



Universitat de Lleida
Escola Politècnica Superior

23/05/2021

Activitat 3

Exercici de disseny

Grau en Enginyeria Informàtica

Joel Farré Cortés (78103400T)

Joel Aumedes Serrano (48051307Y)

ESCOLA POLITÈCNICA SUPERIOR – UNIVERSITAT DE LLEIDA

Disseny

Service locator

Per a la realització d'aquesta part hem implementat la interfície ServiceLocator tan en la classe SimpleServiceLocator com en la CachedServiceLocator.

Per a la primera, hem utilitzat diversos HashMap, un per a emmagatzemar les Factory amb una clau de tipus String i un altre per a emmagatzemar valors a partir d'una clau de tipus String també. Les funcions setService i setConstant es basen en mirar si la clau està guardada com a l'altre tipus. Si ho està, es llença un LocatorError ja que amb la mateixa clau no hi poden haver serveix i constants alhora. Si no ho està s'afegeix al HashMap corresponent. La funció getObject comprova si es una constant el objecte que es demana, si ho es es retorna l'objecte. Si no ho es, però es un servei, es retorna l'objecte creat per aquest servei. Si no es ni una constant ni un servei es llença LocatorError.

Per a la segona classe, CachedServiceLocator, el seu funcionament es similar a l'anterior però en aquest cas, s'ha afegit un HashMap anomenat usedService per a tenir en compte quins serveis ja s'han fet servir i un altre anomenat createdObjects per a guardar els objectes creats pels serveis.

També hem afegit les classes Implementation que es fan servir per a les classes Factory les quals creen implementacions.

Service locator 2

En aquesta segona versió hem fet servir l'ús de genèrics de manera que per a obtenir un objecte d'una classe s'ha d'accedir al ServiceLocator utilitzant «.class». El funcionament tant en la classe SimpleServiceLocator com en la CachedServiceLocator es similar al de la primera versió però aquest cop els HashMaps ja no fan servir strings com a clau sinó que fan servir classes.

Les classes Implementation són les mateixes que les de la versió u i les classes Factory utilitzen un genèric per a poder-los guardar a ServiceLocator com l'objecte que es crea i no com a Factory de l'objecte que es crea.

Tests

Per a la implementació dels tests hem provat totes les funcions en tot els possibles casos d'ús que poden tenir, siguin correctes o incorrectes. Per així, d'aquesta manera assegurar-nos que l'aplicació respon correctament en els casos correctes, i que llança l'error pertinent en els casos incorrectes.

Dificultats i errors

Un dels petits inconvenients han sigut l'ús dels genèrics ja que es tracta d'un tema que feia bastant temps que no utilitzàvem però amb l'ajuda de la documentació de Java i de l'assignatura d'Estructures de Dades hem pogut resoldre aquests inconvenients esmentats anteriorment.

Per l'altra banda a l'hora de realitzar els tests no hem tingut cap tipus d'inconvenient o error.