**DoorGame User Manual**

A program emulating Montey hall’s game.

Cachary Tolentino

CSCI-3327

**Table of Contents**

**Software Description3**

Detailed Description3.1

System Requirements3.2

**Installation Guide4**

**Class Overview7**

**Software Description**

A program emulating Montey Hall’s game.

**Detailed Description**

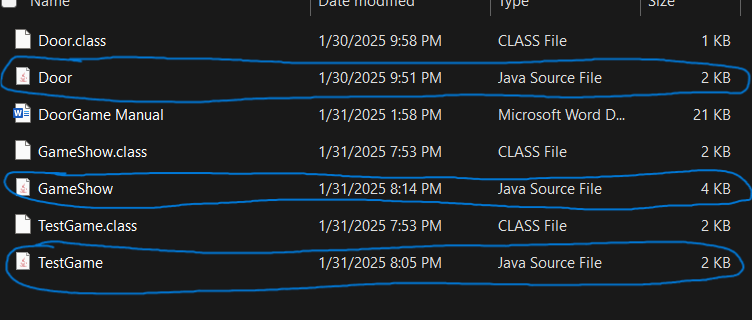
The DoorGame Software is a set of Classes including a Door Class, GameShow class, and TestGame (optional) classes. This set of classes allows for one to emulate Montey Hall’s game. More specifically, learn about the probabilities of a contestant’s odd of winning given switching and non-switching their initial decision.

**System Requirements**

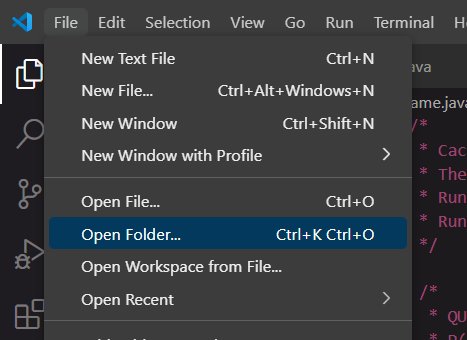
* A working device, primarily a desktop or laptop
* An IDE (ex: VSCode, Eclipse, etc…)
* Java JDK (Ver. 17 & up) & JRE (SE 17 & up)

**Installation Guide**

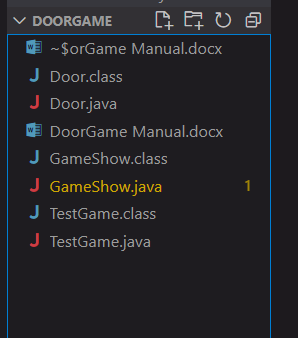
To begin using the DoorGame Software, first download “Door.java”, “GameShow.java”, and “TestGame.java” (optional).

****

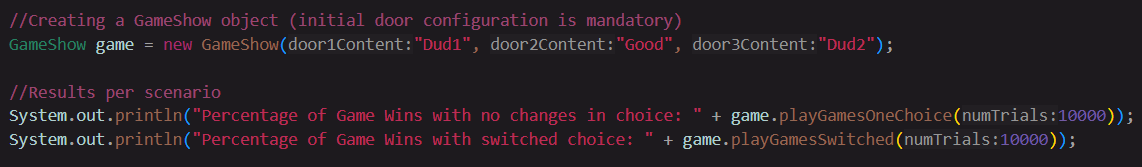
Once downloaded, you can either move the files to your project folder or directly open it in your desired IDE (we will be using VSCode in this tutorial). Open your IDE and open the folder containing the files.



Once opened, it should look similar to the following image.



From here you can simply start using the classes as you would for any Java classes. But if you wanted to quickly test or see the results using the code then open the “TestGame.java” file and run it. The output should look similar to the following image.



**A black background with white letters

Description automatically generated**

**Class Overview**

**Door Class**

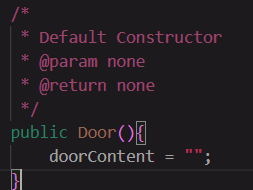
The Door Class consists of multiple functions and constructors. It also has a single global variable, doorContent. doorContent is a private string that would hold the content for the door (i.e Good Prize or Dud).



**Door Class Constructors**

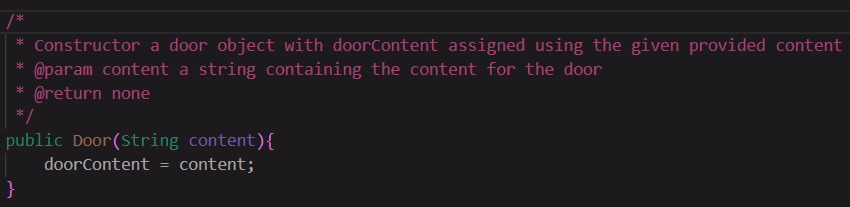
**Default Constructor**

The default constructor takes no parameters. It creates an empty door with no content value.

