Initial Input

The Event and Stat classes are responsible for storing the data in the ‘events.txt’ and ‘stats.txt’ text files in memory. They have fields for every type of data defined in the assignment specifications. String splitting is used extensively here to obtain the values needed, which are then stored in ArrayLists. A default max of 100 is assigned when it is not specified.

Checks for consistency include the sizes (e.g. the number of events and the number of stats), and the event names in both .txt files. If inconsistencies are found, the program outputs an error message and does not execute the further steps.

Activity Simulation Engine and Logs

The events are generated by creating a random object and asking for Gaussian values (random numbers following a normal distribution with a mean 0 and standard deviation of 1) and scaling them to have whatever mean and deviation is specified in Stats.txt for that event.

The actual logs are generated on a per day basis. The event entries are generated sequentially but are given a random timestamp. The events that are of type D or C get a random magnitude until the sum of the magnitudes is equal to the previously generated total. These entries are stored in an ArrayList that then gets sorted with respect to time, thereby randomizing the order of events. This ensures that the values are random.

The data is written to a ‘BaseLine.txt’ text file as it is generated, the format of the data within is simple to maximise ease of processing, as well as readability:

- All values are separated by “:”
- The first value is the names of the event
- The second value is the mean
- The third value is the SD

In order to get the data back we split with “: “ then strip whitespace and convert to numbers. The whitespace is to help with human readability.

Analysis Engine

The analysis engine uses the total values that were created in the activity simulation engine and stored in the BaseLine.txt. Since the document only contains the total values, and the log values are stored in a separate log file it is virtually impossible to pick up log reading anomalies at this stage, and it is improbable that they are apparent as the log values are generated from the total values, and not the other way around.

It reads through the file and calculates the mean and standard deviation on each event. This information is then stored as a 'Stats' object (since the date to be calculated corresponds to a 'Stats' object) and saved in an ArrayList. The engine calculates the anomaly counter by comparing the generated simulation data with the baseline event stats provided:

$$Anomaly\ counter = \frac{Absolute(base\ event\ mean - simulated\ event\ value\ for\ the\ day)}{base\ event\ SD} \times event\ weight$$

If SD = 0,

$$Anomaly\ counter = Absolute(base\ event\ mean - simulated\ event\ value\ for\ the\ day) \times event\ weight$$

Alert Engine

The alert engine uses data from Events.txt and data from the analysis engine. Since the data from the 'Events.txt' text file already contains the total weight, the threshold is obtained by multiplying it by 2 to get the maximum threshold.

It then iterates through the data from analysis to compare the anomaly counter for each day with the threshold and outputs the day for which anomalies are detected (current threshold >= maximum threshold).