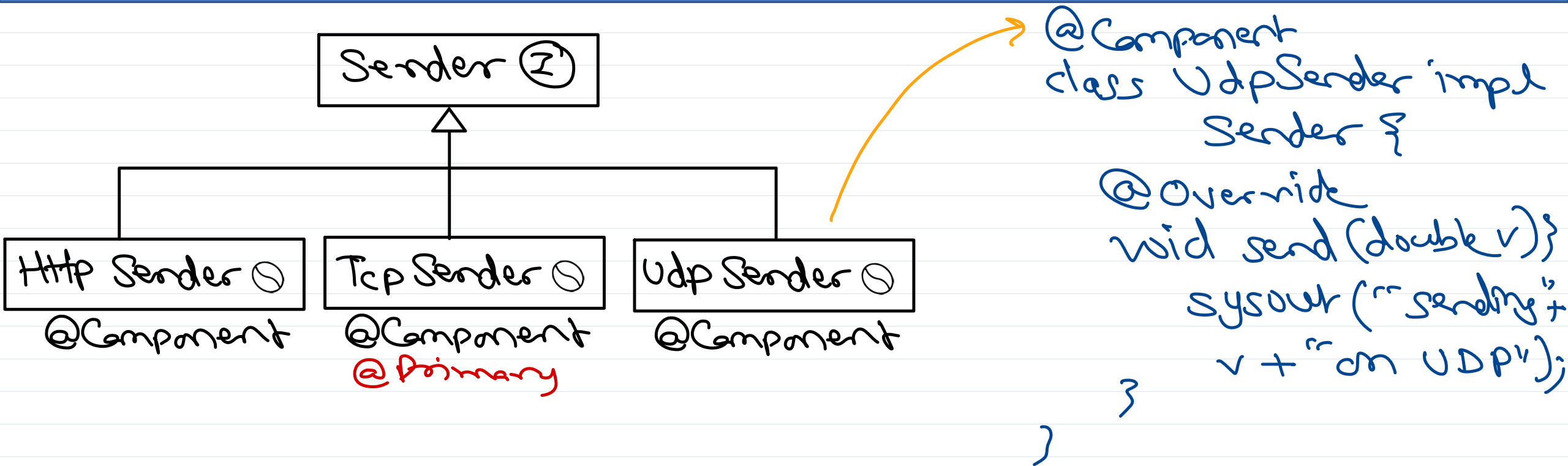




# Advanced Java

*Trainer: Nilesh Ghule*





# Spring Expression Language

\* Syntax:

# {expression}  
or \${expression}

\* Applications:

① access app'n properties/settings.  
@Value("\${key}")

② access spring bean fields.  
@Value("#{bean.field}")

③ access spring bean methods.  
@Value("#{bean.method()}")

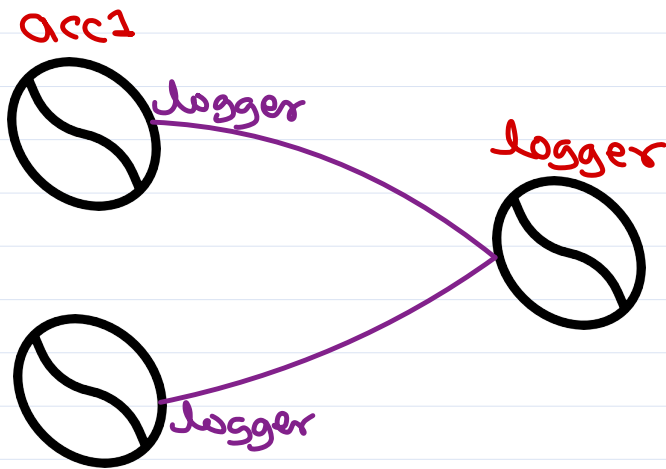
④ access spring bean.  
@Value("#{bean}")

⑤ perform arithmetic operations.  
@Value("#{bean.field+24}")



# Auto-wiring

Spring container can attach appropriate dependency bean into dependent beans → Auto-wiring.



acc2  
class AccountImpl...  
...

