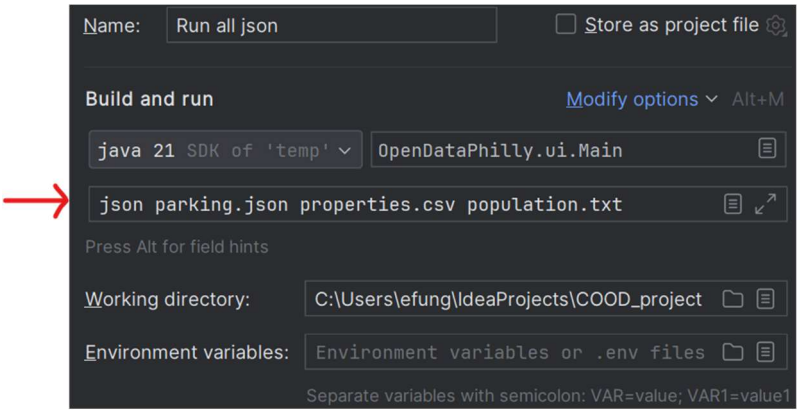
Documentation

To start the program:

The program expects 4 arguments when started, in the format:

- <csv|json> <parking file> <properties file> <population file>

An example of the 4 arguments (in the IntelliJ interface):



After starting, the program shows the Main Menu, which has numbered options:

```
Main Menu
1. Total population for all ZIP Codes
2. Fines per capita for each ZIP Code
3. Average residential market value for a ZIP Code
4. Average residential total livable area for a ZIP Code
5. Residential market value per capita for a ZIP Code
0. Quit
Enter selection:
```

After “Enter selection:”, the user will type in a single number between 0 and 5, followed by “Enter.” Then, the corresponding option will be initiated:

Option 1: Total population for all ZIP Codes

- Input: 1
- Output: <total population>

```
Enter selection: 1
1526206
```

Option 2: Fines per capita for each ZIP Code

- Input: 2
- Output: <ZIP code> <fines per capita>

```
Enter selection: 2
19102 6.9348
19103 4.9118
19104 1.2436
19106 4.1227
19107 6.2616
19111 0.0401
19114 0.0022
19118 0.3169
19119 0.0609
19120 0.1605
19121 0.2898
19122 0.1354
19123 1.9565
19124 0.1740
```

Option 3: Average residential market value for a ZIP Code

- Input: 3
- Prompt: Enter ZIP Code:
- Input: <ZIP Code>
- Output <average residential market value >

```
Enter selection: 3
Enter ZIP Code: 19153
516883.0
```

Option 4: Average residential total liveable area for a ZIP Code

- Input: 4
- Prompt: Enter ZIP Code:
- Input: <ZIP Code>
- Output <average residential total liveable area >

```
Enter selection: 4
Enter ZIP Code: 19153
3882
```

Option 5: Residential market value per capita for a ZIP Code

- Input: 5
- Prompt: Enter ZIP Code:
- Input: <ZIP Code>
- Output <residential market value per capita >

```
Enter selection: 5
Enter ZIP Code: 19153
176960.0
```

After executing the particular option, the main menu screen will appear again. The user can enter their selection again, repeating the above process.

```
Enter selection: 1
1526206

Main Menu
1. Total population for all ZIP Codes
2. Fines per capita for each ZIP Code
3. Average residential market value for a ZIP Code
4. Average residential total livable area for a ZIP Code
5. Residential market value per capita for a ZIP Code
0. Quit
Enter selection:
```

