



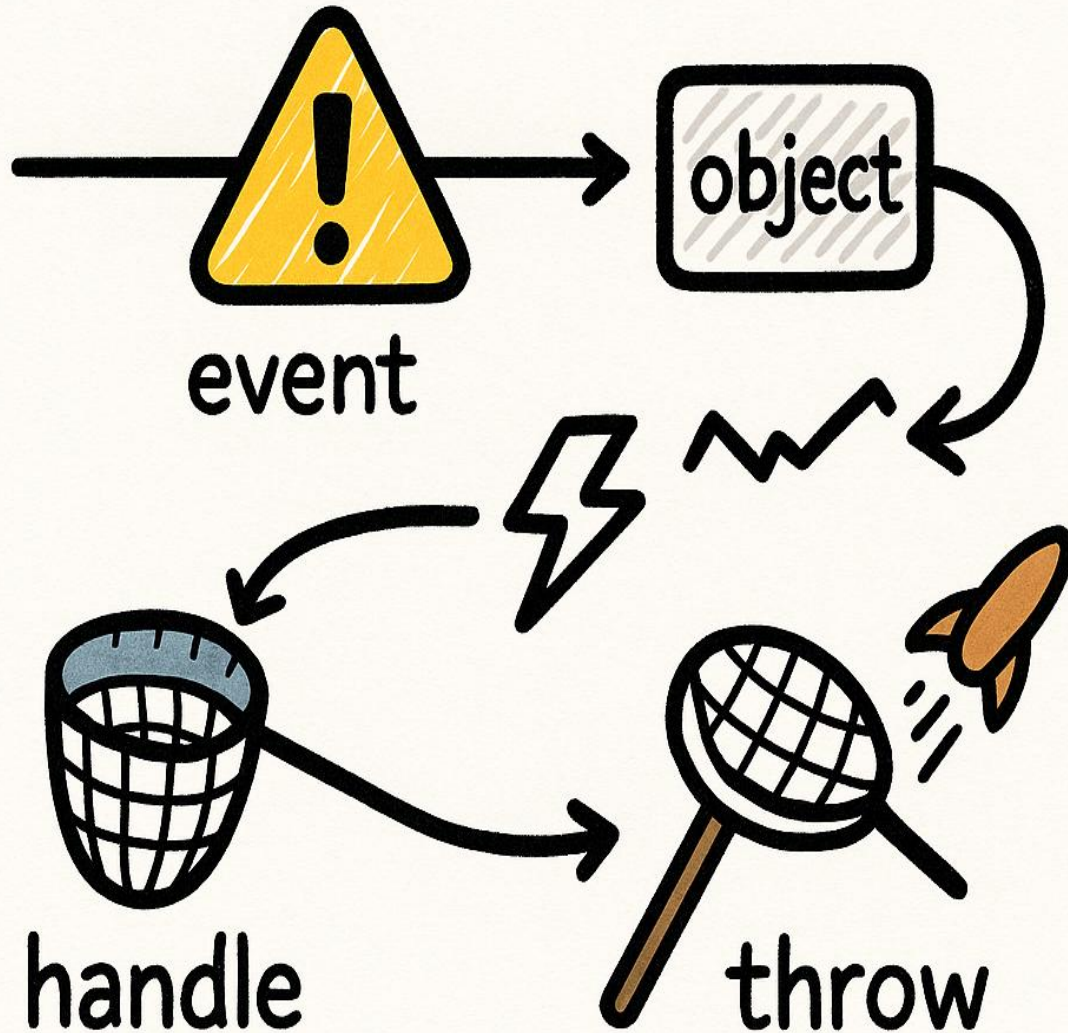
# Java OOP

## 10128

# Exceptions

Pini shlomi

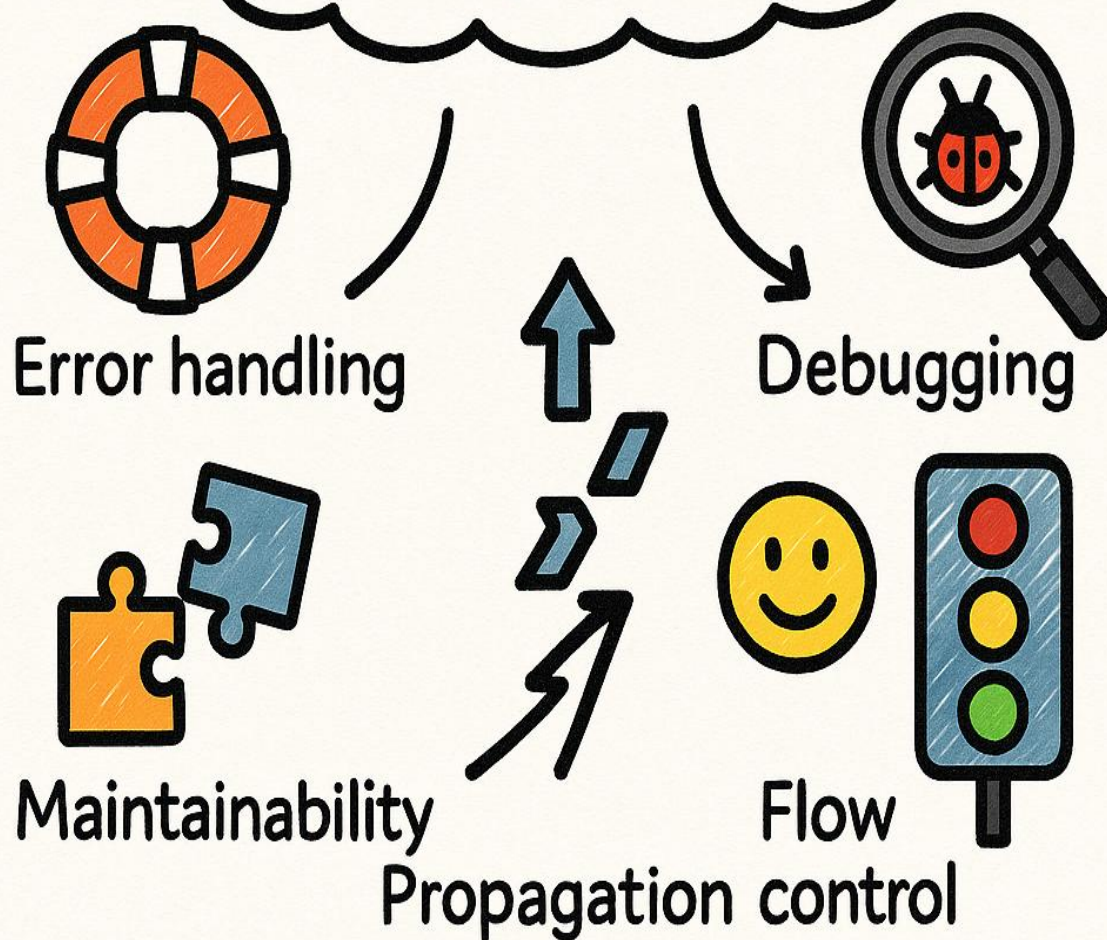
# Exception



- Unexpected runtime event  
→ **disrupts** flow
- Wrapped as Exception object  
(type · message · state)
- Handle with **try / catch**
- Propagate with **throw**



