

1. OOP에 대해서 설명하세요.

: 객체 지향 프로그래밍(OOP)은 프로그램을 객체들의 집합으로 보고, 객체 간의 상호작용으로 프로그램을 구현하는 방법론입니다.

2. 캡슐화 상속 추상화 다형성에 대해 설명하세요

: 캡슐화는 객체의 속성과 메서드를 하나로 묶는 것이며, 상속은 부모 클래스의 속성과 메서드를 자식 클래스가 받아 사용하는 것입니다. 추상화는 복잡한 시스템을 간단한 인터페이스로 표현하는 것이고, 다형성은 하나의 인터페이스나 클래스가 다양한 형태로 동작하는 것을 말합니다.

3. 지역 변수 클래스 변수 인스턴스 변수 비교?

: 지역 변수는 함수 내부에서만 사용되는 변수이며, 함수가 종료되면 사라집니다. 클래스 변수는 클래스에 속하며, 해당 클래스의 모든 인스턴스가 공유합니다. 인스턴스 변수는 클래스의 인스턴스마다 별도로 존재하는 변수입니다.

4. Garbage Collection 특징에 대해 설명하세요.

: Garbage Collection은 동적으로 할당된 메모리 중 사용하지 않는 메모리를 자동으로 회수하는 기능입니다. 이를 통해 개발자는 메모리 관리에 대한 부담을 줄일 수 있습니다.

*취업매니저 메세지

-면접관 분들은 면접자들에게 전공의 여부에 상관없이 공통적으로 Java/Spring/CS의 지식을 알고 있는지 묻는 질문을 가장 많이 합니다.

-"'A'와'B'를 비교해서 설명해보세요" 라는 요청이 많기 때문에 그에 따라 사전준비가 필요합니다.



5. Redis 특징에 대해 설명하세요.

: Redis는 인 메모리 데이터 구조 저장소로, 캐시, 메시지 브로커 등 다양한 용도로 사용됩니다. 높은 성능과 복제, 지속성 등의 기능을 제공합니다.

6. 레디스가 죽었다고 가정하자. 그리고 나서 다시 재부팅을 했는데 그 과정에 100개 요청이 도달했다. 어떻게 처리할 건가요?

: Redis가 재부팅하는 동안 들어온 요청은 임시 큐에 저장하여, Redis가 준비되면 처리하도록 할 수 있습니다.

*참고

: Redis가 재부팅하는 동안에 요청이 들어오면 그 요청들은 처리되지 않습니다. 왜냐하면 Redis는 In-Memory 데이터베이스이기 때문에 메모리에 상주하는 동안만 데이터를 처리할 수 있습니다. 따라서 Redis가 재부팅하는 동안에는 서비스가 중단됩니다.

이러한 문제를 해결하기 위한 방법 중 하나는 Redis Sentinel이나 Cluster를 사용하여고가용성을 확보하는 것입니다. Sentinel은 주 Redis 서버가 실패하면 대기 서버로 자동 전환하는 기능을 제공하며, Cluster는 데이터를 여러 Redis 노드에 분산시켜 서비스 중단을 최소화합니다.

*취업매니저 메세지

-면접관 분들은 면접자들에게 전공의 여부에 상관없이 공통적으로 Java/Spring/CS의 지식을 알고 있는지 묻는 질문을 가장 많이 합니다.

-"'A'와'B'를 비교해서 설명해보세요" 라는 요청이 많기 때문에 그에 따라 사전준비가 필요합니다.



그러나 이러한 방법도 100%의 서비스 중단을 방지할 수는 없습니다. 따라서 클라이언트 측에서는 Redis에 요청을 보낼 때 타임아웃을 설정하거나, 요청 실패 시 재시도하는 로직을 구현해야 합니다.

또한, Redis 앞단에 메시지 큐를 두어 요청을 임시로 저장하고 Redis가 다시 온라인이 되면 그때 요청을 처리하는 방법도 있습니다. 이러한 방법은 서비스의 중단 시간을 최소화하는 데 도움이 될 수 있습니다.

7. JPA를 다뤄본 적이 있나요? 있다면 어떻게 사용 경험에 대해 설명하세요.

: JPA는 Java의 ORM 기술로, 객체와 테이블을 매핑하여 SQL 없이 데이터베이스 작업을 할 수 있게 해줍니다. 이를 통해 개발 속도를 향상시키고, 유지보수를 용이하게 했습니다.

(JPA에 대해 설명하고, 그 사용 경험을 생각해서 덧붙여 말씀하셔야 합니다. 사용 경험에 대한 예시는 각각 준비하시기 바랍니다.)

*취업매니저 메세지

-면접관 분들은 면접자들에게 전공의 여부에 상관없이 공통적으로 Java/Spring/CS의 지식을 알고 있는지 묻는 질문을 가장 많이 합니다.

-"'A'와'B'를 비교해서 설명해보세요" 라는 요청이 많기 때문에 그에 따라 사전준비가 필요합니다.



