Εξαιρέσεις στο κώδικα

Εξαιρέσεις (Exceptions)

Κατά την εκτέλεση ενός προγράμματος μπορεί να συμβούν διάφορα γεγονότα, τα οποία να χαλάσουν τη ροή εκτέλεσης των εντολών. Για παράδειγμα να γίνει μια διαίρεση όπου ο παρανομαστής είναι 0, να διακοπεί η σύνδεση με ένα δικτυακό τόπο ενώ διαβάζουμε ή στέλνουμε δεδομένα κ.λπ.

Εξαιρέσεις στο κώδικα

```
public class JavaExceptions {

    public static void main(String[] args) {
        // TODO code application logic here
        int x = 5;
        int y = 0;
        int k = x / y;
        System.out.println(k);
    }
}
```

```
ut - JavaExceptions (run)
run:
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at JavaExceptions.main(JavaExceptions.java:17)
Java Result: 1
```

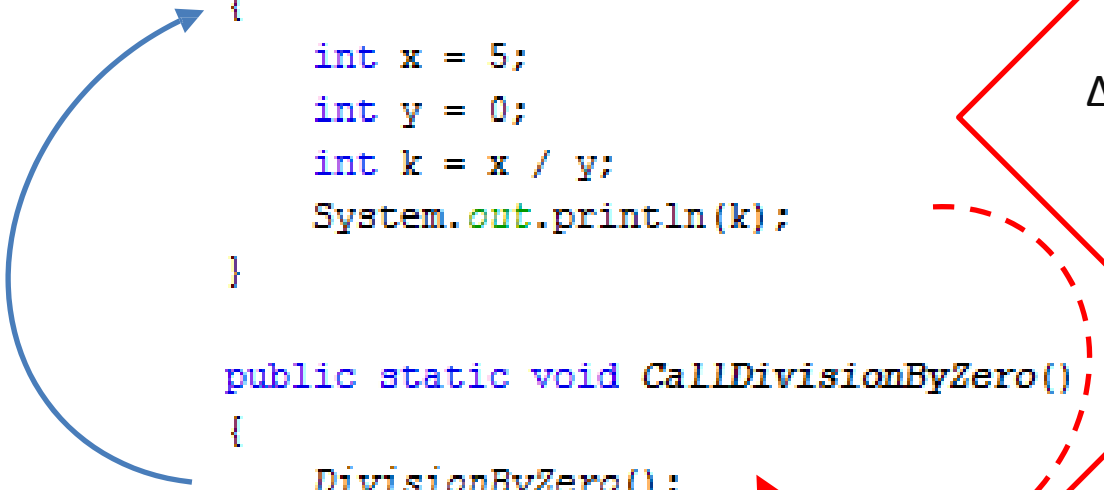
Εξαιρέσεις στο κώδικα

Η δημιουργία ενός σφάλματος στην εκτέλεση του προγράμματος δημιουργεί ένα αντικείμενο εξαίρεσης (exception object), ανάλογα με το σφάλμα που έχει συμβεί, και αν δεν υπάρχει ο κατάλληλος διαχειριστής σφάλματος τότε θα τερματιστεί η εφαρμογή.

```
public class JavaExceptions {

    public static void DivisionByZero()
    {
        int x = 5;
        int y = 0;
        int k = x / y;
        System.out.println(k);
    }

    public static void CallDivisionByZero()
    {
        DivisionByZero();
    }
```



Δεν υπάρχει διαχείριση της εξαίρεσης

Επιστροφή στο σημείο κλήσης για πιθανή διαχείριση της εξαίρεσης

Δεν υπάρχει διαχείριση της εξαίρεσης

Διαχείριση εξαίρεσης

```
    public static void HandleDivisionByZero()
    {
        try
        {
            CallDivisionByZero();
        }
        catch(java.lang.ArrayIndexOutOfBoundsException ex1)
        {
           System.out.println("Error Handler : " + ex1.getMessage());
        }
        catch(java.lang.ArithmeticException ex2)
        {
            System.out.println("Error Handler : " + ex2.getMessage());
        }
    }

    public static void main(String[] args) {
        // TODO code application logic here
        HandleDivisionByZero();
    }
}
```

