

2024년 상반기 K-디지털 트레이닝

람다식

[KB] IT's Your Life



♥ 다음 인터페이스를 함수형 인터페이스로 명시하세요.

```
package ch16.sec01;

public interface Calculable {
   //추상 메소드
   void calculate(int x, int y);
}
```

Calculable.java

```
package ch16.sec01;

@FunctionalInterface
public interface Calculable {
   //추상 메소드
   void calculate(int x, int y);
}
```

- ❷ 앞의 Calculator 인터페이스를 이용해서, 다음 출력이 나오도록 다음 코드를 완성하세요.
 - o 단, 람다식을 이용함

```
package ch16.sec01;
public class LambdaExample {
 public static void main(String[] args) {
   // x + y 계산
   // x - y 계산
 public static void action(Calculable calculable) {
   //데이터
   int x = 10;
   int y = 4;
   //데이터 처리
   calculable.calculate(x, y);
```

```
result: 14
result: 6
```

LambdaExample.java

```
package ch16.sec01;
public class LambdaExample {
 public static void main(String[] args) {
   action((x, y) \rightarrow \{
     int result = x + y;
     System.out.println("result: " + result);
   });
   action((x, y) \rightarrow {
     int result = x - y;
     System.out.println("result: " + result);
   });
 public static void action(Calculable calculable) {
   //데이터
   int x = 10;
   int y = 4;
                                                      result: 14
   //데이터 처리
                                                      result: 6
   calculable.calculate(x, y);
```

♥ 다음과 같이 Workable 인터페이스와 Person 클래스를 정의하세요.

```
package ch16.sec02.exam01;

@FunctionalInterface
public interface Workable {
  void work();
}
```

```
package ch16.sec02.exam01;

public class Person {
   public void action(Workable workable) {
     workable.work();
   }
}
```

앞에서 정의한 인터페이스와 클래스를 이용하여 아래 코드를 완성하세요.

```
package ch16.sec02.exam01;
public class LambdaExample {
 public static void main(String[] args) {
   Person person = new Person();
   //실행문이 두 개 이상인 경우 중괄호 필요
   person.action(
   );
   //실행문이 한 개일 경우 중괄호 생략 가능
   person.action(
```

```
출근을 합니다.
프로그래밍을 합니다.
퇴근합니다.
```

LambdaExample.java

```
package ch16.sec02.exam01;
public class LambdaExample {
 public static void main(String[] args) {
   Person person = new Person();
   //실행문이 두 개 이상인 경우 중괄호 필요
   person.action(() -> {
    System.out.println("출근을 합니다.");
    System.out.println("프로그래밍을 합니다.");
   });
   //실행문이 한 개일 경우 중괄호 생략 가능
   person.action(() -> System.out.println("퇴근합니다."));
```

```
출근을 합니다.
프로그래밍을 합니다.
퇴근합니다.
```

다음과 같이 인터페이스와 클래스를 정의하세요.

```
package ch16.sec03;
@FunctionalInterface
public interface Workable {
 void work(String name, String job);
package ch16.sec03;
@FunctionalInterface
public interface Speakable {
 void speak(String content);
package ch16.sec03;
public class Person {
 public void action1(Workable workable) {
   workable.work("홍길동", "프로그래밍");
 public void action2(Speakable speakable) {
   speakable.speak("안녕하세요");
```

앞에서 정의한 인터페이스와 클래스를 이용하여 아래 코드를 완성하세요.

```
package ch16.sec03;
public class LambdaExample {
 public static void main(String[] args) {
   Person person = new Person();
   //매개변수가 두 개일 경우
   person.action1(
   person.action1(
   //매개변수가 한 개일 경우
   person.action2(
   person.action2(
```

```
홍길동이 프로그래밍을 합니다.
홍길동이 프로그래밍을 하지 않습니다.
"안녕하세요"라고 말합니다.
"안녕하세요"라고 외칩니다.
```

LambdaExample.java

```
package ch16.sec03;
public class LambdaExample {
 public static void main(String[] args) {
   Person person = new Person();
   //매개변수가 두 개일 경우
   person.action1((name, job) -> {
    System.out.print(name + "0| ");
    System.out.println(job + "을 합니다.");
   });
   person.action1((name, job) -> System.out.println(name + "이 " + job + "을 하지 않습니다."));
   //매개변수가 한 개일 경우
   person.action2(word -> {
    System.out.print("\"" + word + "\"");
    System.out.println("라고 말합니다.");
   });
   person.action2(word -> System.out.println("\"" + word + "\"라고 외칩니다."));
       홍길동이 프로그래밍을 합니다.
       홍길동이 프로그래밍을 하지 않습니다.
       "안녕하세요"라고 말합니다.
       "안녕하세요"라고 외칩니다.
```

☑ 다음과 같이 Calcuable 인터페이스와 Person 클래스를 정의하세요.

```
package ch16.sec04;

@FunctionalInterface
public interface Calcuable {
   double calc(double x, double y);
}
package ch16.sec04;
```

```
public class Person {
  public void action(Calcuable calcuable) {
    double result = calcuable.calc(10, 4);
    System.out.println("결과: " + result);
  }
}
```

앞에서 정의한 인터페이스와 클래스를 이용하여 아래 코드를 완성하세요.

```
package ch16.sec04;
public class LambdaExample {
 public static void main(String[] args) {
   Person person = new Person();
   //실행문이 두 개 이상일 경우
   person.action(
   // sum() 메서드를 호출
   person.action( );
 public static double sum(double x, double y) {
   return (x + y);
```

결과: 14.0 결과: 14.0

LambdaExample.java

```
package ch16.sec04;
public class LambdaExample {
 public static void main(String[] args) {
   Person person = new Person();
   //실행문이 두 개 이상일 경우
   person.action((x, y) -> {
     double result = x + y;
     return result;
   });
   person.action((x, y) \rightarrow sum(x, y));
 public static double sum(double x, double y) {
   return (x + y);
```

결과: 14.0 결과: 14.0 결과: 14.0