COMP356/COMP556: Object Modelling and Object Oriented Design Methods (Exam Paper)

Instructions. Credits will be given for the best two answers only.

Question 1 and 2 are based on the following bicycle shop environment.

> We are in a year 3000 bicycle shop. The shop deals with different types of bicycles: race bikes, city bikes, and mountain bikes. A customer enters the shop either to buy a bicycle or to rent a bicycle or to have a bicycle serviced. The process of buying a bicycle has been completely automatised. The customer reads a catalogue, chooses the bicycle by filling in an order form, then inserts the form into an optical form-reader device. This device is connected to a computer that checks the client's details (including her/his credit card number) and the product details and decides whether to execute the required transaction. After the transaction has been validated the computer activates a robot that either finds the bicycle in the store and returns it to the customer or builds a bicycle matching the customer requirements using basic components (such as the frame, the chain set, the gears, the wheels, the tubes) available in the store. Bicycles can be rented at different rates (10 pence per hour for a racer, 5 pence per hour for a mountain bike and 1 pence per hour for a city bike). The process of renting a bike is similar to the process of buying one, except that the customer must insert her/his credit card into a device that will perform the required checks, authorise the deliver of the rented bike (again through the robot) and then keep the card locked until the bike is returned. Bicycles are serviced by a set of workers that perform the various tasks using a set of tools (including spanners, screwdrivers, chain extractors, etc).

Question 1 (a) Define and give an example for each of the following terms: class, instance, associations, *or*-associations. (**16 marks**)

(b) Construct an object diagram for the bicycle shop described above showing the various classes and their relationships (including relationship names, multiplicity, and directions). (**20 marks**)

(c) Bicycles in the example above can be classified according to either their type or their purpose and this leads to join-classes like "race bike for sale" (see Figure 1 below). Explain one possible way to avoid multiple inheritance commenting on its advantages and drawbacks. (**14 marks**)

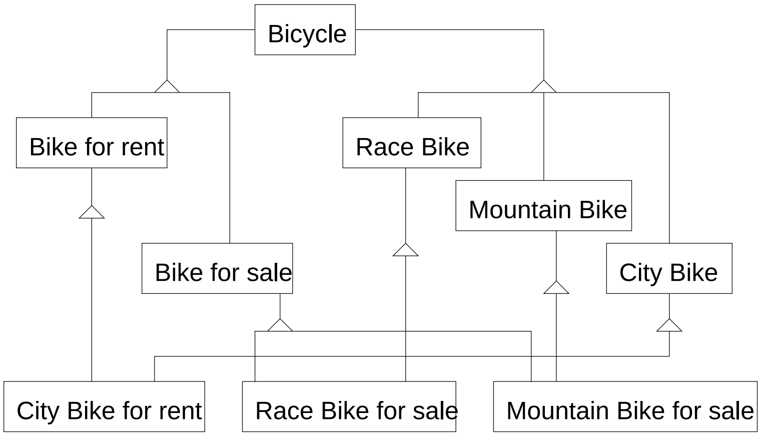

Figure 1: Multiple inheritance in the bicycle example.

Question 2 (a) What are the typical components of a Use-case diagram? What are these components in the bicycle shop example? (**14 marks**)

(b) Describe the buy bicycle process using an object interaction diagram. (**18 marks**)

(c) Describe the buy bicycle process using a collaboration diagram (**18 marks**)

Question 3 (a) Give a definition of state and state diagram. (**9 marks**)

(b) Show with the aid of an example the UML syntax of a state. (**9 marks**)

(c) Describe one possible way to translate the state diagram associated to a particular class into code. (**12 marks**)

(d) The following paragraph describes a simple game.

> There is a player and a target. Initially the player is idle and the target is hidden. The player's goal is to hit as many times as possible a target appearing at a random position in a bidimensional board. The target shows up at a random position on the board at random times sending a signal to the player, stays up until the player fires and then is hidden again from the player's sight. The player stays idle until he receives a signal that a target has shown up, at that point he aims at the target (meaning that he aims at a random position on the board) and then he fires at the aimed position. If the target position matches the player's aimed position the target is hit, the player wins a point and goes into idle state, while the target is hidden. Otherwise the player does not win a point, but again he goes in idle state, and the target is hidden.

Draw the state diagram for the player and the state diagram for the target and relate the two showing the messages that are exchanged between the two objects. (**20 marks**)