DUDEBASE COLLEGES PROGRESS EXAMINATION

Tuesday 18 January 2022 09:00 – 10:00

Computer Science Paper 1 (CST IA)

Answer **one** *question from Section A, and* **one** *question from Section B. Each question is worth the same number of marks.*

Write on **one** *side of the paper only.*

Write your name and the question number at the top of every sheet, and tie your answers into separate bundles (one for each question).

DO NOT TURN OVER THE QUESTION PAPER UNTIL TOLD BY THE INVIGILATOR THAT YOU MAY DO SO

SECTION A

1 Foundations of Computer Science

Consider the following OCaml program (line numbers provided for convenience).

```
.1 type tree = Lf of int | Br of int * tree * tree;;

.2 exception GotIt of int;;

.3 let rec findSix = function
.4  | Lf(x) -> if (x = 6) then raise (GotIt 1) else ()
.5  | Br(v,l,r) -> if (v = 6) then raise (GotIt 1)
.6                 else
.7                   let v1 =
.8                    try findSix l ; 0
.9                    with GotIt(n) -> n
10                   in
11                   let v2 =
12                    try findSix r ; 0
13                    with GotIt(n) -> n
14                   in
15                   let v = v1 + v2
16                   in
17                    if (v = 0) then () else raise (GotIt v)
18 ;;
```

(a) Explain what the `findSix` program does, stating its type and including what happens when it is run on each of the following two trees. [5 marks]

```
let t1 = Br( 4, Lf 6 , Br(6, Lf 3 , Lf 6) );;
let t2 = Br( 4, Lf 7 , Br(8, Lf 3 , Lf 1) );;
```

(b) Write a (Curried) function `find n` that returns a function that computes the same result when $n = 6$ and analogous results for other values of n. Your function should not use exceptions. [7 marks]

(c) Write a program that, given a `tree`, returns `true` if the value in the root appears anywhere beneath it in the tree. [3 marks]

(d) Write a program that returns `true` if any path from the root to a leaf returns the same value more than once. [5 marks]

2 Foundations of Computer Science

We wish to represent and query a relational database using OCaml. A table can be represented as a list of rows; a row can be represented a list of attributes; and an attribute can be represented as a pair of type `string * sqlval` where the string is an attribute name and `sqlval` is defined below.

```
type sqlval = Null | Int of int | String of string
```

(*a*) Write a function `select attrs t` that returns a table containing one row for each row in table `t` but only holding only the attributes named in the `string list attrs`. [3 marks]

(*b*) Write a function `where p t` that returns a table containing the rows from `t` that satisfy predicate `p`. The type of `p` should be `(string * sqlval) list -> bool`. [2 marks]

(*c*) Write a function `from t1 t2` that returns a table containing the rows in the cartesian product of the inputs tables `t1` and `t2`. Raise an exception if the resulting rows contain duplicate attribute names. [4 marks]

(*d*) Write a function `lookup a r` that extracts an attribute (i.e. a value of type `string * sqlval`) from the row `r` where the attribute is named `a`. Raise an exception if no such attribute name exists. [2 marks]

(*e*) Write a function that generates predicates for equality testing. `equals a1 a2` should return a predicate (suitable for use as `p` in your solution to part (b) above) that returns true if and only if a row contains a value for both attribute names `a1` and `a2`, and those values are (structurally) equal in OCaml. Note that `Null` values are not equal to anything, including other `Null`s. [4 marks]

(*f*) Write a function `count` such that the following code returns a table with attributes 'person' and 'COUNT_OF_sport'. Each row should contain a unique value of the 'person' attribute from `t` and the 'COUNT_OF_sport' is the number of non-NULL values of the `sport` attribute appearing in rows of `t` with that value of the 'person' attribute. The first argument to `count` is a list of the attribute names to group by. Your program should use an internal data structure of type `(string * sqlval) list -> int ref`.

```
count ["person"] "sport" t
```

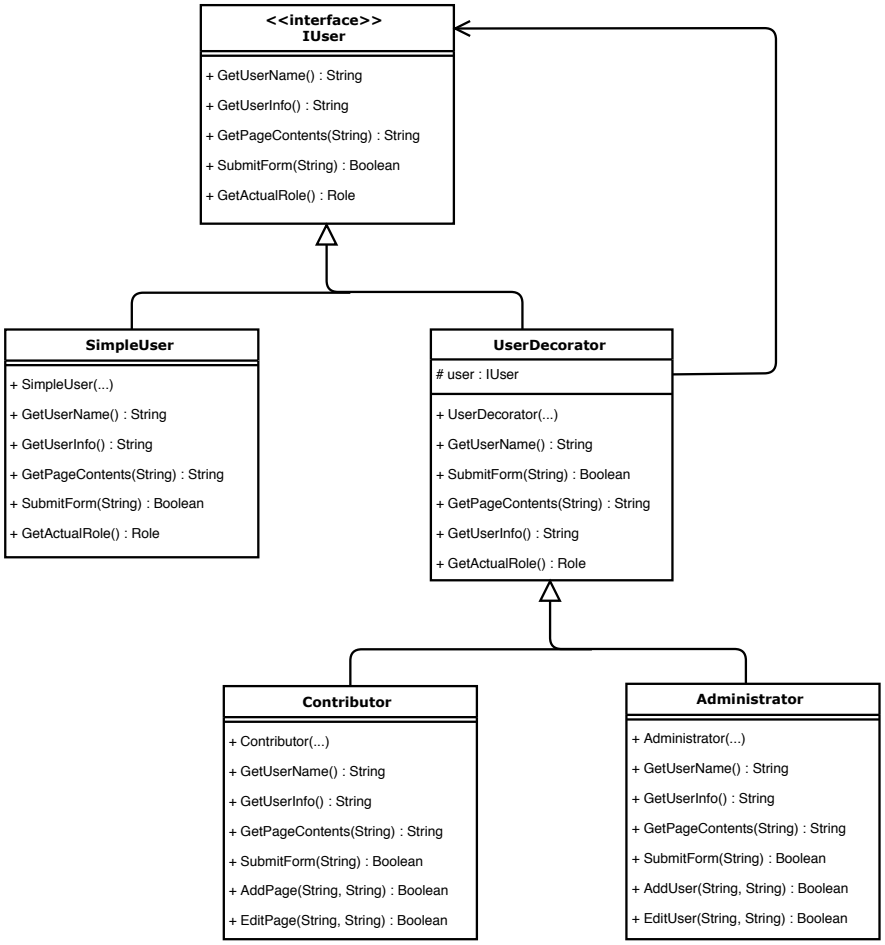
[5 marks]

