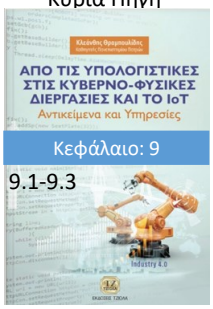
Αντικειμενοστρεφής Προγραμματισμός (Object-Oriented Programming)

(CEID_NNY106)

Διαχείριση Εξαιρέσεων

Exception Handling - Introduction

Κύρια Πηγή



Exception:
an instance or case not conforming to the general rule.

Kleanthis Thramboulidis
Prof. of Software and System Engineering
University of Patras
https://sites.google.com/site/thramboulidiskleanthis/

Java

High-level programming language

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. Wikipedia

Designed by: James Gosling

First appeared: May 23, 1995; 27 years ago

Paradigm: Multi-paradigm: generic, object-oriented (class-based), functional, imperative, reflective, concurrent

NullPointerException

NullPointerExceptionTest.java

```
1
2  public class NullPointerExceptionTest {
3      static Operand op;
4
5      public static void main(String[] args) {
6  //      op = new Operand(1234);
7          System.out.println(op.getVal());
8      }
9  }
10
11 class Operand {
12     private int val;
13
14     public Operand(int val) {
15         this.setVal(val);
16     }
17     public void setVal(int val) {
18         this.val=val;
19     }
20     public int getVal() {
21         return val;
22     }
23 }
```

- Exception in thread "main" java.lang.NullPointerException: Cannot invoke "Operand.getVal()" because "NullPointerExceptionTest.op" is null
- at NullPointerExceptionTest.main(NullPointerExceptionTest.java:7)

Console

```
<terminated> NullPointerExceptionTest [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe  (Apr 13, 2022, 10:07:38 AM – 10:07:39 AM)
Exception in thread "main" java.lang.NullPointerException: Cannot invoke "Operand.getVal()" because "NullPointerExceptionTest.op" is null
        at NullPointerExceptionTest.main(NullPointerExceptionTest.java:7)
```

ArithmeticException

DivideByZero.java

```
1
2  public class DivideByZero {
3
4      public static void main(String[] args) {
5          var d1 = quotient(10,2);
6          System.out.println("quotient(10,2) ->" + d1);
7
8          d1 = quotient(10,0);
9          System.out.println("quotient(10,0) ->" + d1);
10     }
11
12     public static int quotient (int numerator, int denominator)
13     {
14     return numerator / denominator;
15     }
16
17 }
```

Console

```
<terminated> DivideByZero [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe
quotient(10,2) ->5
Exception in thread "main" java.lang.ArithmeticException: / by zero
        at DivideByZero.quotient(DivideByZero.java:14)
        at DivideByZero.main(DivideByZero.java:8)
```


Οργάνωση Διάλεξης

- **Έλεγχος σφαλμάτων στον προγραμματισμό**
- Κλασσικός χειρισμός
  - stack
  - readFile()
- Ανάγκη για Διαχείριση εξαιρέσεων
- Εξαιρέσεις στην καθημερινή πρακτική
  - Βασικές έννοιες

Stack, readFile() - Μία απλή υλοποίηση

```
#define MAXVAL  100
int sp = 0;
double val[MAXVAL];
```

```
void push(double f)
{
val[sp++] = f;
}
```

```
double pop(void)
{
return val[--sp];
}
```

Πηγή : "The C programming Language" Brian Kernighan

Η readFile διαβάζει ένα αρχείο στη μνήμη

```
readFile(filename){
    openTheFile;
    determine its size;
    allocate that much memory;
    read the file into memory;
    closeTheFile;
    }
```

Error handling ?

readFile() - Μειονεκτήματα

- Η read_file() μοιάζει απλή αλλά **αγνοεί μια σειρά από πιθανά σφάλματα**:
  - file can't be opened
  - the length of the file can't be determined
  - enough memory can't be allocated
  - read fails
  - file can't be closed

```
readFile(filename){
    openTheFile;
    determine its size;
    allocate that much memory;
    read the file into memory;
    closeTheFile; }
```

χρειάζεται πολύς κλασσικός κώδικας για να
–ανιχνεύσετε
–αναφέρετε στον χρήστη
–χειριστείτε το κάθε σφάλμα.

