Software design Patterns

TechTalk

basic introduction

01 - intro

02 - good to know

03 - types of design patterns

04 - Conclusion

01 - Introduction

Software design patterns are like recipes for solving design problems.

Software design patterns are established solutions to common software development challenges They provide templates or blueprints that simplify creating, organizing, and coordinating objects in software systems. By using effectively these patterns, developers ensure their code is more reliable, easier to maintain, extensible and promote effective collaboration among team members.

02 - Good to know

Design patterns are not tied to a programming language. They are conceptual solutions that you can use regardless of the language you're working with.

They are particularly useful in complex systems.

Design patterns can improve system performances. But choosing the wrong pattern or overusing them can impact those same performances.

many design patterns are applicable in web development.

03 - types of design patterns

patterns fall into different categories based on the type of problem they solve.

Creational patterns focus on how objects are created.
→ the singleton

Structural patterns deal with how objects are organized and interact with each other.
→ the MVC

Behavioral patterns are about how objects communicate and work together to perform tasks.
→ the observer

04 - Conclusion

Using design patterns has great benefits. By using well-known patterns, your code becomes more understandable and predictable, allowing team members to quickly grasp and contribute to the project.

This approach also promotes modularity, extensibility, and reduces development time and effort, making it a valuable practice for building robust and efficient software systems, as long as they are not wrongly or over used).

MVC, Observer, and Singleton are well-known patterns used to organize and structure code by both frontend and backend developers.

software design patterns are great tools in a developer's toolkit.

Thanks !

resources :

https://refactoring.guru/

https://www.geeksforgeeks.org/software-design-patterns/

Lyne K