

Revisit Java Programming

(Method and Exception Handling)

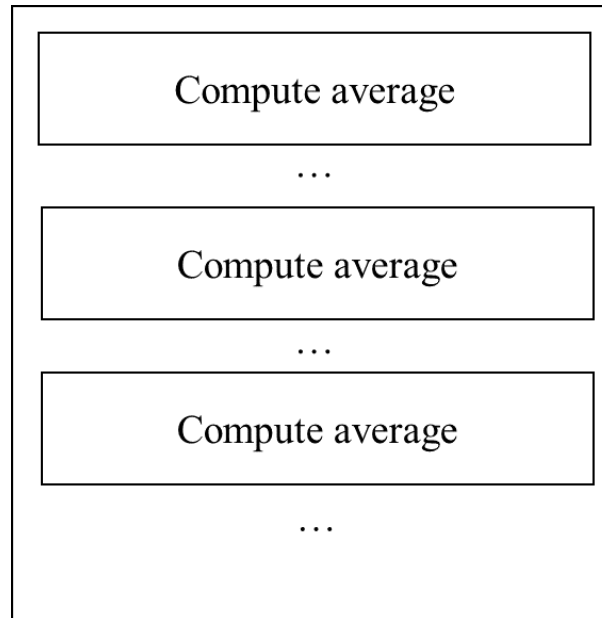
Fall, 2020

Table of Contents

- Method
- Main Method
- Exception Handling

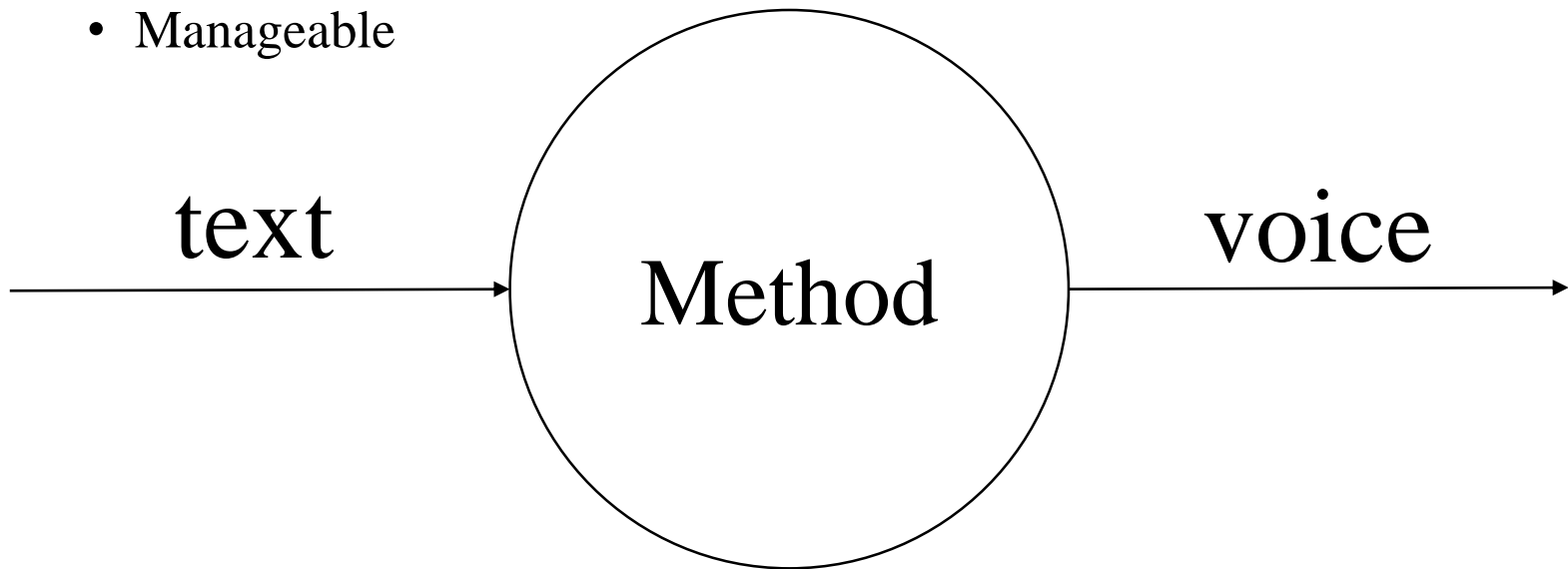
Method

- Motivation
 - Your program needs a logic to compute average of int array
 - Will you write a code to compute average whenever necessary?
 - Verbose
 - What if some logic has an error?
 - ...



