Object-oriented analysis and design: Forestry Management service

—---

1. Analysis
   - Domain
     - Tree, Forest, Forest Management, tree species type
     - *Actors:* Program management(user) interacts with what the Trees can do and the Forest.
   - Function points
     - what things are done by actors: Add trees to the forest, cut trees from the forest, grow the forest, reap the forest, save the forest, load the forest
     - what things are done by non-actors: The Forest has the ability to add random trees, the trees attributes are random
   - Scenarios
     - Walk through typical sequences of events:

       Read forest data from a CSV file, Add a random tree to the forest, Cut a tree from the forest, Grow the forest, Reap the forest, save the forest, load the forest, and display a menu of what to choose.

2. Design
   - Classes and objects (no final variables)
     - Tree: Represents a tree entity, with attributes like species, year planted, height, and growth rate
     - Forest: Represents a collection of trees, with methods for adding, cutting, growing, reaping trees, and reading/writing data from/to files.
   - Data of objects and classes
     - Tree object: Enum species, int yearPlanted, double height, double growthRate. Public accessors for height, private access for other data.
     - Forest object: String name, ArrayList of Trees. Public access for names, private access for the list of trees.
   - Methods of objects and classes
     - Tree

       -Constructor: for initializing tree objects with attributes.

       -Getter method: for getting the height of a tree (public)

       -AnnualGrowthMethod: Simulates annual growth of the tree

       -Generates random tree: Generates a random Tree object with random attributes

       - toString: to display attributes

- Forest

  -Constructor: for initializing given name and list of trees

  - toString:Displays generates a string representation of the forest

  - Methods for adding, cutting, growing, and reaping trees in the forest. (void)

  - Method for Reading, saving, and loading (static)