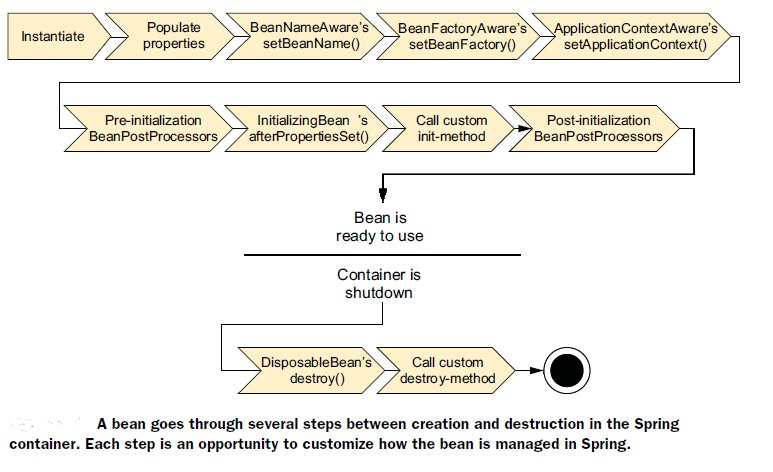
**Spring Life Cycle – 2022**

1. Spring instantiates the bean.
2. Spring populates bean’s properties.
3. If the bean implements **BeanNameAware**, Spring passes the bean’s ID to the **setBeanName()** method.
4. If the bean implements **BeanFactoryAware**, Spring calls the **setBeanFactory()** method, passing in the bean factory.
5. If the bean implements **ApplicationContextAware**, Spring calls the **setApplicationContext()** method, passing in a reference to the enclosing application context.
6. If the bean implements the **BeanPostProcessor** interface, Spring calls **postProcessBeforeInitialization() and postProcessAfterInitialization()** method.
7. If the bean implements the **InitializingBean** interface, Spring calls its **afterPropertiesSet()** method.
8. At this point, the bean is ready to be used by the application and remains in the application context until the application context is destroyed.
9. If the bean implements the **DisposableBean** interface, Spring calls its **destroy()** method. 

Spring provides two annotations: **@PostConstruct** and **@PreDestroy** to perform initialization and end up tasks.

**@PostConstruct**

The @PostConstruct annotation marks a method as initialization method of a bean which runs after dependency injection is completed. @PostConstruct annotation is same as "init-method" we configure in the bean.xml

**@PreDestroy**

The @PreDestroy annotation marks a method that is executed just before the bean is destroyed by Spring container. The method annotated with @PreDestroy can be used to release the resources or preform any destruction task before the container destroys the bean. @PreDestroy annotation is sam eas "destroy-method" we configure in the bean.xml.

@Component

public class DbInit {

private UserRepository userRepository;

private DbConnection dbConnection;

@PostConstruct

private void postConstruct() {

User admin = new User("admin", "admin password");

User normalUser = new User("user", "user password");

userRepository.save(admin, normalUser);

}

**@PreDestroy**

public void preDestroy() {

dbConnection.close();

}

}