Assignment 7

Samuel Godwin, Computer Science BSc, K1630575

November 28, 2016

1 Introduction

The purpose of this assignment is to create a program in which a pirate sails to a series of islands where a set of treasure chests are buried, in pursuit of the contents hidden within each of these chests. The contents of these chests are coins, each of which is completely unique.

(Pseudocode on p2).

2 Pseudocode

GoldCoin Class

```
CLASS ‘Room’:
    Create static int field ‘coins’
    Create final int field ‘coinNumber’
    METHOD ‘GoldCoin’:
        coinNumber = coins++
    END METHOD.
    METHOD ‘getCoins’:
        Return coins
    END METHOD.
    METHOD ‘getCoinNumber’:
        Return coinNumber
    END METHOD.
END CLASS.
```

**See last week’s pseudocode for style of pseudocode

Player Class

```
CLASS ‘Player’:
    Create String field ‘name’
    Create int field ‘lives’
    Create Room field ‘currentRoom’
    Create boolean field ‘outcome’
    METHOD ‘Player’ <lives, currentRoom>:
        Set field ‘lives’ to <lives>
        Set field ‘currentRoom’ to <currentRoom>
    END METHOD.
    METHOD ‘move’ <nextRoom>:
        IF      result of ‘getContainsMonster’ through <nextRoom>  ==  true THEN:
                Set field ‘lives’ = ‘lives’ – 1
                Set field ‘outcome’ = false
        ELSE:
                Set field ‘currentRoom’ to <nextRoom>
                Set field ‘outcome’ = true
        ENDIF.
        Return outcome
    END METHOD.
    METHOD ‘setCurrentRoom’ <currentRoom>:
        Set field ‘currentRoom’ to <currentRoom>
    END METHOD.
    METHOD ‘setName’ <name>:
        Set field ‘name’ to <name>
    END METHOD.
    METHOD ‘getCurrentRoom’:
        Return currentRoom
    END METHOD.
    METHOD ‘getLives’:
        Return lives
    END METHOD.
    METHOD ‘getName’:
        Return name
    END METHOD.
END CLASS.
```

