|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | **Exam 1 Practical - Solutions**   1. **List two of the factors that contributed to the "software crisis" in the early history of software development, and explain how software engineering life-cycle approaches attempt to address those factors.**   Many answers are possible. For example: high cost of maintenance addressed by better documentation, better design for understandability and extensibility, formal reviews, testing procedures, etc; high cost of development addressed by more emphasis on analysis and design before implementation begins; etc.   1. **What software life-cycle model would you use if there is significant technical risk and the customer's requirements are not well-known in advance? Justify your answer in a short essay.**   Either spiral or rapid prototyping, or a combination of both. Spiral has good risk management characteristics, and can be implemented to include early prototypes or simulations to show to the customer. Rapid prototyping is good for clarifying requirements with the customer, and can be used to put early pressure on components with technical risk.   1. **The engineers at hypothetical.com adopted the incremental model for the development of their intelligent Web-based search agent. After implementing version 1.0 (which supported basic keyword searching of a list of sites using keyword query) they found that they could not implement version 2.0 (which supports searching using Boolean combinations of keywords) without completely re-designing the query module. Why do you think this problem might have arisen? Suggest two ways that the problem might have been avoided. Justify your answer in a short essay.**   The problem was probably caused by a lack of detail in the requirements analysis phase, a lack of detail in the design phase, or some combination of both. More attention to detail in these areas could have remedied the problem, assuming that all of the basic functional requirements were known in advance. If the requirements were underspecified and/or not completely known, then a better choice of development model might have been the rapid prototyping model or spiral model. Or, the incremental could have been used after an initial rapid prototype was used as a technique for firming up the requirements. What software life-cycle model would you use if there is significant technical risk and the customer's requirements are not well-known in advance? Justify your answer in a short essay.   1. **Case Study: Consider an automated library circulation system.**    * **Every book has a bar code and every borrower has a card bearing a bar code.**    * **When a borrower wishes to check out a book, the librarian scans the bar code on the book and on the borrower's card, and then enters C at the computer terminal.**    * **When a book is returned, it is again scanned and the librarian enters R.**    * **Librarians can add books (+) to the library collection or remove them (-).**    * **Borrowers can go to a terminal and determine:**      + **all the books in the library by a particular author  (the borrower types A= followed by the author's name)**      + **all the books with a specific title (the borrower types T= followed by the title)**      + **all the books in a particular subject area  (the borrower types S= followed by the subject area)**    * **If a borrower wants a book that is currently checked out, the librarian can place a hold on the book so that, when it is returned, it will be held for the borrower who requested it  (the librarian types H= followed by the number of the book)**   **Questions**   * + **Draw a dataflow diagram for the system, showing as much detail as you can without making assumptions about implementation.**   The dataflow diagram for the entire system is shown immediately below. The Process Command process is shown using a hierarchical expansion in the second diagram.  C:\Users\rongkang\Desktop\ssd9+Exam\exam1\Exam 1 Practical - Solutions.files\DFD-whole.gif    C:\Users\rongkang\Desktop\ssd9+Exam\exam1\Exam 1 Practical - Solutions.files\DFD-detail.gif     * + **Draw a state transition diagram for the system.**   NOTE about the diagram:   * + - The diagram uses separate states for reading commands and bar codes to show the sequence in which the bar codes and commands are expected.   C:\Users\rongkang\Desktop\ssd9+Exam\exam1\Exam 1 Practical - Solutions.files\Std.gif     * + **Draw an entity-relationship diagram for the system.**   C:\Users\rongkang\Desktop\ssd9+Exam\exam1\Exam 1 Practical - Solutions.files\ER-Diagram.gif    NOTES about the diagram:   * + - The Status attribute can take on the values: available, held, checked\_out.     - Placing a hold on a book creates an instance of the relation held\_for.     - Checking out a book creates an instance of the relation borrowed\_by. |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | © Copyright 1999-2004 iCarnegie, Inc. All rights reserved. |  |  |  |  |  |