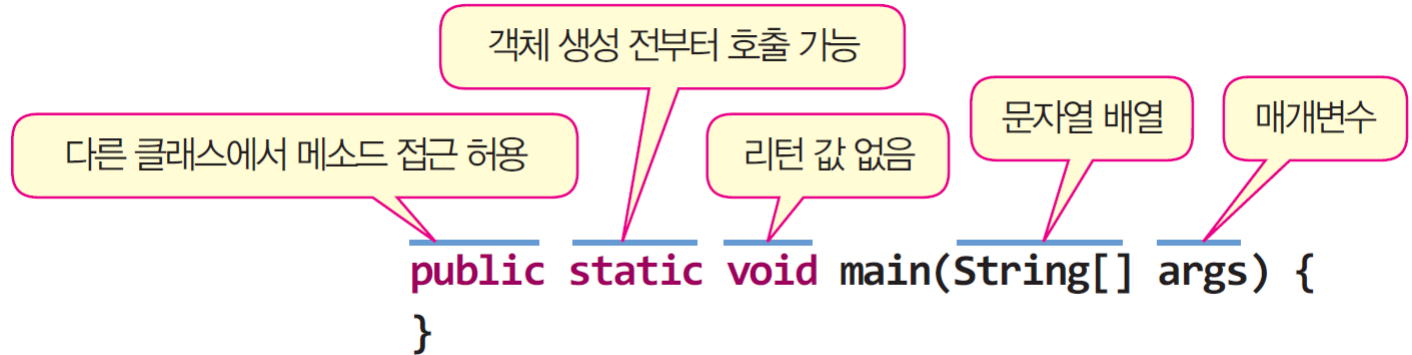
Method

- Motivation
 - Method is an abstraction of a certain procedure
 - E.g., compute average
 - compute summation
 - Text-to-speech
 - Method produces output from inputs
 - Reusable
 - Manageable



Method

- Structure



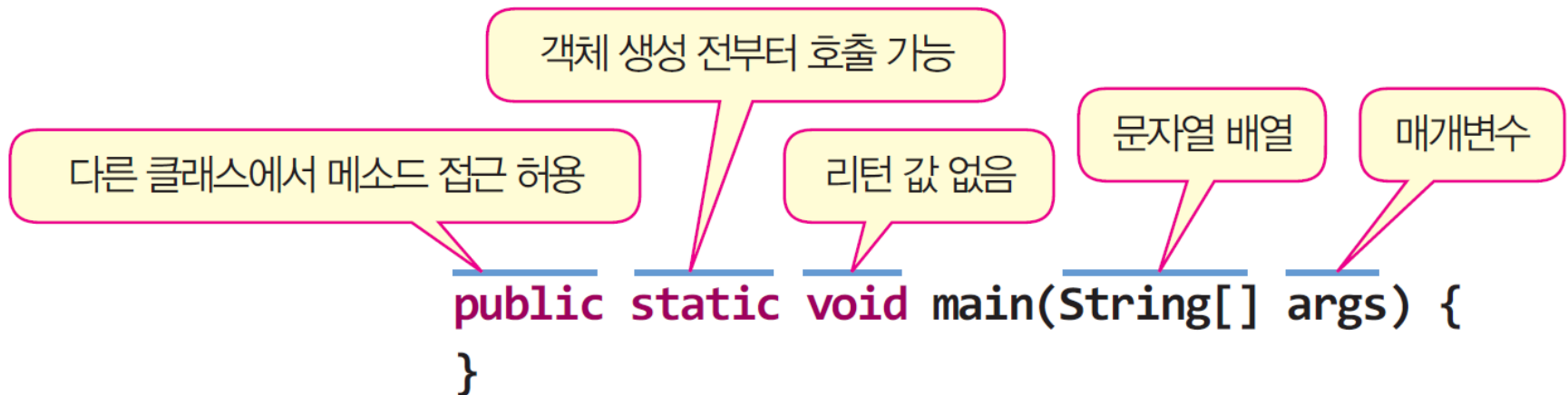
- Write a method
 - `public static int sum(int[] array){`
 - `int sum = 0;`
 - `for(int val: array){`
 - `Sum += val;`
 - `}`
 - `return sum;`
 - `}`
- Call the method
 - `sum(intArray);`

