

Name: _____

Practice Exercise 04a Inheritance and Polymorphism

Overview

In this lab we are going to create a class hierarchy and will investigate the way Java calls methods based on the hierarchy.

Part 1: Define a parent class

We are going to create a simple parent class called Ball about a "ball" object. You should have attributes that contain information about the current state of the ball (For example, a double variable containing the diameter of the ball in inches.)

You also need to define methods for changing the state. An example would be a move method that would change the location of the ball.

The methods should also print a message indicating that they have been called: "Running the Ball class move method." Indicate both the class name and the method name – this will be important when we start looking at polymorphism.

What attributes did you define: location, size, color, weight

What methods did you define: move, paint, resize, changeWeight

Part 2: Child class

Now we are going to create two child classes. The one class will be for a bowling ball, called "BowlingBall". The other class will be for a football (American football, which is an ovoid, not a soccer ball.) Call the second class "Football".

Have the child classes extend the Ball class. They should override at least one of the parent class methods. You must add additional methods and attributes as required.

Once again, your new methods must change the appropriate attributes and must identify themselves by printing out a message.

What new attributes did you define: owner, alley, team, inPlay

What new methods did you define: changeOwner, changeAlley, changeTeam, play

Part 3: Looking at Polymorphism

Now we will create another class, "BallTest", that will be used to demonstrate polymorphism. The main method of this class will perform the following operations:

1. Create a Ball reference variable.
2. Assign a new Football object to the Ball reference variable.
3. Call some of the Football class methods. Observe what happens – In the case of overridden methods, which one gets called – the Ball class method or the Football class method?
the Football class method
4. Assign a BowlingBall object to the Ball reference variable.
5. Call some of the BowlingBall class methods. Observe what happens – In the case of overridden methods, which one gets called – the Ball class method or the BowlingBall class method?
the BowlingBall class method
6. Try calling a method from Ball that was not overridden in the child class. What happens in this case?
these methods cannot be found by the computer
7. Assign a Ball object to the Ball reference variable. Call some of the methods. What happens?
All the methods are called successfully

NOTE: You cannot call a child's method from its parent

Instructor/TA: _____