Faculty of Science, Engineering and Technology

Object Oriented Programming

Pass Task 8.1: Semester Test

Overview

Note: This task is a time-bound test, replacing the usual Semester Test. You have 5 days to complete it (within Week 8 of semester), and only one opportunity to resubmit the task after receiving feedback. Note that this IS NOT a hurdle requirement, but counts as a Pass Task to be included in your final portfolio

You have been using object oriented programming to implement the programs you have created in this unit. In this task you need to express your understanding of the principles associated with this programming paradigm (abstraction, encapsulation, inheritance, and polymorphism).

Purpose:	Test your understanding of Object Oriented Programming and the core concepts of Object Oriented design.
Task:	You must complete two questions. The first is a coding question, to be submitted as C# source code files and a screen shot showing the output. The second will require a written response, and may be submitted as a PDF
Time:	This task should be completed before the start of Week 8.
Resources:	

Submission Details

You must submit the following files to Doubtfire:

- For Question 1:
  - C# code files of the classes created.
  - An image showing your modified UML Diagram
  - A Screenshot of output.
- For Question2:
  - A PDF document with your Question 2 answer

Make sure that your task has the following in your submission:

- Readable code, documented as needed to ensure its understanding.

Question 1

Consider the following program:

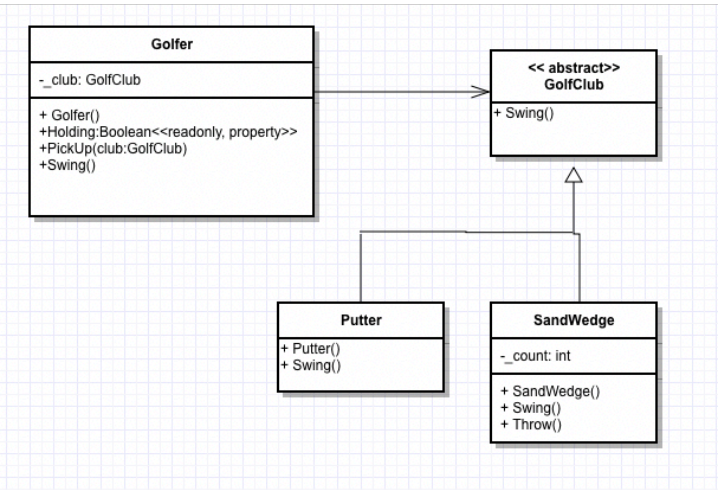

Figure 1: C# Golfer UML Class Diagram

Golfer: A Golfer can be equipped with a Golf Club using the PickUp method; the golf club is then stored in the Golfer's club field. The Golfer's holding property is true when its golf club field is not null. When the Golfer is told to Swing, the following occurs:

1. If the Golfer is holding a golf club:

   "Breathe and focus" is printed to the console. (b) The Golfer tells its club to swing.

2. Else…

   "Where is my caddy?" is printed to the console.

GolfClub: Any golf club that can be swung; a GolfClub can also be held by a Golfer. GolfClub is an abstract base class.

Putter: A Putter is a GolfClub that, when swung, prints "putt putt putt".

SandWedge: A SandWedge is a GolfClub that can be swung 5 times before it is thrown in frustration. (The constructor sets the swing counter to 5 and calling the Throw method resets the play counter to 5.) When a SandWedge is swung, it does one of two things: if the remaining swing count is larger than zero, it prints "I am in my happy place" and the swing count is decremented by one; otherwise it prints "hand me my hockey stick".

Tasks

A change of requirements means the Golfer must now be able to hold multiple golf clubs. Code that uses a Golfer object may therefore ask for a specific golf club to be operated by referring to that GolfClub by name.

You are required to:

a) Provide a new class diagram (UML format) that meets the new requirements (hand drawn is fine).

b) Write the code for all classes, including all methods/fields/constructors required.

c) Write a short program that tests this new design.

Submit a photo or image of your UML, all source code files and screen shorts showing your working program to Doubtfire.

Question 2

Explain the four principles of object oriented programming. For each of the principles, describe a piece of work that you have completed for this unit and explain how it demonstrates the principle. (Tip: Be concise and list points.)

Submit your answer as a PDF document to Doubtfire.