

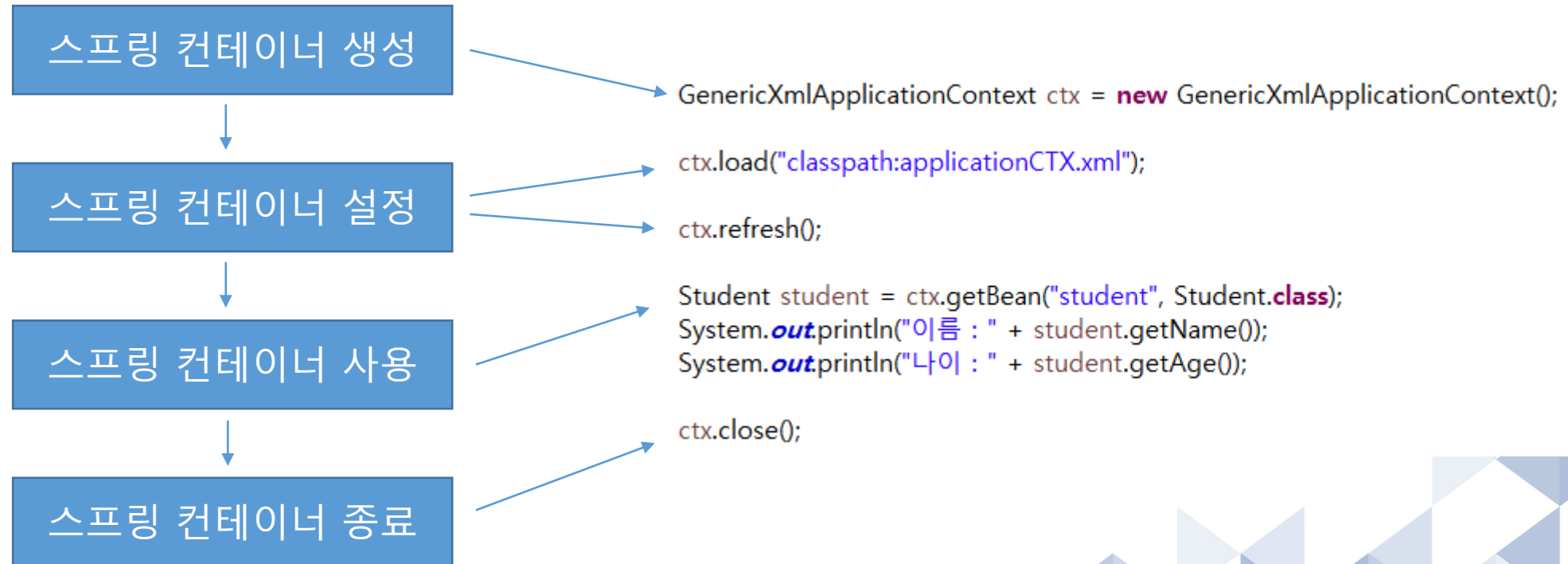
7강. 생명 주기(life cycle) 와 범위(scope)

- 스프링 컨테이너 생명 주기
- 스프링 빈 생명 주기
- 스프링 빈 범위

Lecturer Kim Myoung-Ho
Nickname 불스
blogstudy@naver.com

7-1. 스프링 컨테이너 생명 주기

스프링 컨테이너 생명 주기에 대해서 살펴 봅니다.
(spring_7_1_ex1_springex)



7-2. 스프링 빈 생명 주기

스프링 빈 생명 주기에 대해서 살펴 봅니다.
(spring_7_2_ex1_springex)

1) implements InitializingBean, DisposableBean

```
@Override
public void afterPropertiesSet() throws Exception {
    // TODO Auto-generated method stub
    System.out.println("afterPropertiesSet()");
}
```

빈 초기화 과정에서 호출 됩니다.

```
@Override
public void destroy() throws Exception {
    // TODO Auto-generated method stub
    System.out.println("destroy()");
}
```

빈 소멸 과정에서 생성 됩니다.

```
GenericXmlApplicationContext ctx = new GenericXmlApplicationContext();
ctx.load("classpath:applicationCTX.xml");
ctx.refresh();
ctx.close();
```

[참고 하세요]
ctx.close()의 경우 컨테이너가 소멸 하는 단계입니다.
컨테이너가 소멸 하면, 빈은 자동 소멸 됩니다.
빈만 소멸하게 한다면, student.destroy() API를 이용하
면 됩니다. 한번 해보세요.^^

7-2. 스프링 빈 생명 주기

2) @PostConstruct, @PreDestroy

```
@PostConstruct  
public void initMethod() {  
    System.out.println("initMethod");  
}
```

```
@PreDestroy  
public void destroyMethod() {  
    System.out.println("destroyMethod");  
}
```

빈 초기화 과정에서 호출 됩니다.

```
GenericXmlApplicationContext ctx = new GenericXmlApplicationContext();
```

```
ctx.load("classpath:applicationCTX.xml");
```

```
ctx.refresh();
```

빈 소멸 과정에서 생성 됩니다.

```
ctx.close();
```

7-3. 스프링 빈 범위(scope)

스프링 컨테이너가 생성되고, 스프링 빈이 생성 될 때, 생성된 스프링 빈은 scope을 가지고 있습니다.
범위란 쉽게 생각해서 해당하는 객체가 어디까지 영향을 미치는지 결정하는 것이라고 생각하면 됩니다.
(spring_7_3_ex1_springex)

```
<bean id="student" class="com.javalec.ex.Student" scope="singleton">
  <constructor-arg value="홍길순"></constructor-arg>
  <constructor-arg value="30"></constructor-arg>
</bean>
```

```
Student student1 = ctx.getBean("student", Student.class);
System.out.println("이름 : " + student1.getName());
System.out.println("나이 : " + student1.getAge());

System.out.println("=====");
```

```
Student student2 = ctx.getBean("student", Student.class);
student2.setName("홍길자");
student2.setAge(100);
```

```
System.out.println("이름 : " + student1.getName());
System.out.println("나이 : " + student1.getAge());

System.out.println("=====");
```

```
if(student1.equals(student2)) {
  System.out.println("student1 == student2");
} else {
  System.out.println("student1 != student2");
}
```

이름 : 홍길순

나이 : 30

=====

이름 : 홍길자

나이 : 100

=====

student1 == student2