|  |  |  |
| --- | --- | --- |
| OOPs Design Principles | | |
| DRY (Don’t’ Repeat Yourself) | | |
| One of the most important OOPs Design Principles is DRY, as the name suggests DRY means don’t write duplicate code, instead use Abstraction to abstract common thins in one place. If you are using JDK 8 or later version, you can implement the method in interfaces as well. If you have the same block of code in more than two places, consider making it a separate method. Even if you use a hard-coded value more than once make them public final constant. | | |
| ADVANTAGE | | The main target is in maintenance. It’s important not to abuse it, duplication is not for code, but for functionality. |
| OOP | When it comes to OOP, it suggests us to utilize abstract classes, interfaces, and public constants as much as possible. Whenever there’s a functionality common across classes, one better option is to abstract them away into a common parent class. Another option is to use interfaces to couple their functionality. | |