

수행 테스트 (추상클래스,인터페이스)

3/29

이름 :

닉네임 :

1. 추상클래스와 인터페이스의 공통점과 차이점을 1개 이상씩 적으시오. (2점)

공통점 : 추상메서드를 가질 수 있음. 인스턴스화(new로 객체화) 할 수 없음.

차이점 : 다중상속여부. 인터페이스는 다중상속 가능.

추상클래스는 일반메서드 가질 수 있음. 인터페이스는 모두 추상메서드

추상클래스는 일반 필드를 가질 수 있음. 인터페이스는 **static final** 필드만 가짐.

2. 아래 코드에서 오류가 발생하는 위치와 그 이유를 적으시오.

```
class A {  
    int a = 3;      ①  
    final int b = 5; ②  
    A() {  
        a = 7;      ③  
        b = 9;      ④  
    }  
}
```

4번 : final 변수(=상수)이므로 값의 변경 불가

3. 아래 코드에서 오류가 발생하는 곳의 코드를 수정하시오.

```
abstract class Animal {  
    abstract void cry() {      ①  
    }  
}  
class Cat extends Animal {  
    void cry() {  
        System.out.println("야옹"); ②  
    }  
}  
class Dog extends Animal {  
    void cry() {  
        System.out.println("멍멍"); ③  
    }  
}
```

1번 : 추상메서드는 미완성이어야함. 중괄호도 없어야함
abstract void cry();

4. 네모안에 상속을 수행하는 적당한 키워드를 적으시오.

```
abstract class A {  
    abstract void abc();  
}  
class B extends A {  
    @Override  
    void abc() {  
        System.out.println("테스트");  
    }  
}
```

5. 네모안에 상속을 수행하는 적당한 키워드를 적으시오.

```
interface A {  
    public abstract void abc();  
}  
class B implements A {  
    @Override  
    public void abc() {  
        System.out.println("테스트");  
    }  
}
```

6. 아래 1) 2) 3)은 예외가 발생하는 코드입니다. 각 오류에 해당하는 예외를 보기에서 골라 옆에 각각 적으시오.

[보기]

InterruptedException

ClassNotFoundException

IOException

ArrayIndexOutOfBoundsException

NumberFormatException

NullPointerException

ArithmeticException

- 1) int num = Integer.parseInt("10!"); **NumberFormatException**
2) String a = null;
 System.out.println(a.charAt(2)); **NullPointerException**
3) int[] a = {1,2,3};
 System.out.println(a[3]); **ArrayIndexOutOfBoundsException**

7. 아래의 try catch 구문의 중복된 코드를 finally 구문으로 옮겨서 try catch finally 구문을 완성하시오.

```
try {  
    System.out.println(3/0);  
    System.out.println("프로그램 종료");  
}  
catch(ArithmeticException e) {  
    System.out.println("숫자는 0으로 나눌 수 없습니다.");  
    System.out.println("프로그램 종료");  
}
```



```
try {  
    System.out.println(3/0);  
}  
catch(ArithmeticException e) {  
    System.out.println("숫자는 0으로 나눌 수 없습니다.");  
}  
finally {  
    System.out.println("프로그램 종료");  
}
```

8. 아래 코드의 실행 순서대로 숫자를 적으시오.

```
try {
    System.out.println(3/1);           ①
    int a = Integer.parseInt("20A");   ②
    int[] arr = {1,2,3,4,5};           ③
    arr[5] = 6;                         ④
    System.out.println("프로그램 종료1"); ⑤
} catch (ArithmeticException e) {
    System.out.println("숫자는 0으로 나눌 수 없습니다"); ⑥
    System.out.println("프로그램 종료2"); ⑦
} catch (NumberFormatException e) {
    System.out.println("숫자로 변환할 수 없습니다"); ⑧
} catch (NullPointerException e) {
    System.out.println("Null 관리 좀 잘하자"); ⑨
}
finally {
    System.out.println("프로그램 종료3"); ⑩
}
```

1 - 2 - 8 - 10

9. 사용자 정의 예외 클래스를 선언할때 상속받아야 하는 클래스의 이름을 적으시오.

Exception / RuntimeException

10. 아래 코드를 예외 전가를 이용하여 예외가 발생하는 abc()는 예외를 처리하지 않고 bcd 메소드가 해당 예외를 처리하도록 코드를 수정하시오. (2점)

```
void abc(int num) {
    try {
        if(num>70)
            System.out.println("정상작동");
        else
            throw new MyException();
    } catch (MyException e) {
        System.out.println("예외처리 1");
    }
}
void bcd() {
    abc(65);
}
```



```
void abc(int num) throws MyException {
    if(num>70)
        System.out.println("정상작동");
    else
        throw new MyException();
}
void bcd() {
    try {
        abc(65);
    } catch (MyException e) {
        System.out.println("예외처리 1");
    }
}
```