Видео 1.

Framework vs. Library

Dependency Injection

Inversion of Control

Core container

* Beans
* Core
* Context
* Expression

AOP

* AOP
* Aspects

Instrumentation

* Instrument – Spring Boot??

Data access and integration

* JDBC
* JMS
* ORM
* Transaction

Web and Remoting – MVC тут

* Web
* Servlet
* Struts

Test

* Testing

Модули:

spring-context

spring-context-support

spring-tx - транзакции

spring-jdbc

Вариант приложения 1 – Client – имеет id и name, ConsoleEventLogger – имеет метод logEvent, который выводит сообщение на консоль, App – создает Client и ConsoleEventLogger внутри себя и пользуется их методами.

Проблемы:

1. Сложно редактировать – нужно перекомпилировать при изменении данных
2. Плохое масштабирование – только один логгер, который жестко зашит в приложение. Для изменения нужно перекомпилировать.
3. Сложно тестировать – при тестировании App.eventLogger будет тестироваться и ConsoleEventLogger.

Причина всех проблем – жесткое связывание (tight coupling).

Видео 2.

Решение проблем:

1. Всю статическую информацию лучше вынести во внешние файлы.
2. Сделать интерфейсную развязку - реализация должна зависить от интерфейсов, а не от классов. Но интерфейсы не нужны для классов, которые только хранят информацию (как Client).
3. Внедрение зависимостей – вместо создания экземпляров вручную, просим это сделать у Spring.

SVN???