Practice 1

- Write and call the following methods
 - `public static int sum(int[] array){ }`
 - `public static double average(int[] array){ }`
 - `public static int max(int[] array){ }`
 - `public static int min(int[] array){ }`
 - `public static int[] sort(int[] array){ }`

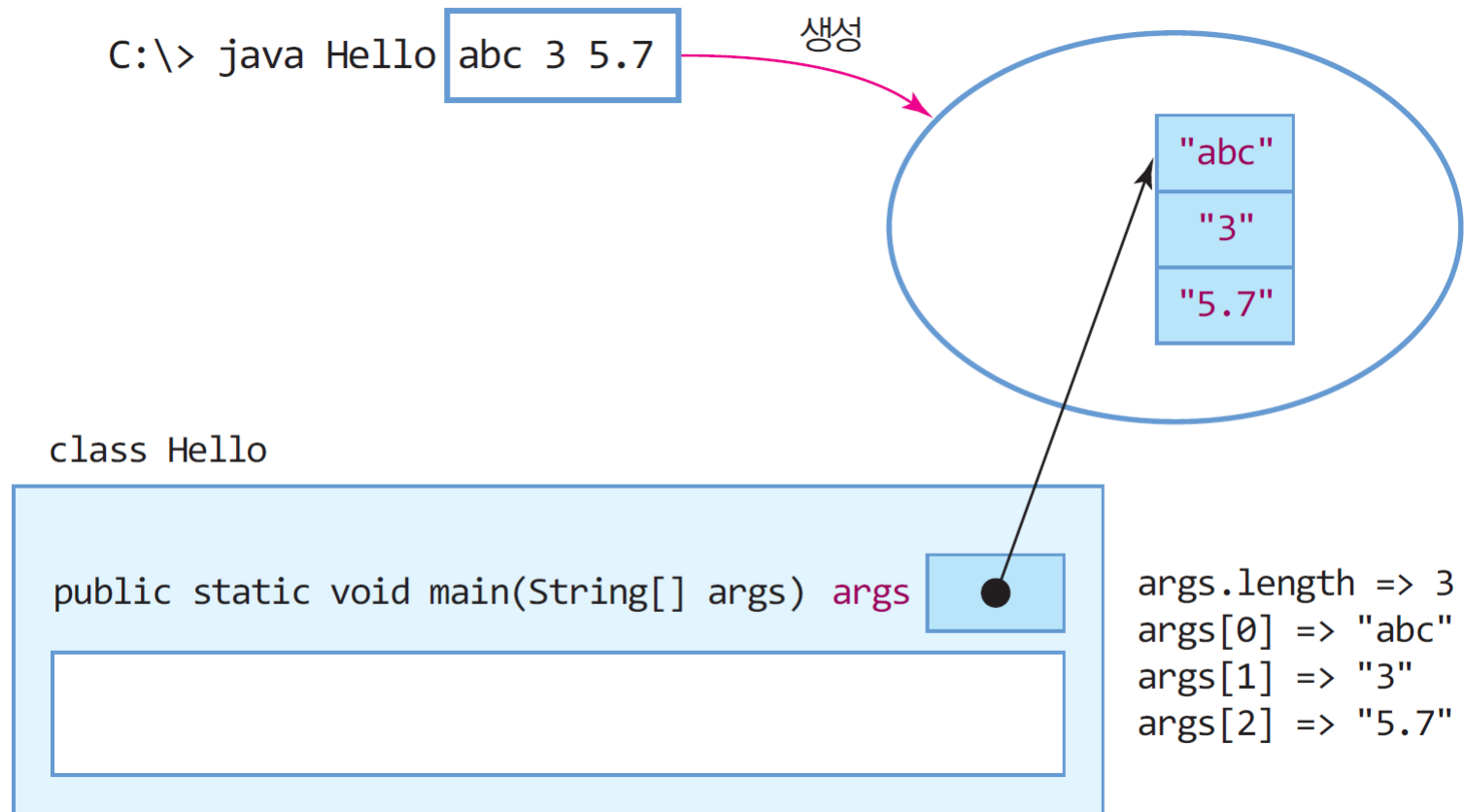
Method: main()

