CS 210
Lab 7: More Generics
55 Points

Submission

- This is an **individual work** lab.
- Submit all source code in a single zip file to the Lab 7 assignment folder on Kodiak.
- Be sure to **comment your code using Javadoc style comments**.

Goal: Utilize generic types and interfaces to represent game of chance scenarios.

This lab asks you to model several games of chance where one person plays the game hoping for a desired outcome. The three games of chance that we will implement include slot machines, lotto, and a simple dice game.

Slot Machine: A slot machine is a gambling machine operated by inserting coins into a slot and pulling a handle or pushing a button that activates a set of spinning symbols on wheels. A **win** occurs in a slot machine when all of the same symbol occur at the same time as shown in Figure 1 below. A stepper wheel is an essential part of the slot machine and they are the mechanism that makes the symbols rotate. Slot machines may have from 3 to 10 stepper wheels and typically have 21 symbols (some repeat) on each wheel. Each wheel also has some "blank" spots shown as the "BAR" in Figure 1 below.

Figure 1. Slot Machine Interface

Lotto: A lotto game is a form of gambling that involves the drawing of numbers at random for a prize. Lotto games may include drawing between three and six numbers. The maximum of the numbers drawn may vary per game, but is typically between 1 and 99. A **win** for a lotto game is when the numbers drawn match a pre-selected set of numbers.

Figure 2. Lotto Example

Die: A simple die game takes between two and nine die and rolls them. A **win** for the die game is if all of the dice have the same value.

The UML for the solution of the lab is shown in Figure 3 below.

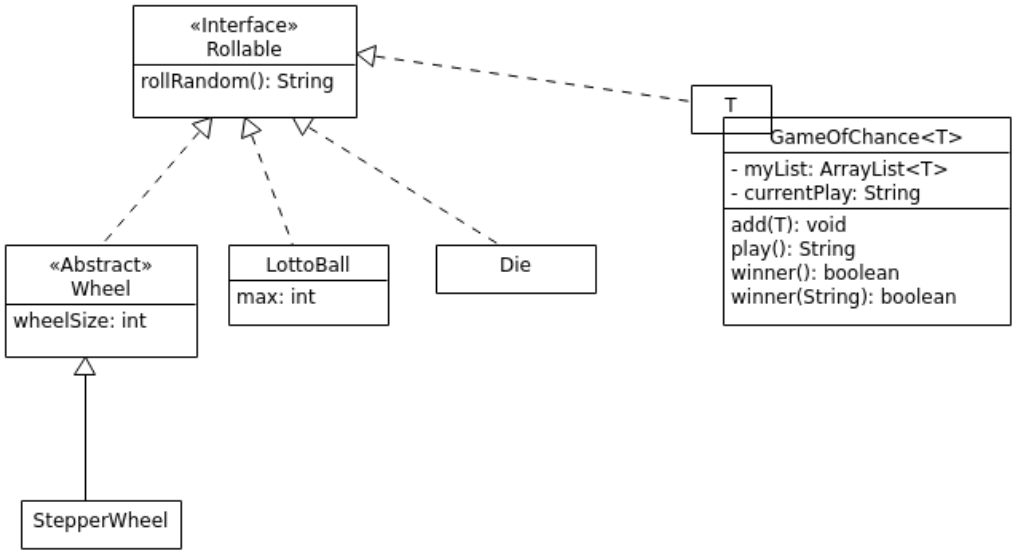

Figure 3. UML for Game of Chance Application

Notes:

- The `GameOfChance` class has two `winner` methods. One method simply determines if all results of `rollRandom()` are the same, and the other checks the results to an established value.
- The `rollRandom` method should return a random symbol in the `StepperWheel` class and a random number in the `LottoBall` and `Die` classes. (Hint: You may want to look up the `Random` class in Java documentation.)
- When creating a `StepperWheel`, you should pass the number of symbols into the constructor.
- You may hard code the values for the symbols into the `StepperWheel` class.
- You should restrict the number of side of a Die to six.
- If there are fewer than 2 entries in a game (e.g., dice or stepperwheels), the `GameOfChance` class should throw an exception and handle that exception within the method when determining the winner.
- You may hard code the test value to test your Lotto instance in `Lab7Driver`.

You should create a `Lab7Driver` class which creates three instances of `GameOfChance`, each instantiated with a different type (`StepperWheel`, `LottoBall`, `Die`). You should add contents to your instances and play each. After each play, you should print out whether the play is a winner or not. Two sample outputs are shown below:

blank blank blank
Winner!
86 39 11 58
No winner.
5 4
No winner.
=========
blank lemons blank
No winner.
9 72 20 53
No winner.
1 1
Winner!