

# 면접예상질문2\_20230628

≡ 과정	java
≡ 회차	3회차
🕒 생성자	영선 김영선
■ 작성자	코리아IT아카데미_김영선강사

## 1. JVM의 구조와 동작과정을 설명해주세요

### ▼ 키워드

1. JVM 구조
2. 동작과정
3. 클래스 로더
4. 실행 엔진
5. 가비지 컬렉션
6. 메모리 영역

### ▼ 참고답안

1. JVM(Java Virtual Machine)은 자바 프로그램 실행을 위한 가상 머신으로 시스템과 하드웨어에 독립적으로 동작합니다.  
구조는 클래스로더, 실행엔진, 가비지 컬렉션, 메모리영역 등으로 나뉘어집니다
2. JVM의 동작과정은 클래스 로더를 통해 필요한 클래스 파일을 로드하면 로드한 클래스 파일을 메모리에 올리게 되고 실행엔진을 통해 바이트 코드를 해석하고 실행합니다. 프로그램이 실행되면서 동적으로 생성되는 객체는 힙영역에 할당되고 메소드 호출과 관련된 데이터는 스택영역에 저장됩니다. JVM이 종료되면 할당되는 메모리를 반환하고 프로그램 실행을 종료합니다.

### ▼ 참고내용

클래스 로더(ClassLoader) : .class 파일을 로드하고, 필요한 클래스를 동적으로 읽어 들입니다.

실행 엔진(Execution Engine) : 로드된 클래스를 해석하여 기계어로 변환하고, 실행합니다. 인터프리터와 JIT 컴파일러 등이 포함됩니다.

가비지 컬렉션(Garbage Collection): 객체의 더 이상 사용되지 않는 메모리를 자동으로 회수합니다.

메모리 영역(Memory Area): 여러 영역으로 구분되어 있으며, 메서드, 클래스, 객체 등을 저장합니다. (예: 메서드 영역, 힙 영역, 스택 영역, PC 레지스터, 네이티브 스택 등)

## 2. JAVA 코드 컴파일 과정에 대해 말해주세요

### ▼ 키워드

소스코드

컴파일

바이트코드

자바 컴파일러

자바 가상 머신(JVM)

### ▼ 참고답안

개발자가 .java 확장자를 가진 파일에 소스코드를 작성한 뒤 자바 컴파일러(javac)를 사용하여 소스코드를 컴파일합니다. 컴파일 과정을 통해 바이트 코드가 생성되고 .class 확장자를 가진 바이너리 코드로 생성됩니다. 생성된 바이트코드는 JVM에서 실행되고 이때 JVM은 런타임환경에서 동작하며 바이트코드를 읽고 해석하여 기계어로 변환하고 실행합니다.

- 꼬리질문 : 컴파일과 컴파일러의 차이를 설명하라 → 컴파일은 소스코드를 바이트 코드로 변환하는 과정을 의미하고 컴파일러는 소스코드를 구문적으로 분석하고 오류를 체크한 후, 바이트 코드를 생성함

## 3. final 키워드가 무엇인지, final을 사용하는 이유와 같이 설명해주세요

### ▼ 키워드

final

상수(Constant)

오버라이딩(Override)

상속(Inheritance)

## ▼ 참고답안

final 키워드는 Java에서 사용되는 특정한 키워드로, 변수, 메소드, 클래스에 적용될 수 있습니다.

final 키워드를 사용하는 이유는 final을 사용하여 값의 변경을 방지하고 안정성, 보안, 일관성을 제공하며, 의도하지 않은 값 변경과 버그 발생 가능성을 줄이고 프로그램의 성능을 최적화할 수 있습니다.

(참고내용을 보면서 개인적으로 답변 만들기)

## ▼ 참고내용

- 첫째로, final을 사용하여 변수를 상수(Constant)로 정의하면 값의 변경을 방지하여 안정성과 보안을 제공할 수 있습니다. 의도하지 않은 값 변경을 막아 프로그램의 예측 가능성을 높이고 버그를 예방할 수 있습니다.
- 둘째로, final을 사용하여 메소드를 정의하면 하위 클래스에서 오버라이드(재정의)를 막을 수 있습니다. final 메소드는 원본 메소드의 수정을 막아 안정성을 제공하고, 클래스의 일관성을 유지할 수 있습니다. 또한, final 메소드는 성능 최적화에도 도움을 줄 수 있습니다.
- 셋째로, final을 사용하여 클래스를 정의하면 상속을 막을 수 있습니다. 다른 클래스에서 이 클래스를 상속받을 수 없으므로 클래스의 확장을 제한하여 API 일관성을 유지하고 버그를 예방할 수 있습니다.
- 넷째로, final을 사용하여 API를 설계할 때 특정 메소드나 클래스를 수정할 수 없도록 제약을 가할 수 있어 API의 일관성을 유지하고 안정성을 확보할 수 있습니다.
- 다섯번째로, final 변수는 JIT 컴파일러에 의해 값을 상수로 인식하여 성능을 최적화할 수 있습니다.

---

final 키워드를 변수, 메소드, 클래스에 사용하면 어떤 차이가 있나?

- 첫째로, 변수에 final 키워드를 사용하면 해당 변수는 상수(Constant)로 정의됩니다. 즉, 한 번 초기화된 이후에는 값을 변경할 수 없습니다. 이를 통해 안정성과 보안을 제공할 수 있으며, 의도하지 않은 값의 변경을 방지할 수 있습니다.
- 둘째로, 메소드에 final 키워드를 사용하면 해당 메소드는 하위 클래스에서 오버라이드(재정의)할 수 없습니다. final 메소드는 상속된 클래스에서 원본 메소드의 수정을 막아 안정성을 제공하고, 성능 최적화를 도모할 수 있습니다.

