Problem 6

1 The basic problem

Suppose there are various different algorithms for carrying out “the same” task which is part of a responsibilities of a class called say Context; perhaps they differ in accuracy, speed, space use or whatever. Clients which create an object of class Context should be able to choose which algorithm they want to used in this Context object, but from then on they interact only with the Context.

For example, a text window might have the responsibility to display text, including calculating linebreaks. It might use a simple algorithm deciding on one linebreak at a time, or one of several better but more expensive algorithms which optimise linebreaks over a paragraph.

2 An initial solution...

3 A standard solution: STRATEGY

We define a separate abstract class – say Strategy – whose objects represent the algorithms which may be in force. The concrete subclasses each implement the behaviour required according to a different algorithm.

The object we’re interested in (which let’s say has class Context) knows about its Strategy. When a context receives a request, it passes the request on to its Strategy. If the algorithm required changes – either at runtime or as a maintenance change – its Strategy object is replaced by a new Strategy object of a different concrete subclass, which implements the newly required behaviour.

To check your understanding of the solution, sketch a class diagram for the case of the text window.

4 When and why is this solution good?