- main() is a starting point of your java application
- Should be public static void main(String[] args)
 - args[] from outside of your application



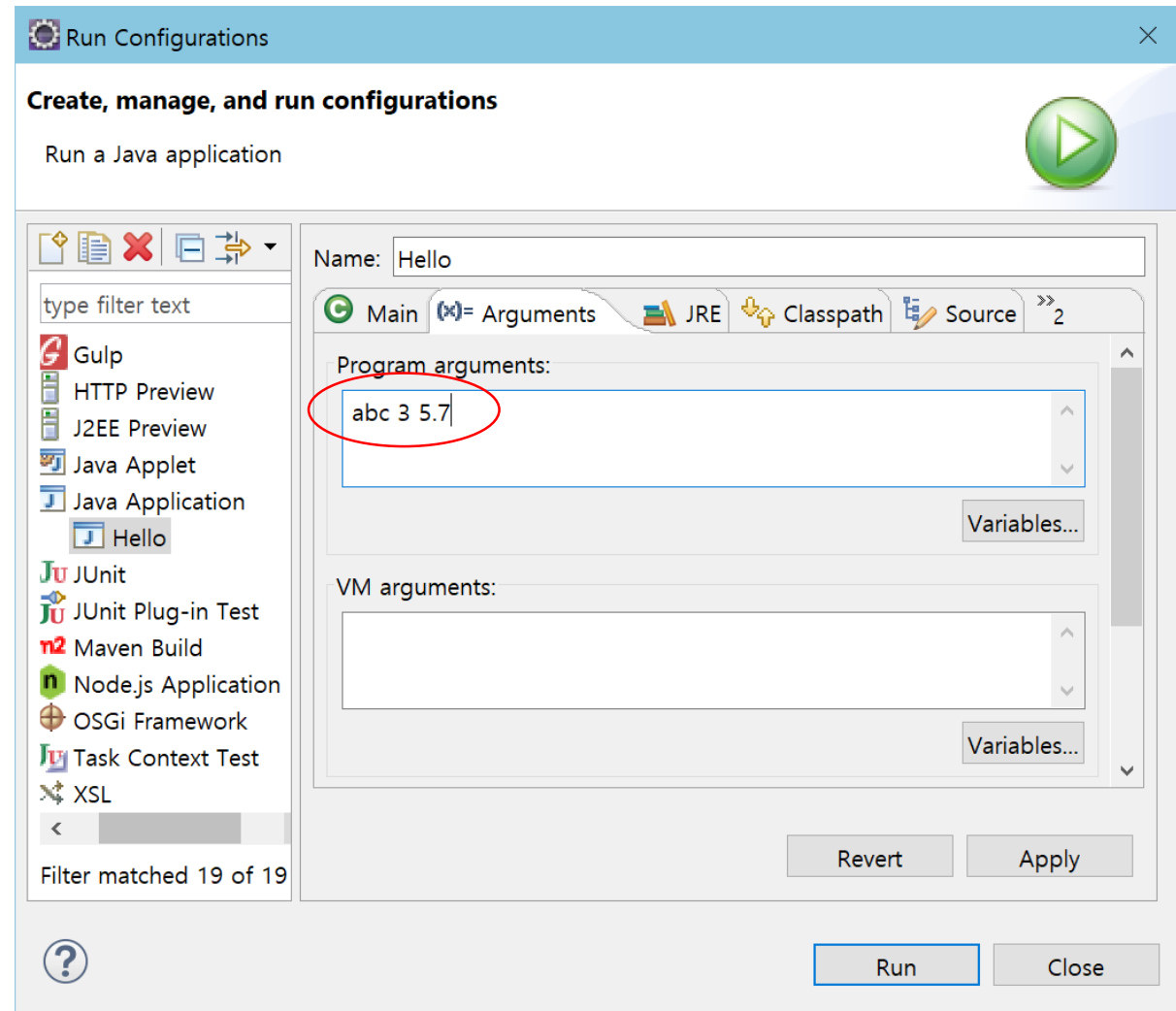
Method: main(string [] args)

