**Inversion of control**

Паттерн абстрактная фабрика или Dependency Injection.

**Spring**

spEL.

Bean – настроенный экземпляр класса, управляемый спрингом.

Xmlconfig.xml

BeanDefinitionReader.

Scope=”singleton” – один bean на context, по умолчанию.

Prototype – beans создаются при запросе, не хранит в контексте, не дестроится.

@Autowired – проставить зависимости. <contex: annotation-config/>

Spring использует рефлексию.

BeanPostProcessor: <name>AnnotationBeanPostProcesor implemets BeanPostProcessing{

Field[] fields = bean.getClass().getFields();

Iter(){

Rand rand = field.getAnnotation(Rand.class);

if(rand!=null){

int min = rand.min;

int max = rand.max;

}

}

}

**Аннотация**

@Retention(ElementType.RUNTIME)

Public @interface Rand{

Int min();

Int max();

}

@interface Profiling

Before {

Class<?> sclass = map.get(beanName);

If sclass != null {

Proxy.newProxyInstance(sclass.getClassLoader(), sclass.getInterfaces() )

@Override invoke

}

}

@PostConstruct

@PreDestroy

DynamicProxy в java

@Transactional

Вызов класса:

1. Invoke constructor
2. postProcessorBefore
3. init-method
4. postProcessorBefore
5. destroy

ApplicationListener: Event