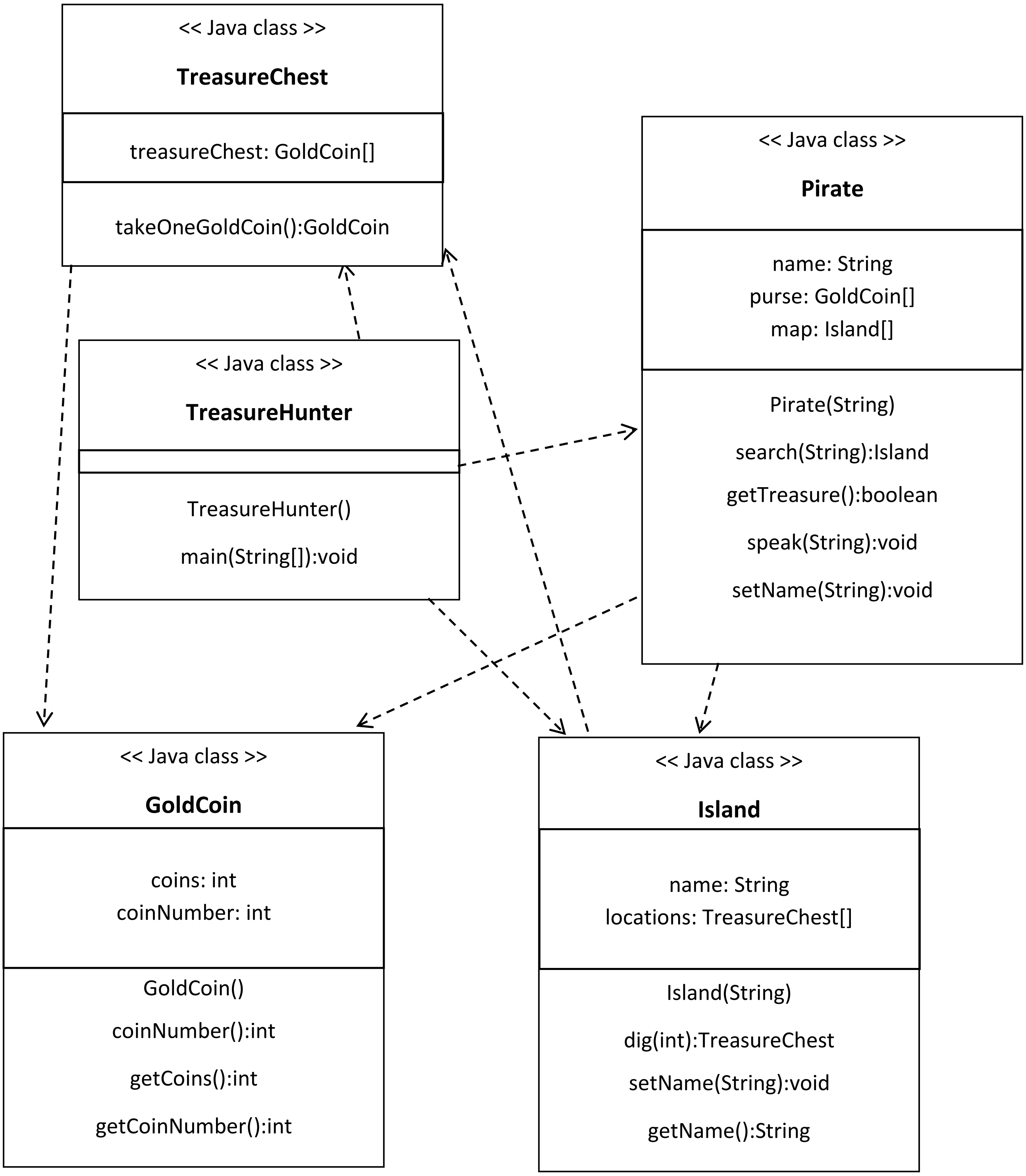
DoorMazeGame Class

```
CLASS ‘DoorMazeGame’:

    Import ‘Scanner’

    METHOD ‘main’:

        Create instance of ‘Scanner’ called ‘in’

        Create object of ‘Room’ called ‘monsterRoom’<name>,
                                    set name = “The Monster Room”

        Go to ‘setContainsMonster’ in ‘monsterRoom’, set containsMonster = true

        Create object of ‘Room’ called ‘room1’<name>, set name = “Cave One”

        Create object of ‘Room’ called ‘room2’<name>, set name = “Cave Two”

        Create object of ‘Room’ called ‘room3’<name>, set name = “Cave Three”

        Create object of ‘Room’ called ‘room4’<name>, set name = “Cave Four”

        Create object of ‘Room’ called ‘room5’<name>, set name = “Cave Five”

        Create object of ‘Room’ called ‘room6’<name>, set name = “Cave Six”

        Go to ‘setBlueDoorRoom’ in ‘room1’, set blueDoorRoom = room2

        Go to ‘setRedDoorRoom’ in ‘room1’, set redDoorRoom = monsterRoom

        Go to ‘setBlueDoorRoom’ in ‘room2’, set blueDoorRoom = monsterRoom

        Go to ‘setRedDoorRoom’ in ‘room2’, set redDoorRoom = room3

        Go to ‘setBlueDoorRoom’ in ‘room3’, set blueDoorRoom = room4

        Go to ‘setRedDoorRoom’ in ‘room3’, set redDoorRoom = monsterRoom

        Go to ‘setBlueDoorRoom’ in ‘room4’, set blueDoorRoom = monsterRoom

        Go to ‘setRedDoorRoom’ in ‘room4’, set redDoorRoom = room5

        Go to ‘setBlueDoorRoom’ in ‘room5’, set blueDoorRoom = room6

        Go to ‘setRedDoorRoom’ in ‘room5’, set redDoorRoom = monsterRoom

        Go to ‘setIsFinalRoom’ in ‘room6’, set isFinalRoom = true

        Create object of ‘Player’ called ‘player’<lives, currentRoom>,
                                    set lives = 2
                                    set currentRoom = room1
```

```
Print “DOOR MAZE GAME. CHOOSE EITHER THE RED OR BLUE DOOR IN EACH ROOM
    TO ATTEMPT TO ENTER THE NEXT ROOM. AVOID MONSTER!”
Print “Enter name: ”
Go to ‘setName’ in ‘player’, set name = String input from user
DO:
    Print result of ‘getName’ and ‘getLives’ in ‘player’
    Print result of ‘getName’ through ‘getCurrentRoom’ in ‘player’
    Print “Please input your choice of door, either 'blue' or 'red':”
    Create String variable ‘nextRoom’, set ‘nextRoom’ = String input from user
    IF ‘nextRoom’ == “red” THEN:
        IF result of ‘getRedDoorRoom’ through ‘getCurrentRoom’
          in ‘player’ == true THEN:
            Print “Correct choice”
        ELSE:
            Print “Not correct choice”
        ENDIF.
    ELSE IF ‘nextRoom’ == “blue” THEN:
        IF result of ‘getBlueDoorRoom’ through ‘getCurrentRoom’
          in ‘player’ == true THEN:
            Print “Correct choice”
        ELSE:
            Print “Not correct choice”
        ENDIF.
    ELSE:
        Print “You must enter either ‘red’ or ‘blue’ for your choice!”
    ENDIF.
```

```
        WHILE result of ‘getIsFinalRoom’ through ‘getCurrentRoom’ in ‘player’ == false
            AND result of ‘getLives’ in ‘player’ != 0.
        ENDWHILE.
        IF result of ‘getLives’ in ‘player’ == 0 THEN:
            Print “You have lost all your lives - YOU LOSE.”
        ELSE:
            Print “You find yourself in the final room! YOU WIN!”
        ENDIF.
        Call the ‘close’ method on my ‘Scanner’ instance
    END METHOD.
END CLASS.
```

