Enumerated Types and UML diagramming

Return to: _____

Members Present: _____

Team Name: _____

Task 1: New Teams

1. Introduce yourselves in the group. Choose one person to be the team leader for today. Recall, this person is charged with keeping the groups on track. Also assign a recorder whose work will be turned in and returned to them.

2. This month on February 28, we have a midterm exam. We will also have another at least one PA due. In your group, discuss some goals that you personally have related to class performance, then decide on a team goal…what would you work to have your team do successfully this month?

3. Finally, choose a Team name.

Task 2: Review from lab yesterday

1. In lab yesterday, you used concurrent arrays to describe allowable grades and their various values. In your groups, describe how you solved some of the problems, and then share any difficulties that you may have had. If you have any remaining questions about this part of the lab, be sure to get some help from the instructor.
2. Answer the following about your solutions:
    a. What was the most difficult part of working with the concurrent arrays?
    b. From what you know about enumerated types, will this be aided by any tools that you have in the enumerated type?
    c. What additional "intelligence" do you want to add to your enum type for grades.

Task 3: UML diagrams

1. Thinking of objects, UML diagramming is one way of describing classes and their relationships to one another. UML stands for Universal Modeling Language and is used to describe processes as well as to describe systems that support those processes. We will explore a very small subset of all possible UML diagrams, those related to classes.
2. Look at the model of the Circle. This describes a single class in isolation. Using this model, answer the following questions:
    a. What is the name and data type of the single Circle instance variable?
    b. What is the visibility of this instance variable?
    c. What methods are provided in the class. What do they return?
    d. What is the visibility of each of the methods?
    e. Do any of these methods take in a parameter?
3. Create a UML diagram of your Rainfall class, choosing to document only a couple of the methods. Choose the constructor, the updateRainfallAmount and one other method. Include visibility, parameters and return type. Choose one model to turn in.

Task 4: Relationships

The second diagram shows two classes, Sphere and Circle. Circle is the same as we have seen before. The line describes the relationship between the two classes.

1. We call the relationship between Sphere and Circle aggregation (or the has-a relationship). Based on what you see in the diagram, when is one class an aggregate of another?
2. Thinking of recent programs or labs, describe other aggregate relationship that you have seen.
3. We can also use shorthand to describe relationships. By eliminating some of the detail (such as the attributes and methods, but still leaving the basic three part rectangle, you can show relationships without worrying about the details of that relationship. With that in mind, draw a UML model of your rainfall tracking system?
4. Looking at the third diagram which includes the class SpherePlay, we see a different kind of relationship. What do you think is different about the relationship between SpherePlay and Sphere and the relationship between Sphere and Circle?
5. Look at the diagram of the Card set of classes. What is different about the Rank and Suit classes from other classes in other diagrams.
6. What is the relationship between Rank, Suit, and Card?
7. Finally, draw a similar diagram for your EnumGrade application that you will be building tomorrow. What will your enum class look like, your EnumGrade method and the relationship between them?

Team Contract for February, 2008

Team Name: _____

Members present: _____

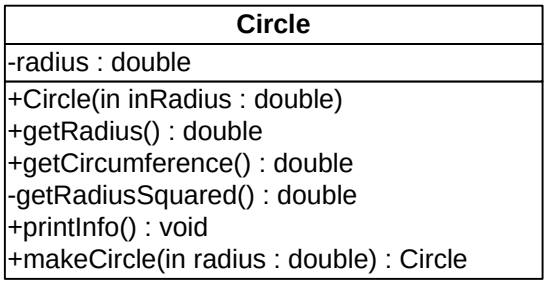
Members absent: _____

TEAM GOAL FOR FEBRUARY: _____

TEAM CONTRACT: We the undersigned will ….

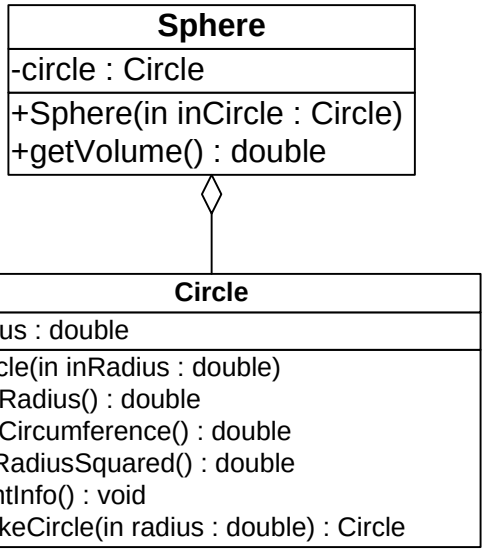
Signatures:

