1. Geocoder.geocode
   1. Ceates an datafile instance (from read input)
2. Readinput.Datafile
   1. Checks if a file was passed in if not gets one via UI
   2. Controller()
      1. Checks file type and sends file to appropriate data type class (csv, xlsx, xls)
         1. Datatype class reads in data and separates the headers
   3. tabs\_with\_intersection\_min
      1. creates a list of tabs that have headers with the intersection pattern and stores in intersection attribute
   4. tabs\_with\_stretch\_min
      1. creates a list of tabs that have headers with the intersection pattern and stores in stretch attribute
   5. tabs\_with\_addresses\_min
      1. creates a list of tabs that have headers with the intersection pattern and stores in address attribute
   6. parse address
      1. checks for address column and then splits into number and street and calls tabs\_with\_addresses\_min
   7. tabs\_with\_nodes\_min
      1. checks for nodeid in the columns – stores in nodes attribute
   8. User selects mapping type
      1. ‘1’ – calls: intersection()
      2. ‘2’ – calls: address()
      3. ‘3’ – calls: block()
      4. ‘4’ – calls: nodes()
3. Geocoder.intersection
   1. Prints length of tabs with intersection pattern
   2. Adds node, x, y, error columns to intersection tab headers
   3. Intersection()
      1. Creates the intersection instance
   4. get\_nodes
      1. Checks that you have two street names
      2. intersection\_request
         1. Calls boro\_format
            1. Converts passed in borough to upper case and stores full name borough in borough attribute
         2. Sends request to api to geocode location
         3. Calls direction\_binary
            1. Attempts without direction and if unsuccessful attempts ‘N’ then ‘E’ then gives up
            2. Resends request to API
         4. Stores result of of geocode in intersection.json
      3. Checks that geocode results contain coordinates or if they have an error.
      4. Populates storage with either result or error message attributes
   5. Updates the data file with the attributes from geocode
   6. Shapify.Points() - Creates shapefile of intersections
      1. Cleans up shapefile name
      2. Delete previous version of shapefile if exists
      3. Creates shapefile
      4. Cleans the shapefile fields
         1. Tries to guess what the datatype of each column is
         2. Cleans up naming conventions
         3. Add fields to shapefile
      5. Adds data to newly added fields