



*Ελληνικό Μεσογειακό Πανεπιστήμιο
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών*

Αντικειμενοστρεφής Προγραμματισμός Εργαστήριο 7

Καπετανάκης Φανούριος

Τι θα δούμε

- Εξαιρέσεις (Exceptions)
- Try
- throw
- Catch
- Declare
- Finally

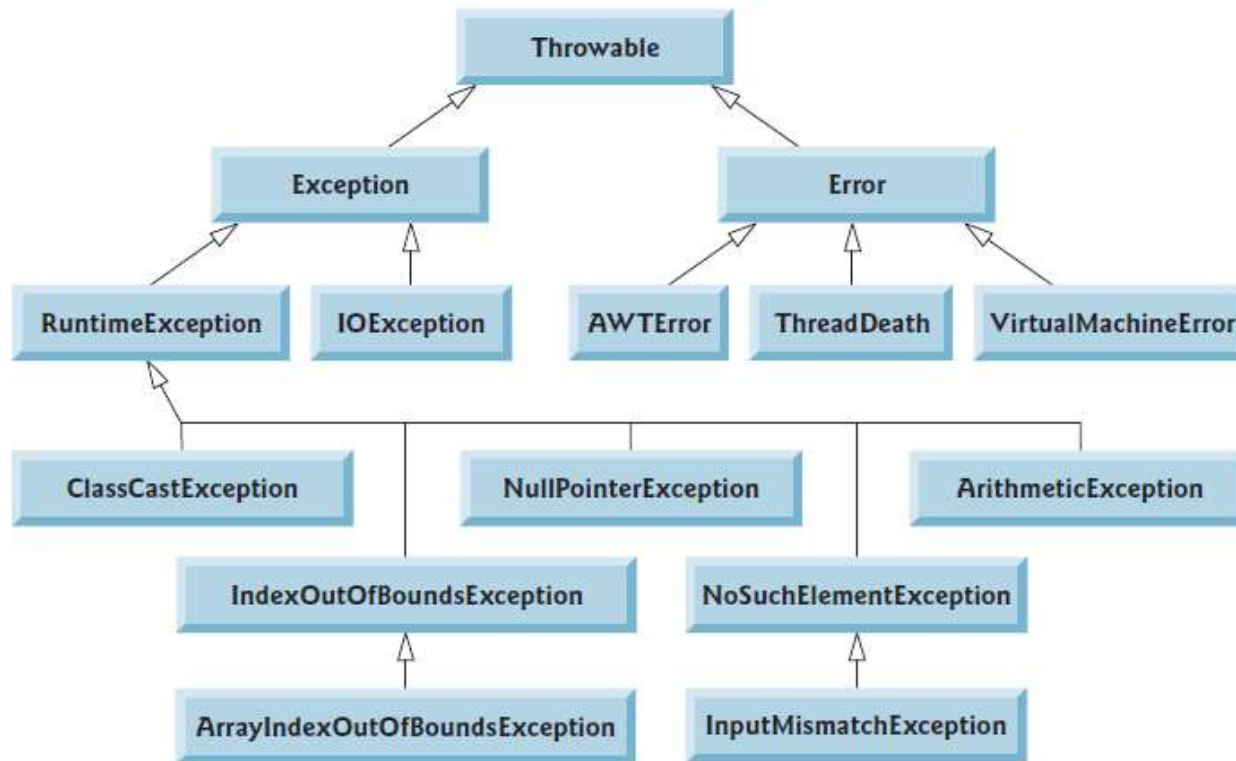
Τι είναι η εξαίρεση;

- Η εξαίρεση είναι ένα ανεπιθύμητο ή μη αναμενόμενο συμβάν, που διακόπτει τη κανονική ροή εκτέλεσης του προγράμματός μας, λόγω κάποιου σφάλματος που μπορεί να προέκυψε.
- Όταν εμφανίζεται ένα σφάλμα, η java δημιουργεί ένα μήνυμα σφάλματος (error message) και παράγει μια εξαίρεση.
- Παράδειγμα, αν το πρόγραμμα περιμένει να διαβάσει ένα ακέραιο για να τον αποθηκεύσει σε μια μεταβλητή, και αντί για ακέραιο πληκτρολογήσουμε ένα String, τι θα γίνει;