# Why Exceptions?



## 🛑 Error handling

*Prevent unexpected crashes or errors*

## 🔍 Debugging

*Provide clues about where and why errors occur*

## 🧩 Maintainability

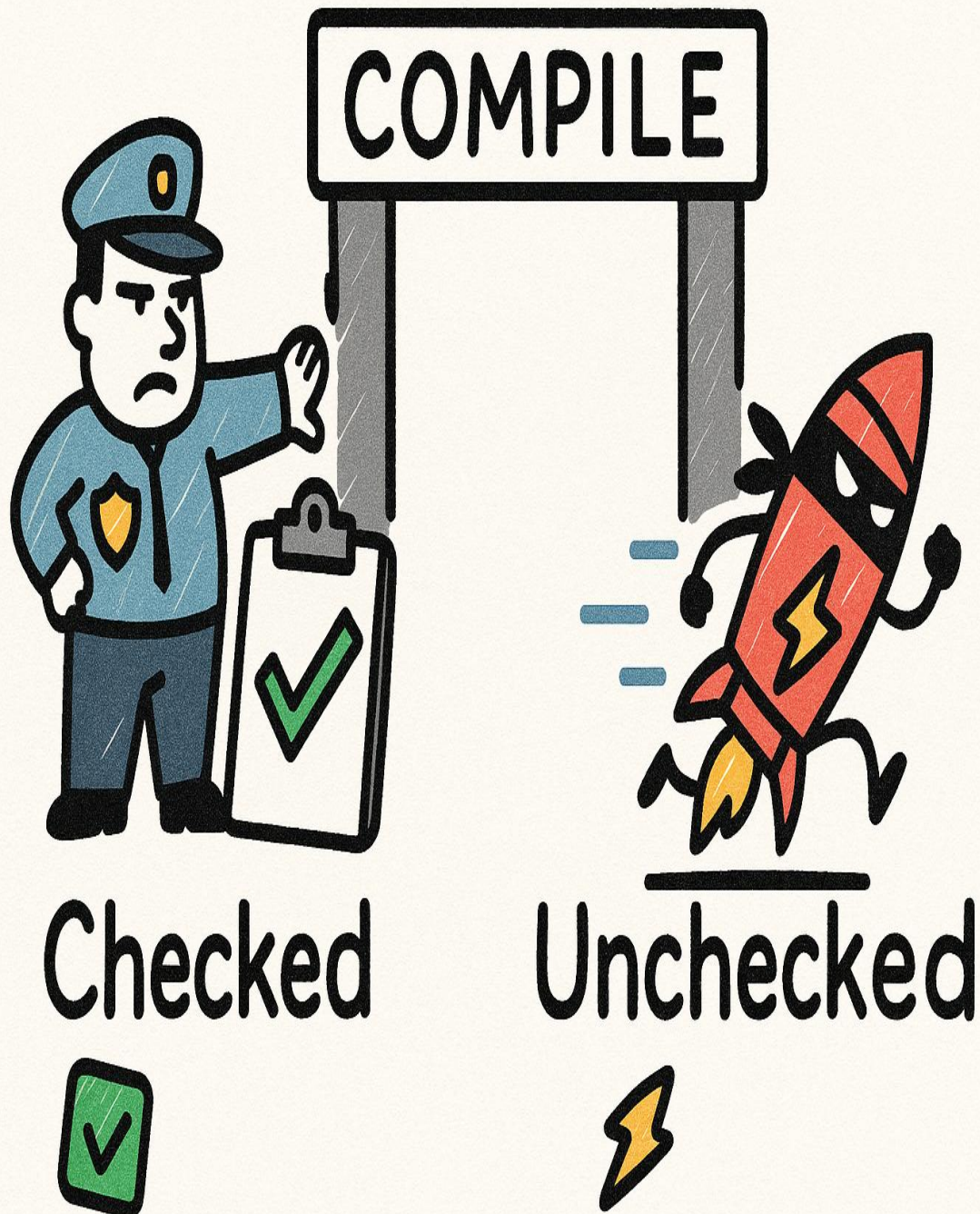
*Keep error-detection code separate from error-handling code*

