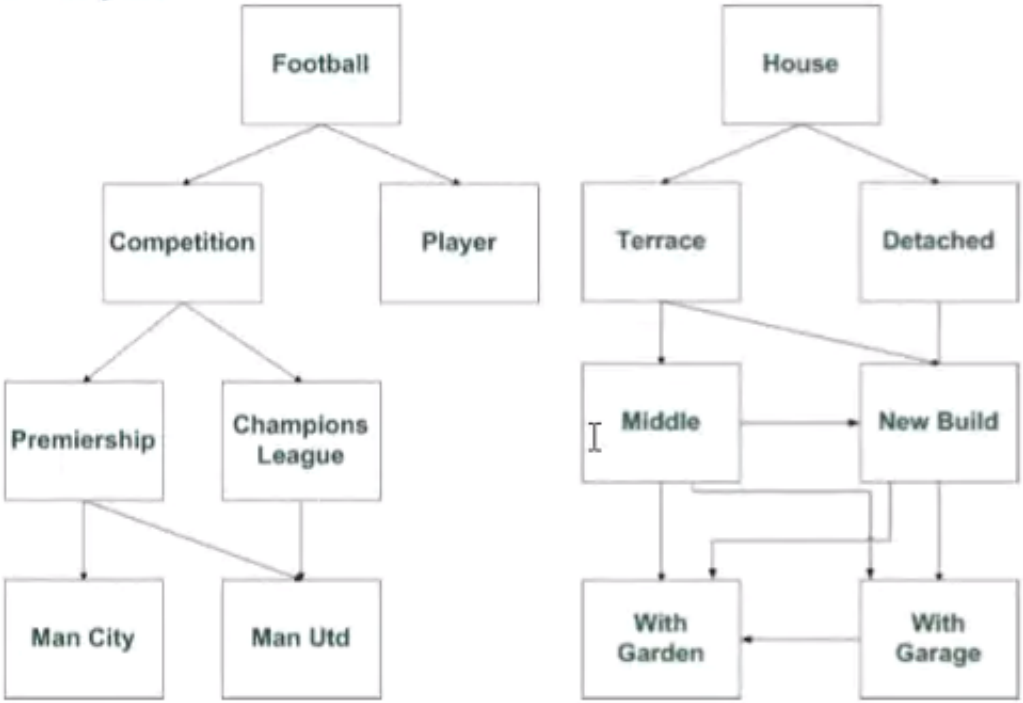
'A **directed acyclic graph (DAG)** is a directed graph with no directed cycles. That is, it is formed by a collection of vertices and directed edges, each edge connecting one vertex to another, such that there is no way to start at some vertex *v* and follow a sequence of edges that eventually loops back to *v* again'

Examples



Exercise

Implement a data structure which the above DAG's could be modelled in. The following functionality is required;

1. Following TDD, ensure unit tests exist for all functionality.
2. Test harness to model one of the above DAG's.
3. The ability to add a child node
4. The ability to add a parent node
5. For any given node, return the node's children
6. For any given node, return the list of all descendant nodes
7. For any given node, return the list of all parent nodes

Additional tasks may be given to you by the developers who you will be working with during the exercise.