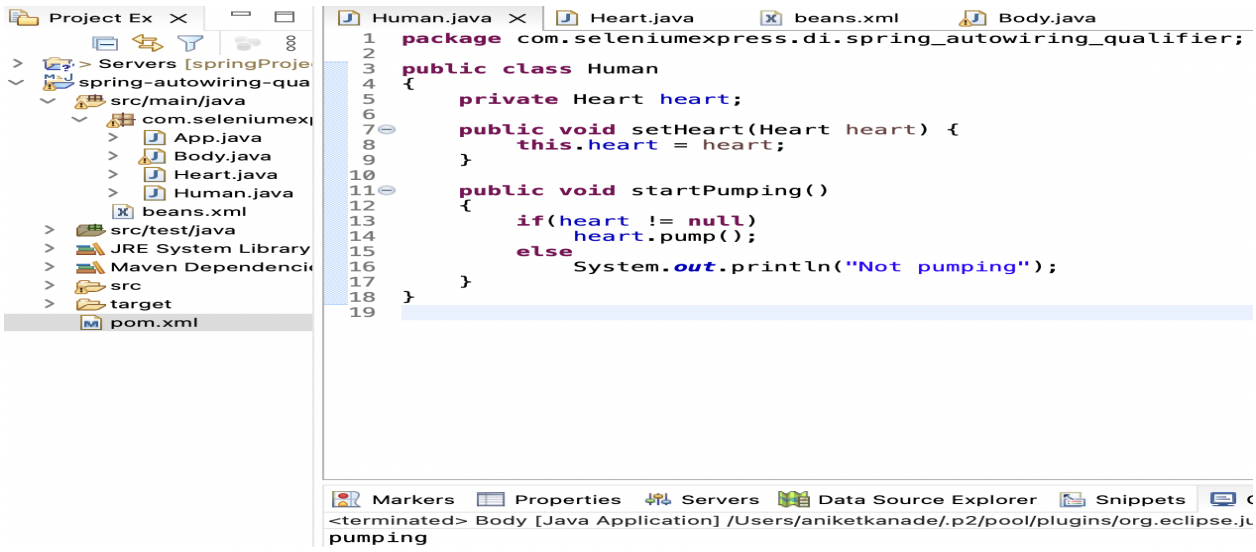
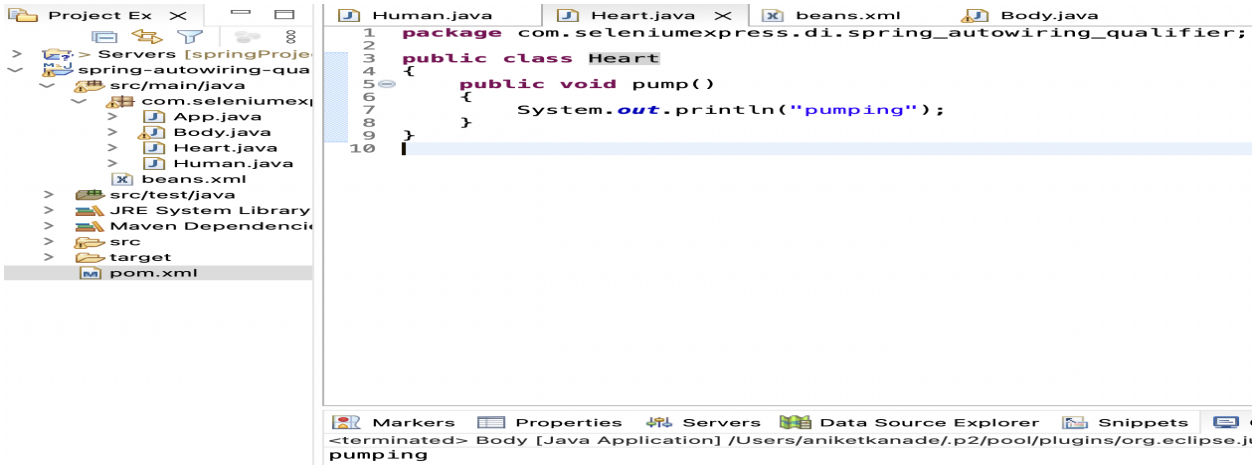