Επιστροφή στο σημείο κλήσης

Όχι κατάλληλος

Κατάλληλος διαχειριστής

Η σύνταξη της Exception

```
try {
        Εκτέλεση εντολών που μπορεί να
         δημιουργήσουν εξαίρεση
}
catch (ExceptionType1 name1)
{
        Διαχείριση της εξαίρεσης τύπου 1
}
catch (ExceptionType2 name2)
{
        Διαχείριση της εξαίρεσης τύπου 2
 }
……..
finally
{
        Εντολές που πάντα θα εκτελεστούν σε ένα
        try .. catch. Συνήθως σε αυτό το σημείο ελέγχουμε
        και απελευθερώνουμε πόρους του προγράμματος
}
```

Unchecked και checked exceptions

Οι εξαιρέσεις διαφοροποιούνται σε δυο κατηγορίες , α) τις μη ελεγχόμενες (unchecked) και β) στις ελεγχόμενες (checked).

Στη πρώτη περίπτωση είναι τα σφάλματα τα οποία μπορεί να συμβούν από λάθος στο προγραμματισμό, π.χ. πρόσβαση εκτός ορίων σε πίνακα, διαίρεση με μηδέν κ.λπ., δηλαδή τα σφάλματα που πιθανόν να μην είχαν συμβεί αν είχαν προηγηθεί οι κατάλληλοι έλεγχοι. Σε αυτή τη περίπτωση, δεν είναι υποχρεωτική η χρήση του try..catch αλλά είναι απαραίτητη για τη διαχείριση του σφάλματος.

Στη δεύτερη περίπτωση είναι τα σφάλματα τα οποία μπορεί να συμβούν εκτός του προγράμματος π.χ. η διαχείριση ενός αρχείου, ή μιας δικτυακής σύνδεσης κ.λπ. Σε αυτή τη περίπτωση, είναι υποχρεωτική η χρήση του try..catch γιατί απαιτούν οι ίδιες οι κλάσεις τη διαχείριση αυτών των εξαιρέσεων.

Δημιουργία νέας εξαίρεσης

```
class myException extends Exception
{
  public String getMessage()
  {
    return "my first exception" ;
  }
}
```

```
public static void DivisionByZero() throws myException
  {

    int x = 5;
    int y = 0;

    if(y == 0)  {
      throw new myException();
    }

    int k = x / y;
    System.out.println(k);
  }
```

```
public static void CallDivisionByZero()
  {
    try
    {
      DivisionByZero();
    }
    catch(myException ex)
    {
     System.out.println("Error Handler : " +ex.getMessage());
    }
  }
```

```
run:
Error Handler : my first exception
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ενεργοποίηση μιας εξαίρεσης

```
public static void DivisionByZero() throws myException
   {

      int x = 5;
      int y = 0;

      if(y == 0)  {
         throw new myException();
      }

      int k = x / y;
      System.out.println(k);
   }
```

Αντικατάσταση με
throw new java.lang.ArithmeticException("You must be careful");

```
run:
Error Handler : You must be careful
BUILD SUCCESSFUL (total time: 0 seconds)
```

Παγίδευση όλων των εξαιρέσεων

```
try {

   // Εκτέλεση εντολών
   // ...

} catch (Exception ex){

    // Το ex μπορεί να είναι
    //οποιουδήποτε τύπου εξαίρεσης
}
```

```
try {

   // Εκτέλεση εντολών
   // ...

}
catch (ExceptionType1 name1)
{
  // Διαχείριση της εξαίρεσης τύπου 1
}
 catch (Exception ex)
{
    // Το ex μπορεί να είναι
    //οποιουδήποτε τύπου εξαίρεσης
}
```