае, когда пользователь не авторизован. Этот параметр может называться **loginPageUrl**, и его значением может быть, например? **«/login.do»**.

Обратите внимание на то, что в рамках основной части задания не предполагается разработка самой страницы авторизации. Суть задания как раз в том, чтобы реализовать универсальный механизм проверки авторизованных пользователей, который можно было бы внедрить в любое приложение, в котором разработчиками уже создана страница авторизации.

Так же фильтр должен выбрасывать исключения (написанные Вами же), в следующих случаях:

1. Исключение **IncorrectAuthAttributeException** должно выбрасываться в том случае, если разработчик не сконфигурировал параметр **authArrributeName** (см. выше).
2. Исключение **IncorrectLoginPageUrlException** должно выбрасываться в том случае, если разработчик не сконфигурировал параметр **loginPageUrl** (см. выше).

Оба исключения должны наследоваться от **RuntimeException** и только выбрасываться Вашим фильтром в некорректных ситуациях, а не обрабатываться им. Я бы рассказал Вам больше об исключительных ситуациях, но, боюсь, это плохо для меня кончится ☺.

Реализовав фильтр, напишите так же небольшое Web-приложение, которое позволит продемонстрировать его работу.