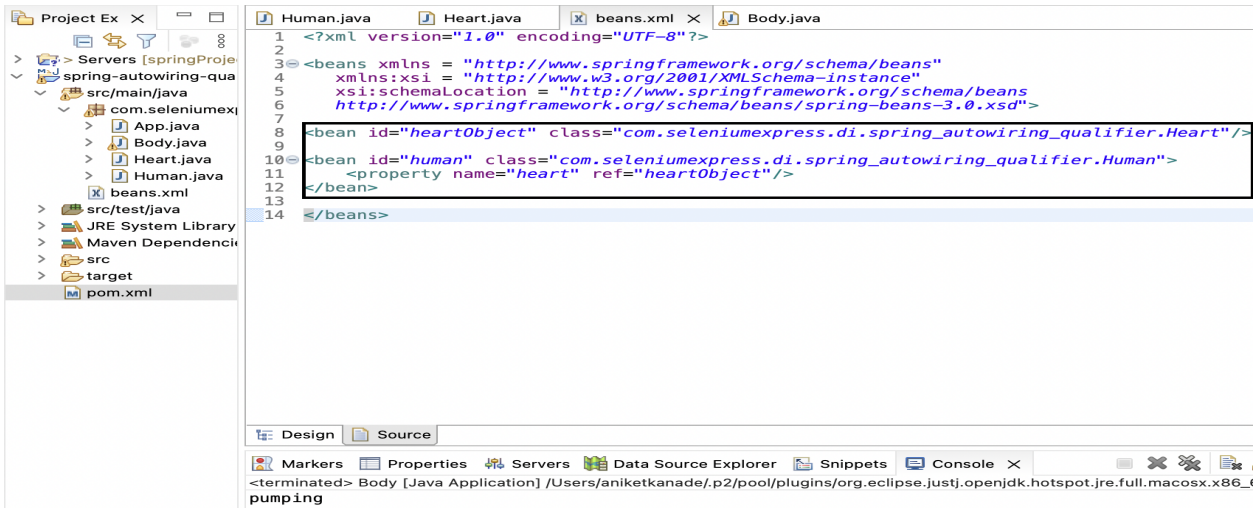
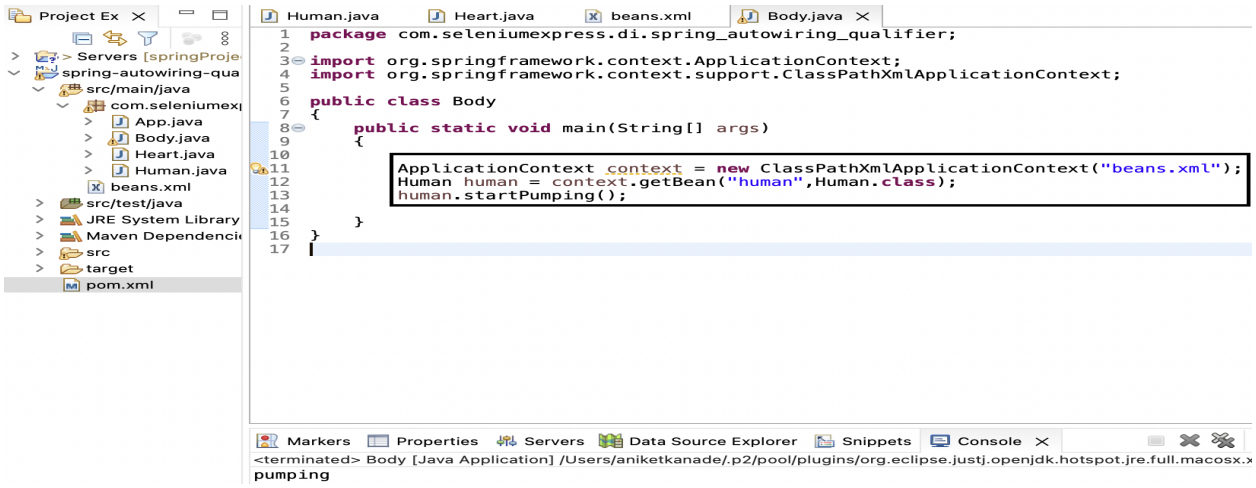
Spring - 4

