Racing Bike by Andrew Redford. Component/Composite class explanation, All classes and code explained and commented, Tests run to confirm program works.

The following table describes

* Which is the component class, and which is the composite class
* What the instance variables of each of these classes are and some examples of their values.
* Identifies and describes the method (not toString()) that allows the composite class to communicate with the component class in order to compute some value.
* And briefly describes other interesting features of the classes

Scenario: Tour de France Competing rider and racing bike.

This scenario models a competing rider in the Tour de France and the rider’s racing bike.

|  |  |
| --- | --- |
| 1(a) Composite class | CompetingRider |
| 1(b) Component class | RacingBike |
| 1(c) Relationship between the two and examples of use | A CompetingRider has-a RacingBike |
| 2(a) Composite class instance variables and examples of use | racingBike of type RacingBike (providing the has-as relationship)  riderName of type String, the name of the competing rider, e.g. “Chris Froome".  currentEnergyLevel of type String, the energy level of the competing rider e.g. "high", “low”, “medium”. |
| 2(b) Component class instance variables and examples of use | currentSpeed of type int, the current speed of the racing bike in km/h e.g. 50  bikeWeight of type double , the weight of the racing bike in kg e.g. 6.9. |
| 3 Description of method (not toString()) that allows the composite class to communicate with the component class in order to compute some value | CompetingRider has a method isInNeedOfFood() that returns true if the speed of the competing rider’s racing bike is below 10km/h and the current energy level of the rider is low. It computes a boolean value based on the current speed of the competing rider’s racing bike and the current energy level of the competing rider. |
| 4 Other interesting features of the class | CompetingRider provides getter and setter methods for its instance variables and also computes 2 other attributes:  isInNeedOfFood() which is set to true if the competing rider’s energy level is “low”.  extraTestIsNeeded() which is set to true if the name of the rider is Lance Armstrong.  There are no accompanying instance variables of these names.  RacingBike provides getters and setters for its instance variables.  RacingBike also contains the method isTooLightToRace() that returns true if the weight of the racing bike is below 6.8 kilos. It computes a boolean value based on the weight of the racing bike.  There are no accompanying instance variables of these names.  Each class includes a toString() method to help describe objects of the class. |

Question 2 (continued)

b)The component class:

/\*\*

\* RacingBike represents a racing bike in the Tour de France.

\* In the context of the CompetingRider, this is its component class.

\* This is for M250 TMA02 Q2.

\*

\* @author Andrew Redford F4730061

\* @version January 10 2019

\*/

public class RacingBike

{

//instance variables

private int currentSpeed;

private double bikeWeight;

/\*\*

\* zero-argument constructor: if we don’t know the racing bike's attributes.

\*/

public RacingBike()

{

this.currentSpeed = 0; //indicates current speed is unknown

this.bikeWeight = 0; //indicates bike weight is unknown

}

/\*\*

\* A constructor for objects of class RacingBike if we know its speed

\* but not its weight. Useful for testing for method isInNeedOfFood()

\* that allows the composite class to communicate with the component

\* class in order to compute some value.

\*/

public RacingBike(int aCurrentSpeed)

{

this.currentSpeed = aCurrentSpeed;

this.bikeWeight = 0;

}

Question 2 (continued)

b)The component class (continued):

/\*\*

\* A constructor for objects of class RacingBike

\* if we know its current speed and weight.

\*/

public RacingBike(int aCurrentSpeed, double aBikeWeight)

{

this.currentSpeed = aCurrentSpeed;

this.bikeWeight = aBikeWeight;

}

/\*\*

\* Setter for a racing bike's speed.

\*/

public void setCurrentSpeed(int aCurrentSpeed)

{

this.currentSpeed = aCurrentSpeed;

}

/\*\*

\* Setter for a racing bike's weight.

\*/

public void setBikeWeight(double aBikeWeight)

{

this.bikeWeight = aBikeWeight;

}

/\*\*

\* Getter for a racing bike's speed.

\*/

public int getCurrentSpeed()

{

return this.currentSpeed;

}

Question 2 (continued)

b)The component class (continued):

/\*\*

\* Getter for an racing bike's weight.

\*/

public double getBikeWeight()

{

return this.bikeWeight;

}

/\*\*

\* This method displays state-dependent behaviour:

\* The value returned depends on part of the state of the receiver.

\* If the receiver's weight is less than 6.8 true is returned since

\* bikes must weigh at least 6.8 kilos in the Tour. Otherwise false

\* is returned.

\*/

public boolean isTooLightToRace()

{

return this.getBikeWeight() < 6.8;

}

/\*\*

\* Returns a description of this object as a string.

\*/

public String toString()

{

return "Current bike speed:" + "(" + currentSpeed + " " + "km/h" + ")"

+ " " + "Bike's Weight:" + "(" + bikeWeight + " " + "kilos" + ")"

+ " " + "Bike too light to race?:" + "(" + isTooLightToRace() + ")";

}

}

Question 2 (continued)

b)The composite class:

/\*\*

\* CompetingRider represents a rider competing in the Tour de France such

\* as "Lance Armstrong". In the context of RacingBike, this is its composite

\* class. This is for M250 TMA02 Q2.

