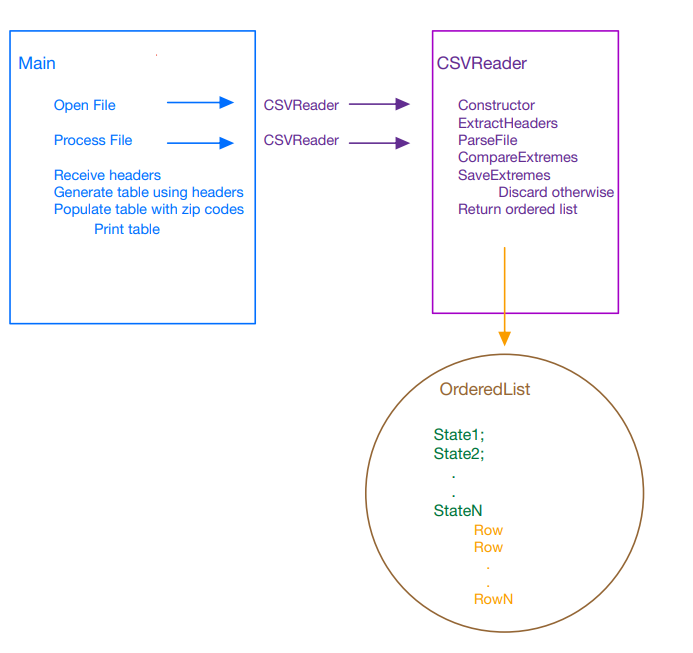
Zip Code Analyzer Design Document

**Overview**

The goal of this project is to read location data of zip US zip codes from a csv and calculate the north, south, east, and westmost locations in the file for each state. These four zip codes will be displayed in a table with their corresponding state in alphabetical order. Each row in the csv has the following header information:

"Zip" (zip code), "Place" (name of location), "State", "County", "Latitude", Longitude"

The project will consist of two main Classes:

* ReadCSV
  + ReadCSV is a buffer class. Its purpose contains methods to open, close, and read from the csv file.
  + The class does not write to the csv.
  + The buffer reads only one row at a time, determines if it is an extreme, then saves the row or discards it if it is not.
* main
  + Calls the buffer class to process a file.
  + Once the file is processed, the table is created.
  + Outputs the table

**Class Design**

ReadCSV

INVOKED: -The class is invoked by main where its constructor takes as input a containing the name or

location of the file to open.

OUTPUTS: -Bool for determining if a csv is opened

-A sorted list with the rows containing the extremes for each state.

-Header information extracted from the file.

Class Parameters:

The class will contain a "Row" struct to contain the csv rows with extremes.

The class will contain a "State" struct to hold the four extremes for each state.

The class will contain a sorted list to organize the State structs.

-The class will contain a method to compare Rows location data

The class will contain methods to open and close a file.

The class will contain a method to parse the header infuriation from the csv.

The class will contain a method to read one row from the csv at a time.

-That one row will be processed by another function to gather the location data

-The location data will be compared to existing data to determine an extreme

-Extreme values for each state will be saved, non-extremes are discarded.

The class will return the ordered list to an invoking class.

The class will rerun the header information from the invoking class.

ASSUMPTIONS

The buffer class assumes the data types of the headers and their corresponding data. Zip codes are assumed to be strings, latitude and longitude to be floats, and the remaining data to be strings.

The buffer class assumes the latitude is between -90 to 90 inclusive, and longitude between -180 to 180 inclusive.

Main

INVOKED: Is not invoked by any other class.

OUTPUT: Produces a table of states ordered alphabetically.

Class Parameters:

The class will specify a file to open.

The class with call the buffer class to open the file.

The class will call the buffer class to process the file into an ordered list.

The class will take the ordered list and extract the state.

-For each state the extremes will be extracted.

-The class will generate a table with headers for the state, eastmost, westmost, northmost, and southmost, zip code in that order.

-Each state will occupy one row in the table.

The table will be populated with the corresponding data from the ordered list.

The table will be output.

**Row Struct Reference**

Represents a row of data in the CSV file. This struct stores information for a single row of data in the CSV file, including the ZIP code, name, state, county, latitude, and longitude.

**Member Data Documentation**

**Headers**

Stores the column headers from the CSV file

**StateMaximums**

Map that stores state ID, as well as the maximum locations

**ZipCSV**

Represents the input CSV file stream used to open and read the CSV file