|  |  |  |
| --- | --- | --- |
| OOPs Design Principles | | |
| PINI (Programming for Interface not for Implementation) | | |
| YAGNI stands for “you aren’t going to need it”: don’t implement something until it is necessary. Always implement things when you actually need them, never when you just foresee that you need them. It leads to code bloat; the software becomes larger and more complicated. | | |
| WHY ? | | * Write unnecessary code and also make assumptions about further destination systems. * require a huge refactor even if other destinations finally come to life. |
| EXAMPLE | * In general, the strategy pattern is a great option to deal with algorithms based on various criteria. However, some developers abuse it by trying to use it everywhere to replace if or switch statements. If the logic is pretty simple and there is just one if - then there is no need to rebuild it and use strategy patterns. | |