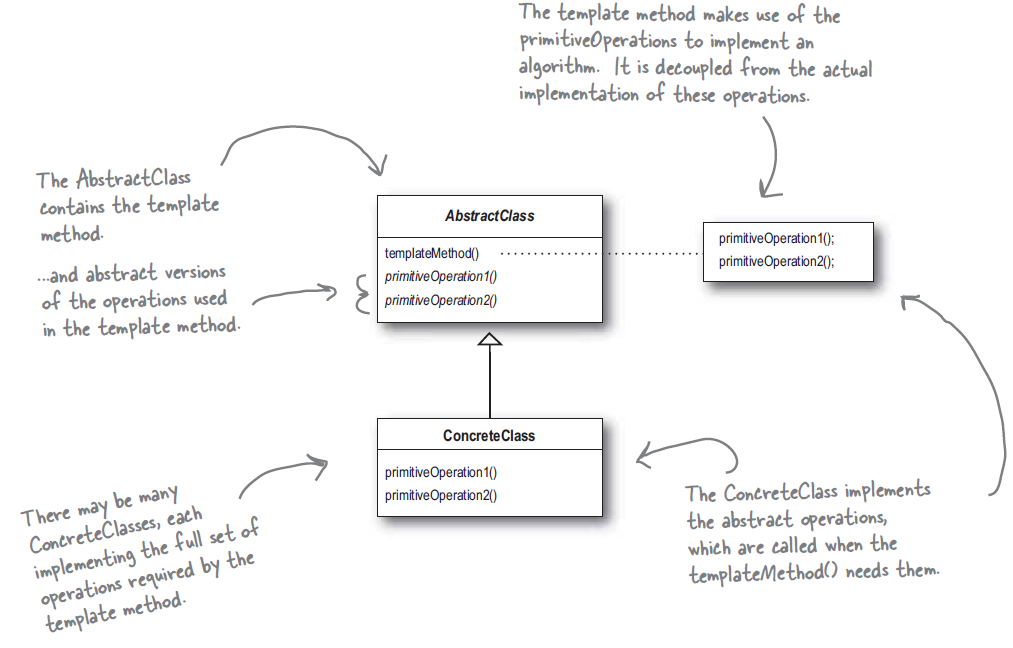
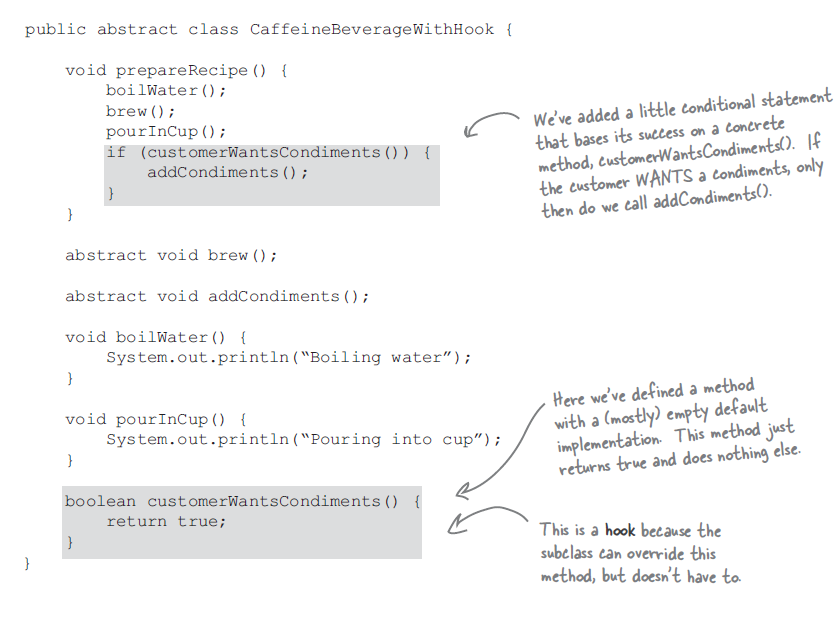
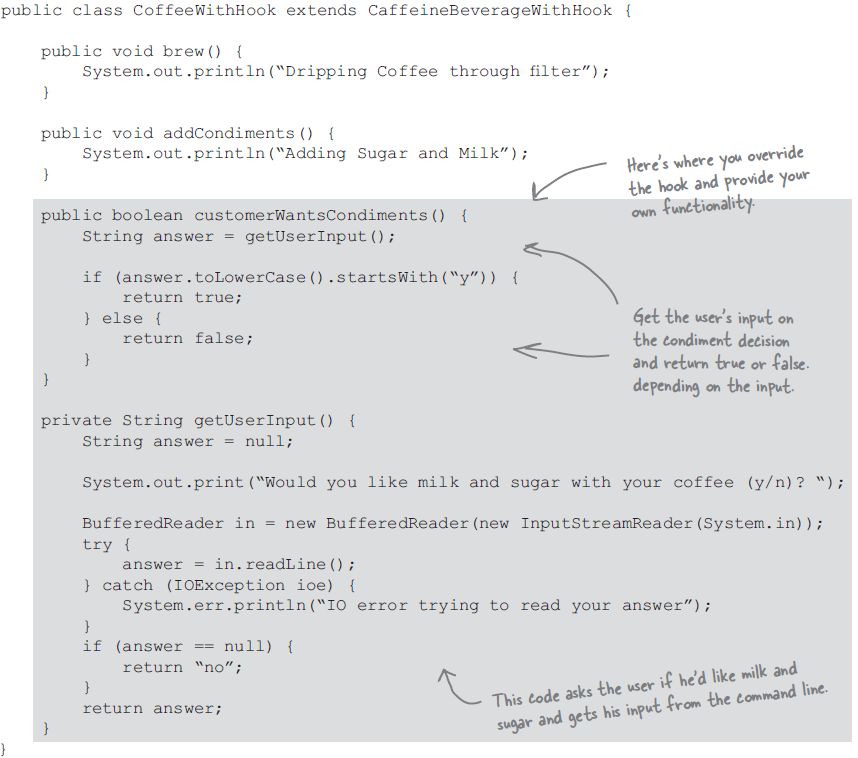
**The Template Method Pattern**

