MSc in Communications Software 2013
Design Patterns

Summer Paper

Eamonn de Leastar
Diarmuid O'Connor

Instructions:

Answer three questions

Section A

Question 1:

The Java SDK implements a ‘classic’ version of the Observer pattern within the *java.util* package, including the *Observable* class and *Observe*r interface. In addition, the *java.beans* package proposes a more sophisticated implementation via *PropertyChangeSupor*t and related types, with a more fine-grained event distribution mechanism.

(a) Review the Observer pattern as implemented by the *java.util* package, outlining the roles and responsibilities of the types concerned. Illustrate with a simple example.

10 Marks

(b) Contrast the *java.util.Observer* implementation with the approach employed by *PropertyChangeSupport* class. In particular, demonstrate how the more fine-grained event control is implemented.

10 Marks

(c) For GUI applications, are there alternatives to using Observer? What modern programming languages features might reduce the need for Observer implementations like the ones discussed above?

5 Marks

Question 2:

Problem:

A CAD application requires a facility whereby components can be incorporated into a design. Components can be structured hierarchically, for example component A can be composed of components B, C and D, and C in turn composed of E, F and G etc… However, there is a requirement that an API be devised such that components can be treated uniformly.

Solution:

(a) Present a high level review the *Composite* design pattern and demonstrate how it could prove suitable for this application.

10 Marks

(b) Elaborate on this high level design with a set of class and interface specifications. In particular present an implementation of a mechanism whereby a single request – draw() or move() for instance – can be dispatched to all objects within a single component via a uniform interface.

15 Marks

Section B

Question 3:

The ActiveRecord component of the Rails framework provides a mini-DSL for expressing associations between model classes, for example, :has_many, belong_to, :has_many :through etc.

Consider the following domain problem excerpt:

'... Moodle's database schema includes four main entities: student, module, topic (a module has many topics) and assignment specification (a module has many assignments). ……….The system's analytics component keeps records (in the database) on student access behaviour. Two access events are recorded: (1) Each time a student clicks a topic link for a module the component will record the duration spent in that topic area; (2) When a student reads an assignment specification (duration is not important in this case).'

These analytics records can be modeled as associations between some of the main entities, and ultimately will appear as join tables and/or foreign keys in the database schema.

You are required to:

(a) Draw an outline data model for the analytics aspect of the above problem, showing clearly the cardinality of model associations. [The classes in your data model will correspond to the main entities from the database schema – student, topic, etc.]

10 Marks

(b) Write excerpts from ActiveRecord implementations of your model class, in particular showing how you used the mini-DSL to declare the associations in your solution to part (I) above.

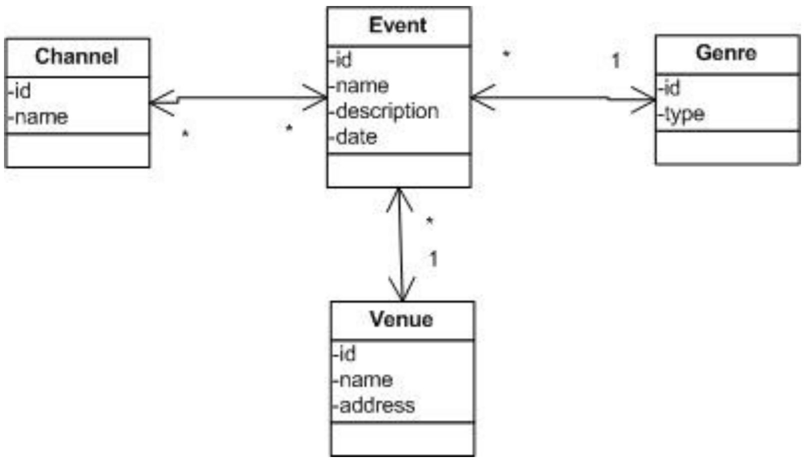
10 Marks

(c) The association DSL statements also add behaviour to a class. Explain and give examples.

5 Marks

Question 4:

The CouchPotato database records the planned transmission of live events (concert, political, sport, etc) by various TV channels An excerpt from its data model is illustrated below:



[Note: More than one channel may be transmitted an event.]

Suppose a web API based on the REST resource-oriented architecture style is required for this data set. You are required to:

(a) Design the read-only URL patterns for the Event resource area of the web API.

5 Marks

(b) Assuming the Grape and Rails frameworks are used as the implementation platform, write the endpoint(s) for updating the list of channels that will be transmitting an event. Two types of updates are possible: adding a channel to the list; removing a channel from the list. Use the code excerpt in Figure 1 below as a starting point.

8 Marks

(c) Explain, with the aid of an outline representation, what hypermedia you would include in the response to a request for a particular event. [Assume JSON formatting.]

7 Marks

(d) Why might partial representations be a useful feature to support for Event related read requests?

5 Marks