[09. 중첩선언과 익명객체]

1. 중첩 멤버 클래스에 대한 설명으로 틀린 것은 무엇입니까? 답: 4

➊ 인스턴스 멤버 클래스는 바깥 클래스의 객체가 있어야 사용될 수 있다.

➋ 정적 멤버 클래스는 바깥 클래스의 객체가 없어도 사용될 수 있다.

➌ 인스턴스 멤버 클래스 내부에는 바깥 클래스의 모든 필드와 메소드를 사용할 수 있다.

➍ 정적 멤버 클래스 내부에는 바깥 클래스의 인스턴스 필드를 사용할 수 있다.

2. 로컬 클래스에 대한 설명으로 틀린 것은 무엇입니까? 답: 3

➊ 로컬 클래스는 메소드 내부에 선언된 클래스를 말한다.

➋ 로컬 클래스는 바깥 클래스의 필드와 메소드를 사용할 수 있다.

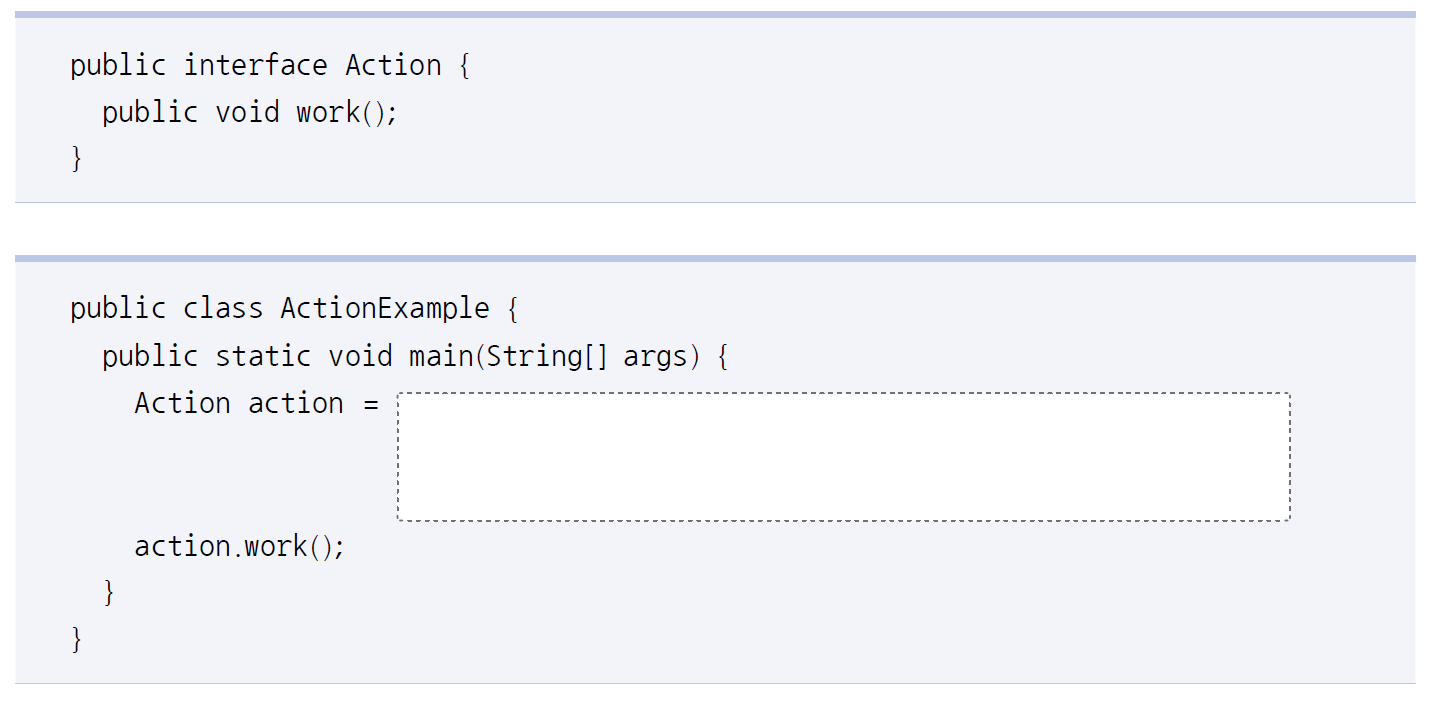
➌ 로컬 클래스는 static 키워드를 이용해서 정적 클래스로 만들 수 있다.

➍ final 특성을 가진 매개변수나 로컬 변수만 로컬 클래스 내부에서 사용할 수 있다.

3. Action 인터페이스는 다음과 같이 work( ) 추상 메소드를 가지고 있습니다. ActionExample

클래스의 main( ) 메소드에서 Action의 익명 구현 객체를 만들어 실행 결과와 동일하게 나오도록

박스 안에 들어갈 코드를 작성해보세요.