## 📈 Propagation

Bubble up the call stack so higher-level code can decide what to do

## 🚦 Flow control

*Let the program change course based on specific conditions*



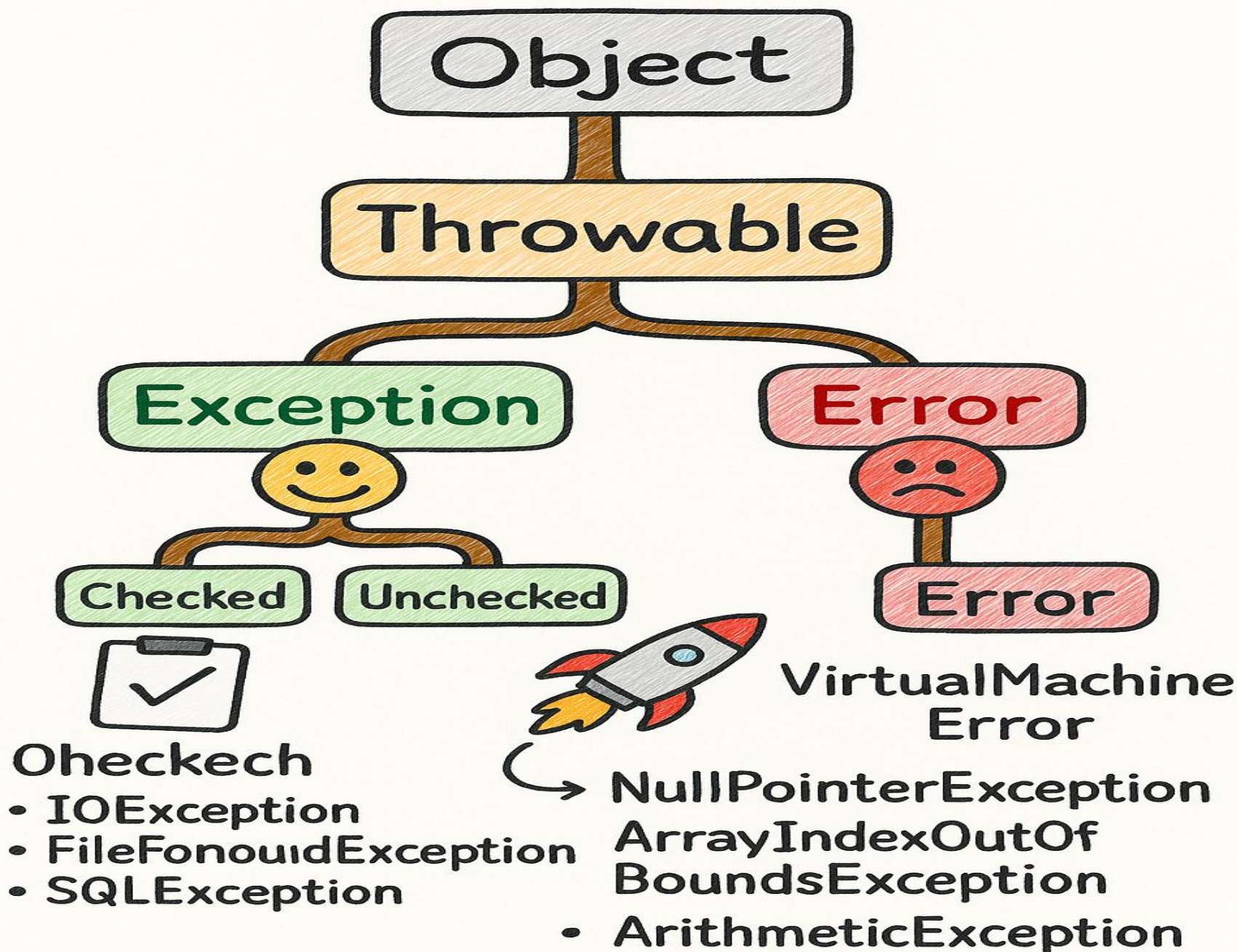
## Checked

Verified at compile-time,  
must be handled (try/catch) or  
declared with throws. Ex:  
IOException, SQLException

## Unchecked

Not checked by the compiler,  
runtime bugs like :  
NullPointerException .  
Handling is optional but recommended  
on specific conditions





# Exception Example

```
public static void main(String[] args) {  
    Scanner s = new Scanner(System.in);  
    double num1, num2, res;  
    System.out.print("Enter 2 numbers: ");  
    num1 = s.nextInt();  
    num2 = s.nextInt();  
    res = num1 / num2;  
    System.out.println(num1 + "/" + num2 + "=" + res);  
    s.close();  
}
```

## console

Enter 2 numbers: 5 2  
5.0/2.0=2.5

## console

Enter 2 numbers: 5 a

Exception in thread "main" java.util.InputMismatchException  
at java.base/java.util.Scanner.throwFor(Scanner.java:939)  
at java.base/java.util.Scanner.next(Scanner.java:1594)  
at java.base/java.util.Scanner.nextInt(Scanner.java:2258)  
at java.base/java.util.Scanner.nextInt(Scanner.java:2212)  
at code\_presentation.Main.main(Main.java:12)