1. **Definition**: The template method pattern defines the skeleton of an algorithm in a method and deferring some steps to subclasses without letting them modify the structure of the algorithm.

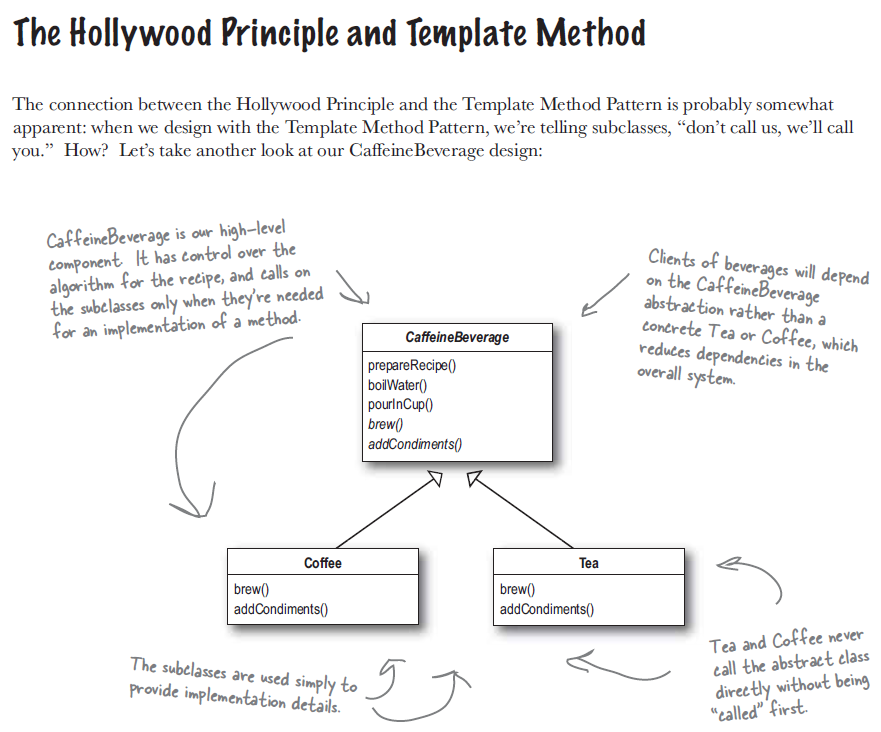


1. **Advantages**:
   * 1. The abstract class containing template method protects the algorithm by defining the template method final.
     2. Any changes in the algorithm need to be done in one place only.
     3. Subclasses can do better code reuse.
     4. Adding new implementation means adding a subclass that gives implementation of abstract methods required by template method.
     5. This is good pattern to build frameworks where framework control how things done but user can specify the working in various steps.
2. **Hooks**: Use hooks to implement optional part of the algorithm. Since abstract class provide the default implementation a subclass can ignore it but it can also use to “hook into” the algorithm at various points.