\* @author Andrew Redford F4730061

\* @version January 10 2019

\*

\*/

public class CompetingRider

{

//instance variables

private String riderName; // name of the rider e.g. "Chris Froome"

private String energyLevel; // energy level of the competing rider, e.g. "high"

private RacingBike racingBike; // a Competing rider has-a racing bike

/\*\*

\* Constructor for objects of class CompetingRider

\* allowing specification of all the required attributes.

\* (We don't check that all the data has been specified

\* correctly however. E.g. aRacingBike could be null.)

\*/

public CompetingRider(String aRiderName, String anEnergyLevel, RacingBike theRacingBike)

{

this.riderName = aRiderName;

this.energyLevel = anEnergyLevel;

this.racingBike = theRacingBike;

}

/\*\*

\* Constructor for objects of class CompetingRider,

\* where we only know the CompetingRider riderName.

\* (We might not know the other attribute values.)

\* This is the minimum information required to construct

\* a CompetingRider object. In this case I chose to make

\* the RacingBike object 'empty'.

Question 2 (b continued : The composite class)

\*/

public CompetingRider(String aRiderName)

{

this.riderName = aRiderName;

this.energyLevel = null; //energy level is a string value e.g. "high" or "low"

this.racingBike = new RacingBike();

}

/\*\*

\* Setter for a competing rider's name.

\*/

public void setRiderName(String aName)

{

this.riderName = aName;

}

/\*\*

\* Setter for a competing rider's energy level.

\*/

public void setEnergyLevel(String aValue)

{

this.energyLevel = aValue;

}

/\*\*

\* Getter for a competing rider's name.

\*/

public String getRiderName()

{

return this.riderName;

}

/\*\*

\* Getter for a competing rider's energy level.

Question 2 (b continued : The composite class)

\* This method returns "Unknown" if the level

\* is unknown (i.e. it has not been set).

\*/

public String getEnergyLevel()

{

if (this.energyLevel == null)

{

return "Unknown";

}

return this.energyLevel;

}

/\*\*

\* This method returns true if the competing rider's name

\* is Lance armstrong, otherwise it returns false.

\*/

public boolean extraTestIsNeeded()

{

if (this.riderName == "Lance Armstrong")

{

return true;

}

else

{

return false;

}

}

/\*\*

\* This method returns true if the energy level of the competing rider

\* is low and the speed of the rider's bike is below 10 km/h, otherwise it

\* returns false. This is the method that allows the composite class to

\* communicate with the component class in order to compute some value.

\*/

Question 2 (b continued : The composite class)

public boolean isInNeedOfFood()

{

if (this.energyLevel == "low" && this.racingBike.getCurrentSpeed() < 10)

{

return true;

}

else

{

return false;

}

}

/\*\*

\* Returns a string describing this object.

\*/

public String toString()

{

return "Competing Rider's Name:" + "(" + this.riderName + ")" + " "

+ "Extra tests needed?:" + "(" + this.extraTestIsNeeded() + ")" + " "

+ "Energy level:" + "(" + this.energyLevel + ")" + " "

+ "Needs food?:" + "(" + this.isInNeedOfFood() + ")" + " "

+ "Current speed of bike:" + "(" + this.racingBike.getCurrentSpeed() + " "

+ "km/h" + ")";

}

}

Question 2 (continued)

c) Here is my code to test toString() from the *composite* class creating new objects with all variables for RacingBike and CompetingRider included:

RacingBike bike1 = new RacingBike(10,6.8);

CompetingRider rider1 = new CompetingRider("Lance Armstrong", "high", bike1);

rider1.toString();

Expected output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(high) Needs food?:(false) Current speed of bike:(10 km/h)

Output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(high) Needs food?:(false) Current speed of bike:(10 km/h)

Test 2: Creating new objects with minimum information for RacingBike and CompetingRider to test both constructors are working for each class:

RacingBike bike2 = new RacingBike();

CompetingRider rider2 = new CompetingRider("Gary Gooseman");

rider2.toString();

Expected Output:

Competing Rider's Name:(Gary Gooseman) Extra tests needed?:(false) Energy level:(null) Needs food?:(false) Current speed of bike:(0 km/h)

Output:

Competing Rider's Name:(Gary Gooseman) Extra tests needed?:(false) Energy level:(null) Needs food?:(false) Current speed of bike:(0 km/h)

Question 2 (c continued)

* The following are tests for the method satisfying the third requirement in part a: isInNeedOfFood() which sends a message to the component object and uses the returned value in a computation.
* I use 6 calls to this method in order to demonstrate its correctness.
* I also change the values of one of the instance variables in the component object between the two calls.
* I test boundary values of current speed of rider’s racing bike, both when the bike’s rider has low energy and medium energy (3 speed values for each energy level).
* The only time the method should return true is when the rider has low energy and the current speed of rider’s racing bike is below 10

In tests 1 to 3 I have a “low” energy level in the instance of the component object and test the boundary value 11,10 and 9 for the speed of the cometing rider’s racing bike.