Σφάλματα προκαλούνται από

- *προγραμματιστικά λάθη*
  - μη επιτρεπτή αριθμητική πράξη, uninitialized data
- *μη αποδεκτή διάρθρωση (configuration)*
  - δε βρέθηκε αρχείο
- *ελλιπείς πόρους συστήματος*
  - δεν υπάρχει αρκετή μνήμη
- *ελαττωματικό υλικό (hardware)*
  - απρόβλεπτα συμπτώματα και συνέπειες

Χαρακτηριστικά σφαλμάτων

- ***απρόσμενα***
  - μπορούν να συμβούν (σχεδόν) σε οποιοδήποτε σημείο του προγράμματος.
- ***αθροιστικά***
- χρειάζονται ***άμεση επέμβαση***
- συνήθως ***δεν μπορούν να αγνοηθούν*** χωρίς σοβαρές συνέπειες
- ***Δεν έχουν εννοιολογική σχέση*** με υπόλοιπο κώδικα

Οργάνωση Διάλεξης

- Έλεγχος σφαλμάτων στον προγραμματισμό
- **Κλασσικός χειρισμός**
  - stack
  - readFile()
- Ανάγκη για Διαχείριση εξαιρέσεων
- Εξαιρέσεις στην καθημερινή πρακτική
  - Βασικές έννοιες

Error Handling στην Stack

```
#define MAXVAL  100
int sp = 0;
double val[MAXVAL];
```

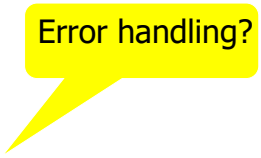
```
void push(double f)
{
if(sp<MAXVAL)
        val[sp++] = f;
else
        printf("error: stack full, can't push %g\n", f);
}

double pop(void)
{
if(sp>0)
        return val[--sp];
else {
        printf("error: stack empty\n");
        return 0.0;
}
```

Πηγή : "The C programming Language" Brian Kernighan

Error Handling στην open()

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#include <stdlib.h>
#include <stdio.h>

int main() {

        int fh;
        char buffer[65];
        int gotten;

        fh = open("abc.txt",O_RDONLY);
        printf ("File handle %d\n",fh);
        while (gotten = read(fh,buffer,64)) {
                buffer[gotten] = '\0';
                printf("******%s",buffer);
                }
        }
```

```
/*
A file opened with 'open' is accessed in
a binary mode ... "read X bytes" stuff.

Note that the buffer is one character
longer that the maximum read to allow
space for us to add a null and turn it
into a string */
```



open() … in case of error

- Function: int **open** (*const char *filename, int flags*[*, mode_t mode*])
  creates and returns a new file descriptor for the file named by *filename*.
- The normal return value from open is a non-negative integer file descriptor.
- In the case of an error, a value of -1 is returned instead.
- In addition to the usual file name errors (see next slide), the following **errno** error conditions are defined for this function:
- EACCES
  - The file exists but is not readable/writable as requested by the *flags* argument, the file does not exist and the directory is unwritable so it cannot be created.
- EEXIST
  - Both O_CREAT and O_EXCL are set, and the named file already exists.
- EINTR
  - The open operation was interrupted by a signal. See Interrupted Primitives.
- EISDIR, EMFILE, ENFILE, ENOENT, EROFS

open() … usual file name errors

- Functions that accept file name arguments usually detect these errno error conditions relating to the file name syntax or trouble finding the named file.

These errors are referred as the *usual file name errors*.

- EACCES
  - The process does not have search permission for a directory component of the file name.
- ENAMETOOLONG
  - This error is used when either the total length of a file name is greater than PATH_MAX, or when an individual file name component has a length greater than NAME_MAX. See Limits for Files. In the GNU system, there is no imposed limit on overall file name length, but some file systems may place limits on the length of a component.
- ENOENT
  - This error is reported when a file referenced as a directory component in the file name doesn't exist, or when a component is a symbolic link whose target file does not exist. See Symbolic Links.
- ENOTDIR
  - A file that is referenced as a directory component in the file name exists, but it isn't a directory.
- ELOOP
  - Too many symbolic links were resolved while trying to look up the file name. The system has an arbitrary limit on the number of symbolic links that may be resolved in looking up a single file name, as a primitive way to detect loops. See Symbolic Links.

Error Handling in readFile()

