1. 인터페이스

* 추상클래스의 발전형
* 모든 메소드가 추상메소드로 선언돼야 함
* 인터페이스의 필드는 값의 변경이 불가능한 상수만 선언 가능
* 인터페이스의 선언 interface 키워드를 이용해서 선언

public interface 인터페이스 명 {

}

* 인터페이스의 상속은 implements 키워드 사용

public Class 클래스명 implements 인터페이스명 {

}

* 인터페이스에서 추상메소드의 선언은 abstract 키워드를 생략한 상태로 선언
* 인터페이스에서 필드는 상수기 때문에 final 키워드를 생략한 상태로 선언
* 상속 > 인터페이스 상속

1. 추상클래스와 인터페이스의 차이

* 추상클래스에는 공통된 변수와 메소드의 선언 및 정의가 가능
* 인터페이스에는 상수필드만 지정 가능, 추상메소드만 생성 가능(정의 불가능)
* 추상클래스는 단일상속만 지원
* 인터페이스는 다중상속 가능

public interface Father {

}

public interface Mother {

}

public Class Child implements Father, Mother {

}

1. 예외처리

* 예외 : 개발자의 잘못된 코드나 조작으로 발생하는 프로그램의 오류
* 예외처리를 할 경우 오류가 발생해도 프로그램 정상적으로 동작
* 예외의 종류

일반 예외 : 프로그램 실행 시 발생하는 예외

소스코드에서 오류로 잡힘

ClassNotFoundException, InterruptedException….

런타임 예외 : 프로그램 실행 시 예측할 수 없는 예외

소스코드에서 오류로 잡히지 않음

OutOfMemoryException, NullPointerException….

* NullPointerException : null값을 갖는 변수에 접근할 때 발생하는 예외
* ArrayIndexOutOfBoundException : 배열의 존재하지 않는 인덱스에 접근 시 발생하는

예외

* NumberFormatException : 문자열을 정수나 실수로 치환할 때 치환불가능한 문자열

포함 시 발생하는 예외

* ClassCastException : 한 객체를 타입변환이 불가능한 클래스로 타입변환 시도 시에

발생하는 예외

* 예외처리 방식

try~catch~finally 블록으로 예외처리하는 방식

throws 키워드를 이용해서 예외를 떠넘기는 방식

* try~catch~finally 블록

try {

실행코드

} catch(NullPointerException ex) {

} catch(ClassCastException ex) {

} finally {  
 예외가 발생하거나 발생하지 않아도 항상 실행될 코드

}

* throws 키워드

메소드선언부에 메소드명 뒤에 throws 키워드를 사용하여 예외처리

public int add(int a, int b) throws 예외클래스1, 예외클래스2 …. {

}

* 메소드 생성 시에 throws로 예외를 떠넘기고 메소드를 호출하는 곳에서

try~catch~finally 블록으로 예외처리를 해주는 것이 일반적인 예외처리 방식

1. java.lang 패키지

* 자바 프로그램의 기본적인 클래스를 담고있는 패키지
* 이 패키지에 포함돼있는 클래스들을 사용할 때는 import 구문 선언 불필요
* Object : 자바 클래스의 최상위 클래스

생성되는 모든 클래스들은 Object 클래스를 상속받도록 설계되어 있음

* System : 입력장치로부터 입력받을 때나 출력장치로 출력할 때 사용하는 클래스

JVM(컴파일러)를 종료할 때 사용

* Class : 클래스를 메모리에 로딩하여 정보를 얻을 때 사용
* String : 문자열 정보를 저장하거나 저장된 문자열을 편집, 조작할 때 사용
* Wrapper : 기본타입의 데이터를 갖는 객체를 만들 때 사용

문자열을 기본타입으로 변환할 때 사용

Integer, Double, Float, Byte, Short, Character ....

* Math : 수학적 계산이 필요한 메소드를 호출할 때 사용

1. Object 클래스

* equals 메소드 : 두 객체가 같은지 다른지 비교하는 메소드

같으면 true, 다르면 false 리턴

* toString() 메소드 : 객체의 문자정보 리턴해주는 메소드

1. System 클래스

* exit() 메소드 : 현재 실행중인 프로세스 강제 종료하는 메소드
* currentTimeMills(), nanoTime() : 시스템의 현재 시각을 밀리초와 나초로 치환하여

리턴하는 메소드