**Exam 2 Practical - Solutions**

1. **Use the technique of noun extraction to identify the possible classes in the following use case scenario for a software system which controls an assembly line:**

**"The sensor sends a STOP message to the control software if a widget reaches the end of the conveyor belt without being lifted by the robot arm. The software sends a signal to the motor controller to stop the assembly line."**

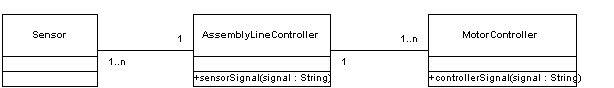
**Which of the candidate classes should not be added to the preliminary class list? Refer explicitly to the refinement principle you used in each case.**

"The **sensor** sends a **STOP message** to the **control software** if a **widget** reaches the **end** of the **conveyor belt** without being lifted by the **robot arm**. The**software** sends a **signal** to the **motor controller** to stop the **assembly line**."

Candidate classes which are eliminated during preliminary class refinement: widget, end, robot arm, control software, assembly line (outside problem scope);STOP message, signal (should be modeled as message or method calls, not as separate classes); also, the pairs of terms (conveyor belt / motor controller) and (software / control software) could be redundant synonyms for each other. The final candidate class list: sensor, motor controller, and assembly line controller.

1. **Draw an object diagram for the classes you identified in Question 1; be sure to include any class attributes or methods that resulted from preliminary class refinement.**

Here is the associated object diagram:

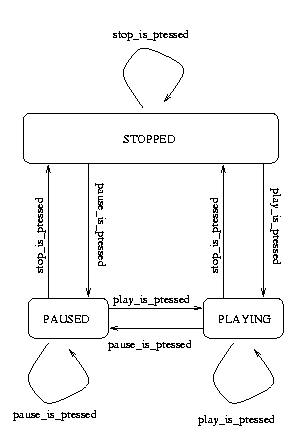


1. **Does the object diagram you drew for Question 2 use the principle of responsibility-driven design in the assignment of messages (method calls) to classes? Justify your answer.**

Yes. Since the controller objects are responsible for responding (modifying their behavior) based on the sensor signals, those signals (messages) should be implemented as method calls inside the controller objects.

1. **Draw a UML state transition diagram to model the control program for a portable CD player. Include three states: stopped, playing, andpaused. Also, include three events, possible in any state: pause\_is\_pressed, stop\_is\_pressed, and play\_is\_pressed.**

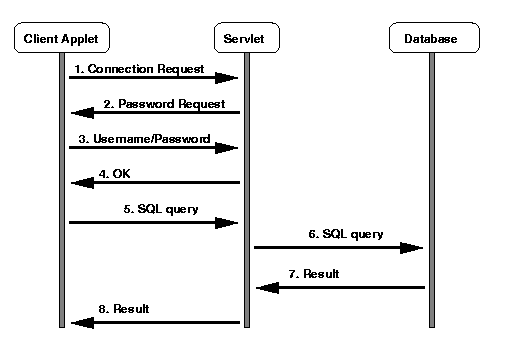
Here is the associated state transition diagram:



1. **Draw a UML sequence diagram that models the following events in a three-tier database client-server architecture:**

**"The client applet sends a connection request to the database servlet, which sends back a password request. The user supplies a username and password, which are sent back to the database servlet. The username and password are correct, so the servlet responds with an 'OK' message. The user enters search parameters in the applet interface, and the client applet translates these to an SQL query, which is sent to the database servlet. The database servlet sends the SQL query to the remote database, which sends back the results of the query. The servlet passes the results back to the client applet."**

Here is the associated sequence diagram:



© 2000 Carnegie Technology Education. All rights reserved.