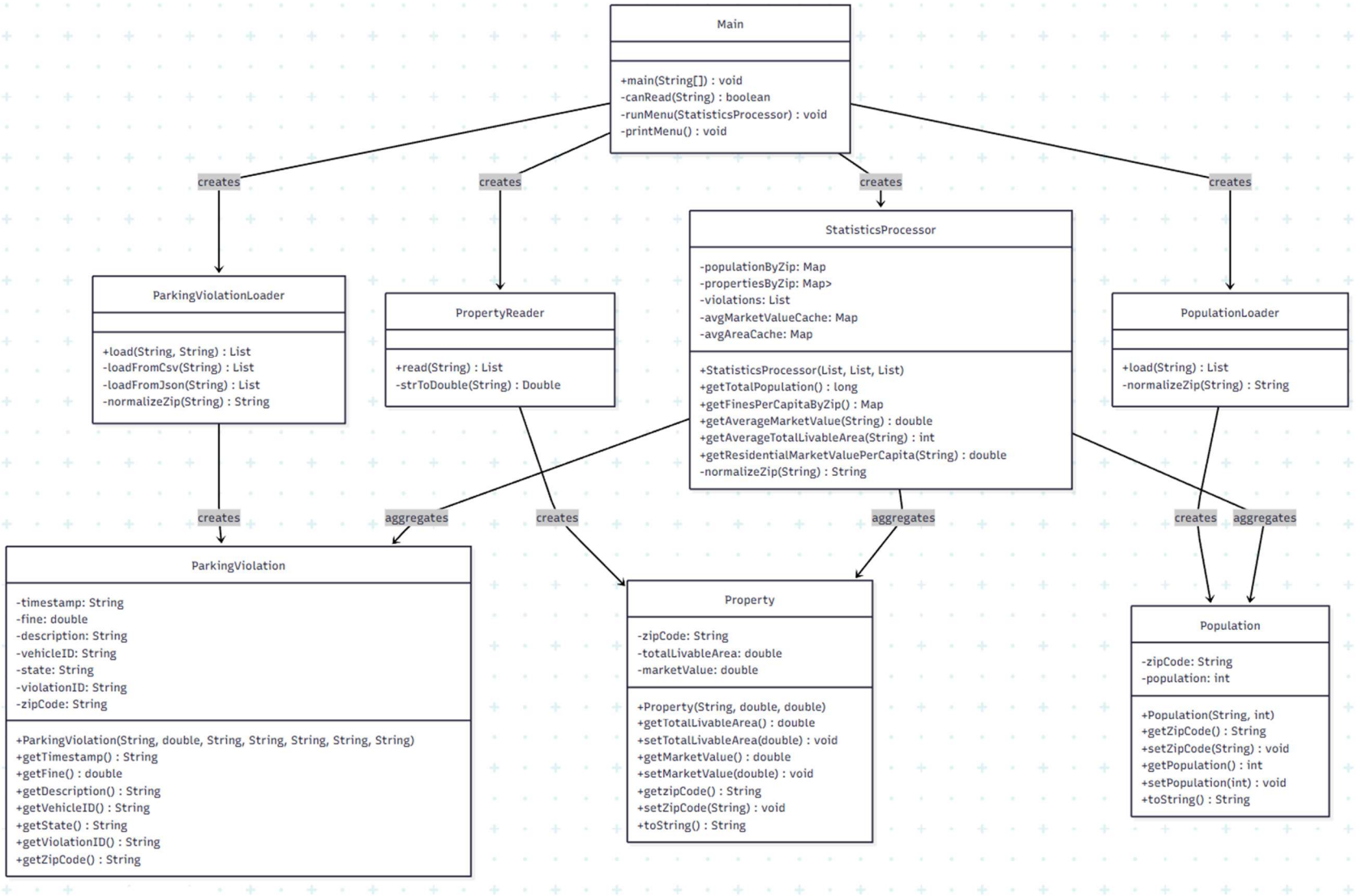
If the user enters any number that is not between 0 and 5, or a non-integer input, they will be ignored and the main menu will be shown again until the user enters a valid selection.

```
Enter selection: 7

Main Menu
1. Total population for all ZIP Codes
2. Fines per capita for each ZIP Code
3. Average residential market value for a ZIP Code
4. Average residential total livable area for a ZIP Code
5. Residential market value per capita for a ZIP Code
0. Quit
Enter selection:
```

To exit the program, the user enters 0. A farewell message will appear and the program will end. An exit code 0 will denote that the program finished successfully without errors.

```
Enter selection: 0
Goodbye!

Process finished with exit code 0
```

UML Diagram:


Main
+main(String[]) : void
-canRead(String) : boolean
-runMenu(StatisticsProcessor) : void
-printMenu() : void
creates
creates
creates
creates
ParkingViolationLoader
+load(String, String) : List
-loadFromCsv(String) : List
-loadFromJson(String) : List
-normalizeZip(String) : String
PropertyReader
+read(String) : List
-strToDouble(String) : Double
StatisticsProcessor
-populationByZip: Map
-propertiesByZip: Map>
-violations: List
-avgMarketValueCache: Map
-avgAreaCache: Map
+StatisticsProcessor(List, List, List)
+getTotalPopulation() : long
+getFinesPerCapitaByZip() : Map
+getAverageMarketValue(String) : double
+getAverageTotalLivableArea(String) : int
+getResidentialMarketValuePerCapita(String) : double
-normalizeZip(String) : String
PopulationLoader
+load(String) : List
-normalizeZip(String) : String
creates
aggregates
creates
aggregates
creates
aggregates
ParkingViolation
-timestamp: String
-fine: double
-description: String
-vehicleID: String
-state: String
-violationID: String
-zipCode: String
+ParkingViolation(String, double, String, String, String, String, String)
+getTimestamp() : String
+getFine() : double
+getDescription() : String
+getVehicleID() : String
+getState() : String
+getViolationID() : String
+getZipCode() : String
Property
-zipCode: String
-totalLivableArea: double
-marketValue: double
+Property(String, double, double)
+getTotalLivableArea() : double
+setTotalLivableArea(double) : void
+getMarketValue() : double
+setMarketValue(double) : void
+getzipCode() : String
+setZipCode(String) : void
+toString() : String
Population
-zipCode: String
-population: int
+Population(String, int)
+getZipCode() : String
+setZipCode(String) : void
+getPopulation() : int
+setPopulation(int) : void
+toString() : String