Information Systems Analysis & Design (M8748)

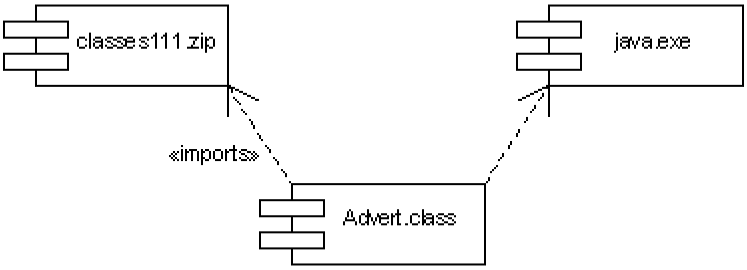
Tutorial 19 Answer

1. List the different categories of software that may be used in developing a system.
   CASE tools; compilers, interpreters and run-time support; visual editors; integrated development environments; class browsers; component managers; DBMS; CORBA; testing tools; conversion tools; documentation generators; configuration management systems; debuggers.

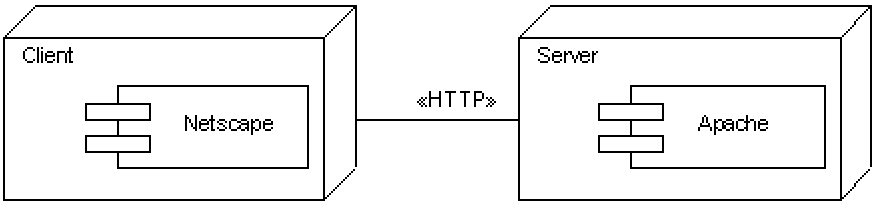
2. What packages have you used and which categories do they fall into?
   The answer to this will depend on the individual. Examples would be: Rational Rose – CASE tool; Java – compiler with run-time support; JBuilder – visual editor/IDE; Microsoft Developer's Studio – IDE; Oracle – DBMS; VisiBroker – CORBA ORB; TestScope – testing tool; ClearCase – configuration mangement tool; gdb – debugger.

3. What is the difference between a package diagram and a component diagram?
   Package diagrams show the logical grouping of classes in a system, whereas component diagrams show the physical components of a system. During implementation, package diagrams can be used to show the grouping of physical components into sub-systems; component diagrams can be combined with deployment diagrams to show the physical location of components of the system.

4. Draw a component diagram to show the run-time dependency between a Java class file, the java.exe run-time program and the Java classes in a zip file.
   Diagram something like this.



5. Draw a deployment diagram to show how a web browser and web server are located on different machines and the communication protocol they use.
Diagram something like this.



6. List five tests that you would carry out on the FoodCo use case Start Line Run.
Possible tests would be to test validation of date, test validation of start time, check validation of job number etc. More detailed validation of time could be to check it is within a certain amount of time of current time—if not, then a warning should be displayed.

7. List the sections that you would include in a post-implementation evaluation report and explain the content of each section.
Review of cost benefit analysis. Summary of functional requirements met and amended. Review of achievement of non-functional requirements. Assessment of user satisfaction. Problems and issues with the system. Positive experiences. Extract of quantitative data for future planning. Identification of candidate components for reuse. Possible future developments. Actions required. (See Section 19.9.2 for the detail.)

8. What is the difference between maintenance work carried out to fix bugs and work carried out to add requested enhancements to a system?
If a software company has developed software specifically to a customer's requirements, then work to fix bugs is usually carried out at the software company's expense. In the case of software that is not developed for a specific customer, bugs are usually fixed in subsequent releases of the software, and the new releases may be available to the customer either as part of the original agreement or under a maintenance contract. The aim of bug fixing is that the software should perform as specified.
Requests to enhance software are usually handled differently. In the case of software developed for a particular customer, then the customer can usually get what they are prepared to pay for. In the case of package software, the software company will consider whether the enhancement will benefit all users, how much it will cost (it may require additional analysis and design, not just coding), and then prioritize enhancements. Major enhancements may emerge as part of new versions of the software that require the customer to upgrade, or they may be available under a maintenance contract.

9. Why should decisions about enhancements not be left to maintenance programmers?

   Because analysts or designers will have the wider view of the system and can ensure that changes fit in and do not have a detrimental effect on other sub-systems.

10. What tasks do maintenance staff undertake?

    Maintaining the system, providing support to users, providing training, improving documentation, solving simple problems, carrying out minor upgrades, recording and analysing bugs, documenting requests for significant enhancements, restoring systems after failures.