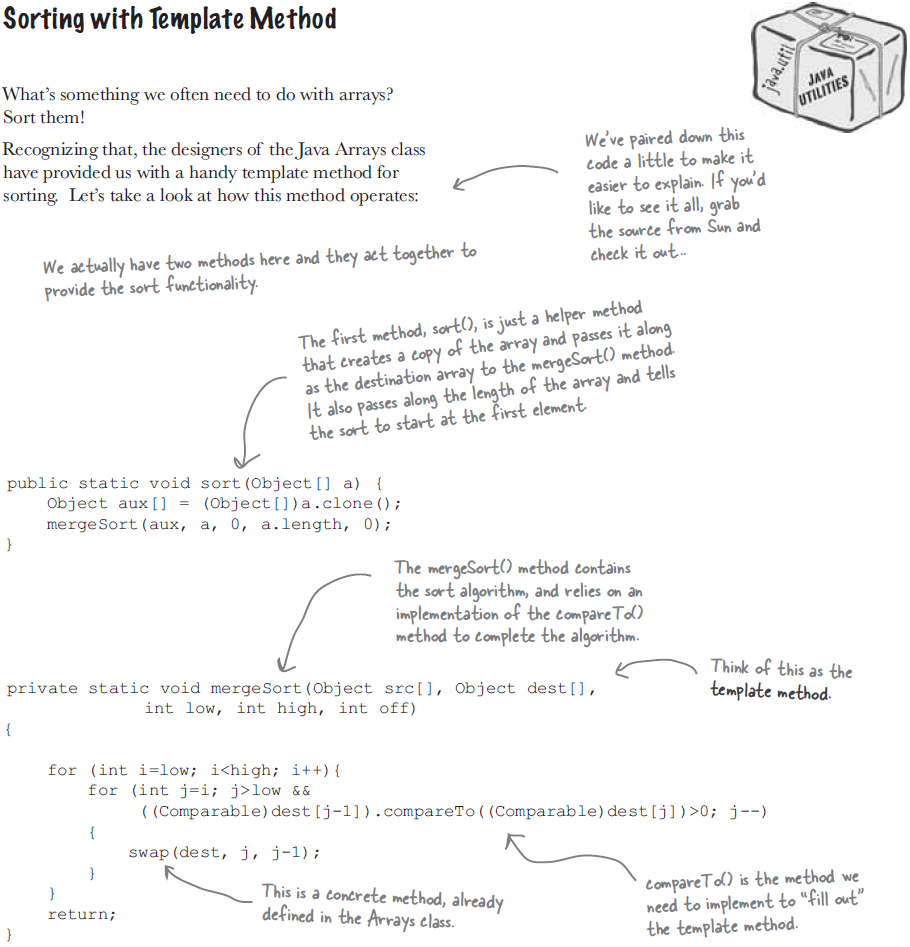




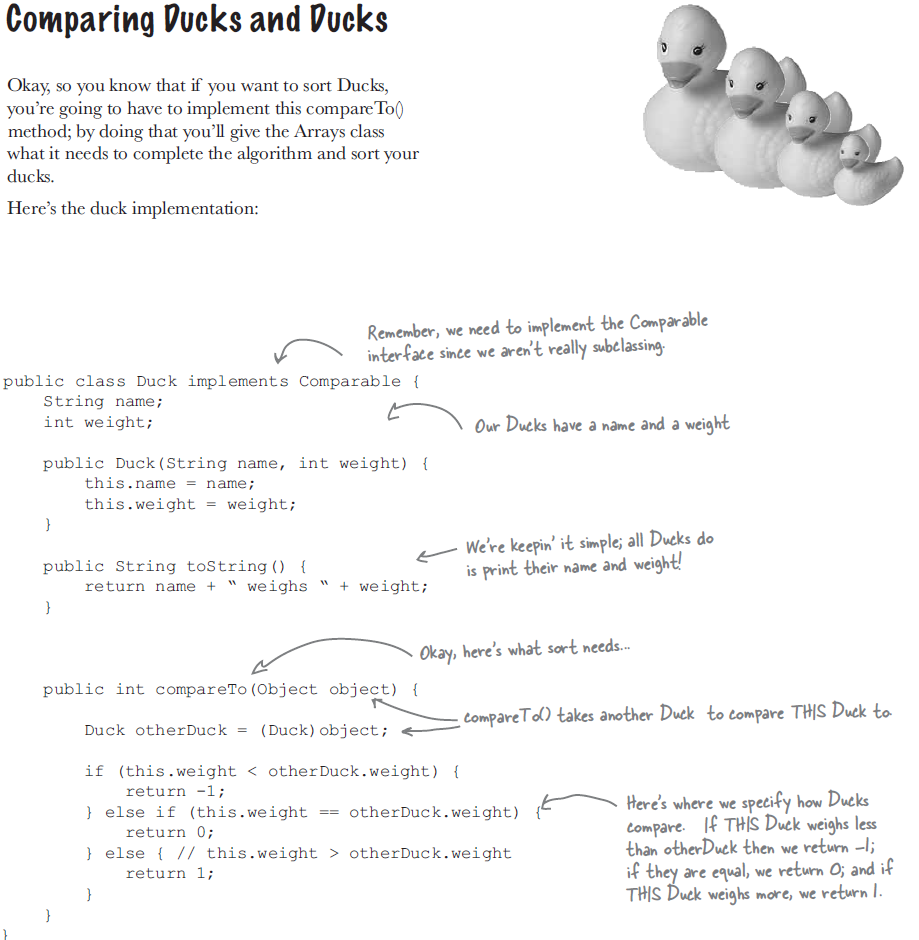
1. The Hollywood Principle: Don’t call us, we’ll call you.
   1. With this we allow lower level component to hook into the higher level component but higher level component determine when to use lower level components.
   2. This design principle gives a technique to decouple higher and lower level components.