****

**Constructor with Parameters**

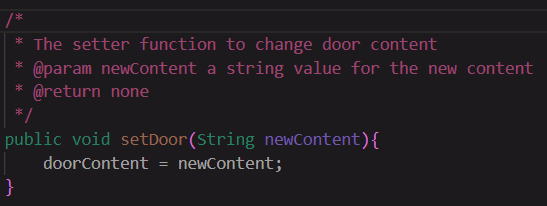
The door class constructor with parameters takes in a **String** value, content. This will create a door object with a filled value using content.

****

**Door Class Functions**

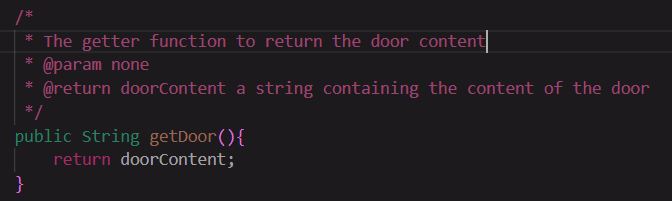
**setDoor Function**

The setDoor function does not return any value. But it takes in a **String** value, newContent, as its parameters. The function will set the new value of the current door object to newContent.

****

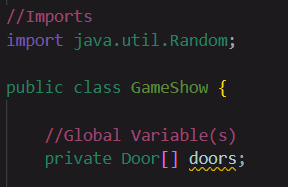
**getDoor Function**

The getDoor function returns a **String** value, doorContent. The function does not take any parameters. The value will return the current door’s content value.

****

**GameShow Class**

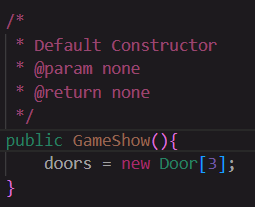
The GameShow Class imports the Random class. It also contains a single global variable, doors, which is an array of Door objects.

****

**GameShow Constructors**

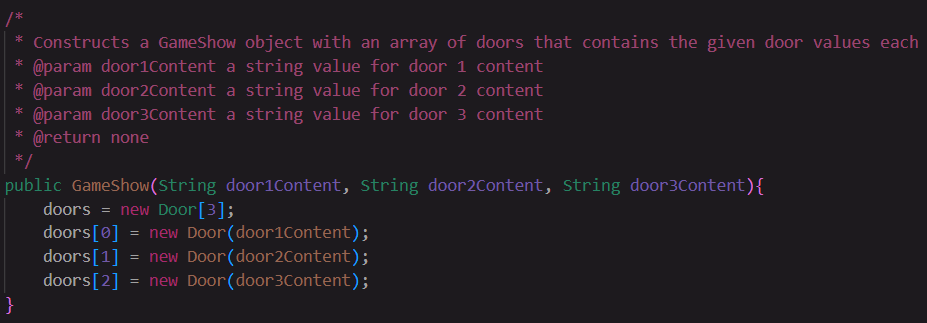
**Default Constructor**

The default constructor creates an empty GameShow object with not filled value for the door array.

****

**Constructor with Parameters**

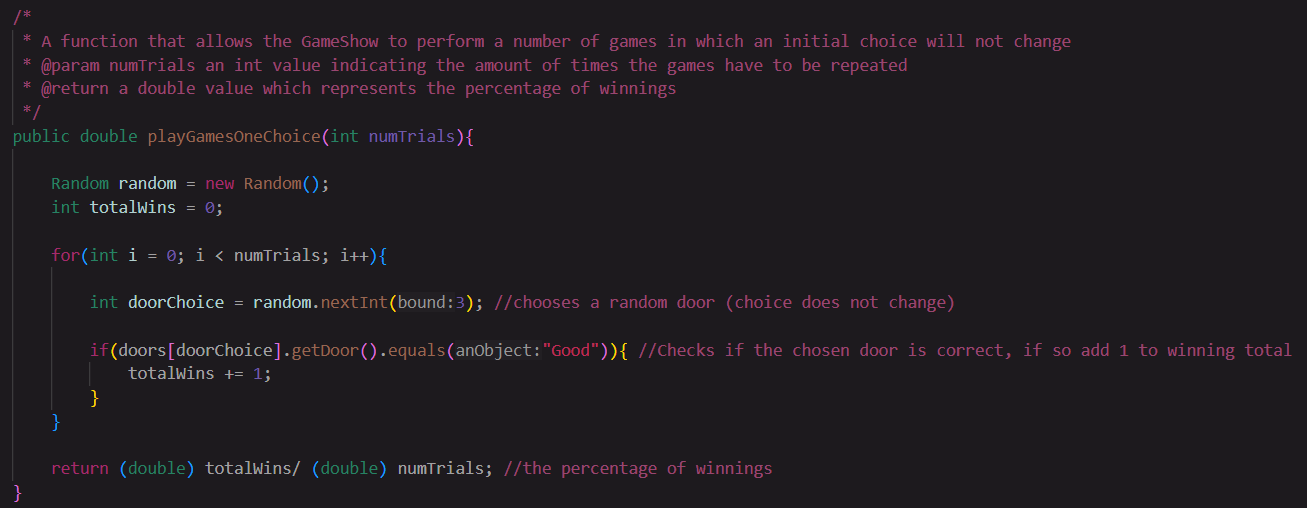
The GameShow constructor with parameters creates a GameShow object with a filled door array with three door objects. Each door object is created using the provided values for each door. It takes three values for its parameters: door1Content, door2Content, and door3Content, all of which are **String** values.

