Java Programs - Exception Handling Examples

1. numberSegregator(String s)

```
import java.util.*;
import java.util.regex.*;

class InvalidInputException extends Exception {
    public InvalidInputException(String message) {
        super(message);
    }
}

public class Samuel {
    public static List<Integer> numberSegregator(String s) throws InvalidInputException {
        List<Integer> result = new ArrayList<>();
        Matcher m = Pattern.compile("\\d+").matcher(s);
        while (m.find()) {
            result.add(Integer.parseInt(m.group()));
        }
        if (result.isEmpty()) {
            throw new InvalidInputException("Invalid Input");
        }
        return result;
    }

    public static void main(String[] args) {
        try {
            System.out.println(numberSegregator("dgarg534htyty6758gh56"));
            System.out.println(numberSegregator("dgarghtytygh"));
        } catch (InvalidInputException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

2. reciprocalSquareSum(String[] input)

```
import java.util.*;

public class Jyoti {
    public static double reciprocalSquareSum(String[] input) throws Exception {
        try {
            double sum = 0.0;
            for (String s : input) {
                int val = Integer.parseInt(s);
                if (val == 0) continue;
                sum += 1.0 / (val * val);
            }
            return Math.round(sum * 10000.0) / 10000.0;
        } catch (Exception e) {
            throw new Exception("Invalid Try");
        }
    }
    public static void main(String[] args) {
        try {
            String[] input1 = {"2", "4", "0", "3", "2"};
            System.out.println(reciprocalSquareSum(input1));

            String[] input2 = {"2", "4", "a"};
            System.out.println(reciprocalSquareSum(input2));
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

3. specialCharacterCounter(String s)

```
import java.util.*;

class NoSpecialCharacterFoundException extends Exception {
    public NoSpecialCharacterFoundException(String message) {
        super(message);
    }
}

public class Ravi {
    public static List<Character> specialCharacterCounter(String s) throws NoSpecialCharacterFoundException {
        List<Character> list = new ArrayList<>();
        for (char c : s.toCharArray()) {
            if (!Character.isLetterOrDigit(c)) {
                list.add(c);
            }
        }
        if (list.isEmpty()) {
            throw new NoSpecialCharacterFoundException("No Special Character Found");
        }
        return list;
    }

    public static void main(String[] args) {
        try {
            System.out.println(specialCharacterCounter("hello@world!"));
            System.out.println(specialCharacterCounter("helloworld"));
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

4. findOddSquares(List<Integer> numbers)

```
import java.util.*;

class NoOddNumberFoundException extends Exception {
    public NoOddNumberFoundException(String message) {
        super(message);
    }
}

public class Anjali {
    public static List<Integer> findOddSquares(List<Integer> numbers) throws NoOddNumberFoundException {
        List<Integer> result = new ArrayList<>();
        for (int n : numbers) {
            if (n % 2 != 0) {
                result.add(n * n);
            }
        }
        if (result.isEmpty()) {
            throw new NoOddNumberFoundException("No Odd Number Found");
        }
        return result;
    }

    public static void main(String[] args) {
        try {
            System.out.println(findOddSquares(Arrays.asList(1, 2, 3, 4, 5)));
            System.out.println(findOddSquares(Arrays.asList(2, 4, 6)));
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```

5. findPrimeCubes(List<Integer> numbers)

```
import java.util.*;

class NoPrimeNumberFoundException extends Exception {
    public NoPrimeNumberFoundException(String message) {
        super(message);
    }
}

public class Vikram {
    public static boolean isPrime(int n) {
        if (n <= 1) return false;
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) return false;
        }
        return true;
    }

    public static List<Integer> findPrimeCubes(List<Integer> numbers) throws NoPrimeNumberFoundException {
        List<Integer> result = new ArrayList<>();
        for (int n : numbers) {
            if (isPrime(n)) {
                result.add(n * n * n);
            }
        }
        if (result.isEmpty()) {
            throw new NoPrimeNumberFoundException("No Prime Number Found");
        }
        return result;
    }

    public static void main(String[] args) {
        try {
            System.out.println(findPrimeCubes(Arrays.asList(2, 3, 4, 5, 6)));
            System.out.println(findPrimeCubes(Arrays.asList(4, 6, 8, 9)));
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}
```