- How parameters of main method are delivered



Method: main(String[] args) in Eclipse

Run →
Run Configurations



Practice 2

- Practice method

```
3 // Method enables you to abstract specific procedures for reusing it!
4 public class P1Method {
5
6     public static int add(int x, int y) {
7         return x + y;
8     }
9
10    public static int minus(int x, int y) {
11        return x - y;
12    }
13
14    public static int multiply(int x, int y) {
15        return x * y;
16    }
17
18    public static float divide(int x, int y) {
19        return (float) x / (float) y;
20    }
21
22    public static void main(String[] args) {
23        System.out.println(add(3, 5));
24        System.out.println(minus(3, 5));
25        System.out.println(multiply(3, 5));
26        System.out.println(divide(3, 5));
27        System.out.println(divide(3, 0));
28    }
29 }
```

Practice 3

- Practice method

```
3  // Revisit in-class assignment
4  public class P2GradeCalculator {
5
6      public static char computeGrade(int point) {
7          return 'f';
8      }
9
10     public static void main(String[] args) {
11         // Abstract a procedure to compute grade via computeGrade
12         // it gets a point and return grade
13     }
14 }
```

Exception Handling

- Compile Error
 - Grammatical Error
 - Found in compilation
- Exception
 - Error can yield malfunction in run-time
 - Example
 - Divide by zero
 - Array index out of bound
 - File not found
 - Number format exception
 - Exception Handling in Java
 - Exception → JVM recognizes it → JVM notifies it to the application
 - No exception handling in the application
 - → Terminate the application

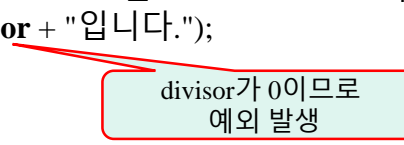
Practice 4: 예제 3-14 : 0으로 나누기 예외 발생으로 프로그램이 강제 종료되는 경우

두 정수를 입력 받아 나눗셈을 하고 몫을 구하는 프로그램 코드이다. 사용자가 나누는 수에 0을 입력하면 `ArithmeticException` 예외가 발생하여 프로그램이 강제 종료된다.

```
import java.util.Scanner;

public class DivideByZero {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int dividend; // 나뉘는수
        int divisor; // 나눗수

        System.out.print("나뉘는수를 입력하시오:");
        dividend = scanner.nextInt(); // 나뉘는수 입력
        System.out.print("나눗수를 입력하시오:");
        divisor = scanner.nextInt(); // 나눗수 입력
        System.out.println(dividend+"를 "+ divisor + "로 나누면 몫은 " +
            dividend/divisor + "입니다.");
        scanner.close();
    }
}
```



나뉘는수를 입력하시오:100

나눗수를 입력하시오:0

Exception in thread "main" [java.lang.ArithmeticException: / by zero](#)
at DivideByZero.main([ExceptionExample1.java:14](#))

Exception Handling: try-catch-finally

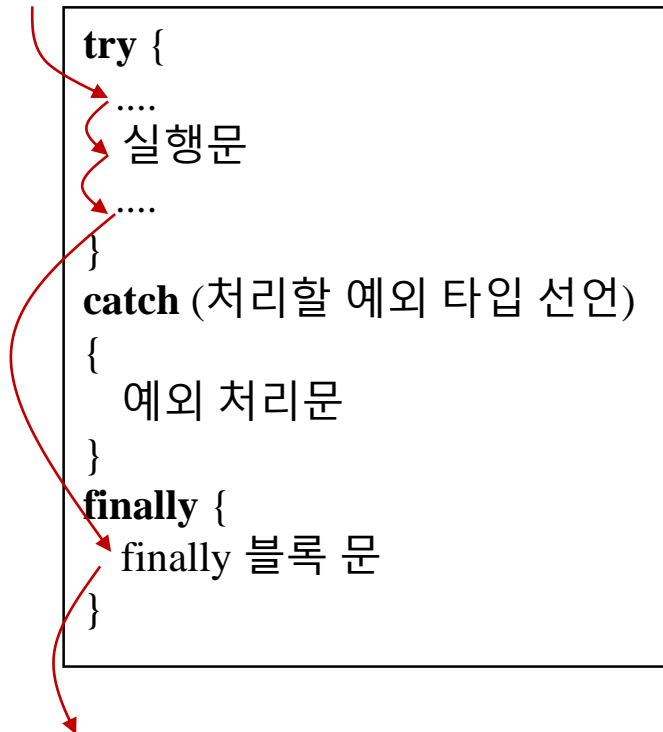
- Exception Handling
 - Proactively handle exceptions
 - try-catch-finally
 - Finally – optional block