****

**GameShow Functions**

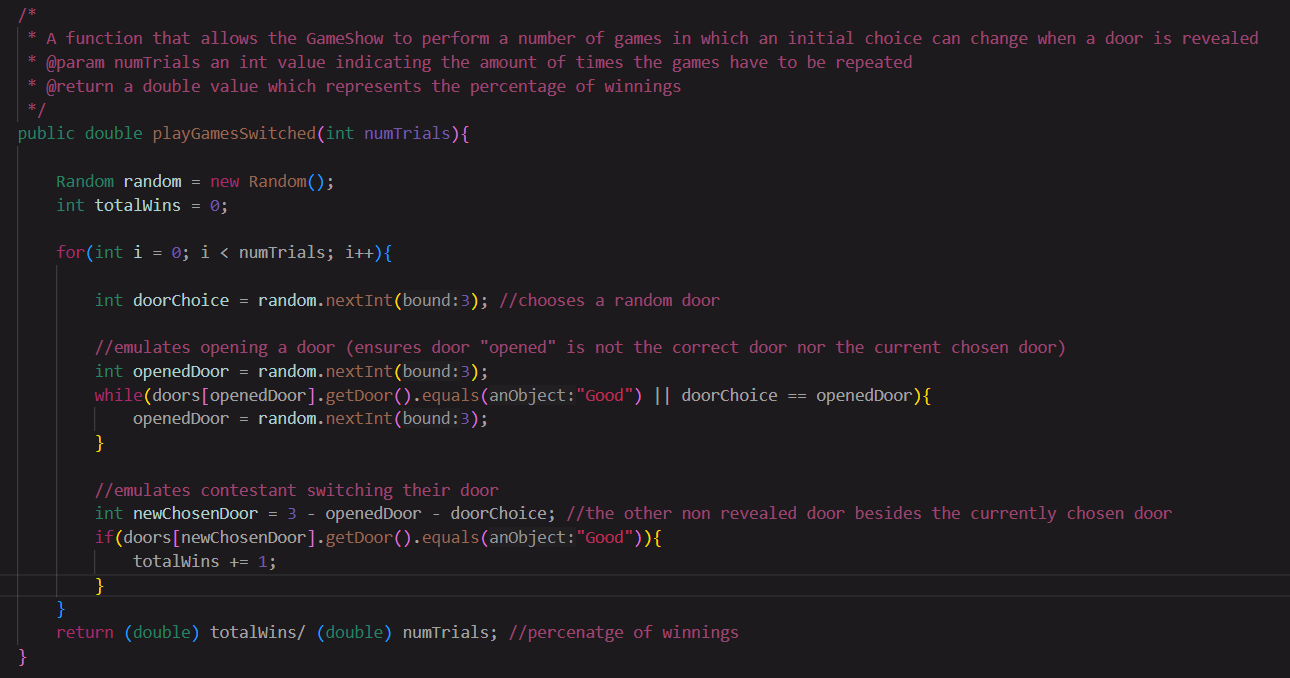
**playGamesOneChoice Function**

The playGamesOneChoice function will run an emulation of a game given the amount of times the user provides. In this game, a contestant is emulated to choose a single door and their choice would not be changed. The function takes in numTrials, an **int** value. This indicates the amount of games are played. The function returns a **double** which is the percentage of winnings through all games played.

****

**playGamesSwitch Function**

The playGamesSwitch function will run an emulation of a game given the amount of times the user provides. In this game, a contestant is emulated to choose a single door and their choice would switch after a random dud door is revealed. The function takes in numTrials, an **int** value. This indicates the amount of games are played. The function returns a **double** which is the percentage of winnings through all games played.

****