Activity 7-2: UML Class Diagrams

Why?

Class models are the central static modeling tool in object-oriented analysis and design. UML class diagrams are the *de facto* notation for expressing class models.

Learning Objectives

- Absorb the syntax and semantics of basic UML class diagrams
- Make class models using basic UML class diagrams

Success Criteria

- Be able to distinguish correct from incorrect basic UML class diagrams
- Be able to explain what the constituents of basic UML class diagrams mean
- Be able to draw UML class diagrams to model entities and their attributes, operations, and associations

Resources

ISED section 7.2

Vocabulary

Name, pathname, class symbol, compartment, suppressed, attribute, multiplicity, order, operation, direction, association, and rolename

Plan

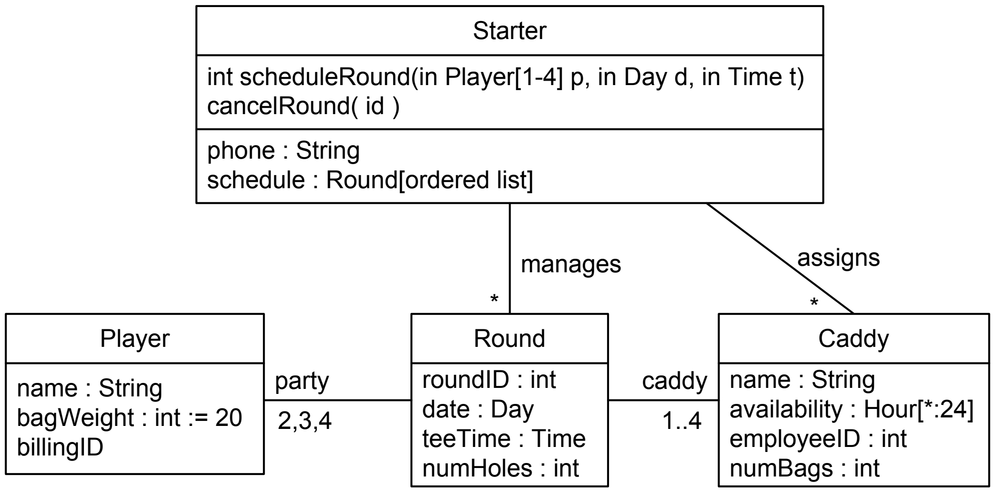
1. Review *ISED* section 7.2 individually.
2. Answer the Key Questions individually, and then evaluate the answers as a team.
3. Do the Exercise as a team, and check your answer with the instructor.
4. Do the Problems and Assessment as a team.
5. Turn in the Problems and Assessment as a team deliverable.

Key Questions

1. What are compartments and what can be inside them?
2. When is an attribute's initial value assigned?
3. What is parameter direction?
4. What is the difference between an association and a relation?
5. What is the difference between the multiplicity * and the multiplicity 0..*?

Exercise

1. The following UML class diagram models entities involved in scheduling a round of golf. Mark all the problems on the diagram.



Problems (Deliverable)

1. According to the diagram above, how many Players can form a party for a Round of golf?
2. What does the Starter's schedule consist of?
3. What is the default weight for a Player's bag?
4. What would be a good name for the two unlabeled associations in this diagram?
5. Do there have to be as many Caddies as Players associated with a Round?
6. Use the Java code skeleton below to do the following exercise: Draw a UML class diagram (using only the features discussed so far) to represent as much of the program fragment in this Java code as possible.

```
class GameClock {
   private RenjuGame game
   private int timeLeft;
   private Player player;
   public void start() { ... }
   public void stop() { ... }
   public void update( Observable masterClock ) { ... }
   public int getTimeLeft() { ... }
   }

class MasterClock {
   private Thread clockThread;
   public void start() { ... }
   public void run() { ... }
   }

class Player {
   private final Offer[] openOffers;
   private final Game[] activeGames;
   private int numMatchesWon;
   private int numMatchesLost;
   private int numMatchesTied;
   public void beginGame( RenjuGame game ) { ... }
   public void endGame( RenjuGame game ) { ... }
   public boolean isPlaying() { ... }
   public void winMatch() { ... }
   public void tieMatch() { ... }
   public void loseMatch() { ... }
   }
```

7. Make a UML class diagram modeling documents, libraries, and patrons that includes classes, attributes, operations, associations, and multiplicities.

Assessment (Deliverable)

1. Did this activity help you achieve the learning objectives?
2. How could the instructor improve this activity?