예외처리(exception handling)

- 프로그램의 안정화를 위해 사용

- 예외 : 코드상에서 보완하여 프로그램을 실행 가능

- 에러 : 코드상에서 보완이 불가(코드의 수정이 필요)

finally

- 예외의 발생여부와 관계없이 무조건 실행된다

throw

- 메서드에서 예외가 발생했을 때 메서드 밖에서 처리하기 위해 사용

- 예외 발생시 throw new 예외객체(); 를 통해 예외를 내보낸다

- throw가 있는 메서드는 메서드 선언부에 throws를 통해 어떤 예외가 발생할 수 있는지 명시해야 한다

- 다만 RuntimeException 예외클래스들은 명시하지 않아도 된다

-> 따로 예외처리를 하지 않아도 실행이 가능하기 때문

toString()

- 문자열이 필요한 부분에 객체를 넘겨주면 해당 객체의 toString 메서드가 자동으로 호출된다

wrapper 클래스

- 일반 자료형을 클래스로 만든 것

- 래퍼 클래스는 각 자료형에 대한 값 + null으로 초기화할 수 있다(일반 자료형으로는 불가능)

- 문자열을 다른 자료형으로 바꾸는 방법

Integer.parseInt(문자열);

Float.parseFloat(문자열); ....

wrapper 클래스 - 8개의 기본형 값을 객체로 다룰 수 있는 클래스

- 기본형 변수를 매개변수의 객체로 요구할 때 사용

- 기본형 값이 아닌 개체로 저장해야 할 때 사용

- null값을 넣을 수 있다