

Activity 11-2: Advanced UML Class Diagrams

Why?

This activity builds on the UML class diagram skills you have already developed and extends them to additional important features of this notation.

Learning Objectives

- Understand the entire syntax and semantics of UML class diagrams
- Make class models using advanced UML class diagram features

Success Criteria

- Be able to distinguish correct from incorrect UML class diagrams
- Be able to explain what the parts of UML class diagrams mean
- Be able to draw UML class diagrams that include inheritance, interfaces, interface realization relationships, abstract operations and classes, visibility markers, class attributes and operations, aggregation and composition, and association navigability markers

Resources

ISED section 11.2

Vocabulary

Inheritance, abstract operation and class, concrete operation and class, interface, realization, visibility, class and instance variables and operations, aggregation, composition, navigability

Plan

1. Review *ISED* section 11.2 individually.
2. Answer the Key Questions individually, and then evaluate the answers as a team.
3. Do the Exercises as a team, and check your answers 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. How is inheritance represented in UML class diagrams?
2. What is the difference between an abstract operation and a concrete operation?
3. What is an *interface* in UML?
4. What is the UML symbol for interface realization?
5. How is the visibility of a protected attribute represented in UML?

6. What is the difference between aggregation and composition in UML?
7. How is the ability to navigate from class *A* to class *B* represented in UML?

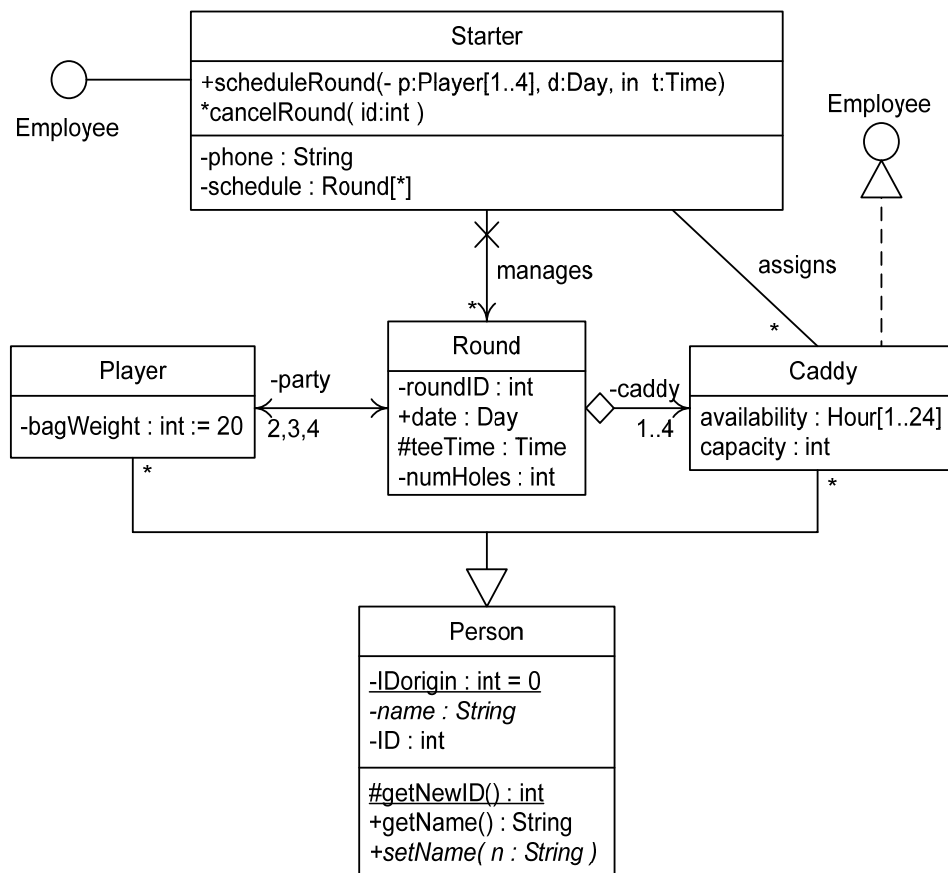
Exercises

True or False

- _____ A class with no abstract operations can still be abstract.
- _____ An association line with no arrows or X on its ends is navigable in both directions.
- _____ An association class requires that there be only a single link between instances of the associated classes.
- _____ An association qualifier requires that there be only one instance of the target class associated with each combination of the qualified class and values of each of its qualifiers.

Problems (Deliverable)

1. The following UML class diagram models entities involved in scheduling a round of golf. Redraw the diagram so that it is correct.



2. According to the diagram above, what attributes and operations must a Player have?
3. Where is the Round teeTime visible?
4. Can a Player instance access a Round instance?
5. If there are 100 instances of (subclasses of) Person, how many IDorigin variables are there? How many name variables?
6. Can Caddy just use the setName() operation it inherits from Person?
7. Draw a UML class diagram to represent as much of the following Java program fragment as possible.

```
class GameClock extends MasterClock {
    protected RenjuGame game;
    protected int timeLeft;
    private Player player;
    public int getTimeLeft() { ... }
}

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

final class Player {
    public static final int EXPERT = 0;
    public static final int NOVICE = 1;
    private final int rating;
    private final Collection activeGames;
    public void beginGame( RenjuGame game ) { ... }
    public void endGame( RenjuGame game ) { ... }
}

interface Subject {
    void update();
}
```

Assessment (Deliverable)

1. How could you improve your team's performance next time?
2. Did this activity help you achieve the learning objectives?
3. How could the instructor improve this activity?