UML 类图练习题

1. On a single diagram, illustrate the following using the UML notation for objects, links and messages.

(*a*) An object of class `Window`, with no attributes shown.

(*b*) An object of class `Rectangle` with attributes *length* and *width*. Assume that the rectangle class supports an operation to return the area of a rectangle object.

(*c*) A link between the window and rectangle objects, modelling the fact that the rectangle defines the screen coordinates of the window.

(*d*) The window object sending a message to the rectangle asking for its area.

Draw a class diagram showing `Window` and `Rectangle` classes with the properties mentioned in this question.

2. Suppose that an environmental monitoring station contains three sensors, namely a thermometer, a rain gauge and a humidity reader. In addition, there is an output device, known as a printer, on which the readings from these three sensors are shown. Readings are taken and transcribed onto the printer every five minutes. This process is known as ‘taking a checkpoint’.

(*a*) Draw an object diagram showing a plausible configuration for these objects, and include on the diagram the messages that might be generated in the system every time a checkpoint is taken. Assume that a checkpoint is initiated by a message sent from a timer object to the monitoring station.

(*b*) Does your diagram clearly show the order in which messages are sent? If not, how might this be shown?

(*c*) Draw a class diagram summarizing the structure of the monitoring station.

3. A workstation currently has three users logged into it, with account names A, B and C. These users are running four processes, with process IDs 1001, 1002, 1003 and 1004. User A is running processes 1001 and 1002, B is running process 1003 and C is running process 1004.

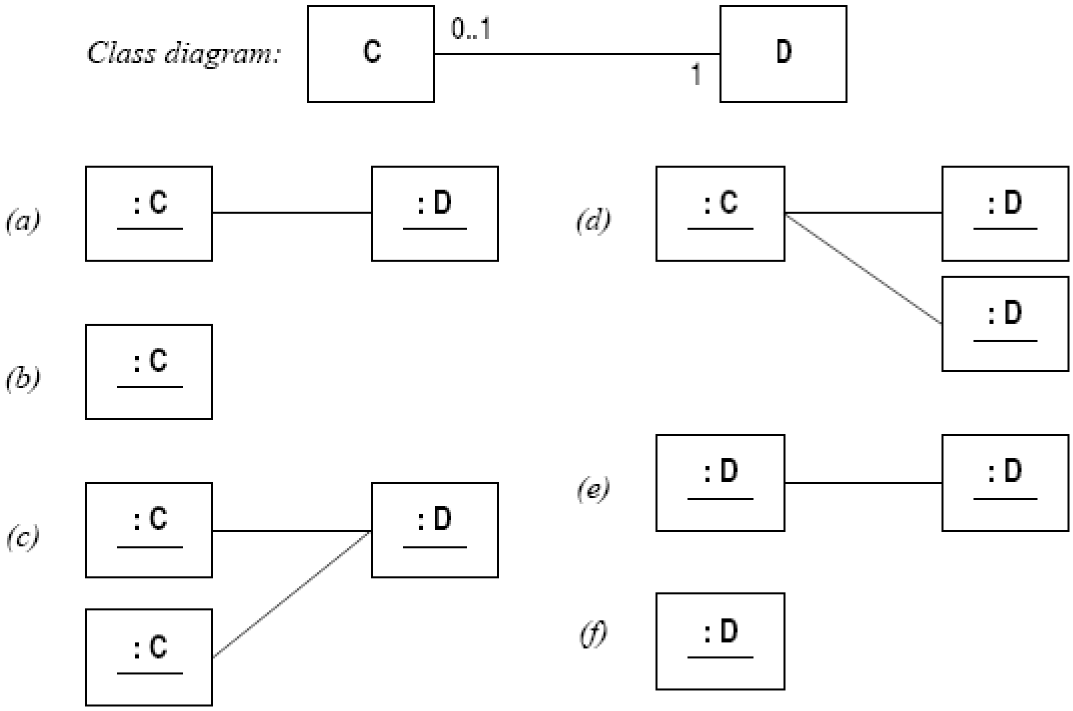
(*a*) Draw an object diagram showing objects representing the workstation, the users and processes, and links to represent the relationships of a process running on a workstation and a user owning a process.