Exception type

Where is happened

# Exception Handling

```
public static void main(String[] args) {  
    Scanner s = new Scanner(System.in);  
    try {  
        double num1, num2, res;  
        System.out.print("Enter 2 numbers: ");  
        num1 = s.nextInt();  
        num2 = s.nextInt();  
        res = num1 / num2;  
        System.out.println(num1 + "/" + num2 + "=" + res);  
    } catch (InputMismatchException e) {  
        System.out.println("Error: invalid input, input must be number");  
        // more catch blocks  
    } finally {  
        System.out.println("finally close scanner resource");  
        s.close();  
    }  
}
```

Always will do even with return exist

## console

Enter 2 numbers: 5 a  
Error: invalid input, input must be number

## console

Enter 2 numbers: 5 2  
5.0/2.0=2.5

## תרגיל 1 טיפול בחריגה של המרת מחרוזת למספר שלם

ממשו את הפונקציה `convertToInt` שמקבלת מערך של מחרוזות, מנסה להמיר כל מחרוזת למספר שלם, ומחזירה מערך חדש של מספרים שלמים עם כל הערכים שהומרו בהצלחה. על הפונקציה לטפל במקרים הבאים:

- אם מחרוזת מסוימת במערך אינה ניתנת להמרה למספר שלם (שגיאה מסוג `NumberFormatException`) יש לדלג עליה ולהדפיס הודעת שגיאה מתאימה.
  - אם המערך ריק - יש להדפיס הודעה מתאימה ולסיים.
- קראו לפונקציה עם מערך מחרוזות והדפיסו את המערך החדש שנוצר לאחר הקריאה לפונקציה.

Enter a list of strings separated by spaces:

4 6.1 6 e 10

Error: '6.1' is not a valid integer.

Error: 'e' is not a valid integer.

Converted numbers: [4, 6, 10]

[קישור ל-starter](#)

[אתר להורדת ספריות מ-github.](#)



## תרגיל 2 טיפול בחריגה של חלוקה באפס.

Enter number 1:  
76  
Enter number 2:  
3  
The result of the division is: 25

Enter number 1:  
76  
Enter number 2:  
0  
Error: Cannot divide by zero.  
Enter number 1:

Enter number 1:  
54  
Enter number 2:  
e  
Error: Please enter valid numbers.  
Enter number 1:

ממשו את הפונקציה `divideNumbers` שמקבלת שני מספרים **שלמים**, מחלקת את המספר הראשון בשני, ומחזירה את התוצאה (**מספר שלם**).  
על הפונקציה לטפל במקרים הבאים:

אם הקלט לא תקין (לדוגמה, אם הקלט אינו מספר) - יש להדפיס הודעת שגיאה מתאימה.

אם מתבצעת חלוקה באפס (`ArithmeticException`) - יש להדפיס הודעת שגיאה מתאימה.

כתבו תוכנית שמבקשת מהמשתמש להכניס שני מספרים, מפעילה את הפונקציה `divideNumbers` ומדפיסה את התוצאה אם לא התרחשה שגיאה. אם התרחשה שגיאה יש לבקש שוב ושוב ערך תקין.

[קישור ל-starter](#)

[אתר להורדת ספריות מ-github.](#)

# Exception Handling with throw



```
public static void main(String[] args) {  
    Scanner s = new Scanner(System.in);  
    try {  
        calculate2Numbers(s);  
    } catch (InputMismatchException e) {  
        System.out.println("Error: invalid input, input must be number");  
    } finally {  
        System.out.println("finally close scanner resource");  
        s.close();  
    }  
}
```

```
private static void calculate2Numbers(Scanner s) throws InputMismatchException {  
    try {  
        double num1, num2, res;  
        System.out.print("Enter 2 numbers: ");  
        num1 = s.nextInt();  
        num2 = s.nextInt();  
        res = num1 / num2;  
        System.out.println(num1 + "/" + num2 + "=" + res);  
    } catch (InputMismatchException e) {  
        System.out.println("Error acquired, I don't know what to do ....");  
        throw new InputMismatchException();  
    }  
}
```

