

Some advice & questions from the recent code review...

advice...

- use explicit access modifiers (unless you have a v.persuasive reason not to)
- use javadoc
- read about collection classes
- read more about collection classes
- read about enums – find out how you can attach values (data) & methods (behaviour) to enums
- decoration: you can also decorate with methods (not just data)
- strategy: if it doesn't change dynamically, it probably doesn't need a strategy pattern
- template: the main template method can be an algorithm with control structures, etc – not just a linear sequence of sub-method calls

questions (read all of these, even those about patterns you have not used)...

- decoration: if some types of decoration can only be applied once – how do you achieve this?
- decoration: if it is only possible to decorate exactly 'twice', 'once' or 'not at all' (ie: maximum of 2 decorations) – is this a decorator pattern?
- how could I argue that a chain (of responsibility) is a decorator? When does this argument get flakey?
- what is the difference between a strategy & a template?
- adapting: are you really adapting if you are just wrapping one class in another?
- how can you use enums to carry strategies – what are the dis/advantages?
- is a linked list simply a decoration? How far can I go with this argument?