3 Class Diagram



4 Description

My solution to this assignment involves the creation of a program in which my driver class ‘TreasureHunter’. My program consists of five classes ‘TreasureHunter’, ‘Island’, ‘Pirate’, ‘TreasureChest’ and ‘GoldCoin’. My GoldCoin class contains two fields, the first of which is of type ‘static int’ called ‘coins’. The second of my fields for this class is called ‘coinNumber’ which is of type ‘final int’. These fields play a role in creating a ‘blockchain’ effect ensuring that every coin created is unique. The constructor for this class simply contains the line ‘coinNumber = coins++;’. My other methods in this class are accessor methods for ‘coinNumber’ and ‘coins’. My TreasureChest class contains an array ‘treasurechest’ made up of 11 GoldCoins initially. This class contains a method ‘takeOneGoldCoin’ which contains a while loop.

My driver class, ‘TreasureHunter’, contains my main method.

The PirateClass contains a field and two arrays. The field is called ‘name’ and is a string type. The first array is called ‘Purse’ and is of the GoldCoin type. The second array is called ‘Map’ and is of the Island type. This class contains three methods called Pirate, search and speak. It also contains an accessor method called getTreasure and a mutator method called setName.

The Island class contains a field Name which is a string type, and an array called Location which is a TreasureChest type. The class contains two methods called Island and dig. This class also contains a mutator method called setName, and an accessor method called getName.

One element of my code I am particularly proud of was using the ‘Random’ java library class to randomly generate a number which I then used as an index in my array of pirate suffixes, since this worked very well for me the first time I tried it. I used this to randomise a pirate suffix on to the end of print statements in my code.

I found this week’s assignment as a whole very difficult to implement in the time available following a test in another module on the course.