Обзор
— XML
— DOM — объектная модель документа
— DOM/SAX парсер
— Reflection API
— Класс Class
— Annotation
— примеры использования

Вебинар proglive.box.com/s/1zy80iaqlcod7e59hdtx

Материалы

XML:

Чтение XML — javaswing.wordpress.com/2010/03/14/java_dom_xml/

DOM/SAX — www.quizful.net/post/getting-started-with-xml-in-java

Reflection:

docs.oracle.com/javase/tutorial/reflect/

www.javenue.info/post/84

Annotation:

habrahabr.ru/post/139736/

habrahabr.ru/company/golovachcourses/blog/217595/

Домашнее задание

Разобраться с SAX/DOM парсерами (есть в материалах). Модифицировать DOM парсер, чтобы он печатал разобранный XML документ с правильными отступами для вложенных элементов

```
<a id="1">
    <b name="B"/>
    <b name="BB">
        <c data="C"/>
    </b>
    <b name="D"/>
</a>
```

```
// отступы для вложенных, аттрибуты в квадратных скобках
node: a [id=1]
	node: b [name="B"]
	node: b [name="BB"]
		node: c [data="C"]
	node: b [name="D"]
```

Разобраться в коде IoC контейнера и добавить новый функционал.

1. Заполнять приватные поля с помощью методов get()/set(). Сейчас ищется поле класса с тем же именем, что и в XML. Это поле инициализируется заданным значением (даже если оно приватное). Если соответствующего метода нет, выбросить исключение. Использовать reflection API для доступа к методу и вызова.

```
// XML code
<property name="count" val="10"/>

// java
class Sample {
	// для поля с заданным именем ищем соответствующий метод set<здесь имя поля с большой буквы>()
	private int count;

	// такой метод есть!
	public void setCount(int count) {
		this.count = count;
	}
}
```

2. Аннотация @PostConstruct

Это аннотация runtime и может быть применена для метода. Так как в нашей модели инициализации объектов мы обходимся пустым конструктором и устанавливаем только поля, нам может потребоваться метод init(), в котором можно провести инициализацию объекта. Считаем, что этот этап должен исполнятся после того, как все property были прочитаны. Чтобы не вызывать метод init() вручную, используем аннотацию @PostConstruct. Метод помеченный такой аннотацией должен вызваться после

инстанцирования всех полей (если такой метод есть). Допустим только один метод с такой аннотацией в классе (если больше одного — кидать исключение)
* имя метода может быть любым, init() — только для примера

```
class Sample {
	// автоматически инициализировать поле класса
	@Auto
	DbHelper helper;

	// метод set для установки поля
	public void setHelper(DbHelper helper) {
		this.helper = helper;
	}

	// этот метод будет вызван нашей системой, когда все поля класса будут инициализированы
	@PostConstruct
	public void init() {
		// какие-то операции по инициализации класса
		// например, создадим таблицы БД, если их нет
		if (!helper.hasTables()) {
			helper.createTables();
		}
	}

}
```

3. Сериализация

Сериализация
http://habrahabr.ru/post/60317/

http://www.skipy.ru/technics/serialization.html

Разобраться с механизмом сериализации, попробовать передать объект по сети (от сервера клиенту и наоборот)