Τύποι σφαλμάτων

- Υπάρχουν 2 τύποι σφαλμάτων
 - Compile time errors (σφάλματα που προκύπτουν κατά τη μεταγλώττιση του προγράμματος)
 - Runtime errors (σφάλματα που προκύπτουν κατά την εκτέλεση του προγράμματος)
- Unchecked Exceptions (RuntimeException και Error. Δεν είναι υποχρεωτικό να χειριστούμε αυτούς τους τύπους εξαιρέσεων)
- Checked Exceptions (πρέπει να συλληφθούν από μία μέθοδο)

Ιεραρχία εξαιρέσεων Java



Εξαιρέσεις- αντιμετώπιση

Ένα πρόγραμμα μπορεί να αντιμετωπίσει μια εξαίρεση με ένα από τους παρακάτω τρόπους:

- Να μην κάνει τίποτα, να την αγνοήσει
- Να την χειριστεί στο σημείο που έγινε η εξαίρεση
- Να την χειριστεί σε κάποιο άλλο σημείο του προγράμματος

Λέξεις - κλειδιά

throws

- περιγράφει τις εξαιρέσεις που μπορεί να παραχθούν από μια μέθοδο

throw

- η εντολή throw μας επιτρέπει να δημιουργήσουμε ένα προσαρμοσμένο σφάλμα
- χρησιμοποιείται με ένα τύπο εξαίρεσης, ArithmeticException, ArrayIndexOutOfBoundsException κλπ.

try

- υποδηλώνει την αρχή ενός block που σχετίζεται με ένα σύνολο χειριστών εξαιρέσεων

catch

- αν το block που περιλαμβάνεται σε ένα try παράγει μια εξαίρεση αντιστοίχου τύπου, η ροή του ελέγχου μεταφέρεται εδώ

finally

- Για την εκτέλεση ενεργειών που πρέπει να εκτελεστούν οπωσδήποτε, ανεξάρτητα από το αν προκληθεί ή όχι εξαίρεση

Παράδειγμα- Διαίρεση με το 0 χωρίς χειρισμό εξαίρεσης

```
import java.util.Scanner;

public class Division {

    public static int phliko(int arithmhths, int paronomasths)
    {
        return arithmhths / paronomasths;
    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.println("Δώσε ένα ακέραιο αριθμό για αριθμητή:");
        int arithmhths = scanner.nextInt();
        System.out.println("Δώσε ένα ακέραιο αριθμό για παρονομαστή:");
        int paronomasths = scanner.nextInt();

        int result = phliko(arithmhths,paronomasths);

        System.out.println("Αποτέλεσμα: "+arithmhths + " / "
            +paronomasths +"= "+result);

    }

}
```

Output (1)

Δώσε ένα ακέραιο αριθμό για αριθμητή:

100

Δώσε ένα ακέραιο αριθμό για παρονομαστή:

7

Αποτέλεσμα: 100 / 7 = 14

Output (2)

Δώσε ένα ακέραιο αριθμό για αριθμητή:

100

Δώσε ένα ακέραιο αριθμό για παρονομαστή:

0

Exception in thread "main"

[java.lang.ArithmeticException: / by zero](#)

at Main.quotient(Main.java:7)

at Main.main(Main.java:17)

Παράδειγμα- Διαίρεση με το 0 και InputMismatchException με χειρισμό εξαίρεσης

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

public class Division {

public static int phliko(int arithmhths, int paronomasths)
throws ArithmeticException ←
{
return arithmhths / paronomasths; ←
}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);
boolean continueLoop = true;

do {
try { ←
System.out.println("Δώσε ένα ακέραιο αριθμό για αριθμητή:");
int arithmhths = scanner.nextInt();
System.out.println("Δώσε ένα ακέραιο αριθμό για
παρονομαστή:");
int paronomasths = scanner.nextInt();

int result = phliko(arithmhths,paronomasths);
System.out.println("Αποτέλεσμα: "+arithmhths + " / "
+paronomasths +" = "+result);
continueLoop = false;