1. Base project:

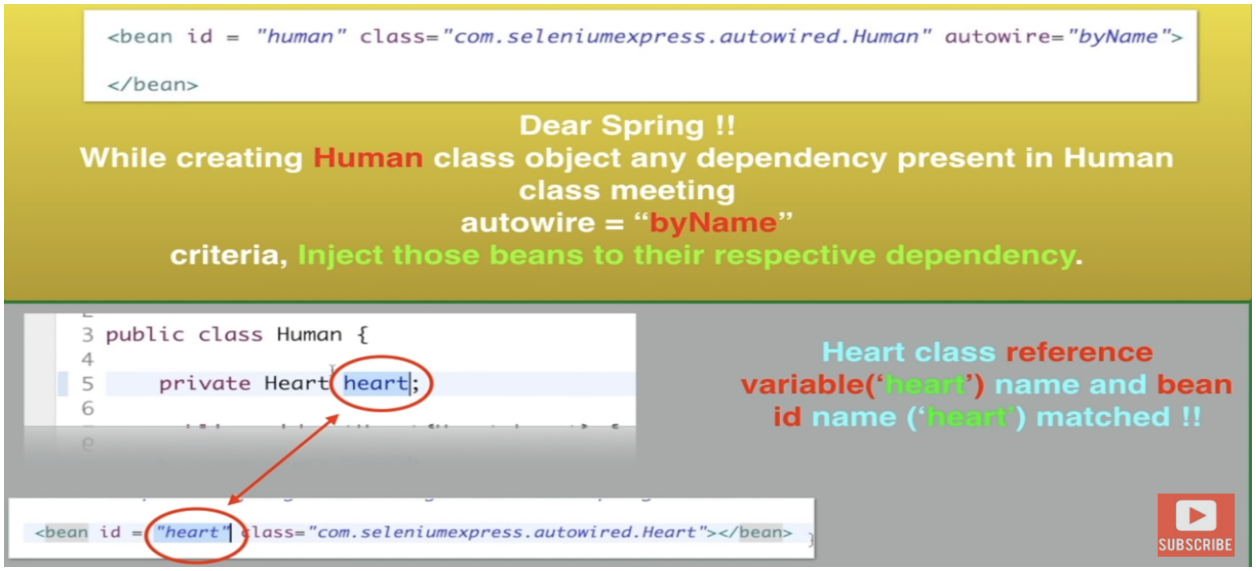


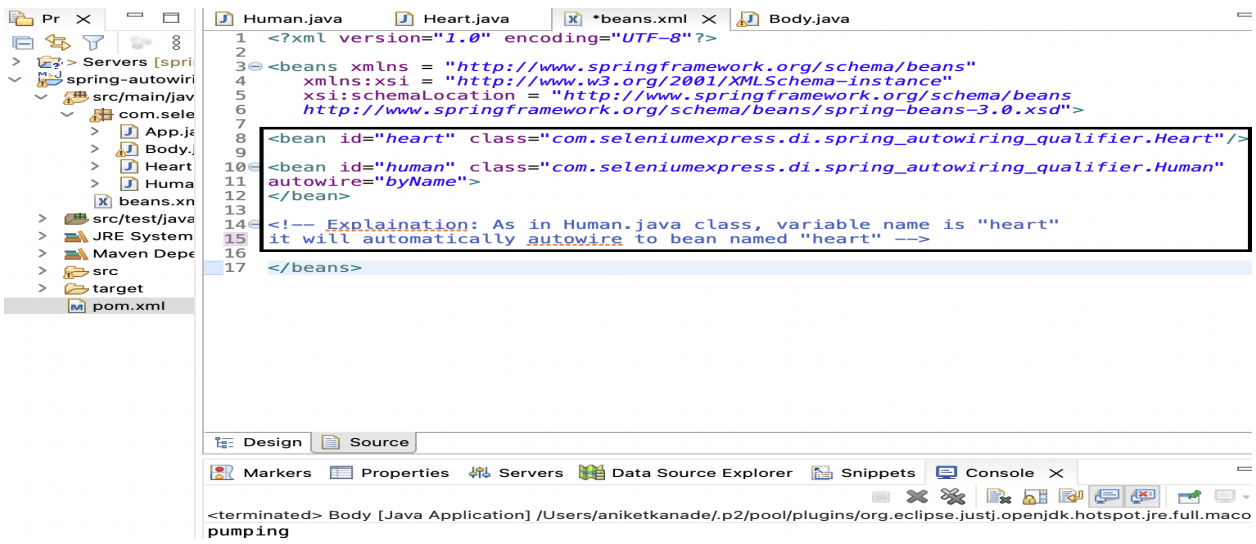




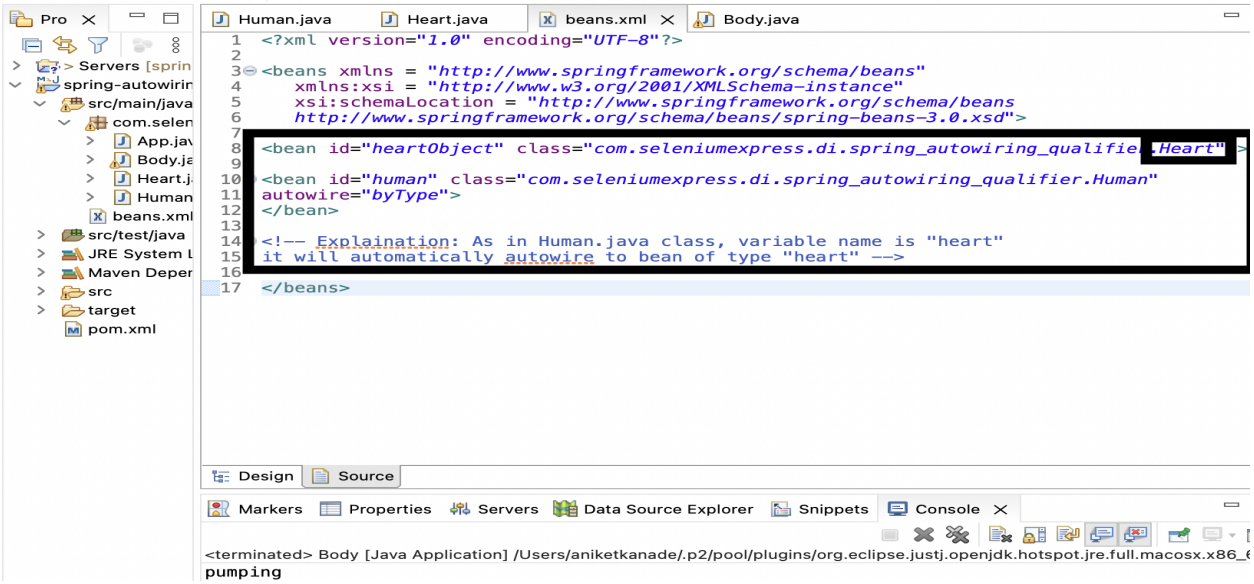


2. Iteration 1: (byName)





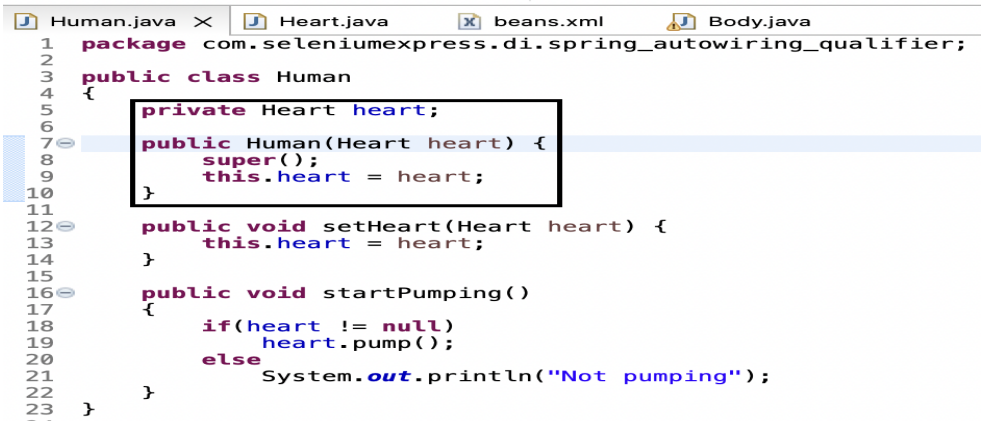
3. Iteration 2: (byType)



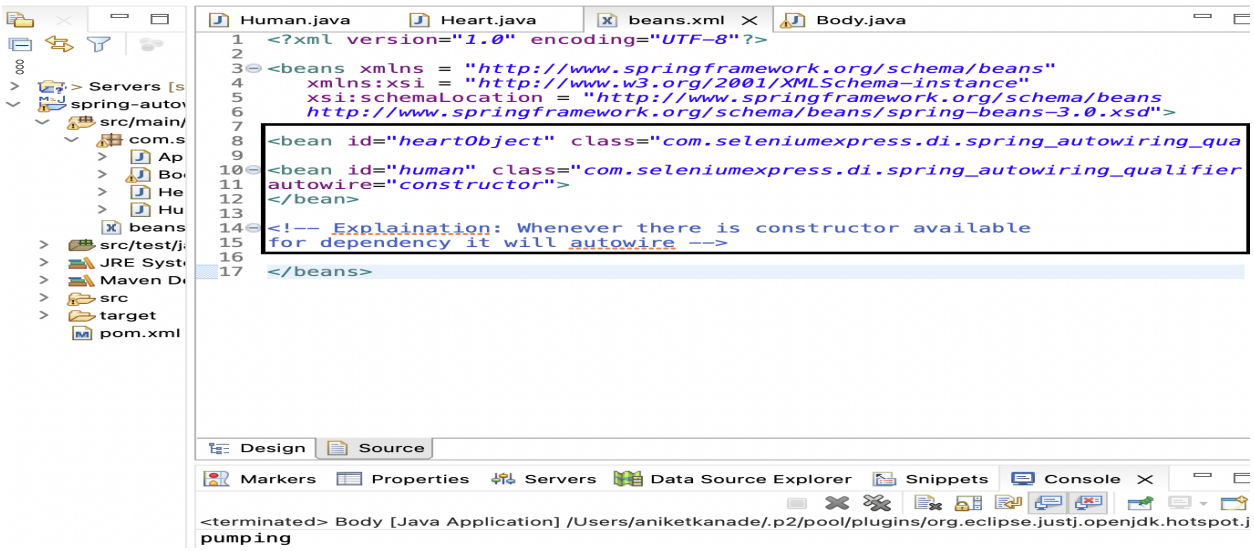
```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns = "http://www.springframework.org/schema/beans"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

 <bean id="heartObject" class="com.seleniumexpress.di.spring_autowiring_qualifier.Heart" >

<bean id="human" class="com.seleniumexpress.di.spring_autowiring_qualifier.Human"
autowire="byType">
</bean>

<!-- Explaination: As in Human.java class, variable name is "heart"
it will automatically autowire to bean of type "heart" -->

 </beans>
```

```
<terminated> Body [Java Application] /Users/aniketkanade/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_6
pumping
```