(*b*) Consider an operation which lists information about the processes that are currently running on a workstation. It can either report on all the current processes or, if invoked with a suitable argument, those for a single specified user. Discuss what messages would need to be passed between the objects shown in part *(a)* in order to implement this operation.

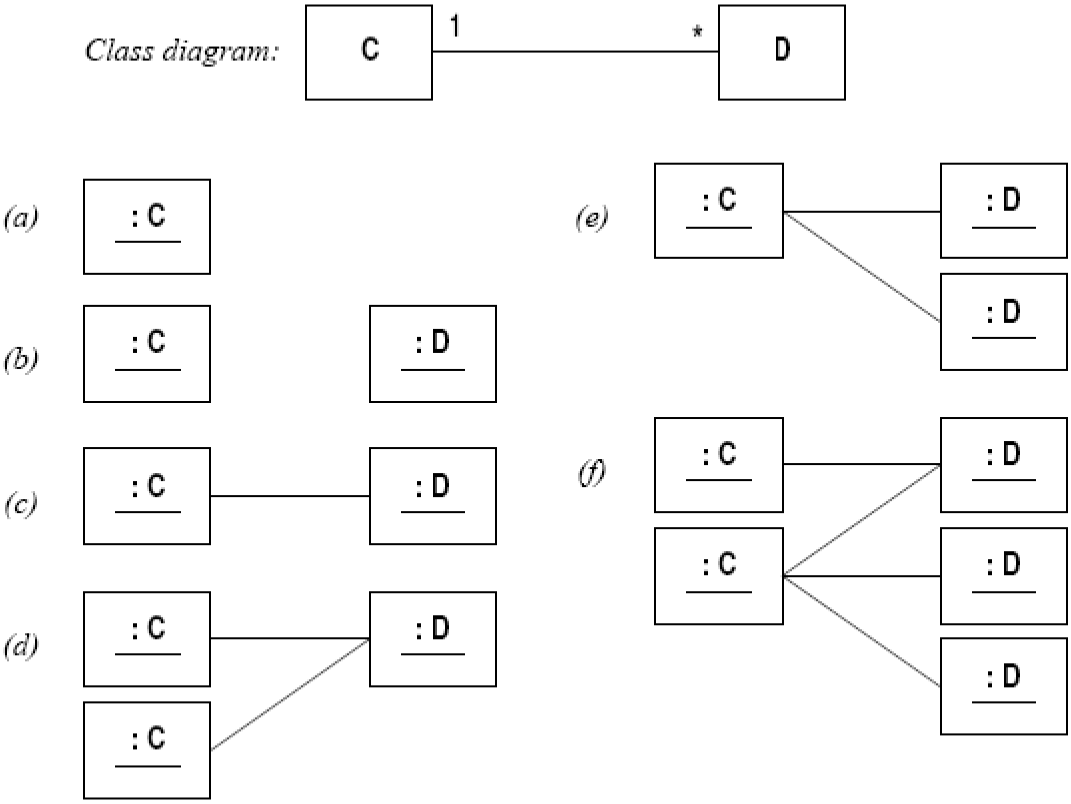
4. Draw UML class and object icons representing each of the following. Where appropriate, use enumerations and programming language types to specify attribute types.
(*a*) A class representing traffic lights, with attributes to record which colors are currently illuminated.
(*b*) A class representing a counter with operations to set or reset the counter to zero, to increment and decrement the value stored by a specified amount, and to return the current value.
(*c*) A class representing a switch which can be turned on or off.

5. Define multiplicities for the following associations:
(*a*) ‘Has on loan’, linking people to books in a library system.
(*b*) ‘Has read’, linking people to books.
(*c*) ‘Is occupying’, linking chess pieces to squares on a chess board.
(*d*) ‘Spouse’, linking class ‘Person’ to itself.
(*e*) ‘Parent’, linking class ‘Person’ to itself.