}
}
```

```
catch (InputMismatchException inputMismatchException) ←
{
System.err.printf("%n Εξαίρεση: %s%n",
inputMismatchException);
scanner.nextLine();
System.out.println("Πρέπει να πληκτρολογήσεις ακέραιο."
+ "Προσπάθησε ξανά");
}
catch(ArithmeticException arithmeticException) ←
{
System.err.printf("Εξαίρεση: "+ arithmeticException);
System.out.println("Το μηδέν είναι ένας μη έγκυρος
παρονομαστής. "
+ "Προσπάθησε ξανά");
}
}while(continueLoop);
}
}
```

Παράδειγμα- Διαίρεση με το 0 και InputMismatchException με χειρισμό εξαίρεσης

Output (1)

Δώσε ένα ακέραιο αριθμό για αριθμητή:

100

Δώσε ένα ακέραιο αριθμό για παρονομαστή:

0

Εξαίρεση: [java.lang.ArithmeticException: / by zero](#)

Το μηδέν είναι ένας μη έγκυρος παρονομαστής. Προσπάθησε ξανά

Δώσε ένα ακέραιο αριθμό για αριθμητή:

100

Δώσε ένα ακέραιο αριθμό για παρονομαστή:

7

Αποτέλεσμα: $100 / 7 = 14$

Output (2)

Δώσε ένα ακέραιο αριθμό για αριθμητή:

100

Δώσε ένα ακέραιο αριθμό για παρονομαστή:

δυο

Εξαίρεση: [java.util.InputMismatchException](#)

Πρέπει να πληκτρολογήσεις ακέραιο. Προσπάθησε ξανά

Δώσε ένα ακέραιο αριθμό για αριθμητή:

100

Δώσε ένα ακέραιο αριθμό για παρονομαστή:

14

Αποτέλεσμα: $100 / 14 = 7$

Παράδειγμα - finally Block

```
public class Demo {  
  
    public static void main(String[] args) {  
        try {  
            int[] myNumbers = {1, 2, 3};  
            System.out.println(myNumbers[10]);  
        }  
        catch (Exception e) {  
            System.out.println("Κάτι πήγε στραβά. Το λάθος είναι:  
"+e.getMessage());  
        }  
        finally {  
            System.out.println("Κώδικας που θα τρέξει ανεξάρτητα  
από το exception.");  
        }  
    }  
}
```

Output

```
Κάτι πήγε στραβά. Το λάθος είναι: Index 10 out of bounds for length 3  
Κώδικας που θα τρέξει ανεξάρτητα από το exception.
```

Ένα ακόμα παράδειγμα για αρχεία

```
finally {  
    if (out != null) {  
        System.out.println("Κλείσιμο αρχείου");  
        out.close();  
    } else {  
        System.out.println("το αρχείο είναι  
κλειστό");  
    }  
}
```

Παράδειγμα Time1

```
package time1;
//Δήλωση κλάσης Timer1 διατηρεί την ώρα σε 24-ωρη μορφή
public class Time1 {

    private int hour;//0-23
    private int minute;//0-59
    private int second;//0-59

    //Ορίζει μια νέα τιμή ώρας χρησιμοποιώντας καθολική ώρα. Εμφανίζει μια
    //εξαίρεση αν είναι άκυρη η ώρα, τα λεπτά ή τα δευτερόλεπτα.
    public void setTime(int hour, int minute, int second)
    {
        //Επικύρωση ώρας, λεπτών και δευτερολέπτων
        if(hour < 0 || hour >=24 || minute < 0 || minute >=60 || second < 0 || second >=60)
        {
            throw new IllegalArgumentException(
                "Η hour, minute ή/και η second ήταν εκτός εμβέλειας ");
        }

        this.hour = hour;
        this.minute = minute;
        this.second = second;
    }