4. Iteration 3: (byConstructor)



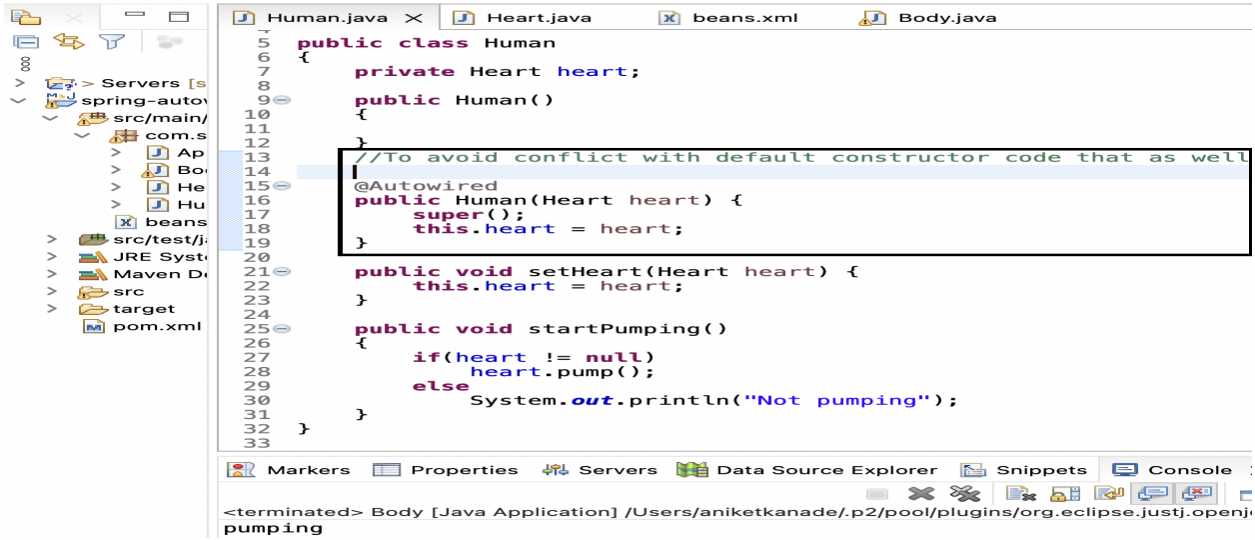
```
package com.seleniumexpress.di.spring_autowiring_qualifier;

public class Human
{
    private Heart heart;

    public Human(Heart heart) {
        super();
        this.heart = heart;
    }

    public void setHeart(Heart heart) {
        this.heart = heart;
    }

    public void startPumping()
    {
        if(heart != null)
            heart.pump();
        else
            System.out.println("Not pumping");
    }
}
```



