What are design patterns?

There are several design patterns used in Software development. They are categorized into three main groups which are Creational Patterns, Structural Patterns and Behavioral patterns. The design pattern where created, to help the developer solve recurring design problems and also improve the code maintainability.

Which types are out there?

Creational Patterns:

The patterns deal with object creation mechanisms, abstracting the process to make the system independent of how its objects are created, composed and represented. So basically they are responsible for creating the objects while hiding the underlying logic of how these objects are created or initialized. You would Encapsulate the details on how an object is constructed within a well defined pattern or component. This will promote that the design is more clear and modular. The clients which will receive the code also don't need to know the details, which means they will get the clean code and all the logic is hidden, which means that this is flexible. You can change the logic at any moment without affecting the clients. Testing the Software is also easier, because it's easier to test the individual Components. These patterns contribute to cleaner, more maintainable, and more adaptable software systems.

Structural Patterns:

These patterns focus on how a class or objects are organized, composed and interact with larger structures. They make the system independent of the internal structure of its objects, ensuring efficient composition and organization. They promote Modularity(Breaking down a software system into separate modules or components. Each module is a discrete unit with clearly defined purpose and responsibility) by organizing components into separate and reusable parts. This modularity simplifies code maintenance and allows you to update or replace individual components without affecting the entire system. They also help in reducing dependencies between components which makes it easier the code to understand, modify or extend.In summary, structural design patterns contribute to more modular, reusable, flexible, and efficient software systems by providing guidelines for organizing and composing objects and classes within larger structures. These patterns are valuable tools for improving software design and architecture.

Behavioral Pattern:

These patterns deal with communication between objects, specifying how they interact and communicate with one another. The benefit of the communication is that they define and streamline the flow of the messages between objects, making the systems communication clearer and more organized. This also improves Collaboration which means when developers work with a common set of behavioral patterns, it enhances collaboration and communication between team members, where everyone will understand the expected interactions and responsibilities of the object.