생략
가능

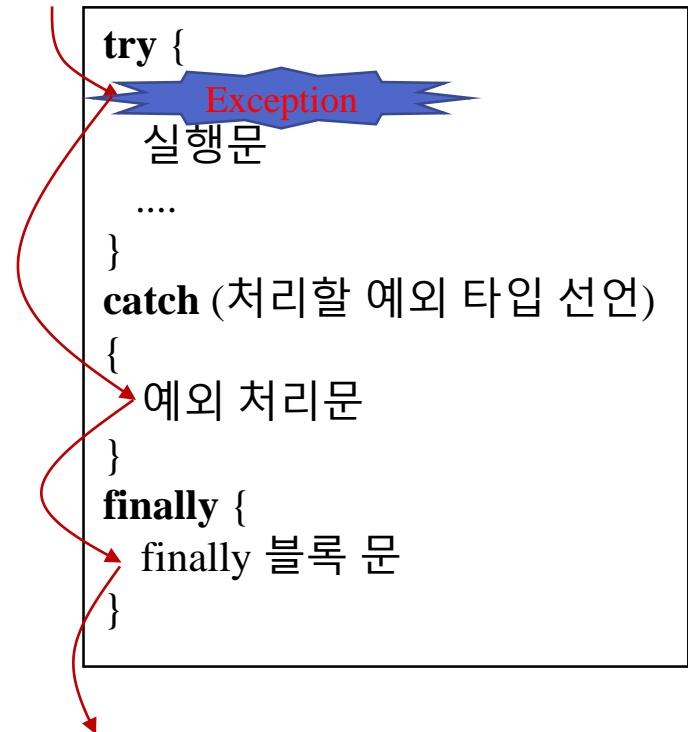
```
try {  
    예외가 발생할 가능성이 있는 실행문(try 블록)  
}  
catch (처리할 예외 타입 선언) {  
    예외 처리문(catch 블록)  
}  
finally {  
    예외 발생 여부와 상관없이 무조건 실행되는 문장  
    (finally 블록)  
}
```

Exception Handling: flow

No Exception



Exception occurs



Exception Handling: Classes

예외 타입(예외 클래스)	예외 발생 경우	패키지
ArithmeticException	정수를 0으로 나눌 때 발생	java.lang
NullPointerException	null 레퍼런스를 참조할 때 발생	java.lang
ClassCastException	변환할 수 없는 타입으로 객체를 변환할 때 발생	java.lang
OutOfMemoryError	메모리가 부족한 경우 발생	java.lang
ArrayIndexOutOfBoundsException	배열의 범위를 벗어난 접근 시 발생	java.lang
IllegalArgumentException	잘못된 인자 전달 시 발생	java.lang
IOException	입출력 동작 실패 또는 인터럽트 시 발생	java.io
NumberFormatException	문자열이 나타내는 숫자와 일치하지 않는 타입의 숫자로 변환 시 발생	java.lang
InputMismatchException	Scanner 클래스의 nextInt()를 호출하여 정수로 입력받고자 하였지만, 사용자가 'a' 등과 같이 문자를 입력한 경우	java.util

