

COSC 241

Programming and Problem Solving

Lecture 2 (1/3/2018)

Review

Michael Albert

michael.albert@cs.otago.ac.nz



Keywords: **interface**



A simple dice game

- ▶ Each player starts with a (randomly rolled) set of three dice. The object is to wind up with all three dice the same (e.g., 444) or in sequence (e.g., 234).
- ▶ In a turn, a player rolls one more die and she may choose to substitute it for one of her current dice.
- ▶ If she does not take the die she rolled, then the next player can either take that number, or roll a die as above. Substituting the rolled die for another of equal value counts as taking it.
- ▶ The players alternate (in the two player game) or play in turn around the table (with more than two players), the winner being the first to achieve the objective.

The problem

Provide a Java program for a text based version of this game that allows:

- ▶ Human v human play
- ▶ Human v computer play
- ▶ Computer v computer play

Allow for computer players that use different strategies from one another. We'll just look at the two player version.

Elements of the program

- ▶ An interface `Player` specifying the methods that a player object must support.
- ▶ A class `Die` that represents a single die.
- ▶ Various classes that implement `Player` and represent different types of players (human, computers with strategy).
- ▶ A class `Manager` that manages the play of the game.
- ▶ The interface ensures that `Manager` does not need to know which sort of player is involved.

What is an interface?

- ▶ A collection of abstract methods (basically method signatures) and constants.
- ▶ But really, a contract.
- ▶ Classes which implement an interface promise to implement its methods.
- ▶ Therefore, different implementations are possible (a key feature in this example), but more importantly other classes can make use of them interchangeably, provided they only draw upon the methods specified by the interface.
- ▶ Unfortunately, there is no way that an interface can ensure that its methods are implemented correctly or appropriately.

Player

What does a player have to do?

- ▶ Provide his, her, or its name when asked.
- ▶ Given the state of the game, and a rolled value decide which die (if any) to replace.

Design principle: Keep interfaces minimal. They should specify only what is absolutely required

Manager

Basic logic:

- ▶ Initialise game and players.
- ▶ While the game is not over:
 - ▶ Offer the previous roll (if not taken) to the current player. If she does not take it (or it's not available) roll a die for her and offer that.
 - ▶ Update based on her response.
 - ▶ Check the winning condition.
- ▶ Congratulate the winner

To do

- ▶ Bomb proofing
- ▶ A way to generate names for computer players
- ▶ Better game management/choice (varying the number of players, or the rules or ...)
- ▶ A GUI?