Lab 11
Simulation Using Random

The following exercises are to be completed during lab class. If you do not have time to finish during lab, they must be completed before the beginning of the following lab session.

Set-Up

- Create a new project in your Eclipse workspace named: **Lab11**
- In the *src* folder, create a package named: **edu.ilstu**

The simulation you will be coding is rolling dice. Actual results approach theory only when the sample size is large. So we will need to repeat rolling the dice a large number of times (we will use 10,000). The theoretical probability of rolling doubles of a specific number is 1 out of 36 or approximately 278 out of 10,000 times that you roll the pair of dice. Since this is a simulation, the numbers will vary a little each time you run it.

Die
- spots:int - generator:Random
+ Die() + roll():void + getSpots():int + equals(Die other):boolean

Die - represents one die. The instance variable spots holds the number that results from "rolling" the die. An `equals` method comparing spots needs to be written to be used in the `DiceSimulator` to determine if the two dice rolled have the same number.

- Constructor – create the generator object here.
- `roll` - use the generator to get a number between 1 and 6 and store the result in the spots variable.
- `getSpots` – getter for the spots instance variable
- `equals` – two `Die` objects are considered to be equal if they have the same number of `spots`

DiceSimulator
+ runSimulation(int numberRolls):DiceAccumulator

DiceSimulator - Controls the running of the dice simulation.

- `runSimulation` – Does the simulation of throwing two die. Determines how many times they both have the same number and accumulates the totals.

Pseudocode :

```
      Dice Simulator runSimulation Algorithm
            CREATE DiceAccumulator object
            CREATE two Die objects

      REPEAT the number of times indicated by the numberRolls value
      passed into the method.

      Roll the first die
      Roll the second die

      IF spots on each die are the same
            IF spots = 1
                  CALL addSnakeEyes
            ELSE IF spots = 2
                  CALL addTwos
            ELSE IF spots = 3
                  CALL addThrees
            ELSE IF spots = 4
                  CALL addFours
            ELSE IF spots = 5
                  CALL addFives
            ELSE
                  CALL addSixes
            END ELSE IF
      END IF
END
```

DiceAccumulator
- snakeEyes:int - twos:int - threes:int - fours:int - fives:int - sixes:int
+ addSnakeEyes():void + addTwos():void + addThrees():void + addFours():void + addFives():void + addSixes():void + getSnakeEyes():int + getTwos():int + getThrees():int + getFours():int + getFives():int + getSixes():int

DiceAccumulator - This class has variables that are the accumulators (or counters) for when both dice roll the same number.

- all `add` methods – add one to the instance variable which is the accumulator
- all `get` methods – get the value of the instance variable

DiceSimulationDriver

Pseudocode:

```
Dice Simulation Driver Algorithm
      CREATE DiceSimulator objects

      SET numberSimulationRolls to 10000

      DiceAccumulator object =
         runSimulation(numberSimulationRolls)

      PRINT output report

END MAIN
```

Output

Prints the report at the end of the simulation showing how many doubles of each number are rolled. These statements can either be done in the driver or in a DiceOutput class with a printReport method.

> *Sample output – NOTE: the actual values for each number will not be exactly the same each time the program is run, but they will be similar.*

```
Number of rolls:     10000
Number snake eyes:     278
Number twos:           280
Number threes:         285
Number fours:          292
Number fives:          257
Number sixes:          261
```

To Be Submitted

The following files should be zipped together into a file called Lab11.zip and submitted to ReggieNet by the beginning of your next lab.

- DiceSimulationDriver.java
- Die.java
- DiceSimulator.java
- DiceAccumulator.java
- Optional: DiceOutput if used