Practice 5: 예제 3-15 : 0으로 나눌 때 발생하는 ArithmeticException 예외 처리

try-catch 블록을 이용하여 예제 3-14를 수정하여, 정수를 0으로 나누는 경우에 "0으로 나눌 수 없습니다!"를 출력하고 다시 입력 받는 프로그램을 작성하라.

```
import java.util.Scanner;

public class DevideByZeroHandling {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while(true) {
            System.out.print("나뉘수를 입력하시오:");
            int dividend = scanner.nextInt(); // 나뉘수 입력
            System.out.print("나눗수를 입력하시오:");
            int divisor = scanner.nextInt(); // 나눗수 입력
            try {
                System.out.println(dividend + "를 " + divisor + "로 나누면 몫은 “ + dividend/divisor + "입니다.");
                break; // 정상적인 나눗기 완료 후 while 벗어나기
            }
            catch(ArithmeticException e) { // ArithmeticException 예외 처리 코드
                System.out.println("0으로 나눌 수 없습니다! 다시 입력하세요");
            }
        }
        scanner.close();
    }
}
```

ArithmeticException
예외 발생

```
나뉘수를 입력하시오:100
나눗수를 입력하시오:0
0으로 나눌 수 없습니다! 다시 입력하세요
나뉘수를 입력하시오:100
나눗수를 입력하시오:5
100를 5로 나누면 몫은 20입니다.
```

Practice 6: 예제 3-16 : 범위를 벗어난 배열의 접근

배열의 인덱스가 범위를 벗어날 때 발생하는 `ArrayIndexOutOfBoundsException`을 처리하는 프로그램을 작성하시오.

```
public class ArrayException {  
    public static void main (String[] args) {  
        int[] intArray = new int[5];  
        intArray[0] = 0;  
        try {  
            for (int i=0; i<5; i++) {  
                intArray[i+1] = i+1 + intArray[i];  
                System.out.println("intArray["+i+"]"+"="+intArray[i]);  
            }  
        }  
        catch (ArrayIndexOutOfBoundsException e) {  
            System.out.println("배열의 인덱스가 범위를 벗어났습니다.");  
        }  
    }  
}
```

i가 4일 때
ArrayIndexOutOfBoundsException 예외 발생

```
intArray[0]=0  
intArray[1]=1  
intArray[2]=3  
intArray[3]=6  
배열의 인덱스가 범위를 벗어났습니다.
```

Practice 7: 예제 3-17 : 입력오류시발생하는 예외 (InputMismatchException)

3개의 정수를 입력받아 합을 구하는 프로그램을 작성하라. 사용자가 정수가 아닌 문자를 입력할 때 발생하는 InputMismatchException 예외를 처리하여 다시 입력받도록 하라.

```
import java.util.Scanner;
import java.util.InputMismatchException;

public class InputException {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("정수 3개를 입력하세요");
        int sum=0, n=0;
        for(int i=0; i<3; i++) {
            System.out.print(i+">>");
            try {
                n = scanner.nextInt(); // 정수 입력
            }
            catch(InputMismatchException e) {
                System.out.println("정수가 아닙니다. 다시 입력하세요!");
                scanner.next(); // 입력 스트림에 있는 정수가 아닌 토큰을 버린다.
                i--; // 인덱스가 증가하지 않도록 미리 감소
                continue; // 다음 루프
            }
            sum += n; // 합하기
        }
        System.out.println("합은 " + sum);
        scanner.close();
    }
}
```

사용자가 문자를 입력하면
InputMismatchException 예외 발생

정수 3개를 입력하세요

0>>5

1>>R

정수가 아닙니다. 다시 입력하세요!

1>>4

2>>6

합은 15

Practice 8: 예제 3-18 : 정수가 아닌 문자열을 정수로 변환할 때 예외 발생(NumberFormatException)

문자열을 정수로 변환할 때 발생하는 NumberFormatException을 처리하는 프로그램을 작성하라.

```
public class NumException {  
    public static void main (String[] args) {  
        String[] stringNumber = {"23", "12", "3.141592", "998"};  
  
        int i=0;  
        try {  
            for (i=0; i<stringNumber.length; i++) {  
                int j = Integer.parseInt(stringNumber[i]);  
                System.out.println("숫자로 변환된 값은 " + j);  
            }  
        }  
        catch (NumberFormatException e) {  
            System.out.println(stringNumber[i] + "는 정수로 변환할 수 없습니다.");  
        }  
    }  
}
```

“3.141592”를 정수로 변환할 때
NumberFormatException
예외 발생

숫자로 변환된 값은 23
숫자로 변환된 값은 12
3.141592는 정수로 변환할 수 없습니다.

Summary

- Method
- Exception Handling