

48024 Object Oriented Design

Self-Study Module: Sequence Diagrams

This module covers exploring and modelling object interactions with UML sequence diagrams.

Objectives:

- To develop skill in understanding and creating sequence diagrams.

References:

- Simon Bennett, Steve McRobb, Ray Farmer, "Object-Oriented Systems Analysis and Design using UML", McGraw-Hill, 2nd edition, 2002. Chapter 9

Contents:

Exercise 1 - Creating sequence diagrams from scenarios.....	2
Exercise 2 - Implementing sequence diagrams in code.....	3

Self-Study Module: Sequence Diagrams

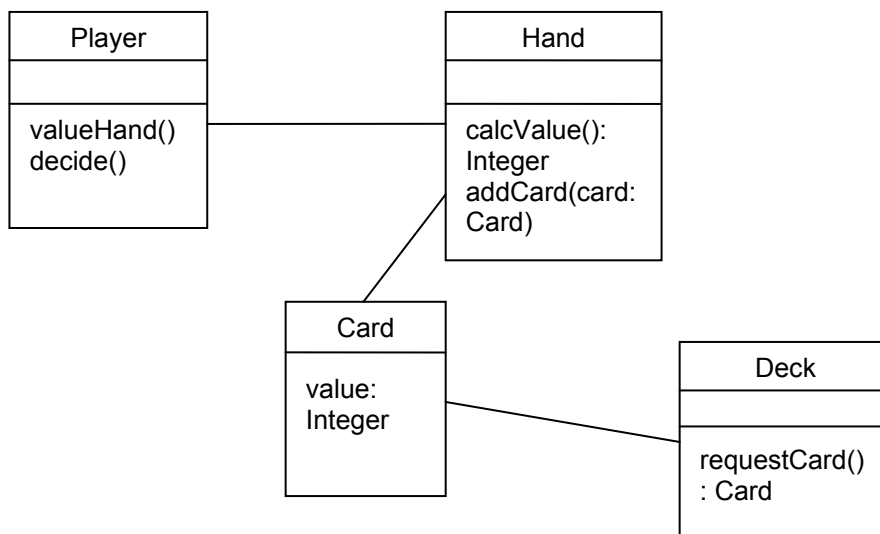
Exercise 1 - Creating sequence diagrams from scenarios

This exercise is based on the BlackJack game. You will draw a sequence diagram for the Use Case *Single Player Turn* – Scenario *Requesting another card*. It may be useful to think of an actual hand with cards (e.g. 2♥ and J♠). Assume that the system is modelled by classes Player, Hand, Card and Deck. This is illustrated in the class diagram below. You may add or change the model illustrated in the class diagram - it is only a starting point.

Scenario *Requesting another card*

The player values their hand. Based on the value they decide to request another card (instead of sitting or going bust). The dealer gives them a new card.

Class diagram *BlackJack*



Discussion ...

What happens when you have an Ace (value = 1 or 11)? Does your sequence diagram handle this case? If not, how could you change it? You might need to redesign some of the classes. In doing this it is useful to think about where the responsibility lies - which class looks after deciding on the value of the hand if it contains an Ace?

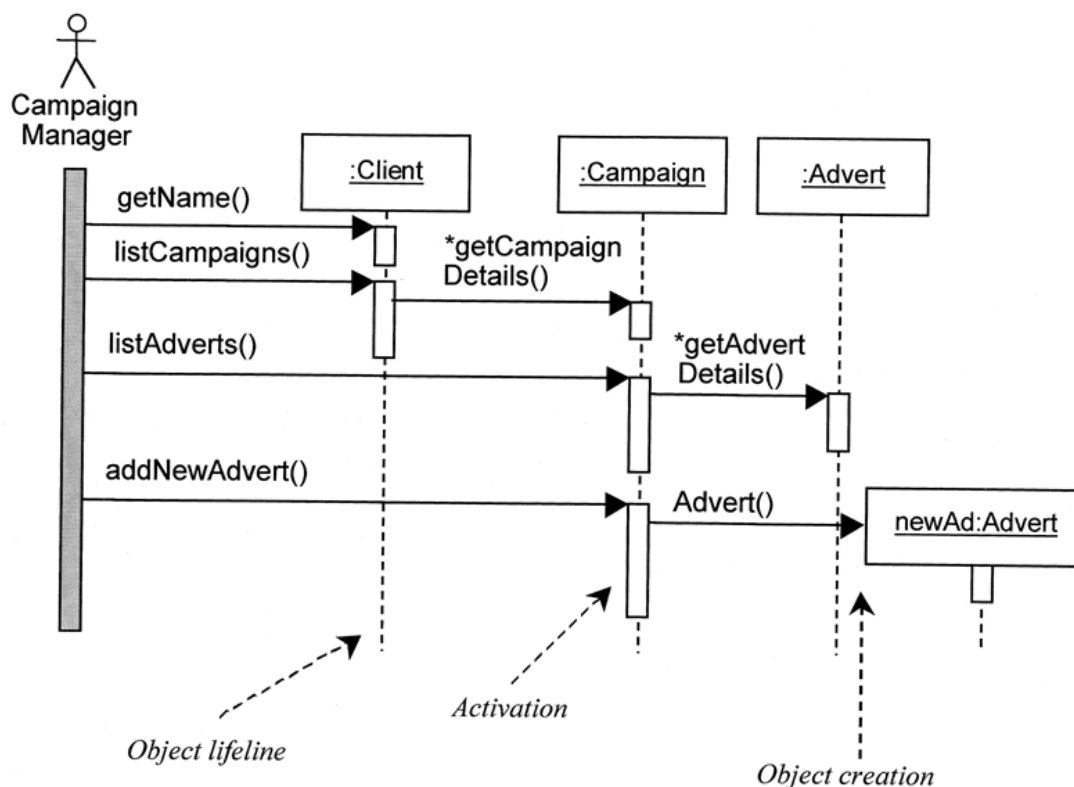
Self-Study Module: Sequence Diagrams

Exercise 2 - Implementing sequence diagrams in code

Write Java code fragments for the sequence diagram for Use Case *Add a new advert to a campaign* below.

Sequence diagram for Use Case *Add a new advert to a campaign*

(ref: Bennett Figure 9.3, p.235)



Note that this sequence diagram omits the user interface (or boundary) classes. In your code, pretend there is a single user interface class that sends messages from the Campaign Manager to the objects shown in the sequence diagram. You can assume that any user input data is available, e.g. to create a new Advert we need the user to enter the title and type and this data is stored in variables title and type – there is no need for the purposes of this exercise to code the user interface. The same applies for selection of a campaign.

Write the code to satisfy the use case depicted in the sequence diagram and code the following methods only:

- Client.getName()
- Client.listCampaigns()
- Campaign.addNewAdvert()
this method creates a new Advert object and then adds it to the adverts array.
- Advert constructor

48024 Object Oriented Design

Self-Study Module: Sequence Diagrams

Suggestions and hints:

- the operation listCampaigns() returns the information for the set of campaigns as a String.
- the operation listAdverts() returns the information for the set of adverts as a String.
- Use the Java toString() method to convert objects to a String.

Use the class diagram for operation signatures.

Class diagram

(ref: Bennett Figure A5.6, p.481)