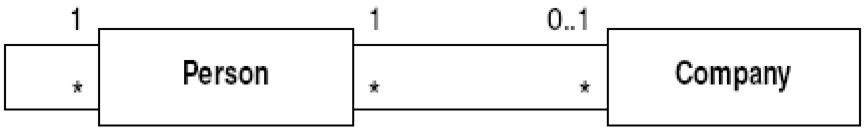
6. In the following Figure, state which of the object diagrams are legitimate instances of the class diagram given. Assume that all links in the object diagrams are instances of the association in the corresponding class diagram. If an object diagram is not a legitimate instance, explain why not.



7. Repeat the previous question for the diagrams given in the following figure .

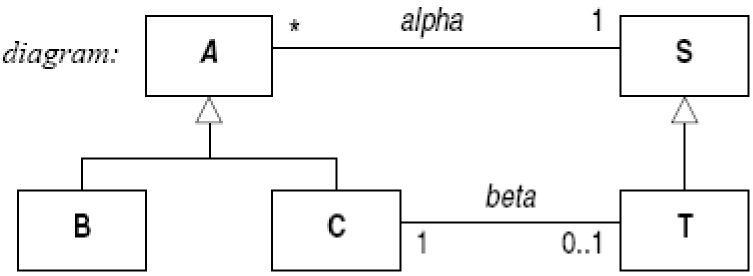


8. Companies may employ many people, and people may work for many companies. Each company has a managing director, and every employee in a company has a manager, who may manage many subordinate employees. Add suitable labelling to the class diagram in the Figure to make this intended meaning clear.

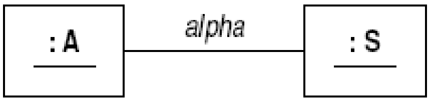


9. In the following Figure, state which of the object diagrams are valid instances of the class diagram given. If an object diagram is not a valid instance, explain why not.

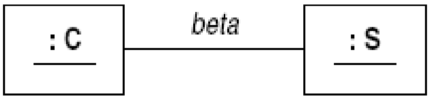
Class diagram:



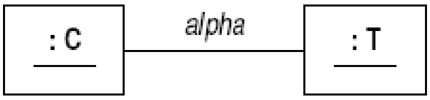
(a)



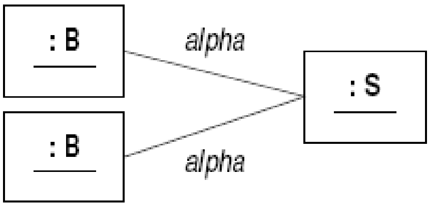
(b)



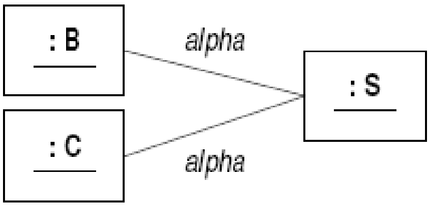
(c)



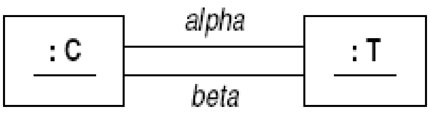
(d)



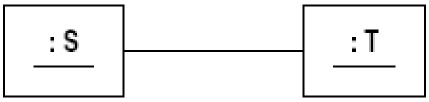
(e)



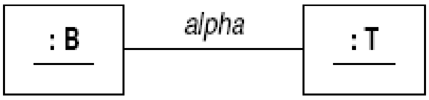
(f)



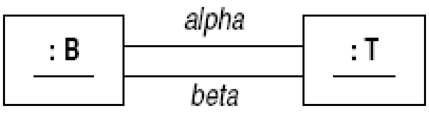
(g)



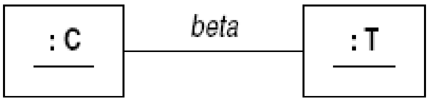
(h)



