\*10chapter , 이것이자바다 p422

오류의 종류

에러(Error) : 컴퓨터 하드웨어의 오동작 또는 고장으로 응용프로그램 실행 오류가 발생하는 것

(JVM실행에 문제가 생겼다는것으로 아무리 견고히만들어도 결국 실행불능)

예외(Exception) : 사용자의 잘못된조작 또는 잘못된 코딩으로 발생함 ->Exception Handling을 통해 예외가 발생하여도 종료하지않고 정상 실행상태가 유지되도록 할수있다

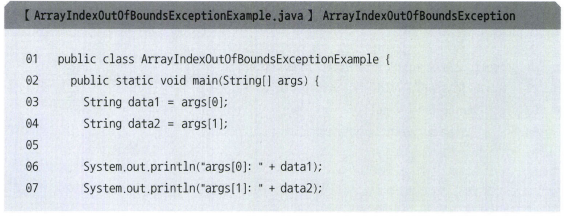
-일반예외 Exception : 컴파일러 체크예외 자바소스를 컴파일 하는 과정에서 예외처리 코드가 필요하는지 검사 -> Exception 클래스를 상속받음

-실행 예외 Runtime Exception : 예외 처리코드를 생략 하더라도 컴파일이 되는 예외 -> RuntimeException을 상속받은 클래스들

-> Exception, RuntimeException 둘다 예외처리를 통해 정상 실행 상태로 돌아갈 수 있음

-NullPointerException 객체 참조가 없는 상태: 해당 객체 참조가 null값인경우나 초기화 시켜주지 않았는데 사용하는 경우 NullPointerException이 발생한다

-ArrayIndexOutOfBoundsException 배열에서 인덱스 범위를 초과하여 사용할경우 발생



두개의 실행 매개값을 주지않았기 때문에 args[0],args[1]의 매개값을 사용할수없어 발생한다

-NumberFormatException : Integer.parseInt(String s), Double.paresDouble(String s)로 각각의문자열을 정수 실수로 변환시켜준다 Integer과 Double은 포장(Wrapper)클래스로 각각의 정적 매소드로 실행시킬수있다 이때 문자열에 숫자로 변환할 수 없는 값이 포함되어있다면 NumberFormatException을 발생시킨다

-ClassCastException 타입변환이 되지 않을 경우에 발생

Ex) 추상클래스 Animal과 Animal을 상속받는 Dog,Cat 클래스가있는경우(Interface도 마찬가지)

객체의 다형성에 의해 Animal animal = new Dog(); ->

Dog dog = (Dog)animal 은 실행되지만

Cat cat = (Cat)animal은 타입변환이 되지않아 ClassCastException이 발생된다

Ex) Class 객체명 = Class.forName(“java.lang.String2”); 해당경로 패키지에 해당클래스가 존재하면 객체생성-> catch(ClassNotFoundException e)로 예외 처리

Finally 부분은 예외가 발생하든 안하든 실행

다중 Catch순서 상위클래스가 나중에 작성되어야한다

-public void method() throws ClassNotFountException{} 하면 method가 실행될 때 예외처리

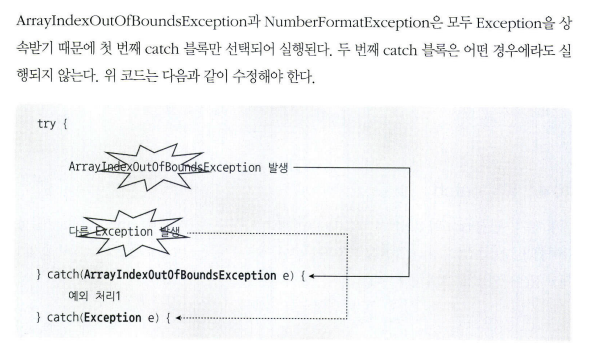
-String 의 단점 Stirng s =의값을 여러 번 바꾸게되면 쓰레기값이 여러 개 생긴다