```
errorCodeType read_file {
    errorCode = 0;
    openTheFile;
    if (theFileIsOpen) {
        determine_file_length();
        if (gotTheFileLength) {
            allocate_that_memory();
            if (gotEnoughMemory) {
                read_file_to_memory();
                if (readFailed)
                    errorCode = -1;
            }
            else
                errorCode = -2;
        } else {
            errorCode = -3;
        }
        closeTheFile;
        if (theFileDidntClose &&
            errorCode == 0)
                errorCode = -4;
        else
            errorCode = errorCode and -4;
    }
    else {
        errorCode = -5;
    }
    return errorCode;
    }
```

Error Handling in readFile()

- **7 γραμμές (bold) αυξήθηκαν σε 29**
  πολύς κώδικας
  - ανίχνευσης,
  - αναφοράς στην καλούσα μέθοδο
- **αρχικές 7 γραμμές χάνονται στην ακαταστασία**
- λογική ροή του κώδικα
  - έχει χαθεί μέσα στο σωρό
  - δύσκολο να δεις αν ο κώδικας κάνει σωστή δουλειά
    - π.χ. κλείνει το αρχείο αν μπορέσει να δεσμεύσει μνήμη;

400% αύξηση!

Πολλοί προγραμματιστές "λύνουν" το πρόβλημα απλά αγνοώντας το.
Τα λάθη εμφανίζονται μαζί με το επόμενο crash.

Κλασσικός χειρισμός λαθών – correctness vs. clarity

π.χ. επιστρεφόμενες τιμές κωδικών σφάλματος

Σοβαρά μειονεκτήματα

- περιπλέκει τον κώδικα
- επιρρεπής σε προγραμματιστικά λάθη
  - π.χ. αγνόηση κωδικών σφάλματος.

- trade-off between
  - correctness (checking for all errors)
  - clarity (not cluttering the basic flow of code with many error checks)

Οργάνωση Διάλεξης

- Έλεγχος σφαλμάτων στον προγραμματισμό
- Κλασσικός χειρισμός
  - stack
  - readFile()
- **Ανάγκη για Διαχείριση εξαιρέσεων**
- Εξαιρέσεις στην καθημερινή πρακτική
  - Βασικές έννοιες

Ανάγκη για ...

- a clean way to check for errors without cluttering the code

- **a mechanism to achieve**
  - **correctness** (checking for all errors)
  - **clarity** (not cluttering the basic flow of code with many error checks)

Εξαίρεση

- είναι
  - γεγονός που συμβαίνει στη διάρκεια εκτέλεσης ενός προγράμματος
- διασπά
  - την φυσική ροή εκτέλεσης των εντολών
- Ο όρος δεν αναφέρεται μόνο σε λάθη αλλά σε συμβάντα των οποίων η εμφάνιση απαιτεί ειδικό χειρισμό