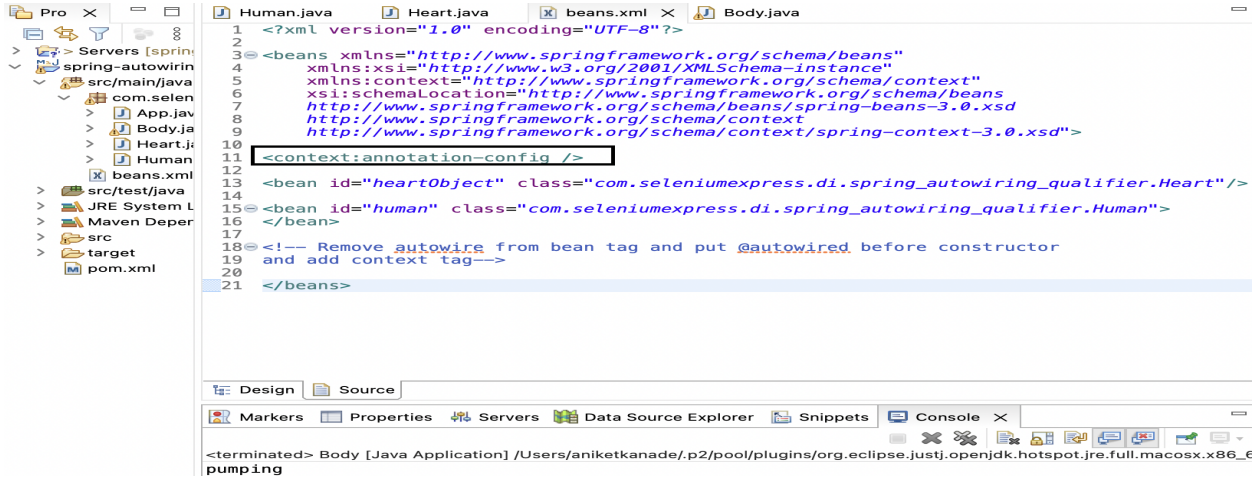
```
<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns = "http://www.springframework.org/schema/beans"
    xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation = "http://www.springframework.org/schema/beans
    http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">

<bean id="heartObject" class="com.seleniumexpress.di.spring_autowiring_qua

<bean id="human" class="com.seleniumexpress.di.spring_autowiring_qualifier
autowire="constructor">
</bean>

<!-- Explaination: Whenever there is constructor available
for dependency it will autowire -->

 </beans>
```

```
<terminated> Body [Java Application] /Users/aniketkanade/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.j
pumping
```

5. @autowire tag: (at constructor)

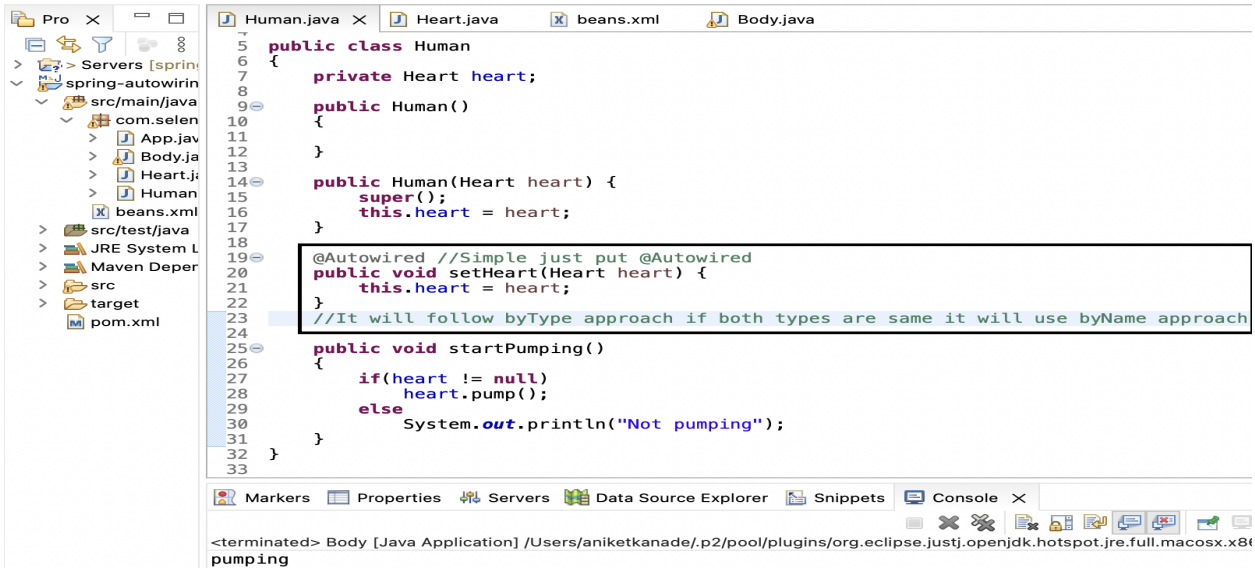


```
Human.java | Heart.java | beans.xml | Body.java

 5  public class Human
 6  {
 7      private Heart heart;
 8
 9      public Human()
10      {
11
12      }
13      //To avoid conflict with default constructor code that as well
14
15      @Autowired
16      public Human(Heart heart) {
17          super();
18          this.heart = heart;
19      }
20
21      public void setHeart(Heart heart) {
22          this.heart = heart;
23      }
24
25      public void startPumping()
26      {
27          if(heart != null)
28              heart.pump();
29          else
30              System.out.println("Not pumping");
31      }
32  }
33

Markers | Properties | Servers | Data Source Explorer | Snippets | Console

<terminated> Body [Java Application] /Users/aniketkanade/.p2/pool/plugins/org.eclipse.justj.openj
pumping
```



```
Human.java | Heart.java | beans.xml | Body.java

 1  <?xml version="1.0" encoding="UTF-8"?>
 2
 3  <beans xmlns="http://www.springframework.org/schema/beans"
 4      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 5      xmlns:context="http://www.springframework.org/schema/context"
 6      xsi:schemaLocation="http://www.springframework.org/schema/beans
 7      http://www.springframework.org/schema/beans/spring-beans-3.0.xsd
 8      http://www.springframework.org/schema/context
 9      http://www.springframework.org/schema/context/spring-context-3.0.xsd">
10
11      <context:annotation-config />
12
13      <bean id="heartObject" class="com.seleniumexpress.di.spring_autowiring_qualifier.Heart"/>
14
15  <bean id="human" class="com.seleniumexpress.di.spring_autowiring_qualifier.Human">
16  </bean>
17
18  <!-- Remove autowire from bean tag and put @autowired before constructor
19  and add context tag-->
20
21  </beans>

Design | Source

Markers | Properties | Servers | Data Source Explorer | Snippets | Console

<terminated> Body [Java Application] /Users/aniketkanade/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_6
pumping
```