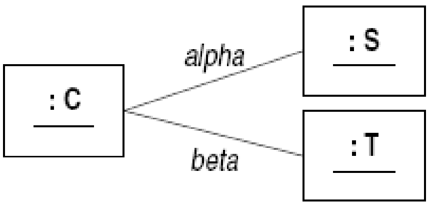
(i)



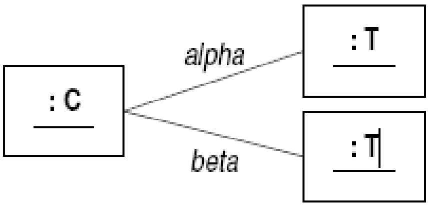
(j)



(k)



(l)



10. Draw a class diagram summarizing the following facts about a library. Discuss your design decisions, and any limitations of your model.

> For each book held by the library, the catalogue contains the title, author's name and ISBN number of the book. There may be multiple copies of a book in the library. Each copy of a book has a unique accession number. There are many registered readers belonging to the library, each of whom is issued with a number of tickets. The system records the name and address of each reader, and the number of tickets that they have been issued with. Readers can borrow one book for each ticket that they possess, and the system keeps a record of which books a reader has borrowed, along with the date that the book must be returned by.

11. Here is an extract from the documentation of a drawing tool for graphical workstations.

> The objects in the package are divided into *primitive objects* and *compound objects*. The primitive objects are: *arc*, *ellipse*, *polyline*, *polygon*, *box* and *text*. A primitive object can be moved, rotated, flipped vertically or horizontally, copied or erased. The *text* primitive may not be flipped. A compound object is composed of primitive objects. The primitive objects that constitute a compound cannot be individually modified, but they can be manipulated as an entity; a compound can be moved, rotated, flipped vertically or horizontally, copied or erased. A compound that contains any boxes may only be rotated by 90 degrees.

Based on this description, draw a class diagram using generalization to show the relationships between the various different kinds of graphical object in the drawing package. Discuss the limitations of your model, and any significant design decisions that you make.

12. A customer places an order with a supplier. The order is for various numbers of different kinds of parts; the distinction between different kinds of parts can be ignored for the purposes of this exercise. An order consists of a number of order lines; each line specifies a particular part from the suppliers catalogue, and says how many are to be ordered. In response to an order, the supplier makes up a delivery, consisting of all the ordered parts. Describe this situation in a class diagram, and discuss the possible use of aggregation in your diagram.

13. The UK banking system consists of a number of banks. Each bank has a number

of branches, each identified by a unique sort code. Banks maintain accounts, each with a unique account number; in addition, each account is held at a particular branch of the bank. Some accounts allow cheques to be written; each cheque is identified with a cheque number.
Draw a class diagram to represent this information, paying particular attention to the use of qualified associations. Explain any assumptions or design decisions that you make.

14. Based on the following description of part of the syntax of a programming language, construct a class diagram showing the structure of programs written in the language.

A *module* consists of a collection of named *features*. A feature can either be a *variable,* a *routine* or a nested module. Routines consist of a *declaration part* and a *statement part*. Features local to the routine can be declared in the declaration part, and the statement part consists of a non-empty sequence of statements. Statements can be *loops*, *conditionals* or *assignments,* and each assignment contains a reference to the variable which is being assigned to.

15. Draw a class diagram modelling the system described in the following:

A company has decided to computerize the circulation of documents round its offices, and to do this by installing a network of *electronic desks*. Each desk provides the following services:

a *blotting pad*, which can hold a document that the user is currently working on. The blotting pad provides basic word-processing facilities;

a *filing cabinet*, which models a physical filing cabinet. It is divided into drawers, and each drawer is divided into folders. Documents can be stored either in drawers or in folders within drawers;

a *mail service,* which allows the user to communicate with other users on the network. Each desk is provided with three *trays*, corresponding to the IN, OUT and PENDING trays in traditional offices. The network will automatically put new mail in a user's IN tray, and periodically take documents from the OUT tray and mail them to their recipients.

