

NAME - Miran Member

Template Method

Template Method requires you to build a skeleton code which then the subclasses can override the steps without changing the overall algorithm. Template Method is far less complex than other styles in java and is also fairly popular. If you work on a team this would be the ideal method to use because it would allow the team to work on the skeleton together and then the team can work on individual subclasses without changing the overall structure.

In my Code, I have a skeleton code for a class called computer. This class has a method to make a computer which has a bunch of if statements that only become true when the subclass overrides the `customerwantssomething()`; Also the skeleton has abstract methods `addGPU` which the subclass must override. So basically the subclass can choose what to add to the computer from a bunch of computer parts to specify a need for that type of computer.

In my case, I have a `GamingComputer` that extends `Computer` which requires a GPU, CPU, Motherboard, PSU, but the skeleton class also has 2 more components that the gaming computer doesn't really need. So in the `GamingComputer` class I set `customerwantsSoundCard` to false which will make the if statement in the `Computer` class not execute which means the Sound Card will not be added to the `GamingComputer`. The `GamingComputer` will choose a random string from a String array and return it to `Computer` which will compile a `ListView` then the `Computer` class method `makeComputer` will return the `ListView` that can be displayed on the screen.

The same can be said for the `OfficeComputer` but in this case this computer uses all the components except for the GPU. So I follow the same directions from above.

This is a really Easy Style that can let me make as many type of computers as i want The only drawback I can come up with is that everytime you want to add a new component in the main class you would have to add that component in every subclass and set it to false if that subclass will not use it.

In conclusion, I think that this Style is great when you want to work with a big UserBase that requires different types of things such as a sandwich, computer, car etc.