8. Spring DI / IoC가 무엇인지 설명하세요.

: DI(Dependency Injection)는 의존성 주입을, IoC(Inversion of Control)는 제어의 역전을 의미합니다. Spring에서는 IoC 컨테이너가 객체의 생명 주기를 관리하고, DI를 통해 객체 간의 의존성을 자동으로 처리합니다.

9. 동기와 비동기의 차이를 비교해서 설명하세요.

동기는 작업이 순서대로 실행되며, 한 작업이 끝나야 다음 작업이 시작되는 반면, 비동기는 작업이 동시에 실행되며, 작업의 완료 순서와 상관없이 다음 작업이 시작됩니다.

10. HTTP의 메소드에 대해 설명하세요.

HTTP 메소드에는 GET, POST, PUT, DELETE 등이 있습니다. 각각 정보 조회, 정보 추가, 정보 수정, 정보 삭제를 위해 사용됩니다.

11. REST API가 무엇을 의미하는지 설명하세요.

REST API는 Representational State Transfer의 원칙을 따르는 API로, 자원 중심의 구조, 상태 정보를 가진 표현, 표현을 처리하는 메소드 등의 특성을 가집니다.

*취업매니저 메세지

-면접관 분들은 면접자들에게 전공의 여부에 상관없이 공통적으로 Java/Spring/CS의 지식을 알고 있는지 묻는 질문을 가장 많이 합니다.

-"'A'와'B'를 비교해서 설명해보세요" 라는 요청이 많기 때문에 그에 따라 사전준비가 필요합니다.



12. 사용해본 개발 방법론이나 패턴이 있나요? 왜 그 방법론을 사용했나요?

: Agile 방법론과 MVC 패턴을 사용한 경험이 있습니다. Agile은 고객의 요구사항 변화에 유연하게 대응하며, MVC 패턴은 코드의 재사용과 유지보수를 용이하게 하기 위해 사용했습니다.

[참고]

개발 방법론 중 가장 널리 알려진 것은 'Agile'이나 'Scrum'입니다. Agile은 빠른 시간 내에 높은 품질의 소프트웨어를 개발하고 지속적으로 개선하는 것에 초점을 맞추며, Scrum은 Agile 개발을 위한 구체적인 프레임워크를 제공합니다. 이러한 방법론은 고객의 요구사항이 빠르게 변하는 환경에서 효과적입니다.

패턴 중에는 'MVC' (Model-View-Controller)가 널리 사용됩니다. MVC는 애플리케이션을 모델, 뷰, 컨트롤러 세 부분으로 나누어 개발하는 디자인 패턴으로, 각 부분의 역할을 명확하게 분리함으로써 코드의 재사용성과 유지보수성을 높입니다.

이러한 방법론이나 패턴을 선택하는 이유는 프로젝트의 목표, 요구사항, 팀의 구성 등 다양한 요소에 따라 달라집니다. 이들을 적절히 활용하면 개발 프로세스를 효율화하고, 코드의 품질을 향상시킬 수 있습니다.

*취업매니저 메세지

-면접관 분들은 면접자들에게 전공의 여부에 상관없이 공통적으로 Java/Spring/CS의 지식을 알고 있는지 묻는 질문을 가장 많이 합니다.

-"'A'와'B'를 비교해서 설명해보세요" 라는 요청이 많기 때문에 그에 따라 사전준비가 필요합니다.



13. 멀티쓰레드 멀티프로세스 차이를 설명하세요.

: 멀티쓰레드는 하나의 프로세스 내에서 여러 쓰레드가 동작하는 것이며, 멀티프로세스는 여러 프로세스가 동시에 실행되는 것입니다. 멀티쓰레드는 메모리 공간을 공유하지만, 멀티프로세스는 각 프로세스가 독립적인 메모리 공간을 가집니다.

14. bfs의 성능복잡도나 빅오로 표현한다면?

: bfs(Breadth-First Search)의 시간 복잡도는 $O(V+E)$ 입니다. 여기서 V 는 정점의 수, E 는 간선의 수입니다.

15. Spring MVC 패턴이 요청을 주고 받을 때 어떤 흐름인지 설명하세요.

: Spring MVC는 클라이언트의 요청을 `DispatcherServlet`이 받아 `HandlerMapping`에게 처리할 `Controller`를 물어보고, `Controller`는 로직을 처리한 후 `Model`과 `View`를 반환하며, `ViewResolver`는 `View` 이름을 바탕으로 실제 `View`를 찾아 렌더링합니다.

16. 데이터베이스 정규화가 무엇이고, 왜 하는지 설명하세요.

: 데이터베이스 정규화는 데이터의 중복을 제거하고, 데이터의 무결성을 유지하기 위해 데이터를 구조적으로 설계하는 과정입니다. 이를 통해 데이터의 추가, 수정, 삭제 시 발생할 수 있는 문제를 최소화합니다.

*취업매니저 메세지

-면접관 분들은 면접자들에게 전공의 여부에 상관없이 공통적으로 Java/Spring/CS의 지식을 알고 있는지 묻는 질문을 가장 많이 합니다.

-”A와 B를 비교해서 설명해보세요” 라는 요청이 많기 때문에 그에 따라 사전준비가 필요합니다.