Logger logger;

}

XML config:

```
<bean id="logger" class="pkg.DbLogger">
```

```
</bean>
```

```
<bean id="acc1" class="pkg.AccountImpl"
      autowire="byType">
```

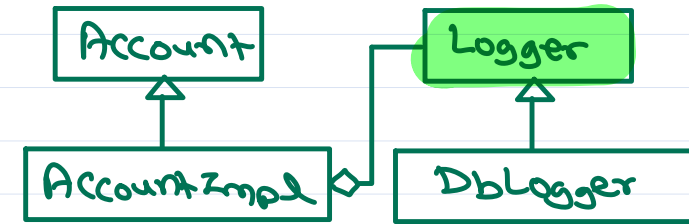
```
<property name="id" value="101"/>
```

```
<property name="type" value="Saving"/>
```

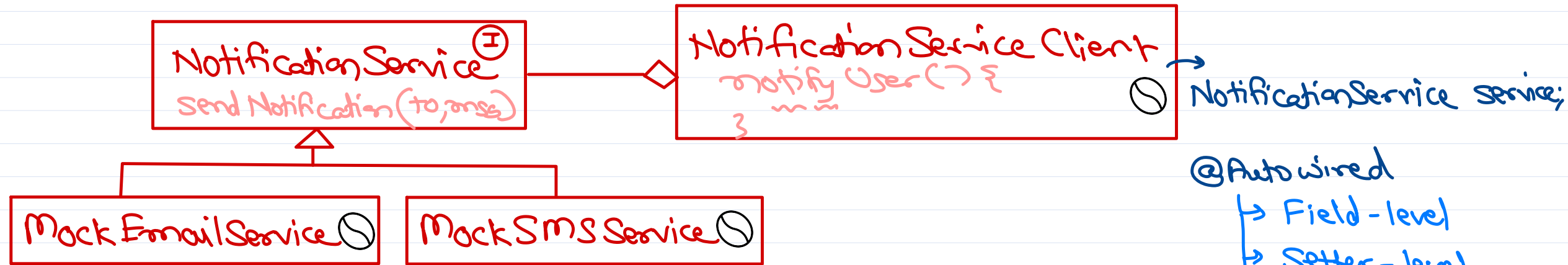
```
<property name="balance" value="5000"/>
```

```
</bean>
```

```
<bean id="acc2" class="pkg.AccountImpl"
      autowire="byName"> ... </bean>
```



# Auto-wiring



**@Autowired**

- Field-level
- Setter-level
- Constructor-level (recommended in spring docs).

@Autowired resolution (which dependency bean to be attached?):

- ① by type: Container find bean of given type. e.g. **NotificationService** service;
- ② by Qualifier: Container find bean of name given in `@Qualifier("...")`. The `@Qualifier` on field is always used with `@Autowired`,
- ③ by name: if multiple beans of same type avail, it will look for bean of given name.

Note: Spring 5 & above  
single arg ctors are  
auto considered as  
`@Autowired`.



# Bean Scopes

- ① singleton → Single bean object (instance) is created for given bean definition when Spring Container is ready. \* This is default.
  - ② prototype → No bean is created when container is ready. New bean is created on each access i.e. call to `ctx.getBean()`.
  - ③ request
  - ④ session
  - ⑤ application
  - ⑥ websocket
- Spring web mvc appln. Not destroyed by container during shutdown.

\* XML bean defn.

```
<bean id="b1" class="pkg.BoxImpl" scope="singleton">  
  </bean>  
OR  
scope="prototype"
```

return address of same bean.

\* Annotation bean defn

```
@Bean @Scope("singleton") or  
public Box b2() { @Scope("prototype")  
  ...  
}
```

Container calls destroy / PreDestroy for beans on shutdown.





*Thank you!*

Nilesh Ghule <nilesh@sunbeaminfo.com>

