UML Sample Tasks¶

Classes and Interactions of their Objects¶

Describe the following classes, objects, and their interactions in UML Class and Sequence Diagrams.

Variant A¶

Class Diagram¶
Assume you are developing an information system.

When implementing a UCx scenario, you used CA, CB, CC1, CC2, and CD classes. The CB class is a singleton, that is, there can exist only one instance of the class which is returned by the static method `ob()` of this class. The object of the CB class provides also `ob1()` and `ob2()` operations. You have designed an IC interface which is implemented by CC1 and CC2 classes. The IC interface includes the `oc()` operation, which is polymorphic.
The CA class implements `oa1(x)` operation with one parameter `x` and a nonparametric operation `oa2()`. Objects of the CA class furthermore "see" the collection of objects with the IC interface through the association between CA and IC.
Each object of the CC1 class "sees" by an association one CD object that provides an `od()` operation.

Interaction Diagram¶
Consider the class diagram above and an implementation of the `oa1(x)` and `oa2()` operations of the CA class that are called within the UCx scenario.
In **the first operation** `oa1(x)`, where `x` is the call parameter of this operation, an object of the CA class calls the static `ob()` method of the CB class that returns a single instance of the CB class (it is a singleton).
The CA object then asks the just-received instance of CB to perform the `ob1()` operation if the `x` parameter is zero or the `ob2()` operation for other values of `x`.
In the case of the `ob1()` operation, the object of the CB class creates and returns a new CC1 object. Otherwise, in the case of `ob2()`, the object of the CB class creates and returns a new CC2 object.
The created object CC1 or CC2 (hidden behind their common interface IC) is then stored by the object CA into its collection of IC objects.

Subsequently, in **the second operation** `oa2()` of the CA object in the given scenario, for each object of the collection of IC objects described above, the `oc()` operation is executed. In the case of a CC1 object, the execution of the `oc()` operation results into the CC1 object calling the `od()` operation of its associated CD object.

Variant B¶

Class Diagram¶

When implementing a UCx scenario, you used CA, CB, CC1, CC2, and CD classes. The CB class is a singleton, that is, there can exist only one instance of the class which is returned by the static method `ob()` of this class. The object if the CB class provides also `ob2(x)` operation with one parameter `x`.
You have designed an IC interface which is implemented by CC1 and CC2 classes. The IC interface includes the `oc()` operation, which is polymorphic.
The CA class implements `oa1(x)` operation with one parameter `x` and a nonparametric operation `oa2()`. An object of the CA class furthermore "sees" an object with the IC interface through the association between CA and IC.
An object of the CC2 object class "sees" by an association a collection of CD objects that provide an `od()` operation.

Interaction Diagram¶

Consider the class diagram above and an implementation of the `oa1(x)` and `oa2()` operations of the CA class that are called within the UCx scenario.
In **the first operation** `oa1(x)`, where `x` is the call parameter of this operation, an object of the CA class calls the static `ob()` method of the CB class that returns a single instance of the CB class (it is a singleton).
The CA object then asks the just-received instance of CB to perform the `ob2()` operation.
If the `x` parameter is zero, the object of the CB class creates and returns a new CC1 object.
Otherwise, the object of the CB class creates and returns a new CC2 object.
The created object CC1 or CC2 (hidden behind their common interface IC) is then stored by the object CA in its attribute as an IC object.

Subsequently, in **the second operation** `oa2()` of the CA object in the given scenario, the CA object call an `oc()` operation of the stored IC object. In the case of a CC2 object, the execution of the `oc()` operation results into the CC2 object calling the `od()` operation of each object of a collection of associated CD objects.