Tyler Stephens (tds153)

CS OOP 3354

Dependency Injection

Due: 05/03/2021

What is dependency injection? Dependency injection like the name implies is a concept which uses dependencies. This is to say that one java class depends on another class or that one class uses an instance of another class it is therefore dependent on that class. Dependency injection itself is a design pattern that extends two other design principles one being the dependency inversion principle and the other being the single responsibility principle [1]. But what is it we have discussed what it is an extension of and what concepts its uses, so what is dependency injection?

The definition of dependency injection essentially states the dependency is a software engineering concept in which “one object (or static method) supplies the dependencies of another object. A dependency is an object that can be used (a service).[2]” What does this definition mean? Essentially from my understanding dependency injection boils down to being the filling of an Oreo cookie it connects two separate classes and links the two of them together. Now what does that mean, dependency injection has three responsibilities. Firstly, it creates the object or objects to be supplied. Secondly, dependency injection is responsible for locating the classes that require the created objects. And lastly the injector supplies the created objects or “injects” them into the required classes [3]. Therefore, dependency injection is mainly used to decrease coupling between classes by separating the class and its dependency.

So, what are the advantages of using the dependency injection design pattern? Well firstly as I said earlier if enables loose coupling which is very useful in object-oriented programming [3]. Also by using dependency injection it helps in unit testing by making clients independent which eases the unit testing process. Another plus of using dependency injection is that using tends to help reduce the amount of repetition and repeated use of sections of code [2].

Now that we have discussed the various advantages, I would be remised to not mention the disadvantages of using this design patter. The first disadvantage I would like to discuss is that it makes reading code and tracing more difficult because behavior and bulk of the code is separated from its constructor [2]. This can especially troubling if you have a dependency injector that is dealing with multiple classes. Another problem that we find when dealing with dependency injection is that it can be difficult and complex to learn. I can personally attest to this as after reading articles about it from various websites and even reviewing the wiki I’m still not certain that I’m one hundred percent on how to implement it and other smaller details that deal with dependency injection [3].

So in conclusion dependency injection is a design pattern in which an object supplies or creates objects that other classes will use. It has various advantages which are useful in OOP and relatively few disadvantages making it a ideal design pattern if one is looking to decouple classes from one another.

[1] Janssen, T. (2021, March 31). Design patterns explained – dependency injection with code examples. Retrieved May 03, 2021, from https://stackify.com/dependency-injection/

[2] Dependency injection. (2021, March 26). Retrieved May 03, 2021, from https://en.wikipedia.org/wiki/Dependency\_injection

[3] Karia, B. (2020, March 19). A quick intro to dependency injection: What it is, and when to use it. Retrieved May 03, 2021, from https://www.freecodecamp.org/news/a-quick-intro-to-dependency-injection-what-it-is-and-when-to-use-it-7578c84fa88f/

[4] Vogel, L. (n.d.). Get more... Retrieved May 03, 2021, from https://www.vogella.com/tutorials/DependencyInjection/article.html#:~:text=Dependency%20injection%20(DI)%20is%20the,an%20instance%20of%20this%20class. https://stackify.com/dependency-injection/