OLLSCOIL NA hÉIREANN

THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

SUMMER EXAMINATIONS 2000

B.Sc. Honours in Computer Science

CS4983 Object-Oriented Design

Dr. D.G. Bridge,
Professor C.J. Sreenan,
Professor J.G. Hughes

Answer **all** questions.

Time allowed: One-and-a-half hours

1. (*5%*) State briefly what people mean when they say that programmers should *program to an interface, not an implementation.*

2. (*15%*) What problem is the *iterator* design pattern supposed to solve?

   What are the *consequences* of using this design pattern (i.e. the advantages/benefits and disadvantages/costs)?

3. (*30%*) Look at the `Quote` class definition at the end of this exam paper. The program uses a text file of famous quotations. Each quotation is one line in the file, the quotation being separated from its author by a vertical bar, e.g.:

   ```
   Bah, Humbug|Scrooge
   Yo ho ho|Father Christmas
   Brrrr|Rudolph
   ```

   When you run the program, it is supposed to display, on two lines of the screen, a quotation that has been randomly taken from the file. For example, if it generates random number 1, the quotation displayed is:

   ```
   Yo ho ho
   Father Christmas
   ```

   (The first line of the file is line zero.)

   Give a detailed critique of the program from the point of view of its use of *defensive programming.* Mention all the places where it falls short of the ideal (if any) but also mention places where it does a proper job (if any).

4. (*50%*) At the end of this paper, we describe a computerised game, inspired by the pen-and-paper game of Battleships.

   Design a program to play this game.

   Present the structure of your design using text and one or more UML Class Diagrams. (If you cannot remember exact UML notation, invent your own, but include comments to explain the intention of your invented notation.)

   Your design should be detailed enough that a programmer could begin coding in Java from it. This means that you should discuss and/or show on your diagrams such things as attributes, operations, data structures, the main GUI classes, interfaces, packages, etc., in addition to classes and relationships from the problem domain. (Your diagrams can be accompanied by pseudocode and/or fragments of Java but only if you think these will help to explain your design to me and to make the design more detailed.)

   There will be extra credit for designs that make proper use of design patterns.

   It is anticipated that the game will be made more sophisticated in future (e.g. further different types of vessel, limited range sonar and radar facilities, further types of obstacle, etc.) Your design should not include these, but there will be extra credit for designs that are general enough that they could relatively easily accommodate extensions of these kinds in the future.

   Important: The examiners will not answer questions about the rules of the Battleships game during the examination. If you feel that there are problems with the description of the game (e.g. vagueness, ambiguity or incompleteness), resolve them yourself by stating assumptions in your answer.

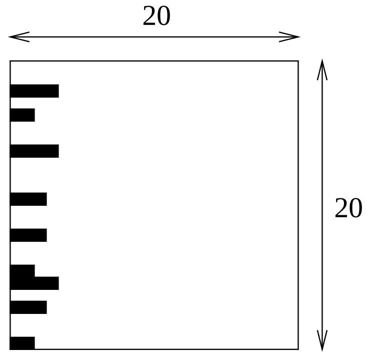
`Quote` **class definition (Question 3)**

```
import java.io.*;

public class Quote
{
   public Quote() throws IOException
   {  int quoteNum = (int) (Math.random() * FILE_SIZE);
      String line = readNthLine(quoteNum);
      quote = line.substring(0, line.indexOf("|"));
      author = line.substring(line.indexOf("|") + 1);
      if (quote == null || author == null)
      {  quote = BACKUP_QUOTE;
         author = BACKUP_AUTHOR;
      }
   }

   public void display()
   {  System.out.println(quote);
      System.out.println(author);
   }

   private static String readNthLine(int n) throws IOException
   {  int lineNum = -1;
      String line = "";
      boolean eof = false;
      BufferedReader reader =
         new BufferedReader(new FileReader(QUOTE_FILE));
      do
      {  line = reader.readLine();
         lineNum++;
         eof = line == null;
      } while (! eof && lineNum != n);
      reader.close();
      return line;
   }

   private String quote;
   private String author;
   private static final String QUOTE_FILE = "quotes.txt";
   private static final int FILE_SIZE = 100;
   private static final String BACKUP_QUOTE = "Will never mark the marble";
   private static final String BACKUP_AUTHOR = "Alexander Pope";

   public static void main(String[] args) throws IOException
   {  Quote q = new Quote();
      q.display();
   }
}
```

The Game of Battleships++ (Question 4)

In this game, the user plays against the computer. Both players begin the game with a fleet of vessels: three destroyers, three submarines and three minesweepers. The different types of vessel have different capabilities (see below). The aim of the game is to destroy the whole of the enemy fleet.

At all times, the program's GUI presents the user with a 20×20 grid, showing the position of his/her fleet. At no time can the user see the computer's fleet. At the start of the game, the user's fleet is lined up, at random positions along the left-hand side of the grid, facing the open sea. Destroyers are represented by a row of four adjacent cells, minesweepers by three cells, submarines by two cells, and mines by one cell.



The computer's fleet is similarly randomly positioned along the right-hand side of the grid, but cannot be seen by the user.

Players take alternate turns and, in each turn, can do *one* of the following:

- **Move:** The player can choose to move one of his/her vessels a certain number of cells in the direction in which the vessel is facing. Different types of vessel have different maximum distances that they can travel in a single move.

  Suppose the path of the vessel would pass through an already occupied cell. If the cell is occupied by a mine, then destroyers and submarines are destroyed by the mine, and both the vessel and the mine are removed from the grid; minesweepers, on the other hand, are not destroyed and have the option of removing the mine from the sea (so that it can be dropped again at a different position later). If the cell is occupied by another vessel (whether it be one of the player's own or one of the opponent's), then destroyers and minesweepers will be stopped prematurely by the obstruction; submarines complete their full move since they can pass under these obstructions.

- **Rotate:** The player can choose to rotate one of his/her vessels so that it is facing a different direction. A vessel can turn either 45° or 90°. If turning the vessel makes part of the vessel pass through a cell that is already occupied by a mine or another vessel, the rules that apply are those described for **Move** above.

- **Fire:** The player can choose to fire a torpedo from one of his/her vessels. The torpedo will have a certain maximum distance that it can travel, depending on the type of vessel from which it is fired. If unimpeded, it will travel its maximum distance and then explode harmlessly in the sea. But, if its path is obstructed by a mine or a vessel, the torpedo destroys the obstruction, and both are removed from the grid.

- **DropMine:** The player can choose to drop a mine from one of his/her minesweepers. The minesweepers begin the game with 5 mines each. Once dropped, mines remain immobile, unless collected on board again by a minesweeper.