Classes and Relations

There are 2 interfaces, 1 abstract class, and 9 concrete classes in the project:

1) Two interfaces: **Visitable.java** and **Visitor.java** that layouts the visitor pattern used in this assignment. Specifically, **Visitable.java** is implemented by **Household.java**, an abstract class inherited by four concrete classes representing four types of household: **HouseholdDetachedHouse.java, HouseholdDuplex.java, HouseholdMansion.java, HouseholdTownHome.java**. **Visitor.java** is implemented by **HouseVisitor.java**.
2) One main runnable class: **HalloweenNeighborhoodTraversal.java**, which is the main entry point of the program. It initializes historic information of households, read lists of desired candies from input CSV files, and tries to find and print traversal for each candy list.
3) One abstract class: **Household.java**, which implements **Visitable.java**, and has a method for encoding historic data of four household types into trie-like tree. This method is inherited in each of the four inheriting household classes.
4) Four visitable classes representing household types: **HouseholdDetachedHouse.java, HouseholdDuplex.java, HouseholdMansion.java, HouseholdTownHome.java**, each of the four classes encodes historic data as tree.
5) One visitor class: **HouseVisitor.java**, which has overloaded methods that accept four types of visitable objects representing four household types. It also has a method to traverse trie-like tree in each visitable class that encodes historic data about each household.
6) Two data structure classes that are used throughout the program: **Candy.java, CandyTreeNode.java**. Candy.java is the temporary data structure for storing candy information (candy size and type) when the program reads candy lists from CSV files; It also serves as the medium for encoding historic household data into CandyTreeNode.java data structure. CandyTreeNode.java is the trie-like tree data structure for encoding list of candies. The second level and third level of the tree in each household class represents respectively the “candy size” and “candy type” information.
7) Finally, one IO class **CSVReader.java** for reading and parsing candy lists from CSV files.