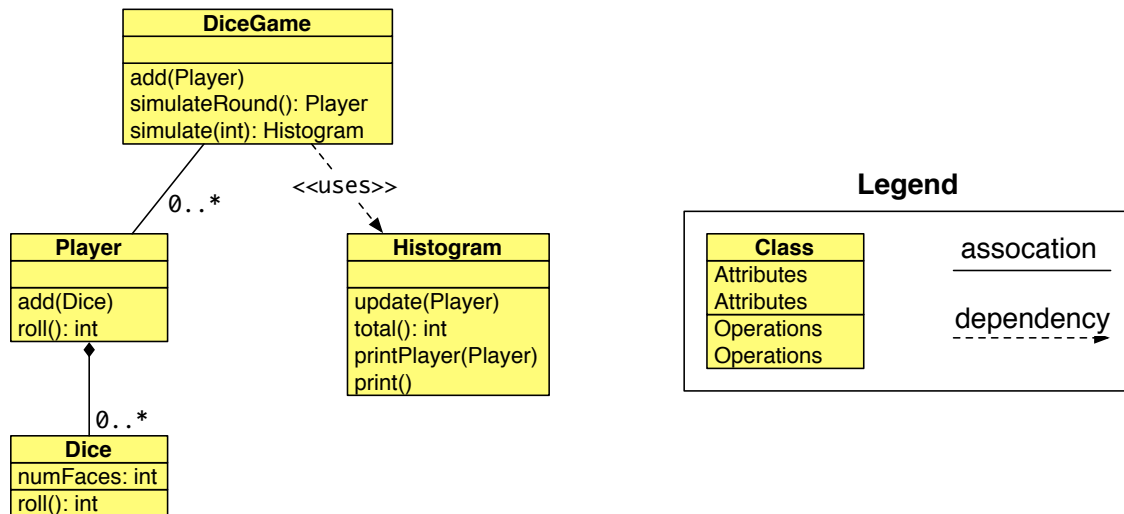


# Homework Assignment

## 5 Object-Oriented Dice Game

In this assignment you are going to deliver an object-oriented version of the Dice Game program. One possible approach is sketched here. You may deviate from this approach if you can motivate it. See the following class diagram.

**Class Diagram for Simple OO DiceGame Program**



### 5.1 Dice

Design a class `Dice` that represents a single dice (or die) with an arbitrary number of faces, numbered 1,2,ldots. Give it a reference to a `Random` object (not shown in the class diagram). A `roll()` should return the number on one of the faces, each face occurring with equal probability. Test it.

### 5.2 Player

Design a class `Player` that represents a player in the game. A player has one or more dice (implemented by an `ArrayList` of `Dice` objects). A `roll()` of a player rolls each of these dice and returns the total of the rolls.

### 5.3 DiceGame

A `DiceGame` object has a number of players (again an `ArrayList`). One round lets all players roll and determines the winner. If there is no winner, it returns `null`. A `HashMap` may be helpful here. Test this first. Then add a `Histogram` object (see below) that accumulates the results of many rounds.

## 5.4 Histogram

A `Histogram` records for each player the number of wins so far and the number of draws (when there is no winner). A `HashMap` can be used here (not shown in the class diagram). You can use the key `null` to record the number of draws. The method `update(Player)` updates the record of a `Player` with another win.

The method `total()` gives the number of rounds recorded so far. Although the class diagram shows an `int` as the return type, a `long` may be considered here as well.

## 5.5 Specification

Give pre/postconditions to all methods.