CS 124 / Spring 2020
Project 1
Egbert Porcupine

Data Set

The data are a subset of the NASA Exoplanet Archive's list of exoplanets as of 27 September 2018. The data are originally from NASA's Exoplanet Science Institute (at CalTech) but were discovered on Kaggle.[1] The original data included 50 fields, I have reduced this to the following nine fields by deleting columns from the CSV file:

- `rowid` - a unique integer serial number indicating row in the database
- `kepid` - a unique integer; the target identification number in the Kepler Input Catalog (KIC)
- `kepoi_name` - a string; a number prefixed by the letter "K"; the integer part of the number designates the target star, and the two-digit decimal part designates the unique transiting object of interest; example "K00747.01"
- `kepler_name` - a string in the form "Kepler-N n" where n designates the planet in question; example "Kepler-661 b"
- `koi_disposition` - a string; one of "CANDIDATE", "CONFIRMED", "FALSE POSITIVE" or "NOT DISPOSITIONED" indicating the status of the object of interest
- `koi_score` - a float between 0 and 1 indicating the confidence in the KOI disposition
- `ra` - a float representing KIC right ascension (degrees)
- `dec` - a float representing KIC declination (degrees)
- `koi_kepmag` - a float representing the estimated magnitude in the Kepler band

Entries are ordered by row ID (`rowid`), ascending.

Peculiarities: `kepler_name`, `koi_score`, or `koi_kepmag` may be empty, but even if all are empty, there are still more than five fields with values, and the desired data types are represented. Only one record is missing a value for `koi_kepmag`.

I chose this data set because the Earth is a very tiny place and it's good to get to know the neighbors!

Demonstration of correctness

How can you demonstrate in your code that your program successfully reads and stores 1000+ objects from your data file?

I added a counter to verify that no records were skipped due to inconsistencies in the data, and checked the vector after reading from the file to confirm that it reports the correct number of elements in the vector. *I.e.*, in the console you will see

```
9564 records read from file
The last row ID read was 1
The last row ID read was 9564
...
```

[1] Original source: https://exoplanetarchive.ipac.caltech.edu/
Kaggle: https://www.kaggle.com/nasa/kepler-exoplanet-search-results.
For more on the Kepler Input Catalog
see: http://tdc-www.harvard.edu/software/catalogs/kic.html

Also, if there is a mismatch at any point between `rowid` and the counter, the function will exit with a value of false, and the program will indicate that an error has occurred. There is also a test `verifyAllFieldsSet()` that checks to make sure all fields are properly assigned. This is called from `main()`.

How can you demonstrate in your code that all your class methods function correctly?

I have written tests for object initialization, setters and getters, and auxiliary functions. These are in a function called `keplerTest()` which is called from `main()`. If any tests were to fail, the program would terminate with an "Assertion failed" error and non-zero exit code.

Calculation Function

My calculation function calculates total records by disposition. Here is the result:

```
Disposition statistics
--------------------------
Candidates           2248
Confirmed            2293
False positives      5023
Not dispositioned       0
Other                   0
--------------------------
Total                9564
==========================
```

How can you demonstrate in your code that your calculation function works correctly?

I have verified the result of my calculations by reading my data into spreadsheet software and performing independent calculation. All values are in agreement. Also, I calculate the sum of all the values I add up by disposition and this total agrees with the total number of records.