**Exam 3 Practical - Solutions**

1. **Which of the artifacts in your project system are the most likely candidates for reuse? Justify your answer using the principles of reuse you learned in the course.**

Potentially reusable artifacts include code and noncode items (specifications, models, etc.). Hence the wording of the question. The most likely candidates for reuse are the database middleware and the generic aspects of the client applet. The independence (low coupling) and high cohesion of these modules make them good candidates for reuse. The entity-relationship model and backend database model are probably the least reusable in another context, because they are completely dependent on the particular problem to be solved.

1. **Use the technique of boundary value analysis to describe the different equivalence classes of input to the three objects in a three-tier database client-server architecture (client applet, database servlet, database). Use this information to sketch a brief outline for an execution-based test plan for such a system.**

Boundary value analysis and equivalence classes:

* + At the client applet: good data from the user, bad data from the user.
  + At the database servlet: valid SQL queries, invalid SQL queries.
  + At the database backend: valid SQL queries, invalid SQL queries

(The student may also mention the control messages that are sent back/forth for things like login, connecting to DB, etc.).

A basic execution-based test plan would include two main phases: a phase to verify that all the acceptable input is handled correctly (functional requirements are satisfied), and a phase to verify that all the invalid input is handled gracefully (robustness requirements are satisfied).

1. **Assume that you are a software procurement officer for a company that sells automobiles online, and that you are interested in contracting with a software developer to create a three-tier client-server system for e-commerce like the one you built for your project. It is your job to write a set of acceptance criteria for the contract, based on your idea of what comprises adequate product testing for the system. What kinds of product testing are appropriate in this context? Give examples specific to the three-tier e-commerce system.**

The following answers are representative of the types of student answers that are acceptable. Correctness: the system must handle any valid set of input it is given, based on the requirements specification for the system. Reliability: the system must run 24 hours a day, 7 days a week, with a maximum acceptable down time of 10 minutes every 7 days. Robustness: the system must be able to handle at least 100,000 transactions per day, with a maximum allowable burst rate of 100 transactions per minutes. Documentation: the system must include complete documentation for the installer, maintainer, and end user, each of which will be subjected to formal review before acceptance. Installation: the client and server programs will be installed and tested under expected conditions before acceptance.

1. **Briefly describe the types of documentation activity during the various life-cycle phases of a software product.**

The answer should include a mention of at least one kind of document/type of documentation activity during each life-cycle phase (requirements, specification, design, implementation, integration, maintenance). More detail should be given for implementation and integration phases. Testing documents should be mentioned.

**Discuss the role of documentation across phases.**

Documentation for the current phase of the current version of the product should be completed before the next phase begins, ideally by the same individuals who worked on the phase. Documentation provides continuity when personnel changes. Accurate documentation is a necessity for product development and maintenance.

**If you were a project manager, what would you do in order to ensure that good documentation is available?**

Plan for a thorough documentation process from the beginning. Schedule time in the plan for writing and maintaining the documentation. Remind the client that good documentation is a priority for both sides. Establish or adopt documentation standards and train people in their use. Use processes and/or CASE technology that facilitate documentation maintenance and make available online the most recent versions of documents to all interested parties.

1. **What is maintenance and are there different kinds of maintenance? What activities does maintenance involve and what skills does it require? Describe the different aspects of managing maintenance including fault reports, CASE technology, and metrics.**

Maintenance is any changes to a software product after it has been accepted by the client. Students should mention and briefly describe corrective, perfective, and adaptive maintenance. Releases should try not to mix different types of maintenance. Activities may involve changes to any aspect of the product's life cycle, including documentation, and therefore maintainers must have a broad range of skills in addition to good diagnostic skills.

Students should talk a bit about managing fault reports (prioritization, workarounds). CASE technology is useful throughout the process, but version-control and configuration-control tools in particular will help keep track of the various versions of a changing product. Fault-tracking systems help with fault management.

Useful metrics are:

* + number of faults found in total and in a given period
  + a classification of faults
  + the status of faults

© 2000 Carnegie Technology Education. All rights reserved.