Παραδείγματα

αναμενόμενο αλλά σπάνιο γεγονός, π.χ.

- διαίρεση με μηδέν
- τέλος αρχείου
- αναφορά σε στοιχείο πίνακα έξω από τα όρια του

Exception handling

- είναι ο μηχανισμός της γλώσσας που
  - **παρέχει** ένα **ειδικό τρόπο μεταφοράς** του **ελέγχου** μεταξύ διαδικασιών (διαφορετικό από αυτόν της κλήσης συνάρτησης και επιστροφής)
  - **επιτρέπει** την **ανταλλαγή πληροφοριών** ανάμεσα σε διαδικασίες διαφορετικών επιπέδων

"is the process of responding to the occurrence of *exceptions* – anomalous or exceptional conditions requiring special processing - during the execution of a program."
wikipedia

Βασικός στόχος μηχανισμού χειρισμού εξαιρέσεων

- να επιτρέψει την συγγραφή μεθόδων οι οποίες είναι σε θέση
  - **να ανιχνεύσουν** τις εμφανίσεις εξαιρετικών περιπτώσεων (π.χ. λαθών) τις οποίες αν δεν είναι σε θέση να αντιμετωπίσουν, μπορούν να μεταδώσουν στην καλούσα διαδικασία
  - **να συλλάβουν και να διαχειριστούν** εξαιρέσεις που μεταδίδονται από καλούμενες διαδικασίες

Οργάνωση Διάλεξης

- Έλεγχος σφαλμάτων στον προγραμματισμό
- Κλασσικός χειρισμός
  - stack
  - readFile()
- Ανάγκη για Διαχείριση εξαιρέσεων
- **Εξαιρέσεις στην καθημερινή πρακτική**
  - Βασικές έννοιες

Exceptions στα Goody's

- Η Μαίρη όταν προσλήφθηκε πήρε οδηγίες για το **πως θα ενεργεί σε απόκριση ορισμένων μηνυμάτων** που θα δέχεται ή συμβάντων που θα αναγνωρίζει.
- Θα θεωρήσουμε ένα από τα μηνύματα αυτά και συγκεκριμένα το «ένα τοστ με ζαμπόν-τυρί παρακαλώ»
- Είναι η συμπεριφορά της Μαίρης πλήρως ορισμένη;

- Στην εκτέλεση της ενέργειας «πάρε τυρί» διαπιστώνει ότι το τυρί έχει τελειώσει.
  - Το συμβάν αυτό χαρακτηρίζεται ως **εξαιρετική περίπτωση** που όμως είναι πιθανή καθώς η Μαίρη προετοιμάζει το τοστ
- αν η Μαίρη δεν έχει ενημερωθεί πως να την αναγνωρίσει και να την χειριστεί το σύστημα κινδυνεύει να μεταπέσει σε **κατάσταση απροσδιόριστης συμπεριφοράς**.

Exception: an instance or case not conforming to the general rule.

Αναγνώριση Εξαίρεσης (Exception)

- Υποθέτουμε πως η Μαίρη αναγνωρίζει το συμβάν αυτό ως εξαιρετική περίπτωση.
- Ποια μπορεί να είναι η συμπεριφορά της στην περίπτωση αυτή;

- Σενάριο 1 – **Αναγνώριση** και **διαχείριση** εξαίρεσης