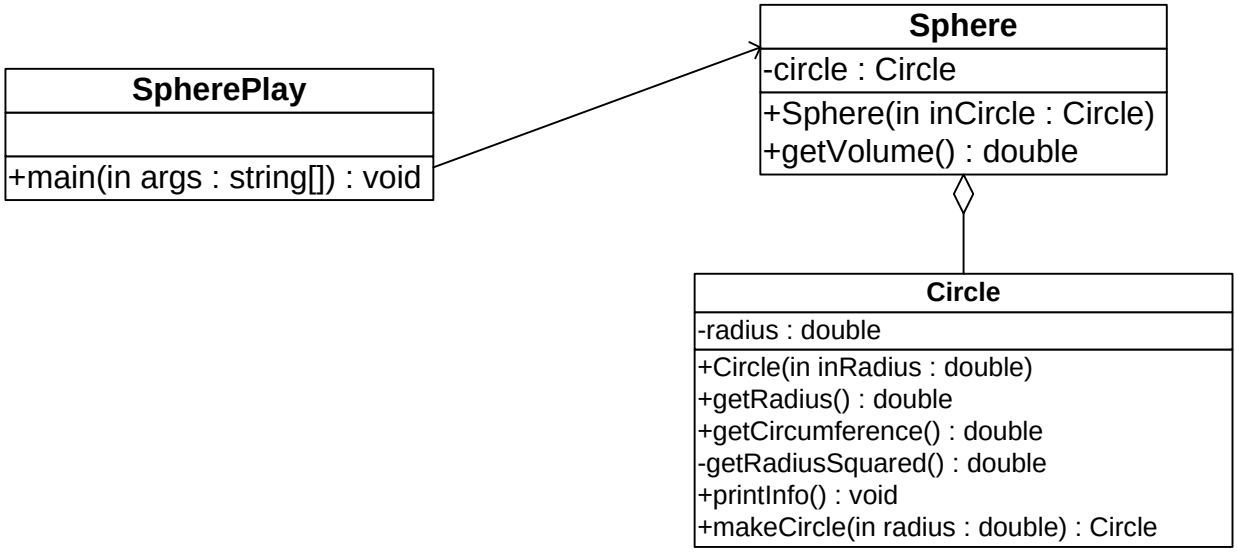
UML Diagrams

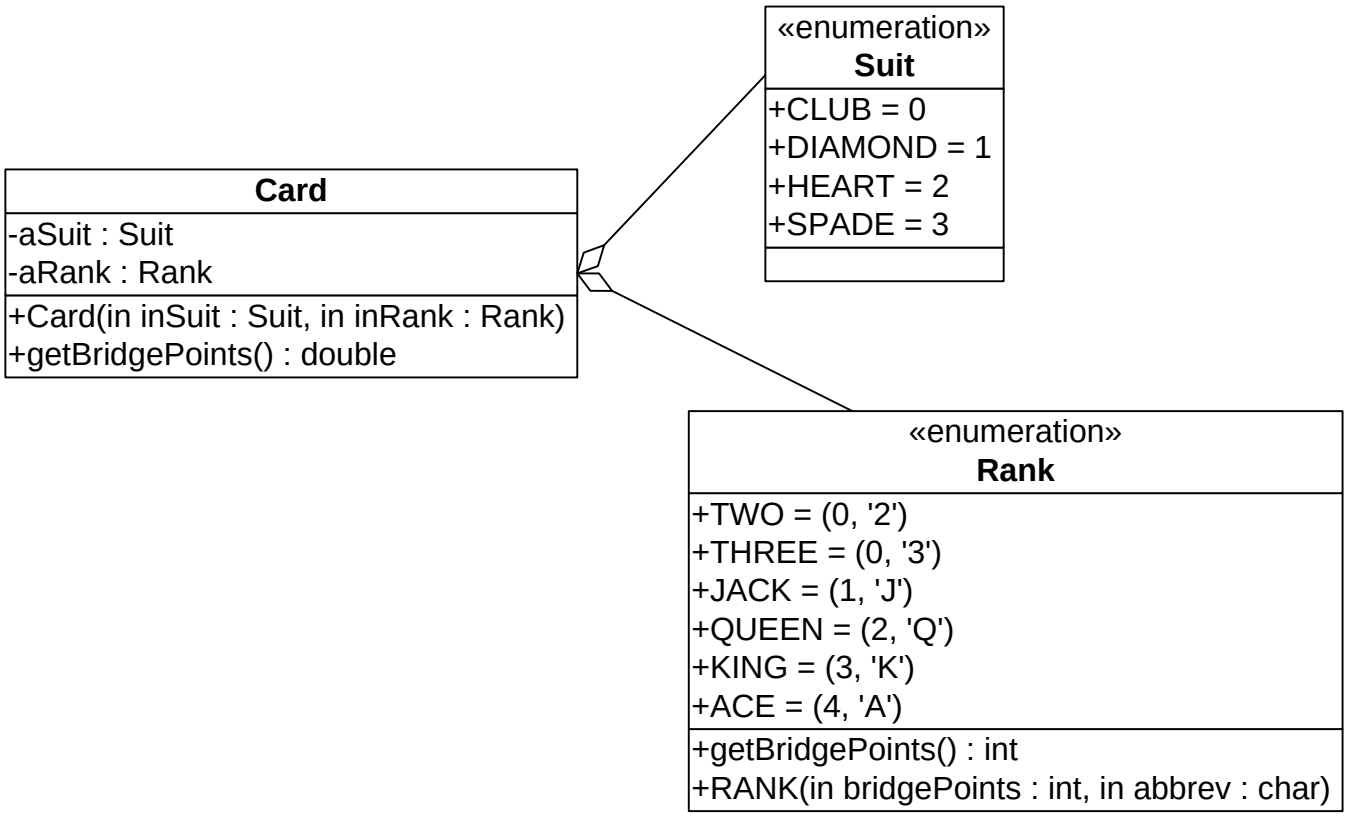


Class Name
Attributes, type, visibility

Methods, their parameters, their return, visibility






«enumeration»
Suit
+CLUB = 0
+DIAMOND = 1
+HEART = 2
+SPADE = 3
Card
-aSuit : Suit
-aRank : Rank
+Card(in inSuit : Suit, in inRank : Rank)
+getBridgePoints() : double
«enumeration»
Rank
+TWO = (0, '2')
+THREE = (0, '3')
+JACK = (1, 'J')
+QUEEN = (2, 'Q')
+KING = (3, 'K')
+ACE = (4, 'A')
+getBridgePoints() : int
+RANK(in bridgePoints : int, in abbrev : char)