**BirthdayProgram User Manual**

A collection of classes for finding birthday probabilities.

Cachary Tolentino

CSCI-3327

**Table of Contents**

**Software Description3**

Detailed Description3.1

System Requirements3.2

**Installation Guide4**

**Class Overview7**

**Software Description**

A collection of classes for finding birthday probabilities.

**Detailed Description**

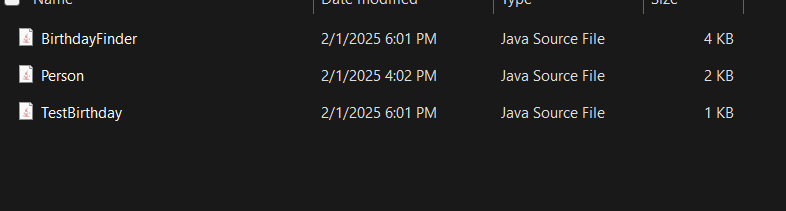
BirthdayProgram consists of three classes: Person, BirthdayFinder, and TestBirthday(optional). Person allows for the creation of Person objects that have a birthday. While BirthdayFinder emulates having a lists of person objects. Here you can find the function for finding the probability of a pari of person objects that have the same birthday. TestBirthday contains testing code for these functions.

**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 BirthdayProgram you must download “Person.java”, “BirthdayFinder.java”, and “TestBirthday.java” (optional).



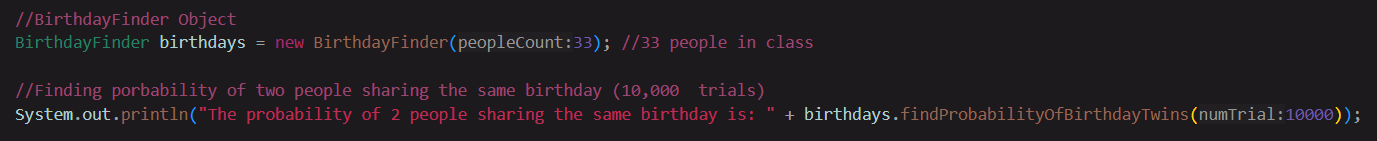
Once downloaded, you can place them in your project folder or open them directly through your IDE of choice (this will be shown through VSCode).

A screenshot of a computer

Description automatically generated **A screenshot of a computer program

Description automatically generated**

Once you’ve opened the folder with the files and your project, you may begin to use the classes freely. If you want to quickly see some tests, then open TestBirthday and run the file. It should output the following:





**Class Overview**

**Person Class**

The Peron class consists of a single global variable, birthday. It is an int value that would hold a person object’s birthday value. The class does not use any other imported packages.

**A computer code with different colored text

Description automatically generated with medium confidence**

**Default Constructor**

The default constructor creates an empty person object with a birthday value of 0000.

**A computer screen with text

Description automatically generated**

**Constructor with Parameters**

The constructor with Parameters creates a person object with a filled birthday value. The parameter takes a int value called newBirthday. This is used to assign the birthday value of the object.

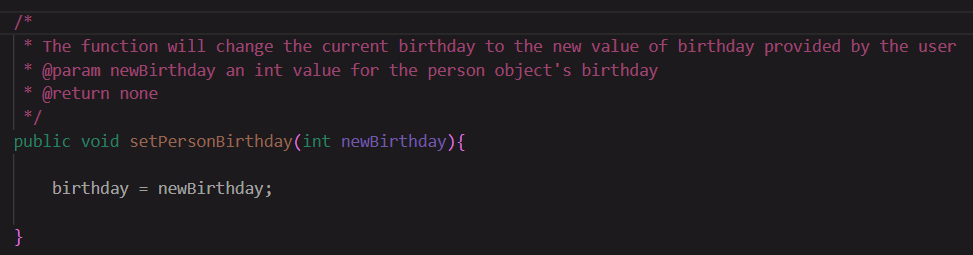
**A computer screen with text

Description automatically generated**

**Person Functions**

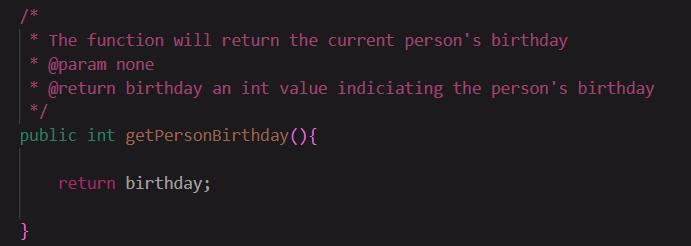
**setPersonBirthday Function**

This function allows the user to manually set the birthday value to their desire for their person object. Its parameter takes an int value called newBirthday. This is used to update the current birthday value.

****

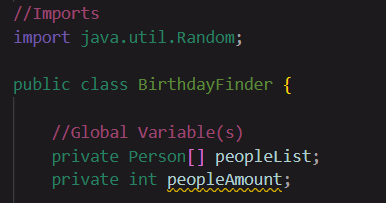
**getPersonBirthday Function**

This function allows the user to retrieve the current person object’s birthday value. A double value is returned containing The birthday value. This function requires no parameters.

****

**BirthdayFinder Class**

The BirthdayFinder class uses a single import, Random from the java.util package. The class also uses two global variables: peopleList and peopleAmount. The peopleList is an array of person objects. The peopleAmount is just an int value.

****

**Default Constructor**

The default constructor creates an empty BirthdayFinder object. It initializes the peopleList array with 0 size and peopleAmount is set to 0.

