

1. Vehicle 인터페이스의 익명 구현 객체를 이용해 AnonymouseExample 클래스의 필드, 로컬 변수의 초기값, 메서드의 매개값을 대입해 프로그램을 완성하라. 단, 람다식 사용하지 말 것.

<Vehicle.java>

```
public interface Vehicle {  
    public void run();  
}
```

<실행결과>

```
자전거가 달립니다.  
승용차가 달립니다.  
트럭이 달립니다.
```

```
public class Anonymouse {  
    Vehicle field = ;  
  
    void method1() {  
        Vehicle localVar = ;  
        localVar.run();  
    }  
  
    void method2(Vehicle v) {  
        v.run();  
    }  
}
```

```
public class AnonymouseExample {  
  
    public static void main(String[] args) {  
        Anonymouse anony = new Anonymouse();  
        anony.field.run();  
        anony.method1();  
        anony.method2();  
    }  
}
```

2. 실행 결과를 참고하여 MathOperation의 구현 객체를 생성해 프로그램을 완성하라. 단, 주석의 요구에 따라 랴다식을 사용하라.

<실행 결과>

```
10 + 5 = 15
10 - 5 = 5
10 x 5 = 50
10 / 5 = 2
Hello
Java
```

```
public class LambdaEx {
    public static void main(String args[]) {
        LambdaEx tester = new LambdaEx();

        //매개 변수 타입 명시, 중괄호와 return 사용
        MathOperation addition = [ ];

        //매개 변수 타입 생략, 중괄호와 return 생략
        MathOperation subtraction = [ ];

        //매개 변수 타입 명시, 중괄호와 return 생략
        MathOperation multiplication = [ ];

        //매개변수 타입 생략, 중괄호와 return 사용
        MathOperation division = [ ];

        System.out.println("10 + 5 = " + tester.operate(10, 5, addition));
        System.out.println("10 - 5 = " + tester.operate(10, 5, subtraction));
        System.out.println("10 x 5 = " + tester.operate(10, 5, multiplication));
        System.out.println("10 / 5 = " + tester.operate(10, 5, division));

        //기본 랴다식 사용
        GreetingService greetService1 = message -> [ ];

        //람다식 축약형 메서드 사용
        GreetingService greetService2 = [ ];

        greetService1.sendMessage("Hello");
        greetService2.sendMessage("Java");
    }

    interface MathOperation {
        int operation(int a, int b);
    }

    interface GreetingService {
        void sendMessage(String message);
    }

    private int operate(int a, int b, MathOperation mathOperation) {
        return mathOperation.operation(a, b);
    }
}
```