



Ćwiczenie 1:

Utworzenie nowego projektu Spring.

Instrukcja rozwiązania

1. Tworzymy nowy projekt: Java Project
2. Do wymaganych bibliotek dodajemy następujące zależności: spring-core, spring-context, spring-beans, spring-asm, spring-expression i commons-logging oraz commons-logging-api
3. Do katalogu ze źródłami dodajemy plik context.xml, w do którego zapisujemy pustą konfigurację.
4. Taka pusta aplikacja może zostać uruchomiona.

Rozwiązanie

```
public class App {  
  
    public static void main(String[] arg) {  
        AbstractApplicationContext context =  
            new ClassPathXmlApplicationContext("context.xml");  
    }  
}
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<beans xmlns="http://www.springframework.org/schema/beans"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
    xmlns:context="http://www.springframework.org/schema/context"  
    xsi:schemaLocation="http://www.springframework.org/schema/beans  
http://www.springframework.org/schema/beans/spring-beans-3.0.xsd  
http://www.springframework.org/schema/context  
http://www.springframework.org/schema/context/spring-context-3.0.xsd">  
  
    <context:component-scan base-package="pl.devcastzone.spring.*" />  
  
</beans>
```

Ćwiczenie 2:

Utwórz komponent Library, będący implementacją biblioteki. Biblioteka udostępnia dwie metody: openingHours() i isOpen(). Komponent ma zostać dodany do kontekstu automatycznie za pomocą adnotacji.

Zweryfikuj utworzenie komponentu poprzez odpowiedni wpis w logach systemu.



Instrukcja rozwiązania

1. Tworzymy nową klasę w pakiecie pl.devcastzone.spring, o nazwie Library
2. Klasę oznaczamy adnotacją @Component
3. Dodajemy wymagane metody.
4. Po uruchomieniu aplikacji w logach pojawi się potwierdzenie dodania komponentu:
`defining beans [library...]`

Rozwiązanie

```
@Component
public class Library {

    public String openingHours() {
        return "Weekday 9:00 - 17:00";
    }

    public boolean isOpen() {
        return false;
    }
}
```