**Nested Conditional Refactoring**

**public** **double** **getPayAmount**() {

**if** (isDead){

**return** deadAmount();

}

**if** (isSeparated){

**return** separatedAmount();

}

**if** (isRetired){

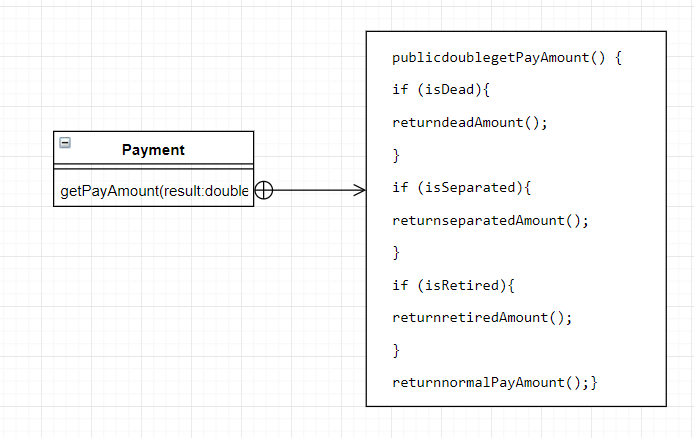
**return** retiredAmount();

}

**return** normalPayAmount();

}

**UML Diagram**



The conditionals have been replaced with guard clauses that follow one after another instead of being nested within each other. This makes it easier to read and find errors in. Each statement immediately ends execution and returns a null value if the guard clauses are true.

**Testing**

The method should be tested to ensure it returns the same values as it did before it was refactored. This can be done using a unit test an asserting each expected value for each conditional. Afterwards, the system should be tested to ensure it still works as before.