SECTION B

3 Object Oriented Programming

(a) An array can be used to represent the decimal digits of a fraction. For example, the number $1/7$ has decimal representation $0.142857\ldots$ and (a finite approximation of) the value could be stored in an int array $[1, 4, 2, 8, 5, 7, \ldots]$ where each digit $\in [0, 9]$. Write a Java method that multiplies a fraction stored as an int array by a single decimial digit integer, returning the whole number part of the result and leaving the input array modified in-place to hold the updated fractional part of the product. [4 marks]

(b) Consider the following UML class diagram. It describes a class hierarchy for users in a "*web app*". "*Simple users*" have the ability to get page contents and submit forms. "*Contributors*" can additionally add and edit pages. "*Administrators*" can add and edit users.



To avoid an *administrator* or *contributor* having the ability to make an accidental change when idly using the app, they always start as *simple users*, but can be elevated at runtime to gain the additional functionality of their actual rôle.

`Role` is an enumerated type, defined as follows:

```
public enum Role { USER, CONTRIBUTOR, ADMINISTRATOR }
```

(*i*) Why is it useful for there to exist a "*has-a*" relationship between `UserDecorator` and `IUser`, in addition to the "*is-a*" relationship? [2 marks]

(*ii*) Write Java code to show how the `UserDecorator` class would be implemented. Show explicitly the parameters you would pass to the class's constructor. [4 marks]

(*iii*) Show how the `Contributor` class would be implemented (again, including a definition of the constructor with each of its formal parameters listed explicitly). Write a Java method

```
public static IUser ElevateToContributor(IUser user) throws
InsufficientPrivilegesException
```

that elevates a `SimpleUser` to a `Contributor` at runtime, or throws an exception (for which you should include a definition explicitly) if the user does not have sufficient privileges to become a `Contributor` (i.e. if the user's actual role is not a `CONTRIBUTOR`). [5 marks]

(*c*) It is required to print the result of calling `GetUserInfo()` on each user at the end of each day. The user info should appear ordered alphabetically by name.

Show how this can be achieved using the Composite pattern and the `Comparator` interface. You should include definitions for the `PrintUserInfo()`, `AddUser(...)` and `RemoveUser(...)` methods in your Composite class. You can assume that the `AddUser(...)` method will be called whenever a new user is created and the `PrintUserInfo(...)` method will be called at the end of each day by the system. [5 marks]

4 Object Oriented Programming

(*a*) Provide outline code for the following aspects of a smart home system, explaining your design choices.

(*i*) An exception, `smarthome.MalfunctioningDeviceException`, having a constructor of two String arguments naming the device and describing the nature of its malfunction. [2 marks]

(*ii*) Provide a class `smarthome.Device` that stores a device's name and a list of its configuration settings. You should use at least one Map and may assume that the names of settings are Strings and that their values are integers representing percentages, but your code should check that the values are `>= 0` and `<= 100`. [4 marks]

(*b*) Provide two classes, each with a single method taking arguments `(String setting, Device [] devices)` to process a set of Devices as follows.

(*i*) Print to `System.out` the devices sorted by their value for the 'setting'. Devices with no configured value for that setting should appear in alphabetical order at the end of the output. [3 marks]

(*ii*) Print to `System.out` an alphabetical list of devices, one line each, with their minimum config value, maximum value and number of configured parameters. Separate name, min, max, and count with commas. [3 marks]

(*c*) A Caesar Cipher encypts a message composed of capital letters and spaces by adding some fixed amount (the key) to each letter. If a letter plus the key extends beyond Z, you wrap back to A. For example, if the key is 2 then plaintext "JOHN" results in ciphertext "LQJP" and "ZOO" becomes "BQQ". Spaces between words remain as spaces. Given that 'E' is the most common letter of the English alphabet, write a Java method to guess the key used to encrypt a given ciphertext. (You may provide additional helper methods if you wish.) [4 marks]

(*d*) Given a rectangular 2D array of integers (positive and negative), find how many rectangular sub-arrays (of non-zero size) sum to zero. Correctness is more important than efficiency. [4 marks]

END OF PAPER