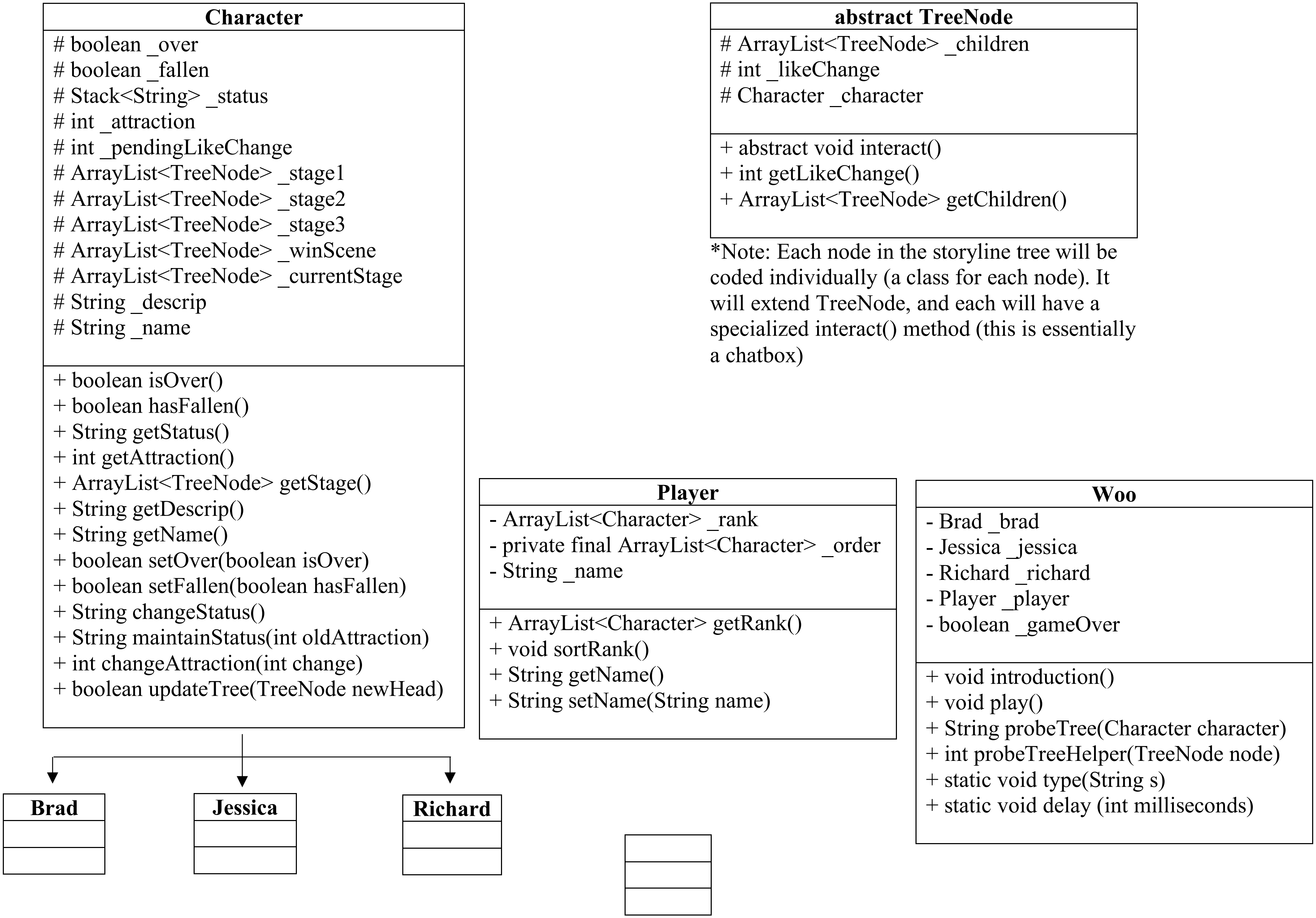

Character

# boolean _over
# boolean _fallen
# Stack<String> _status
# int _attraction
# int _pendingLikeChange
# ArrayList<TreeNode> _stage1
# ArrayList<TreeNode> _stage2
# ArrayList<TreeNode> _stage3
# ArrayList<TreeNode> _winScene
# ArrayList<TreeNode> _currentStage
# String _descrip
# String _name

+ boolean isOver()
+ boolean hasFallen()
+ String getStatus()
+ int getAttraction()
+ ArrayList<TreeNode> getStage()
+ String getDescrip()
+ String getName()
+ boolean setOver(boolean isOver)
+ boolean setFallen(boolean hasFallen)
+ String changeStatus()
+ String maintainStatus(int oldAttraction)
+ int changeAttraction(int change)
+ boolean updateTree(TreeNode newHead)

Brad

Jessica

Richard

abstract TreeNode

# ArrayList<TreeNode> _children
# int _likeChange
# Character _character

+ abstract void interact()
+ int getLikeChange()
+ ArrayList<TreeNode> getChildren()

*Note: Each node in the storyline tree will be coded individually (a class for each node). It will extend TreeNode, and each will have a specialized interact() method (this is essentially a chatbox)

Player

- ArrayList<Character> _rank
- private final ArrayList<Character> _order
- String _name

+ ArrayList<Character> getRank()
+ void sortRank()
+ String getName()
+ String setName(String name)

Woo

- Brad _brad
- Jessica _jessica
- Richard _richard
- Player _player
- boolean _gameOver

+ void introduction()
+ void play()
+ String probeTree(Character character)
+ int probeTreeHelper(TreeNode node)
+ static void type(String s)
+ static void delay (int milliseconds)