## Solution Projects:

1. AnalysisLibrary: Contains codes for designing and calibrating high-rise buildings to NMFS models.
2. ResultsVisualizationUtility: helper to launch the AnalysisLibrary (Empty Project.)
3. GISChapter: Reading Neighbours properties from legacy data (previous), distributing models to Areas inventory, then reading the result from R2D resulting .CSV files.

**AnalysisLibrary:**

**Main Run Function:**

This function in (Project.cs) is the trigger for designing and calibrating tall building archetypes. The enum ModelState is used to track the current stage in this process, a backup binary file is saved after accomplishing each stage.

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with medium confidence

**How to define Archetypes:**

Define material properties in Projects.cs, units are in Newton, Meters and AED.

A screenshot of a computer program

Description automatically generated with medium confidence

A screen shot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

Define detailed archetype

A number of archetypes should be appended to “\_models” list property in a project instance.

A screenshot of a computer program

Description automatically generated with medium confidence

The detailed archetype is defined by the number of floors, material properties and layout utility (objects encapsulate the location of columns and shear wall, and loads)

A screenshot of a computer program

Description automatically generated with medium confidence

**How to Wrap Opensees Commands:**

Create a Class that inherits from the BaseCommand class and overrides the “Write Command” Function

A screenshot of a computer program

Description automatically generated with low confidence

|  |  |
| --- | --- |
| **Class/ Module** | **Function** |
| Project | The start class contains the main run function. |
| ModelState | Enum defines the current stage in the process, |
| LayoutUtility | The properties of the defined layout (location of columns and shear walls and dead and live loads) |
| DetailedModel | Class to represent a model for its detailed modeling criteria |
| FloorsGroup | Group of similar floors. |
| ReinforcementUtility | The class contains functions for designing columns and shear walls |
| GridSystem | Class Capture X, Y, and Z axis and generate nodes’ locations so they are not repeated. |
| ModeShapeData | Class for capturing a solved mode shape |
| PushOverResults | Class for capturing a solved pushover case. |
| IDsManager | The class generates IDs for opensees elements. |
| ModelMaterialsInfo | The class captures the properties of all material in the model |
| LoadCase | Base class for a load case in opensees, the parent for GravityLoadCase, ModalAnalysisLoadCase, PushOverLoadCase |
| ModelEstimationEntity | The class contains functions to calculate a bill of materials after design. |
| OpenSeesTranslator | Class organizes writing tcl files and runs opensees as a background process. |
| BaseCommand | Base Class for all opensees wrapping commands, such as BaseNode, Element, OpenSeesMaterial |
| Element | Base Class for Element commands, the parent for ZeroLengthElement,FrameElasticElement, FrameNonLinearElement |
| OpenSeesMaterial | Base Class for material command in opensees, the parent for  PlateFromPlaneStressMaterial, PlaneStressUserMaterial, ElasticMaterial, HystericMaterial, ParallelMaterial, Concrete02Material, ConcreteUniaxialMaterial, PlateRebarMaterial, SteelUniAxialMaterial |
| Region | Base class for opensees region commands wrapping classes; ElementsRegion, Nodes2DRegion, MasterNodesRegion, BaseNodesRegion |
| Section | Base Class for opensees sections commands; SectiongAggregator, LayeredShell, ElasticShellSection, FiberSection |
| ExcelWriterUtiliy | A class contains functions for writing results in Excel sheets in the bin folder. |
| SimplifiedModel | A class represents a simplified models (NMFS) properties |
| CalibrationUnit | A class contains functions for calibrating a detailed model to NMFS model. |

**GISChapter:**

**Main Run Function:**

This function in (GISProject.cs) is the trigger for designing and calibrating tall building archetypes. The enum ModelState is used to track the current stage in this process, a backup binary file is saved after accomplishing each stage.

A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with low confidence

|  |  |
| --- | --- |
| **Class/ Module** | **Function** |
| GISProject | The start class contains the main run function. |
| ProjectState | Enum defines the current stage in the process, |
| ArchTypesGroup | A class represents a group of archetypes model as defined in Chapter 5 |
| ArchType | A class represents a model archetype. The properties are Label, YearBuilt, NumberOfStories, Occupancy, StructureType, and PlanArea |
| Neighbour | A class represents a neighbor in the map, the properties are Name, Usage, NoOfbuildings, Polygon (Footprint), and ID |
| BuildingModel | Class represents a model of an archetype existing in a certain neighbour. The properties are ArchType, BuildingResult, and Count (how many it is repeated in the same area) |
| BuildingResult | A Class contains loss estimation parameter |
| DeterminsticParameter | A class contains functions to add/multiply a deterministic parameter |
| PropbabilisticParameter | A class contains functions to add/multiply a probabilistic parameter |

**Define Neighbour Entity:**

The Neighbour Entity is defined by the ID, Name, usage classification, central location, footprint polygon, and accumulative results.

A screenshot of a computer program

Description automatically generated with medium confidence

**Define Results For an Entity:**

The Byulding Result class contains a Decision variable for a certain scale. To accumulate the result among different entities (for example to get the average variables for the entire neighbor), the static function of Add is used.

A screenshot of a computer program

Description automatically generated with medium confidence