**A computer screen shot of a program

Description automatically generated**

**Constructor with Parameters**

The constructor with parameters takes an int value parameter, peopleCount. peopleCount is the number of person object needed to be created. The constructor then assigns peopleCount to peopleAmount. peopleList is also filled with the same amount as peopleCount. This array is then filled with person objects with random birthday values.

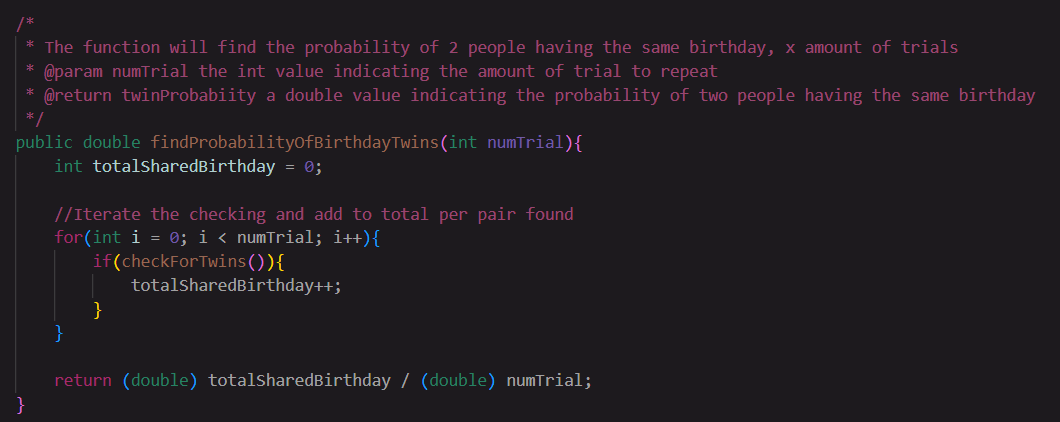
**A screenshot of a computer program

Description automatically generated**

**BirthdayFinder Functions**

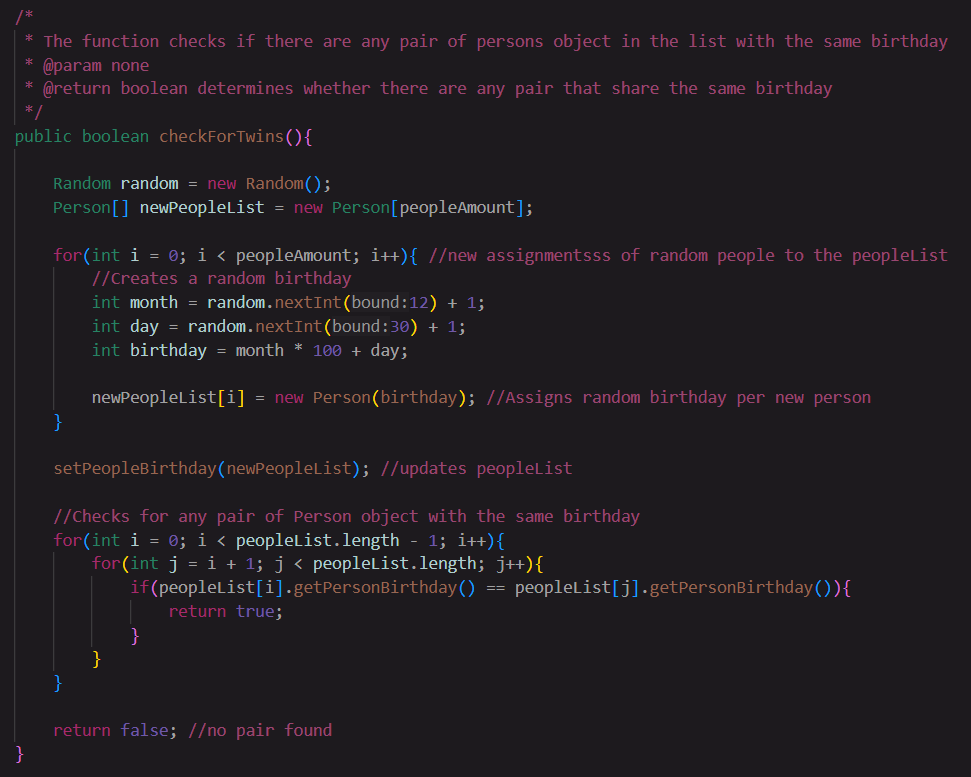
**findProbabilityOfBirthdayTwins Function**

This function calculates the percentage of how many pairs of person objects have the same birthday value. The function parameters takes in an int value, numTrial, which indicates how many repetitions the checking has to occur. This function returns a double value containing the probability of having a pair of person object with the same birthday value given an x amount of trials.

****

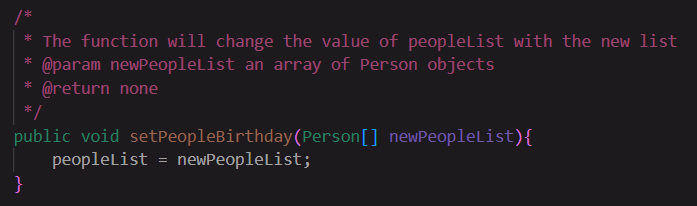
**checkForTwin Function**

This function first refills the peopleList with new persons object with random birthday values. It then checks for any pair of person objects that share the same birthday values. This function returns a Boolean deciding whether or not a pair was found.

****

**setPeopleBirthday Function**

This function assigns a new array of persons object to the current array. Its parameters takes an person array, newPeopleList.

****