- Σενάριο 2- Αναγνώριση, **μετάδοση**, **σύλληψη** και διαχείριση εξαίρεσης
- Σενάριο 3 - Αναγνώριση, μετάδοση, μετάδοση, … σύλληψη εξαίρεσης

Διαχείριση Εξαίρεσης – 1ο σενάριο

- Η Μαίρη έχει την γνώση να χειριστεί τη συγκεκριμένη εξαίρεση, πρέπει να εκτελέσει ένα σύνολο από ενέργειες όπως «πάρε από το ψυγείο της αποθήκης μια πλάκα τυρί», «κόψε το σε φέτες, », κ.λ.π. Λέμε πως

catch

- η Μαίρη αναγνωρίζει την εξαίρεση και την αντιμετωπίζει.
- Δεν είναι απαραίτητο να ενημερώσει τον Γιώργο (προϊστάμενος τμήματος) για την εξαιρετική αυτή περίπτωση.

Διαχείριση Εξαίρεσης – 2ο σενάριο

- Ο Γιώργος δεν θέλει να αναθέσει στην Μαίρη την αντιμετώπιση της εξαιρετικής αυτής περίπτωσης για διάφορους λόγους. Στην περίπτωση αυτή η Μαίρη αναγνωρίζει την εξαιρετική περίπτωση, δημιουργεί ένα μήνυμα όπως «δεν υπάρχει τυρί» (ένα αντικείμενο εξαίρεσης θα έλεγα) και το στέλνει στον Γιώργο που είναι ο προϊστάμενος του τμήματος. Στην περίπτωση αυτή λέμε πως

- **η Μαίρη ανιχνεύει την εξαιρετική περίπτωση, εγείρει ένα αντικείμενο εξαίρεσης (throw an exception) και την μεταδίδει (throws).**
- Ο Γιώργος θα πρέπει να **συλλάβει (catch) την εξαίρεση και να την αντιμετωπίσει κατάλληλα.**

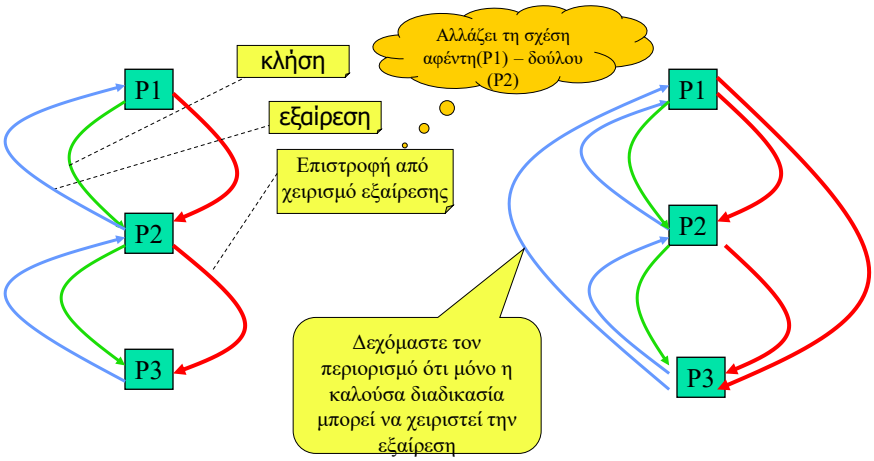
Διαχείριση Εξαίρεσης – 3ο σενάριο

- όπως η επιλογή Νο 2 με τη διαφορά πως αν ο Γιώργος δεν μπορεί να χειρισθεί την εξαίρεση την μεταδίδει (**throws)**

πολιτική διαχείρισης εξαιρετικών περιπτώσεων στα Goody'

- Θέματα όπως
  - «σε ποιόν θα αναφέρει η Μαίρη την εξαίρεση;» «σε αυτόν που της ζήτησε την εξυπηρέτηση ή πάντα στον Γιώργο;»
  - «συνεχίζει την δουλειά της μετά την έγερση της εξαίρεσης ή περιμένει να ολοκληρωθεί η αντιμετώπιση της πρώτα;», κ.λ.π.
- καθορίζουν
  - τη **πολιτική διαχείρισης των εξαιρετικών περιπτώσεων** στη λειτουργία ενός συστήματος.

Πιθανές εκδοχές χειρισμού εξαιρέσεων



Η σχεδίαση μιας ευέλικτης μεθόδου χειρισμού εξαιρέσεων είναι πολύπλοκη υπόθεση

Πηγή: Horowitz 84

Μοντέλα χειρισμού εξαιρέσεων

Υπάρχουν διάφορα μοντέλα χειρισμού εξαιρέσεων

- μοντέλο επαν-ενεργοποίησης
- μοντέλο τερματισμού
- μπορούν να εγερθούν εξαιρέσεις μέσα σε ένα χειριστή εξαιρέσεων; – συνέπειες
- μπορεί οι εξαιρέσεις να έχουν παραμέτρους;
- ...