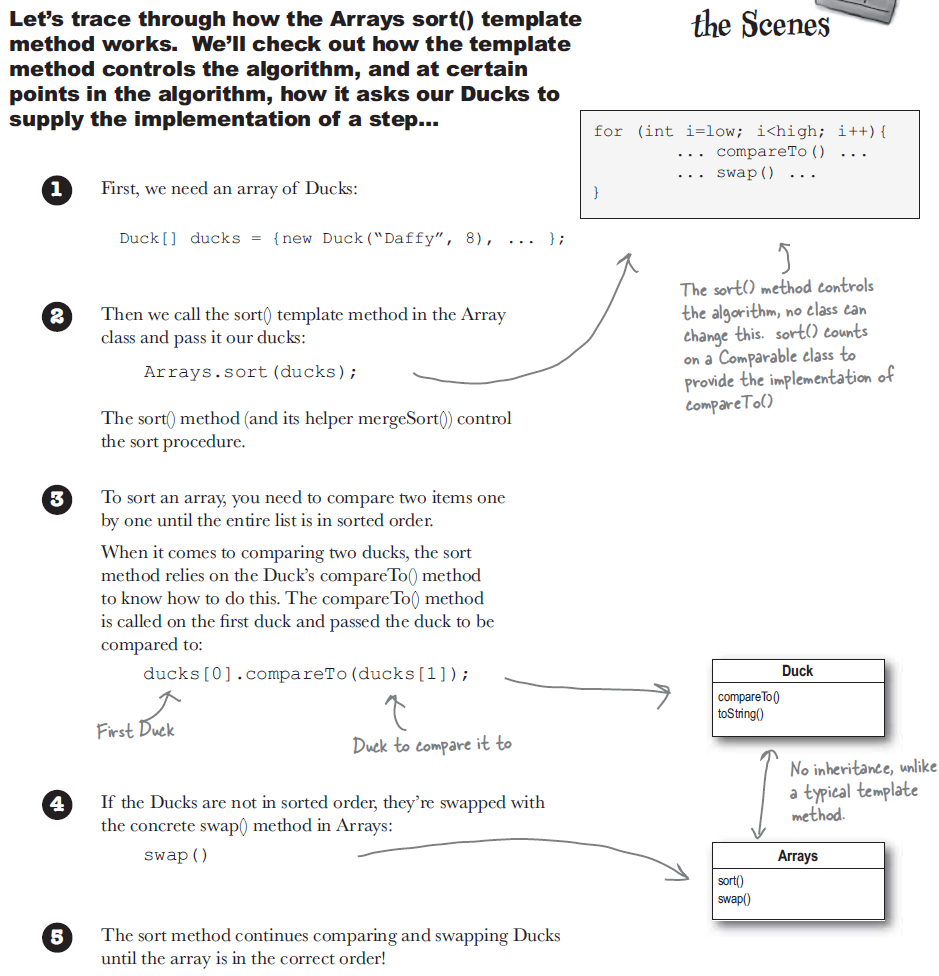


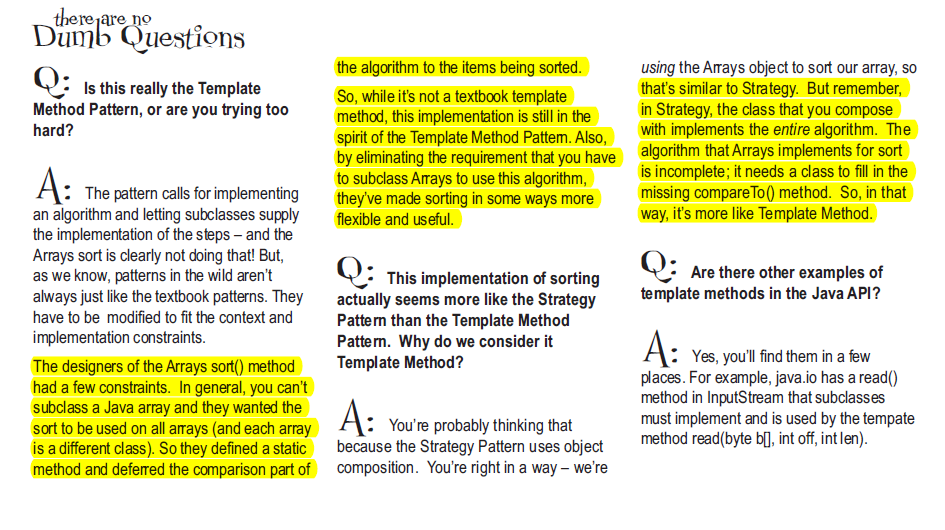
1. Java api usage:

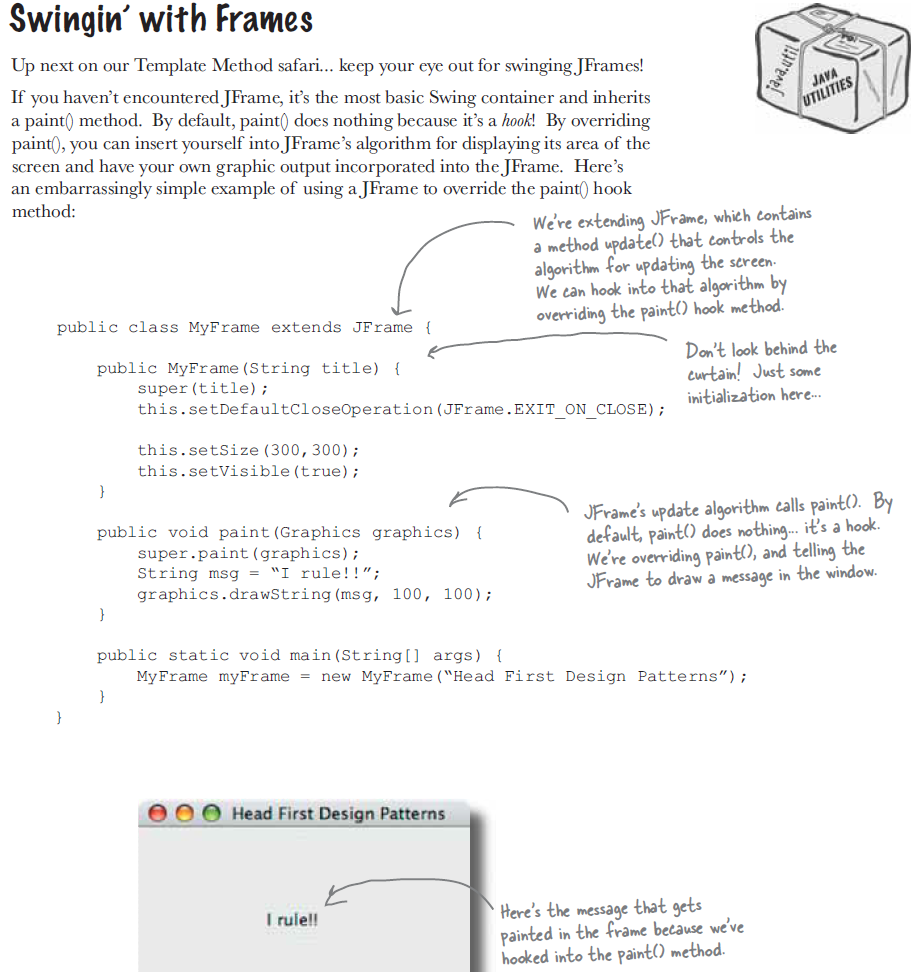


This is legacy merge sort used to sort array of objects. Recent api uses tim sort for object array sorting and dual pivot quick sort for primitive array sorting.









1. Comparison:
   1. The strategy and template method pattern both encapsulates algorithms but one by inheritance and other by composition.
   2. The factory method is a specialize template method.