- 셋째로, 클래스에 final 키워드를 사용하면 해당 클래스는 상속될 수 없습니다. 다른 클래스에서 이 클래스를 상속받을 수 없으므로 클래스의 확장을 제한하는 역할을 합니다. 이는 설계 의도를 강조하고 API의 제약을 통해 일관성을 유지할 수 있도록 도와줍니다.
- 또한, final을 사용하여 API를 설계할 때 특정 메소드나 클래스를 수정할 수 없도록 제약을 가할 수 있어 API의 일관성을 유지하고 안정성을 확보할 수 있습니다.
- final 키워드를 사용할 때에는 적절한 사용이 필요합니다. 변수나 메소드에는 필요한 경우에만 final을 사용하고, 클래스에 final을 사용할 경우 상속이 필요한 경우에는 사용하지 않는 것이 좋습니다.

### 3. static 키워드는 무엇을 의미하고 언제 사용하나요?

#### ▼ 키워드

정적(Static)

클래스 변수(Class variable)

정적 메소드(Static method)

메모리 공간(Static memory space)

객체(Object)

클래스(Class)

#### ▼ 참고답안

- static 키워드는 클래스 수준에서 사용되며, 정적(Static)이라는 의미를 가지고 있습니다. static 변수는 클래스 변수로, 객체마다 독립적인 값이 아닌 모든 객체가 공유하는 값입니다. static 메소드는 인스턴스(Object)에 속하는 것이 아니라 클래스(Class)에 속하는 메소드로, 객체의 생성 없이도 호출할 수 있습니다.
- static 키워드를 사용하는 이유는 여러가지가 있지만 몇가지를 말씀드리면
- 첫번째로, static 멤버는 객체의 생성 없이도 클래스 수준에서 접근 가능하고, 여러 객체 간에 공통된 값을 공유할 수 있습니다.
- 두번째로, static 멤버는 객체마다 독립적인 메모리 공간을 가지지 않으므로 메모리 공간을 절약하고, 객체의 생성 없이도 메소드를 호출할 수 있어 효율적인 메소드 호출이 가능합니다.

- 세번째로, static 메소드는 객체의 상태에 의존하지 않고 독립적으로 동작하는 특성을 가지며, 객체의 생성 없이도 호출할 수 있습니다.

#### ▼ 참고내용

(추가 답변) static 키워드는 여러 의미와 사용 방식을 가지고 있습니다.

- 첫째로, static 키워드는 정적(Static)을 의미합니다. 클래스 멤버인 변수와 메소드에 static 키워드를 사용할 수 있습니다.
- 둘째로, static 변수는 클래스 변수(Class variable)라고도 불리며, 클래스 수준에서 공유되는 변수입니다. 객체(Object)가 생성되기 이전에 메모리에 할당되어 프로그램 실행 동안 유지됩니다. 객체마다 독립적인 값이 아니라 모든 객체가 공유하는 값입니다.
- 셋째로, static 메소드는 인스턴스(Object)에 속하는 것이 아니라 클래스(Class)에 속하는 메소드입니다. static 메소드는 클래스 수준에서 호출되며, 객체의 생성 없이도 호출할 수 있습니다. 따라서, 객체의 상태에 의존하지 않는 독립적인 기능을 수행하는 메소드를 정의할 때 사용합니다.

---

static 키워드를 사용하는 이유도 여러가지가 있습니다

- 첫째로, static 변수와 static 메소드는 객체의 생성 없이도 클래스 자체에 속해 있기 때문에 전역적으로 접근 가능하며, 공통된 값을 공유하거나 독립적인 기능을 수행할 수 있습니다.
- 둘째로, static 멤버는 클래스 수준에 존재하기 때문에 객체가 생성되지 않아도 사용할 수 있습니다. 이를 통해 메모리 공간을 절약하고 메소드 호출의 효율을 높일 수 있습니다.
- 셋째로, static 메소드는 객체의 상태에 의존하지 않고 독립적으로 동작하는 기능을 제공합니다. 따라서, 객체의 생성 없이도 호출할 수 있고 유틸리티 메소드 등에 유용하게 사용됩니다.

## 4. static 장단점에 대해 설명해주세요

### ▼ 키워드

static

장점

단점

메모리 효율

접근 속도

유연성 저하

상속

유지보수

## ▼ 참고답안

static 키워드는 변수와 메소드에 대해 클래스에서 공유되는 고정된 값을 제공합니다.

자주 사용되는 메서드나 공통적으로 사용되어야 하는 변수를 static으로 선언하며, 시스템의 안정성과 유지 보수성을 고려하여 적용할 수 있습니다.

인스턴스 생성 없이도 접근 가능하므로 실행 속도가 빠르다는 것입니다

공통적으로 사용되는 변수나 메소드이므로 메모리 사용이 효율적인 것이 장점인 반면, 단점은 과도한 사용으로 인해 유연성이 저하되고 변경 시 영향 범위가 증가하여 유지 보수가 어려워질 수 있습니다. static 메서드는 오버라이드가 불가능하므로, 상속 관계에서 기능 추가 및 변경에 제한이 있습니다.

## 5. java의 main 메소드가 static인 이유를 안다면 설명해주세요

### ▼ 키워드

1. static
2. JVM
3. main 메소드

### ▼ 참고답안

static은 메모리 선언을 하지 않아도 사용할 수 있습니다. main 메소드는 자바가상머신(JVM)에 의해 호출되는 것이므로 반드시 static으로 선언 되어 미리 메모리에 올라가 있어야합니다. 만일 메모리에 있지 않다면 시작점(entry-point)인 main메소드를 호출하려고 할 때 메모리에 없기 때문에 실행되지 않습니다