

Study questions for Lecture 8 - Answers

1. Consider the sample design for the Dating System, and the collaboration diagrams handed out for the lecture on this material. There were two diagrams, one had three collaboration diagrams and one had a diagram for :GUIFrame.

a) identify the occurrences of object visibility in the diagrams

i) three diagram page

-dL, the Domain Logic interface object has method return visibility to dateRequest. This is because it calls its constructor, who returns an instance

- dL, the Domain Logic interface object, has method parameter and class variable visibility to the DataBase interface object. This is because it is passed in the constructor when dL is created, and in that constructor the passed-in value is remembered by setting the value of a class variable to it.

- third diagram same visibility as second

ii) larger diagram

- the GUIFrame object could have either class variable or local variable visibility to the LogOnDialog object, depending how it gets created. For example, if a permanent object is created by the GUIFrame constructor, and assigned to a class variable, it would have class variable visibility. If a temporary instance was created in the :GUIFrame method that wanted to display it, then it would probably be a local variable in that method, so that when the method was exited the instance of LogOnDialog would disappear. However, there is no create() method on the 1b edge, which there would be if it was being created by the GUIFrame method that makes it visible, which implies it is probably the first of the two alternatives listed here, i.e. class variable visibility.

- :GUIFrame will have method parameter and class variable visibility to dL because dL is passed as a parameter in the constructor for the GUIFrame interface class, which then assigns it to a GUIFrame class variable.

- :GUIFrame will have class variable visibility to :DaterOptionSelectionDialog, the instance of DaterOptionSelectionDialog, because it is created in the constructor for :GUIFrame and assigned to a class variable for GUIFrame. We can guess this is the case because there is no create() message on the 3a edge, indicating the prior existence of the object.

- :GUIFrame has local variable visibility to :MessageDialog. We can infer this to be the case because the method in :GUIFrame that is sending the message numbered 4 to this object is also creating it. Presumably it is assigned to a local variable while it is being used inside that method. Note that the create() message could either be a call on a constructor or a factory method. In this case, MessageDialog could be a singleton, with an actual create() method.

b) for each occurrence, describe how the object in question comes/came into existence

- this is covered in the above

c) for each example of object visibility describe how the required class visibility was satisfied

i) three diagram page

- first diagram: we could make DateRequest an inner class of DomainLogic, the interface class for the DomainLogic subsystem, so methods in the DomainLogic interface class could create instances of DateRequest
- second diagram: we could import the DataBase subsystem package into the DomainLogic package, which would make the DataBase interface class, called DataBase, visible to classes inside that package.
- in the third diagram, same thing

ii) Larger diagram

- GUIFrame can have class visibility to LogOnDialog because it can be an inner class
- it will have class visibility to DomainLogic because the GUIFrame package imports the DomainLogic package.
- it will have class visibility to DateOptionSelectionDialog because it is an inner class
- it can have inner class visibility to MessageDialog if that class is declared inside of GUIFrame, or it could have global/import visibility if that class is described in a Globals package that is imported into all other classes. This comment about globals could apply to other class visibilities but it is only here that it seems it might be used.