Information Systems Analysis & Design (M8748)

Tutorial 15 Answer

1. What is the difference between a pattern and a framework and how is each used?

   A pattern is used to resolve a problem by applying a suitably contextualized solution whereas a framework is the partially developed system that is essentially the core to a set of similar applications and is instantiated by adding the necessary program code.

   The major differences between patterns and frameworks can be summarized as follows.

   - Patterns are more abstract and general than frameworks. A pattern is a description of the way that a type of problem can be solved, but the pattern is not itself a solution.
   - Unlike a framework, a pattern cannot be directly implemented in a particular software environment. A successful implementation is only an example of a design pattern.
   - Patterns are more primitive than frameworks. A framework can employ several patterns but a pattern cannot incorporate a framework.

2. What are the main aspects of changeability?

   The main aspects of changeability are maintainability, extensibility, restructuring and portability.

3. Why is the class constructor private in the Singleton pattern?

   The class constructor in the Singleton pattern is private so that it can only be accessed the class-scope instance() method. This ensures that the Singleton class has total control over its own instantiation.

4. What are the advantages of using the Singleton pattern?

   The Singleton pattern ensures that only one instance of a class exists and provides system wide access to that instance.

5. What are the disadvantages of using the Singleton pattern?

   The disadvantages of the Singleton pattern are as follows.

   - Using the pattern introduces some additional message passing. To access the singleton instance the class scope method has to be accessed first rather than accessing the instance directly.
   - The pattern limits the flexibility of the application. If requirements change and as a result the singleton class may have many instances then accommodating this new requirement necessitates significant modification to the system.

- The singleton pattern is quite well known and developers are tempted to use it in circumstances that are inappropriate. Patterns must be used with care.

6. What implementation problems may occur when using the State pattern?

   The State pattern may have the following implementation problems.

   - If State objects cannot be shared amongst Context objects (i.e. are not pure state objects) then each Context object will have to have its own State object thus increasing storage requirements.
   - State objects may have to be created and deleted as the Context object changes state increasing the processing requirement.
   - The use of the State pattern introduces one additional message, which also possibly increases the processing requirement.

7. What are the differences between a pattern language and a pattern catalogue?

   A pattern language is a group of patterns that work together to solve problems in a particular domain. A pattern catalogue is a less closely related group of patterns that may be used together to some extent or may be used independently.

8. List two general dangers and two general benefits of the use of patterns.

   Two general dangers of using patterns are the inappropriate application of a pattern and some limitation on creativity. Two advantages of using patterns are the introduction of a reuse culture at the design level and the rich source of development experience they offer.

9. What seven steps are suggested by Gamma et al. for the effective use of patterns?

   The seven steps from Gamma et al. are as follows.

   1) Read the pattern to get a complete overview.
   2) Study the Structure, Participants and Collaborations of the pattern in detail.
   3) Examine the Sample Code to see an example of the pattern in use.
   4) Choose names for the pattern's participants (i.e. classes) that are meaningful to the application.
   5) Define the classes.
   6) Choose application specific names for the operations.
   7) Implement operations that perform the responsibilities and collaborations in the pattern.