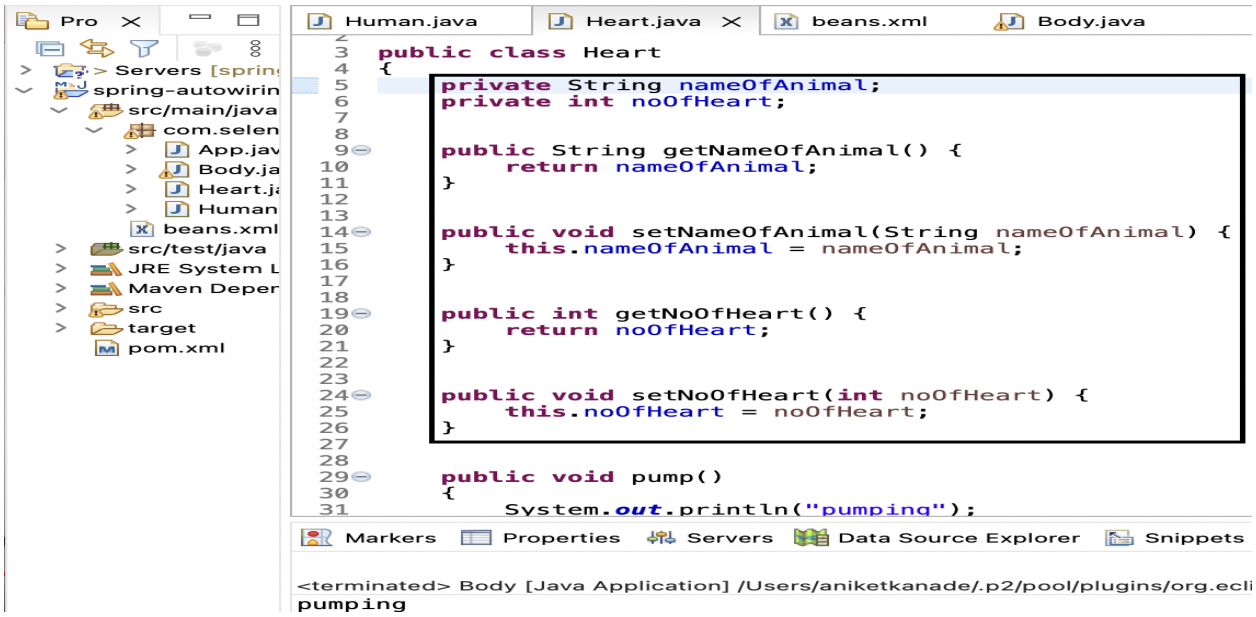
6. @Autowired: (at setter)