-> 가장많이 쓰는 메소드 equals()값을비교

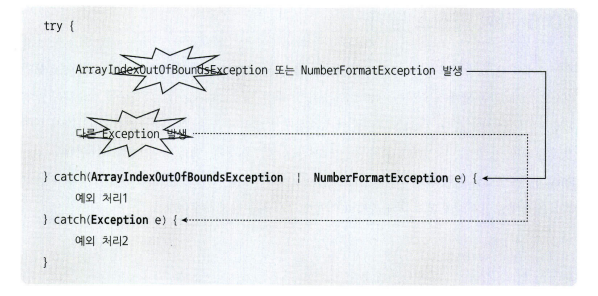
-예외 처리 코드 try- catch-finally 각각의 try{}문안에는 예외가 발생할 코드, catch(에러코드 e){}안에는 예외발생시 처리코드, finally{}안에는 예외발생여부와 상관없이 항상실행되며 생략가능하다

->실제 코드작성시 빨간밑줄에 (Unhandled exception 에러코드) 오류로 발생하며 해당오류는 try catch문으로 예외처리하여 정상 실행시킬수있다

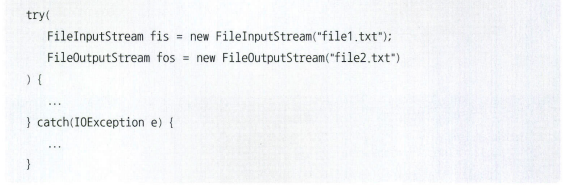
-다중 catch : try{}안에 코드가 여러 예외가 발생할경우 다중 catch로 잡아줘야하는데 먼저 쓰여지는 catch에서 상위예외클래스로 잡을경우 하단의 catch에서 에러를 잡지못하기 때문에 하위클래스의 예외를 먼저 실행해준뒤 하단catch에 상위예외클래스를 잡아줘야한다



-멀티 catch -> 아래와같이 | or연산자를 사용하여 멀티 catch 기능을 추가할수있다(자바7부터 사용가능)



-자동 리소스닫기 try-with-resource 다중으로도 사용가능하며 try( )가실행되면 자동으로 close()를호출시킬 코드를()안에 작성하고, 리소스 객체는 java.lang.AutoClseable 인터페이스를 구현하고있어야하여 해당클래스에 implements 시켜줘야한다 , 예외 발생여부와 상관없이 close()메소드를 호출한다 🡪각종 입출력스트림, 서버소켓, 소켓, 각종 채널을 사용할 때 사용한다



-throw new Exception(); or throw new Exception(“메시지”)를 명시하면 예외를 강제적으로 발생시킬수있다

-예외 떠넘기기(throws) 보통 메소드 내부에서 try-catch블록으로 예외처리를 바로하지만 메소드를 호출한곳으로 떠넘길수있다 -> 리턴타입 메소드면(매개변수) throws 예외클래스1, 예외클래스2 : 떠넘길 예외클래스1,예외클래스2를 나열하여 사용한다

or 리턴타입 메소드면(매개변수) throws Exception{} Exception을 사용하여 모든예외를 떠넘기며

받은 클래스에서는 발생한 모든예외에 대하여 try-catch하여 작성하여야한다

-catch(Exception e){

String message = e.getMessage(); // getMessage()를 사용하여 메소드의 리턴값을 얻어 messaged 저장시킨후 필요시 출력하여 에러값을 없을수있다

e.printStackTrace(); // printStackTrace();를 사용하여 예외발생 코드를 추적하여 모두 콘솔에 출력할수있다

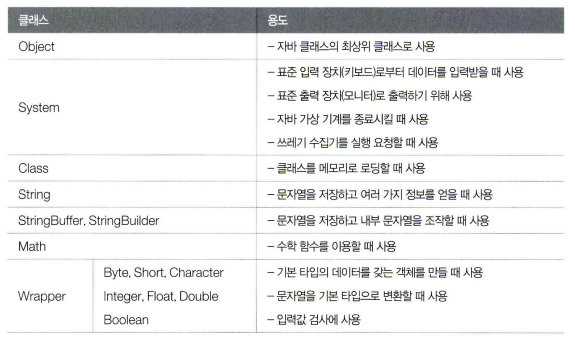
}

\*11 chapter 기본API클래스 p454~p516

API는 라이브러리(library)라고 부르기도 하는데 프로그램 개발에 자주 사용되는 클래스 및 인터페이스 모음을 말한다

<https://docs.oracle.com/en/java/javase/11/docs/api/> 와같이 API Document는 API를 쉽게 찾아 이용할수 있도록 문서화 한 것을 말한다

-java.lang 패키지



-java.util 패키지