답:

public class ActionExample {

public static void main(String[] args) {

Action action = new Action() {

@Override

public void work() {

System.out.println("복사를 합니다.");

}

};

action.work();

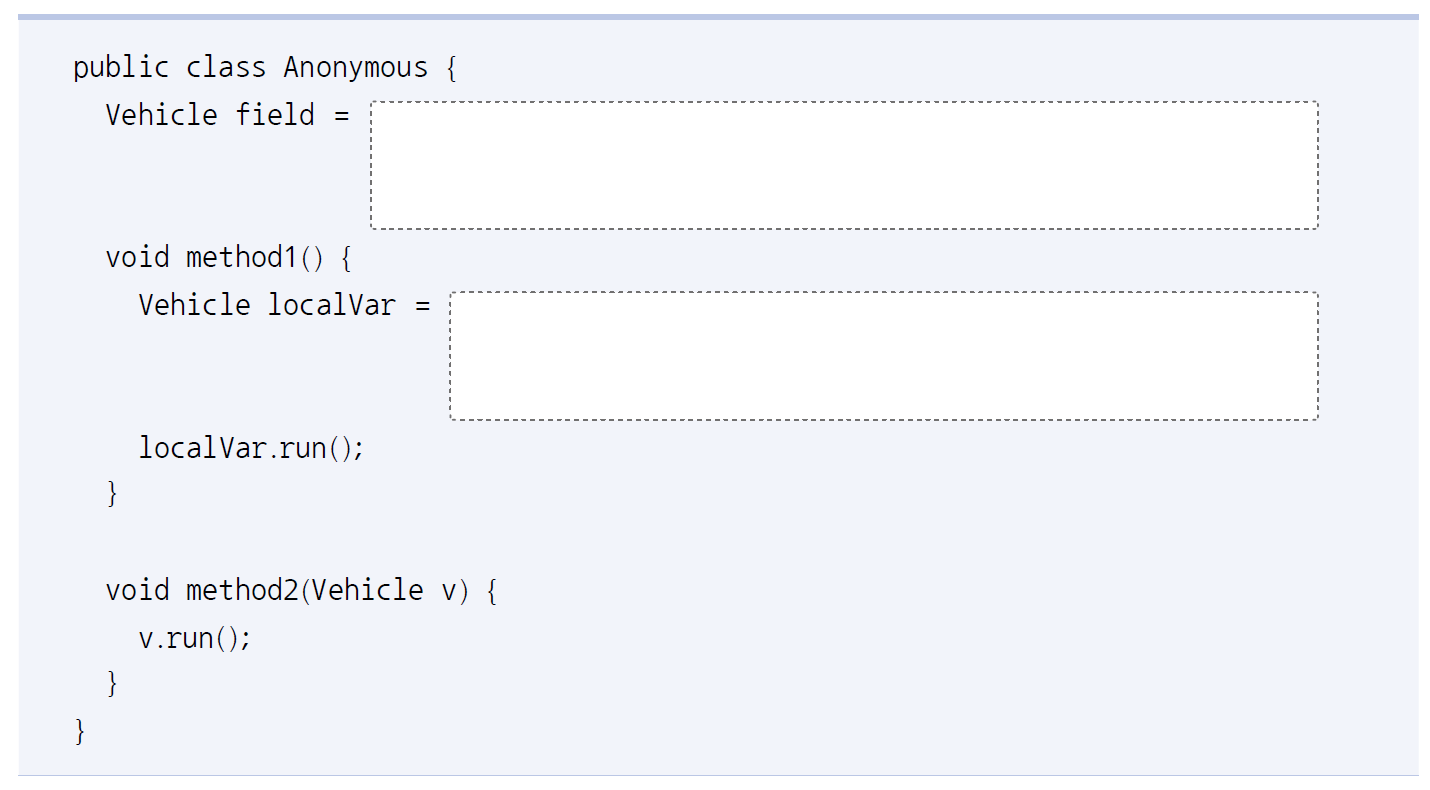
}

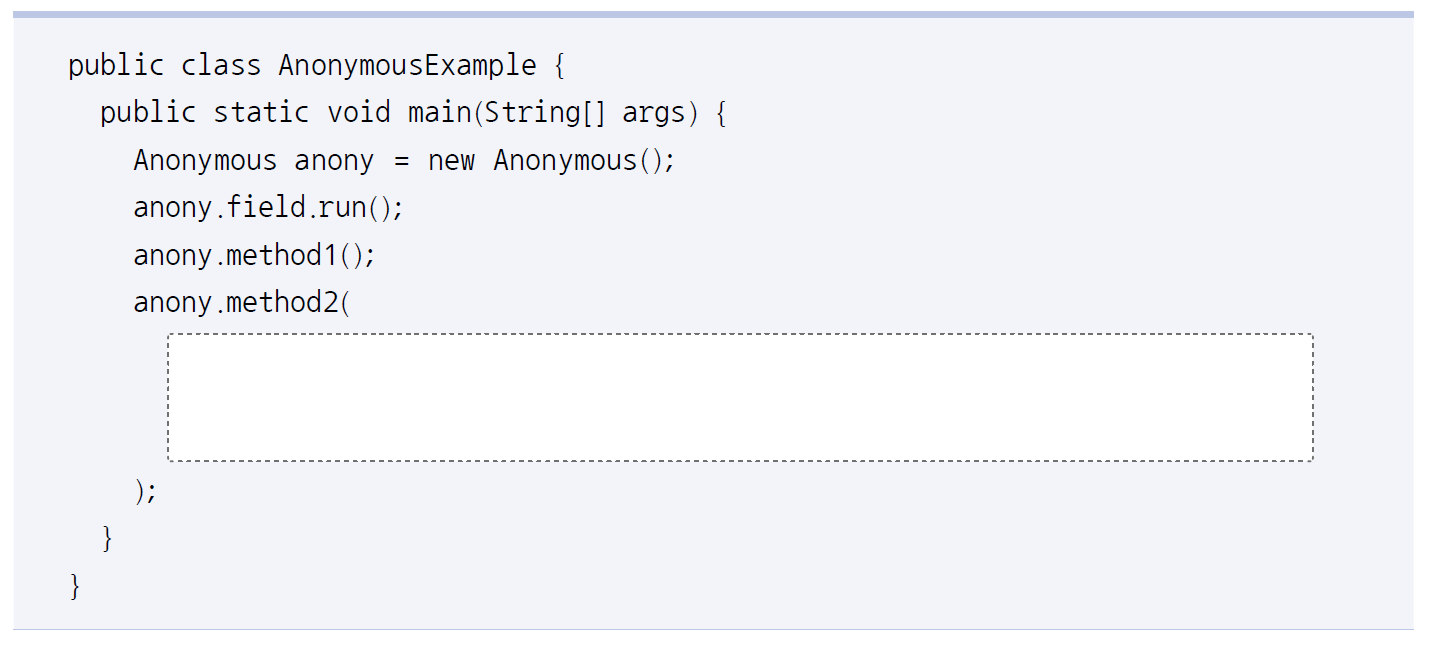
}

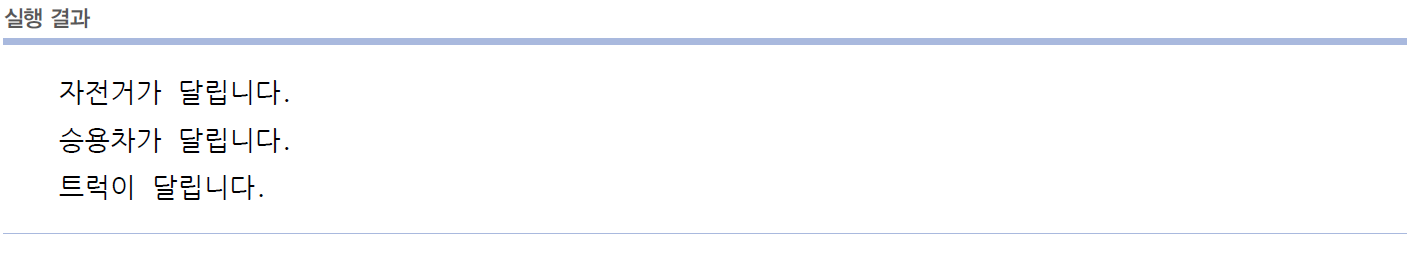
4. AnonymousExample 클래스의 실행 결과를 보고, Vehicle 인터페이스의 익명 구현 객체를

필드와 로컬 변수의 초기값 그리고 메소드의 매개값으로 대입해보세요.









답:

| public class Anonymous {  Vehicle field = new Vehicle() {  @Override  public void run() {  System.out.println("자전거가 달립니다.");  }  };  void method1() {  Vehicle localVar = new Vehicle() {  @Override  public void run() {  System.out.println("승용차가 달립니다.");  }  };  localVar.run();  }  void method2(Vehicle v) {  v.run();  }  } |
| --- |

| public class AnonymousExample {  public static void main(String[] args) {  Anonymous anony = new Anonymous();  //익명 객체 필드 사용  anony.field.run();  //익명 객체 로컬변수 사용  anony.method1();  //익명 객체 매개값 사용  anony.method2(new Vehicle() {  @Override  public void run() {  System.out.println("트럭이 달립니다.");  }  });  }  } |
| --- |