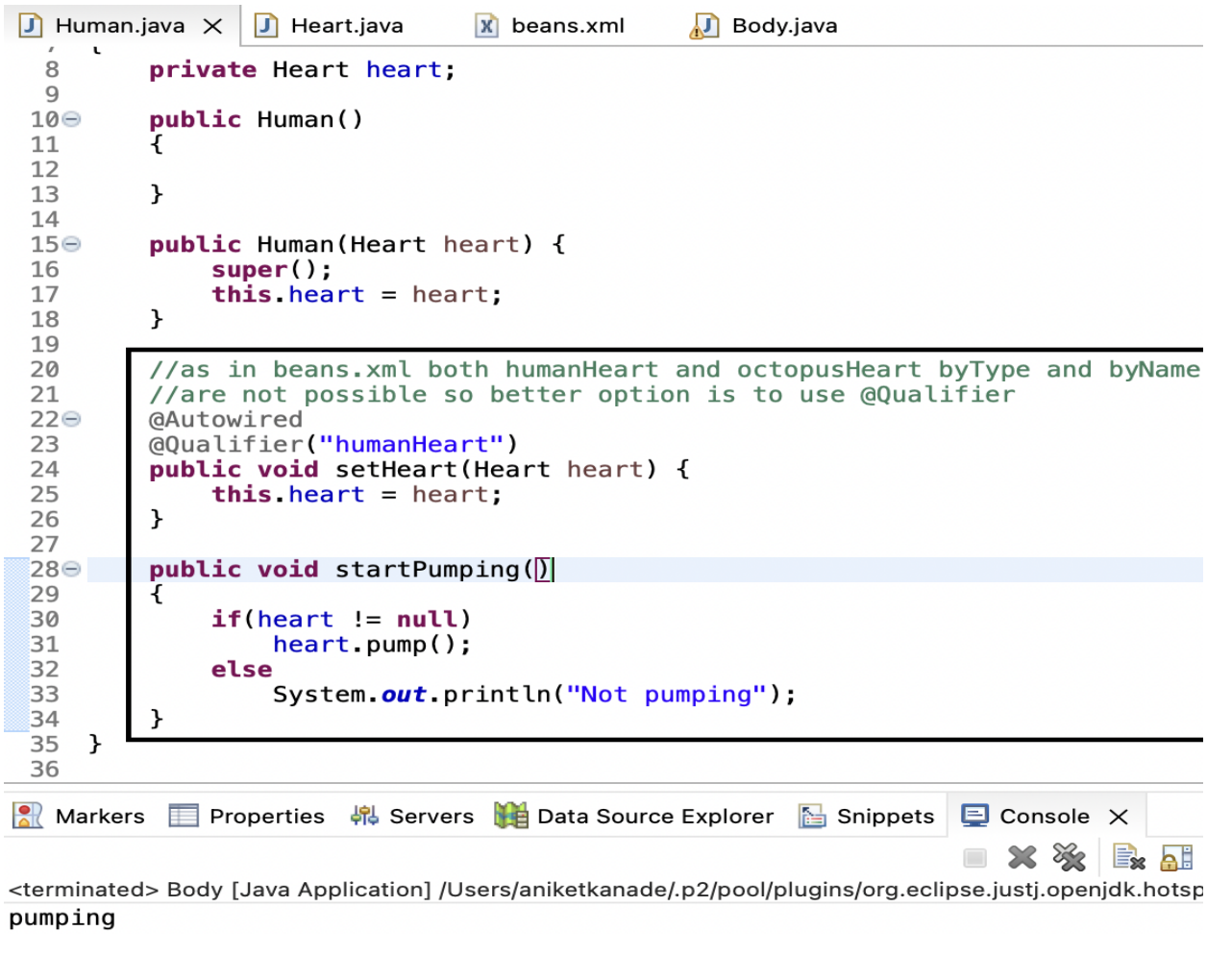


```
Human.java | Heart.java | beans.xml | Body.java

 5  public class Human
 6  {
 7      private Heart heart;
 8
 9      public Human()
10      {
11
12      }
13
14      public Human(Heart heart) {
15          super();
16          this.heart = heart;
17      }
18
19      @Autowired //Simple just put @Autowired
20      public void setHeart(Heart heart) {
21          this.heart = heart;
22      }
23      //It will follow byType approach if both types are same it will use byName approach
24
25      public void startPumping()
26      {
27          if(heart != null)
28              heart.pump();
29          else
30              System.out.println("Not pumping");
31      }
32  }
33

Markers | Properties | Servers | Data Source Explorer | Snippets | Console

<terminated> Body [Java Application] /Users/aniketkanade/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x8
pumping
```

- Autowiring is not possible for primitive types. Its only for reference types.

7. Qualifier:





If we remove setter methods then also it will work: