**Day2\_Assignments(SpringCore)**

**Problem:**

Day 2: Spring Core - Policy Administration Backend

Task 1: Refactor policy-related operations to utilize Spring Beans and Dependency Injection.

Task 2: Implement Spring validation on the server side to ensure policy data integrity.

Task 3: Set up Application Context and Bean Factory for a scalable backend structure.

**Solution**:

**Policy Interface**

package org.example;

public interface Policy {

void createPolicy(String policyName);

void updatePolicy(String policyName);

void deletePolicy(String policyName);

}

**HealthPolicy Class**

package org.example;

import org.springframework.stereotype.Component;

@Component("healthPolicy")

public class HealthPolicy implements Policy {

@Override

public void createPolicy(String policyName) {

System.out.println("Health Policy Created: " + policyName);

}

@Override

public void updatePolicy(String policyName) {

System.out.println("Health Policy Updated: " + policyName);

}

@Override

public void deletePolicy(String policyName) {

System.out.println("Health Policy Deleted: " + policyName);

}

}

**LifePolicy Class**

package org.example;

import org.springframework.stereotype.Component;

@Component("lifePolicy")

public class LifePolicy implements Policy {

@Override

public void createPolicy(String policyName) {

System.out.println("Life Policy Created: " + policyName);

}

@Override

public void updatePolicy(String policyName) {

System.out.println("Life Policy Updated: " + policyName);

}

@Override

public void deletePolicy(String policyName) {

System.out.println("Life Policy Deleted: " + policyName);

}

}

**PolicyService Class**

package org.example;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

@Service

public class PolicyService {

private final Policy policy;

@Autowired

public PolicyService(Policy policy) {

this.policy = policy;

}

public void performOperations(String policyName) {

policy.createPolicy(policyName);

policy.updatePolicy(policyName);

policy.deletePolicy(policyName);

}

}

**AppConfig Class**

package org.example;

import org.springframework.context.annotation.ComponentScan;

import org.springframework.context.annotation.Configuration;

@Configuration

@ComponentScan("org.example")

public class AppConfig {

}

**Application Class**

package org.example;

import org.springframework.context.ApplicationContext;

import org.springframework.context.annotation.AnnotationConfigApplicationContext;

public class Application {

public static void main(String[] args) {

ApplicationContext context = new AnnotationConfigApplicationContext(AppConfig.class);

PolicyService policyService = context.getBean(PolicyService.class);

// Perform operations on Health Policy

policyService.performOperations("Health Insurance");

// Perform operations on Life Policy

// Uncomment below line and comment the above line to switch to Life Policy

// PolicyService lifePolicyService = context.getBean(LifePolicyService.class);

// lifePolicyService.performOperations("Life Insurance");

}

}