Documents can be moved between the mail trays and the blotting pad, and between the blotting pad and the filing cabinet. There is no provision to move documents directly between the trays and the filing cabinet. Only one document can be on the blotting pad at any given time.

16. Construct a class diagram based on the information contained in the following description of the `info` system provided with the Emacs text editor. Where appropriate, give justifications for the design decisions you make.

The `info` system in Emacs provides a simple hypertext reading facility enabling on-line documentation to be browsed from within the editor. The `info` system consists of a number of *files*, each roughly corresponding to a single document. Each file is divided into a number of *nodes*, each node containing a small amount of text describing a particular topic. One node in the system is identified as the *directory* node: it contains information about the files available in the `info` system, and is presented to the user when the system is started.

Nodes are connected together by *links*: each link connects two nodes, and operations are provided enabling the user to move from one node to another via a link: in this way it is possible to browse through the complete document. There are three important kinds of links, called *up*, *next* and *previous* links: the names imply that the nodes in a document will be structured in a hierarchical manner, but this is not enforced by the system. In addition to links, a node can contain a *menu*.

A menu consists of a number of entries, each of which connects to another node in the system.

When the system is running, a record is kept in a *history list* of all the nodes that have been visited, thus enabling users to retrace their path through the document.

17. Construct a class diagram summarizing the aspects of the window manager described below:

When you are running the system, your work takes place on the *desktop*, the screen space occupied by the window manager. At any time you will be running a number of applications; each application will be displayed on the desktop either as an icon or in an application window. The control application starts when the window manager is started up; terminating this application terminates the current session.

As you work with applications, two kinds of windows appear on your desktop: application windows, and windows contained within the application windows, often known as *document windows*. All windows contain a title bar, maximize and minimize buttons, a control-menu box, and optionally horizontal and vertical scroll bars. In addition, application windows contain a menu bar; operations chosen from the menu bar can affect either the application window or any of the document windows contained within it.

18. Users of a network are authorized to use certain workstations. For each such machine, they are issued with an account and password. Draw a class model describing this situation, and discuss any assumptions that you make.

19. Imagine a library full of books. Each book contains a bibliography; each bibliography consists of a number of references to other books. Typically, a book will be referred to in many places, and therefore a reference can appear in more than one bibliography.

Draw a class diagram for this situation, and discuss the possible uses of aggregation in the diagram.

20. Construct a class diagram for the Electronic Noticeboard (EN) system described in the following:

> The EN is a system designed to facilitate communication between a group of users of a computer system. It enables notices to be posted that all users can read, and permits discussions to take place between the users.
>
> When a user logs on to the EN, they are presented with a user workspace that contains two main areas, the *noticeboard* and the *discussion groups*. The user has to choose which area they want to access; it is possible to move freely between the areas at any later stage.
>
> The noticeboard contains a list of notices, and a user can choose between reading any of the existing notices, or adding a new notice to the noticeboard. A notice must have an expiry date; after that date it will be archived and will no longer appear on the standard noticeboard. By default, all notices appear to all users until their expiry dates; a user can choose to remove specified notices from their private view of the noticeboard, however, although this is not recommended.
>
> The system also maintains discussion groups each dealing with a particular topic. Each discussion consists of a number of contributions. Users can choose which, if any, of the discussions they wish to read. By default, only unread contributions in a particular discussion will be presented to the user. A discussion is started when a user posts an initial contribution. Other users can respond to this by posting *followup contributions*, which in turn can generate further followups. If a followup

is deemed to have moved sufficiently far away from the original topic of discussion, it can be identified as the initial contribution of a new discussion group; it will then be accessible through both the old and new groups. All notices and contributions are archived, together with their date and in the case of contributions, a record of the contributions, if any, that they are following up.

A user starting a new discussion can specify that only a subset of the registered users of the EN can access the group. It is possible to allow users to have read-only, read-write or no access to a group. By contrast, all notices are readable by all users. A user can only read an archived contribution if they had read access to the group it was originally posted to.