CS 기술 면접 대비

Library와 Framework를 설명

* Library와 Framework 모두 특정한 기능들을 모듈화한 미리 작성된 코드이다. flow의 제어권이 프로그래머에게 있는지 아니면 framework에 있는지로 Library와 Framework가 구별된다.

Singleton Pattern

* 클래스 하나에 오직 하나의 인스턴스를 가지도록 하는 패턴. 인스턴스를 마구 생성하는 것에 비해 메모리와 속도 측면에서 이점을 가질 수 있으며 하나의 인스턴스이기에 객체 상태의 공유가 유의미한 환경에서 잘 사용된다.

Factory Method Pattern

* 객체를 생성하기 위한 인터페이스를 정의하고 어떤 클래스의 인스턴스를 만들지 하위 클래스에서 결정하게 만드는 패턴. PhoneFactory 인터페이스가 있고 이를 상속 받는 IphoneFactory, GalaxyFactory를 구현하여 각각이 다른 인스턴스를 생성하게 했다면 하위 클래스가 어떤 인스턴스를 생성할지 정하도록 한 것

Strategy Pattern

* 캡슐화된 알고리즘을 갈아낄 수 있도록 함으로써 코드의 동작을 다양화하는 패턴. 다형성을 이용해서 구현할 수 있다. OCP를 지킬 수 있음

의존성 주입

* 의존성 만을 주입하는 중간자(Ioc Container)에 필요한 모듈들을 등록하고 의존성을 주입시키는 방법. (개발자가 직접 주입하는 것이 아니라 중간자가 개입함으로 제어의 역전이 일어남) 모듈간의 결합을 느슨히 하여 모듈간 전환을 부드럽게 하며 SRP OCP를 위배하지 않도록 도움을 준다.

Observer Pattern

* 관찰자가 객체의 상태변화를 관찰하고 상태가 변화할 때 마다 옵저버들에게 변화 상황을 알리는 패턴

자바 상속 vs 구현

* 상속: 자식 클래스가 부모 클래스의 속성과 메서드를 상속받아 추가 및 확장을 가능하도록 하는 기능. 재사용성, 중복성의 최소화를 통해 객체지향의 특징을 강화해준다.
* 구현: 부모 인터페이스를 자식 클래스에서 재정의하여 구현하는 것을 말함 상속과 달리 반드시 부모 클래스의 메서드를 재정의해야 함
* 자바에서는 diamond problem으로 다중 상속을 허용하지 않는다. 하지만 인터페이스는 다중 구현이 가능함 이것 또한 상속과 구현의 차이
* Diamond problem
  + 텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

    자동 생성된 설명
  + Child는 Father를 extends하고 Mather도 extends 다만 Person의 특정 메소드가 Father와 Mother에서 다르게 구현되어 있다면 child는 Person의 특정메소드를 실행할 때 Father 것으로? Mother의 것으로? -> diamond problem

프록시 패턴, 프록시 서버

* 프록시 패턴이란 대상 객체의 앞단에 흐름을 가로챌 수 있는 가상 객체를 두어 인터페이스 역할을 할 수 있도록 설계한 패턴을 말함. 이를 통해 프록시 서버를 만들 경우 보안적인 측면에서 공격의도가 있는 요청을 걸러낼 수도 있고 CDN등을 구축하여 서버 응답성을 높일 수도 있음 또한 모든 요청을 곧이곧대로 서버에 보내어 SPOF문제를 직면하기 보다 SPOF 문제를 해결할 수 있는 것임

DDOS 공격

* Distributed Denial Of Service: 공격대상이 되는 서버의 서비스 중단을 목적으로 표적 서버에 대량의 트래픽을 유발하는 공격 -> 프록시 패턴을 이용한 CloudFlare를 이용하면 사용자가 아닌 시스템으로부터 오는 트래픽을 자동으로 차단하여 공격을 쉽게 막아낼 수 있다 (프록시 패턴 이용)

CDN (Contend Delivery Network)

* 분산된 형태의 서버 네트워크를 말함 콘텐츠를 사용자에게 가까운 여러 프록시 서버에 데이터를 캐싱, 배포해놓음으로써 요청에 대한 응답성을 높일 수 있음(SPOF 문제 해결) 이러한 특징 때문에 주로 CDN은 한번 배포하면 수정이 적은 특징의 데이터를 위한 서버 네트워크를 구축할 때 사용이 됨. 영상 컨텐츠 등이 그러하기에 넷플릭스 왓챠 등 여러 OTT에서 CDN이 사용되는 것을 확인할 수 있음. CDN의 정책에는 여러가지가 있는데 사용자가 여러 서버노드들 중 핑을 날려보고 응답성이 가장 좋은 프록시 서버에게 요청하도록 하는 정책을 NETFLIX에서 사용한다고 분산시스템 과목에서 배운 기억이 있음