    //Μετατρέπει σε String, σε μορφή καθολικής ώρας(HH:MM:SS)
    public String toUniversalString()
    {
        return String.format("%02d: %02d: %02d:", hour, minute, second);
    }

    //Μετατρέπει σε String, σε τυπική μορφή ώρας(HH:MM:SS)
    public String toString()
    {
        return String.format("%d:%02d:%02d %s",
            ((hour == 0 || hour == 12) ? 12 : hour % 12),
            minute, second, (hour < 12 ? "AM" : "PM") );
    }
}
```

- Η μέθοδος format είναι παρόμοια με την μέθοδο System.out.printf με την διαφορά ότι η format επιστέφει ένα μορφοποιημένο String αντί να το εμφανίζει σ' ένα παράθυρο εντολών
- Τελεστής υπό όρους (?):
 - για να προσδιορίσει την τιμή της hour στο String
 - αν η hour είναι 0 ή 12, η hour εμφανίζεται ως 12, διαφορετικά η hour εμφανίζεται ως μια τιμή μεταξύ 1 και 11.
 - Ο τελεστής υπό όρους προσδιορίζει αν θα επιστραφεί το AM ή το PM σαν μέρος του String
 - If... then (?) else (:) ...

Παράδειγμα Time1

```
package time1;

public class TimeTest {

public static void main(String[] args) {
//Δημιουργεί και αρχικοποιεί ένα αντικείμενο Time1
Time1 time= new Time1();//καλεί την συνάρτηση δημιουργίας Time1

//Εμφανίζει τις String αναπαραστάσεις της ώρας
displayTime("Μετά την δημιουργία αντικειμένου ώρας", time);
System.out.println();

//Αλλάζει την ώρα και εμφανίζει την αλλαγμένη ώρα
time.setTime(13, 27, 6);
displayTime("Μετά την κλήση setTime", time);
System.out.println();

//Προσπαθεί να ορίσει την ώρα με άκυρες τιμές
try
{
time.setTime(99, 99, 99);//όλες οι τιμές εκτός εύρους
}
catch(IllegalArgumentException e)
{
System.out.printf("Exception: %s%n", e.getMessage());
}

//Εμφανίζει την ώρα μετά από προσπάθεια να ορισθούν άκυρες τιμές
displayTime("Μετά την κλήση της setTime με άκυρες τιμές",time);
}

//Εμφανίζει ένα αντικείμενο Time1 σε 24-ωρη μορφή
private static void displayTime(String header, Time1 t)
{
System.out.printf("%s\n Καθολική ώρα: %s\n Τυπική ώρα %s\n",
header, t.toUniversalString(), t.toString());
} }
}
```

Output

Μετά την δημιουργία αντικειμένου ώρας
Καθολική ώρα: 00: 00: 00:
Τυπική ώρα 12:00:00 AM

Μετά την κλήση setTime
Καθολική ώρα: 13: 27: 06:
Τυπική ώρα 1:27:06 PM

Exception: H hour, minute και η second ήταν εκτός εμβέλειας

Μετά την κλήση της setTime με άκυρες τιμές
Καθολική ώρα: 13: 27: 06:
Τυπική ώρα 1:27:06 PM

Άσκηση για παρουσίες μαθήματος

Να δημιουργήσετε μία κλάση, την `Test` που θα υλοποιεί την κλάση `main`. Στη `main`, να διαβάσετε από το πληκτρολόγιο έναν ακέραιο και να φροντίσετε σε περίπτωση που διαβαστεί ένα `String`, να εμφανιστεί κατάλληλο μήνυμα κάνοντας χρήση της `InputMismatchException`, να μην τερματιστεί το πρόγραμμα και να συνεχίσει να ζητάει από το χρήστη να δώσει ακέραιο.