Summary

Project Name

Patterns

Project Repository

https://github.com/Lukeekul/wahlzeit

Project CI

https://travis-ci.org/Lukeekul/wahlzeit

Current Tag

adap-hw11 on master

Diff to last tag

https://github.com/Lukeekul/wahlzeit/compare/adap-hw10...Lukeekul:adap-hw11

Overview

- Adding new Classes *Pattern*, *PatternType* and *PatternManager* in package *org.wahlzeit.model*
- Adding new member *m_type* to Class *PatternPhoto*
- Amending Creation of *PatternPhoto* with *Typename*
- Adding small Test Case for PatternPhoto and PatternType

Details

Implementation of Pattern Class

- Members of this Class are protected PatternType and ID, which can only be set by the Constructor
- Getter Methods for Id (getId) and PatternType (getType) are implemented

Implementation of PatternType Class

- The private Member m_name represents the name of the Type and has its own getter methode (getTypeAsString)
- The protected Member m_superType represents the SuperType of the Type, its default is null. It has its own getter (getSuperType) and setter (setSuperType) methodes

- For Subtypes a protected HashSet m_subTypes is implemented into which every new subtype of the instance has to entered in. This can be done with the methode addSubType. The methode isSubtype checks wheter m_superType is null, therefore the instance is not a subType. By given a Pattern instance, the methode hasInstance checks wheter this given instance of Pattern has set a PatternType. The methode getSubTypeIterator returns an iterator to the member subTypes and the methode createInstance returns a new Pattern instance.

Implementation of PatternManager Class

- The PatternManager takes care of the creation of new Pattern, by putting them into its private HasMap by assuring that the Id of the Pattern instances are consecutive.

Test

- A simple Test Case (testPatternType) in ValueTests asserts that the set Types at object creation are the same as attached to the objects.