Exception



## console

```
Enter 2 numbers: 5 a  
Error acquired, I don't know what to do ....  
Error: invalid input, input must be number  
finally close scanner resource
```

# Read numbers example

```
public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    int[] arr = new int [5];
    boolean isNumOK = false;
    int i= 0 ;
    while (!isNumOK ) {
        try {
            System.out.println ("Enter "+ (arr.length - i) + " numbers:");
            for (; i < arr.length ; i ++) {
                arr[i] = s.nextInt();
            }
            isNumOK = true;
        } catch (InputMismatchException e ) {
            System.out.println ("You were asked to enter a number");
            s.nextLine (); // cleans buffer
        }
    }
    System.out.println ("The numbers: "+ Arrays.toString(arr));
    s.close();
}
```

## console

```
Enter 5 numbers:
1
2
a
You were asked to enter a number
Enter 3 numbers:
3
4
5
The numbers: [1, 2, 3, 4, 5]
```



# Read positive numbers example

```
public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    int[] arr = new int [5];
    boolean isNumOK = false;
    int i= 0, num ;
    while (!isNumOK ) {
        try {
            System.out.println ("Enter "+ (arr.length - i) + " numbers:");
            for (; i < arr.length ; i ++) {
                num = s.nextInt();
                if (num >= 0) {
                    arr[i] = num;
                } else {
                    throw new Exception("Only positive numbers allowed");
                }
            }
            isNumOK = true;
        } catch (InputMismatchException e ) {
            System.out.println ("You were asked to enter a number");
            s.nextLine (); // cleans buffer
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
    System.out.println ("The numbers: "+ Arrays.toString(arr));
    s.close();
}
```

## console

Enter 5 numbers:

1  
2  
a

You were asked to enter a number

Enter 3 numbers:

3  
-5

Only positive numbers allowed

Enter 2 numbers:

4  
5

The numbers: [1, 2, 3, 4, 5]

# Custom Exception

- handle a specific error condition that is not already covered by the built-in exception classes in Java.
- define a new class that extends the "Exception" class or one of its subclasses
- add any additional properties or methods that are required to handle the specific error condition

```
public class NotPositiveException extends Exception {  
  
    private static final String NOT_POSITIVE_EXCEPTION_MESSAGE="Only positive numbers allowed";  
  
    public NotPositiveException() {  
        super(NOT_POSITIVE_EXCEPTION_MESSAGE);  
    }  
}
```

# Read positive numbers example

```
public static void main(String[] args) {
    Scanner s = new Scanner(System.in);
    int[] arr = new int [5];
    boolean isNumOK = false;
    int i= 0, num ;
    while (!isNumOK ) {
        try {
            System.out.println ("Enter "+ (arr.length - i) + " numbers:");
            for (; i < arr.length ; i ++) {
                num = s.nextInt();
                if (num >= 0) {
                    arr[i] = num;
                }else {
                    throw new NotPositiveExeption();
                }
            }
            isNumOK = true;
        } catch (InputMismatchException e ) {
            System.out.println ("You were asked to enter a number");
            s.nextLine (); // cleans buffer
        } catch (NotPositiveExeption e) {
            System.out.println(e.getMessage());
        }
    }
    System.out.println ("The numbers: "+ Arrays.toString(arr));
    s.close();
}
```

## console

Enter 5 numbers:

1  
2  
-5

Only positive number allowed

Enter 3 numbers:

3  
4  
a

You were asked to enter a number

Enter 1 numbers:

5

The numbers: [1, 2, 3, 4, 5]



## תרגיל 3: תוכנית לעדכון ציוני סטודנטים בכיתה

[קישור לקובץ התרגיל](#)

[קישור ל-starter](#)

[אתר להורדת ספריות מ-github](#)

תרגיל 4: תוכנית ניהול לשמירת ספרים בספריה

[קישור לקובץ התרגיל](#)

[קישור ל-starter](#)

[אתר להורדת ספריות מ-github](#)