Project Description:

Collect data from all JSON files in a folder, grab the data you have selected from the settings tab, sort and output the information into a csv file for excel.

Resources used:

- Eclipse IDE (for Java)
- Jackson Library (for xml):
  - Jar files: https://mvnrepository.com/artifact/com.fasterxml.jackson.dataformat/jackson-dataformat-xml/2.9.7
  - Javadocs: https://www.javadoc.io/doc/com.fasterxml.jackson.core/jackson-databind/2.9.7/index.html
- Simple JSON:
- Swing (for GUI)

Action Items:

- Phase one: Initialization
  - Customize GUI to allow for all the following via main menu settings.
  - Settings, simple exclusion filters (Name, duration, and location. Any information that can easily be accessed before gathering all the tables.)
  - ~~Xml settings import and export into/from a simple class.~~
  - JSON Read files in selected forlder, using simple exclusion filter.
  - Export csv file with the JSON information. Just put relevent settings up top and then all of the found JSON info in columns below.
- Phase Two: Core
  - GUI to allow for following features. Going into the settings tab for some of them.
  - Create display filters. Could use lists to show what output sections (object) should be displayed. (e.g. “Healing” would exclude damage.)
    - Get the base in for boons, but only allow “no boons” or “all boons”? This should be a list of checkable boons, with “(un)select all” and a selection option for different boon settings (active phase, all, etc.)
    - Get the base for different features, such as consumable in to a selectable table. Also a list of checkable options. No need to get far or find all values.
  - Export csv with spaces between “Areas” (general, damage, heals, boons, conditions, etc.)
- Phase Three: Features
  - GUI to allow for the following features.
  - Further all filters:
    - Boon table needs to be able to select specific boons.
    - Finish the selection for different features, such as consumables.
    - Add in selection for class specific stats. Everything avaiable in Arc DPS’s boon table should be available for selection.
  - Add further exclusion filtering (weapon selecitons, Profession, percent alive. Etc.)
  - Selecting a Directory for JSON reading should count the number of files available for reading?
  - Add a loading bar while reading files.
  - Persist settings on close: Save and load previously used settings when closing and loading the application.
- Phase Four: Expand
  - Expand options for Strikes, raids, and other environments. I do not currently know all of the changes that would be made, but there are many examples to read from.
  - Expand the output options as well.
- Phase Five: Finalize
  - Time and optimize running.
  - Simplify for building.
  - Create readme highlighting utility and use.
- Phase Six: Build and deploy.

Sections:

- Menu:
    - Main:
        - Note: stats will be collected from Active Phase Duration, discounting dead and dc time.
        - Output/input file
        - Character/account name.
        - Order by:
            - Duration
            - Location
            - Profession
            - Ouput (by damage, or healing if healing statistics are checked in settings)
            - Kills
        - Inclusions: (?)
            - Could go through all of your log files and only parse out from logs matching this filter? That may result in searching through to many logs tough.
        - Exclusion:
            - Min log duration
            - Min number of players
            - Min % time alive
            - k/d ratio (?)
            - Profession (Numbered as 0, 1, 2, 3 according to expansion)
            - Min tag distance (recommended 800)
            -
        -
    - Settings:
        - Boon selection options:
            - Uptime, goup generation, squad generation options.
            - All boons (DEFAULT)
            - Support boons
            - A scrollable list with selectable boons
        - Other selection options:
            - A scrollable list with with selectable options that aren't boons, but effects. Can have classes and then the misc able to be selected individually.
        - Desired stat selection options:
            - All stats (DEFAULT)
            - Damage statistics
            - Advanced damage (conditions)
            - Healing statistics

        - Enable enhancements/personal buffs display (sigils, nourishments, enhancements/ conumables)
        - Enable

The ouptut will be formatted for excel csv. Basically, a comma between each column. To move down a row use a new line, or “\n”. More details found at https://www.spreadsheetsmadeeasy.com/understanding-csv-files-in-excel/

Basic details and room for comments will be left at the top of the spreadsheet, then the format for a table will be inserted, the the headers stating the information below it. It may end up being a wide table, or long if many options or logs are parsed. To help this, we could use a spacing between different sections on the x axis (still have the same information in the row), between general, dps, healing, and boons. This will make the format more readable.

Settings chosen by the user may be saved or loaded via xml file. By using the Jackson library, we are able to easily read from or fill a class object that stores the settings.

Information for each option from the JSON format. The Report file will be skiped to save clutter, but all accesses start with the JSON variable:

Player is found in JSON.players with the according name.

- Gneral (for every log and any setting)
    - Location (fightName)
    - Duration (duration)
    - % time alive (…)
    - Party deaths(for player of players (+= player.defences[0].deadCount;
    - Party downs (for player of players (+= player.defences[0].downCount;
    - Enemy deaths (for player of players(for stat of player.statsTargets (+= stat[0].killed)))
    - Report link (hyperlink, don’t clutter)
    - .
    - Times Weapon Swapped
    - Tag distance
    - Times died
    - Times downed
    - Number of dodges
    - Number of evades
    - Number of blocked attacks
    - Number of boon strips
    - Resurrects (?)
- Damage
    - Damage (for stat of player.dpsTargets (damage += stat[0].dps))
    - Dps (for stat of player.dpsTargets (dps += stat[0].dps))
    - Enemy downed (for stat of player.statsTargets (stat[0].downed))
    - Enemy killed (for stat of player.statsTargets (stat[0].killed))

    -
- Healing
    - Cleanses on Squad mates
    - Self Cleanses
    - Outgoing healing All
    - Downed Healing
- General boon selection
    - …
- …