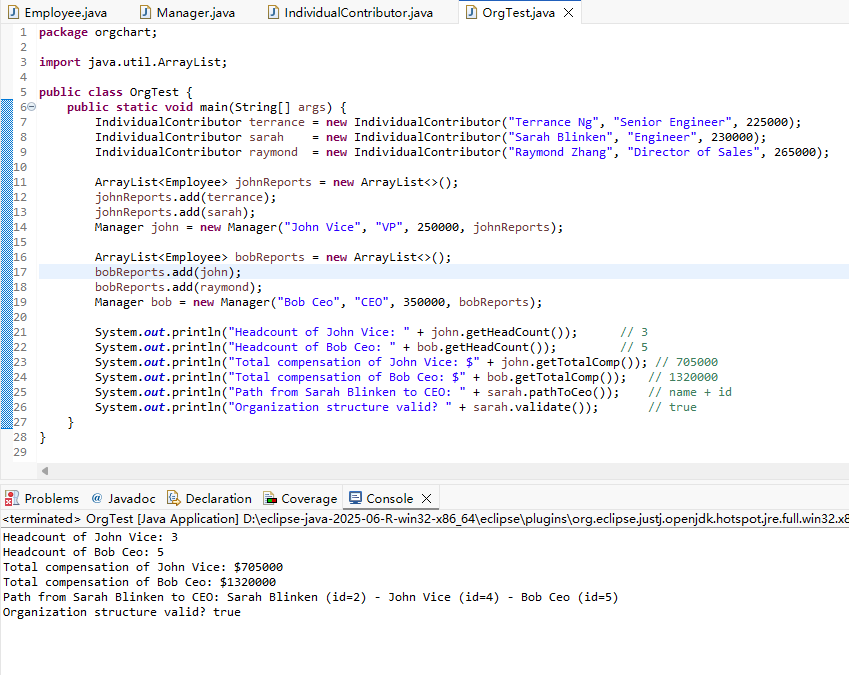
Part1

The org chart implementation demonstrates the use of inheritance, abstraction, and recursive computation in building a hierarchical employee management model. The design centers on an abstract Employee superclass that encapsulates shared attributes such as ID, name, title, and salary, while specialized subclasses Manager and IndividualContributor extend functionality to represent leadership and non-leadership roles. Through recursive traversal, the program dynamically calculates organizational headcount and total compensation within any managerial subtree. A validation method ensures structural integrity by verifying bidirectional manager-report relationships and detecting potential cycles. The resulting hierarchy accurately models reporting lines from individual contributors to the CEO and produces consistent analytical outputs for hierarchy size, compensation aggregation, and upward navigation paths.



Part2

The movie analyzer showcases robust file parsing and functional data processing using exception handling and lambda-based sorting. It reads a CSV dataset into a structured list of Movie objects, gracefully handling malformed records through a custom unchecked exception while maintaining processing continuity. The analyzer applies functional programming principles to sort and display the dataset by different criteria—descending ratings and ascending genres—reflecting both analytical precision and code efficiency. This implementation highlights practical data-cleaning resilience, modular decomposition between parsing and analysis, and the expressive power of Java’s lambda syntax for flexible dataset manipulation.