In tests 4 to 6 I have a “medium” energy level in the instance of the component object and test the boundary value 11,10 and 9 for the speed of the cometing rider’s racing bike.

All tests were successful, and they are documented below of the following 6 pages. See the project’s readme file for the opportunity to run these same tests.

Question 2 (c continued)

**Test 1 of 6**

First I put a “low” energy level in the instance of the composite object:

Create a competing rider with energy level: “low” and current speed of rider’s racing bike: 11

I set the rider’s energy level, executing the line:

rider1.setEnergyLevel("low");

Then I get the energy level:

rider1.getEnergyLevel();

expected output:

low

actual output:

low

Next I set the rider’s bike’s current speed to 11:

bike1.setCurrentSpeed(11);

Then I get it to check:

bike1.getCurrentSpeed();

expected output:

11

Actual output:

11

Now I see if the rider needs food, executing the line:

rider1.isInNeedOfFood();

expected output:

false

actual output:

false

I execute toString for the rider to see all this information together:

rider1.toString();

expected output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(low) Needs food?:(false) Current speed of bike:(11 km/h)

actual output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(low) Needs food?:(false) Current speed of bike:(11 km/h)

Question 2 (c continued)

**Test 2 of 6**

I keep the “low” energy level in the instance of the composite object but I change current speed of rider’s racing bike to : 10

I set the rider’s bike’s current speed to 10:

bike1.setCurrentSpeed(10);

Then I get it to check:

bike1.getCurrentSpeed();

expected output:

10

Actual output:

10

Now I see if the rider needs food, executing the line:

rider1.isInNeedOfFood();

expected output:

false

actual output:

false

I execute toString for the rider to see all this information together:

rider1.toString();

expected output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(low) Needs food?:(false) Current speed of bike:(10 km/h)

actual output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(low) Needs food?:(false) Current speed of bike:(10 km/h)

Question 2 (c continued)

**Test 3 of 6**

Competing rider still has energy level: “low” this time I

Change current speed of rider’s racing bike to: 9

I set the rider’s bike’s current speed to 9:

bike1.setCurrentSpeed(9);

Then I get it to check:

bike1.getCurrentSpeed();

expected output:

9

Actual output:

9

Now I see if the rider needs food, executing the line:

rider1.isInNeedOfFood();

expected output:

true

actual output:

true

I execute toString for the rider to see all this information together:

rider1.toString();

In the expected output below I expect the Needs food to say true after it like this:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(low) Needs food?:(true) Current speed of bike:(9 km/h)

actual output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(low) Needs food?:(true) Current speed of bike:(9 km/h)

Question 2 (c continued)

In tests 4 to 6 I have a “medium” energy level in the instance of the component object:

**Test 4 of 6**

Create a competing rider with energy level: “medium” and current speed of rider’s racing bike: 11

I set the rider’s energy level, executing the line:

rider1.setEnergyLevel("medium");

Then I get the energy level:

rider1.getEnergyLevel();

expected output:

medium

actual output:

medium

Next I set the rider’s bike’s current speed to 11:

bike1.setCurrentSpeed(11);

Then I get it to check:

bike1.getCurrentSpeed();

expected output:

11

Actual output:

11

Now I see if the rider needs food, executing the line:

rider1.isInNeedOfFood();

expected output:

false

actual output:

false

I execute toString for the rider to see all this information together:

rider1.toString();

expected output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(medium) Needs food?:(false) Current speed of bike:(11 km/h)

actual output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(medium) Needs food?:(false) Current speed of bike:(11 km/h)

Question 2 (c continued)

**Test 5 of 6**

I keep the “medium” energy level in the instance of the composite object but I change current speed of rider’s racing bike to : 10

I set the rider’s bike’s current speed to 10:

bike1.setCurrentSpeed(10);

Then I get it to check:

bike1.getCurrentSpeed();

expected output:

10

Actual output:

10

Now I see if the rider needs food, executing the line:

rider1.isInNeedOfFood();

expected output:

false

actual output:

false

I execute toString for the rider to see all this information together:

rider1.toString();

expected output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(medium) Needs food?:(false) Current speed of bike:(10 km/h)

actual output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(medium) Needs food?:(false) Current speed of bike:(10 km/h)

Question 2 (c continued)

**Test 6 of 6**

Competing rider still has energy level: “medium” this time I change current speed of rider’s racing bike to: 9

I set the rider’s bike’s current speed to 9:

bike1.setCurrentSpeed(9);

Then I get it to check:

bike1.getCurrentSpeed();

expected output:

9

Actual output:

9

Now I see if the rider needs food, executing the line:

rider1.isInNeedOfFood();

expected output:

false

actual output:

false

I execute toString for the rider to see all this information together:

rider1.toString();

expected output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(medium) Needs food?:(false) Current speed of bike:(9 km/h)actual output:

Actual output:

Competing Rider's Name:(Lance Armstrong) Extra tests needed?:(true) Energy level:(medium) Needs food?:(false) Current speed of bike:(9 km/h)

This is the end of the tests.