

10 Χειρισμός Εξαιρέσεων

Συγγραφή ανθεκτικού κώδικα

Σφάλματα Κατά την Εκτέλεση

- Τα λογικά σφάλματα είναι δύσκολο να εντοπιστούν και να αντιμετωπιστούν.
- Μερικά σφάλματα προκύπτουν από συνθήκες που δεν είναι στον έλεγχο του προγραμματιστή. Πχ
 - Απόπειρα εγγραφής αρχείου σε συσκευή μόνον ανάγνωσης ή χωρίς ελεύθερο χώρο.
 - Διακοπή της σύνδεσης του δικτύου.
- Θα δούμε μεθόδους αντιμετώπισης σφαλμάτων κατά την εκτέλεση του προγράμματος (runtime errors).

251

Παράδειγμα Πελάτη-Εξυπηρέτη

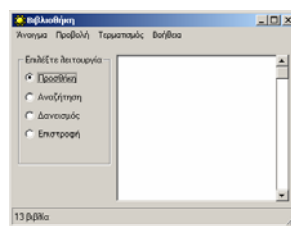
- Εξυπηρέτης: υλοποιεί τη βασική λειτουργικότητα του προγράμματος
 - Παρέχει υπηρεσίες: κατασκευή, προσθήκη, κατάλογος
- Πελάτης: μια διεπαφή χρήστη (γραφική ή διαταγών) με λειτουργίες που καλούν τις υπηρεσίες (μεθόδους) του εξυπηρέτη

```
public Library()

public void addBook(
    String author,
    String title,
    String bookCode)

public String toString()

public Book[] searchBooks(
    String criteria)
```



252

Αλληλεπίδραση Εξυπηρέτη / Πελάτη

- Μπορεί ο εξυπηρέτης να θεωρήσει ότι οι πελάτες των υπηρεσιών του θα συμπεριφέρονται κατά τον αναμενόμενο και προδιαγεγραμμένο τρόπο;
 - Ή αντίθετα, θα πρέπει να παίρνει ως δεδομένο ότι είναι κακόβουλο;
- Το πιθανότερο σενάριο είναι κάπου ενδιάμεσα.
- Μήπως και οι πελάτες θα πρέπει να ελέγχουν τα δεδομένα που τους επιστρέφει ο εξυπηρέτης;

253

Ζητήματα Προς Αντιμετώπιση

- Πόσο εξαντλητικοί πρέπει να είναι οι έλεγχοι των απαιτήσεων του πελάτη μέσα στις μεθόδους του εξυπηρέτη;
- Πώς θα ανακοινώνει ο εξυπηρέτης τα σφάλματα στους πελάτες του;
- Πώς θα αντιλαμβάνεται ο πελάτης την αποτυχία ενός αιτήματός του;
 - Τίνος είναι η ευθύνη;
- Πώς θα ξεπερνάει ο πελάτης την άρνηση του εξυπηρέτη;
 - Με διακοπή εκτέλεσης (crash) του προγράμματος;

254

Έλεγχος Τιμών Παραμέτρων

- Οι παράμετροι είναι ένα ευαίσθητο σημείο του εξυπηρέτη.
 - Έχουν «νόμιμες» τιμές;
- Ο έλεγχος των τιμών των παραμέτρων μέσα στη μέθοδο του εξυπηρέτη είναι ένα μέσο άμυνας.
 - Οι τιμές των παραμέτρων προδιαγράφονται από την τεκμηρίωση ως προς τον τύπο τους και ως προς το εύρος αποδεκτών τιμών τους.
 - Η μέθοδος του εξυπηρέτη, προτού τις χρησιμοποιήσει, καλό είναι να ελέγξει ότι είναι «νόμιμες».

255

Αναφορά Σφάλματος

- Πώς θα αναφέρει ο εξυπηρετής τυχόν παράνομες παραμέτρους;
 - Προς τον χρήστη;
 - Υπάρχει άνθρωπος εκεί έξω;
 - Μπορεί να αντιμετωπίσει το πρόβλημα;
 - Προς το πρόγραμμα πελάτη;
 - Επιστρέφει μια τιμή που ο πελάτης αναγνωρίζει ως ένδειξη σφάλματος
 - Πχ `null`, `(-1)` ή κάποια άλλη τιμή -μη αναμενόμενη- σε κανονικές περιστάσεις
 - **Εγείρει μια εξαίρεση** (raise exception)

256

Αντίδραση του Πελάτη

- Ελέγχει την επιστρεφόμενη ένδειξη σφάλματος
 - Προσπαθεί να ανανήψει
 - Αποφεύγει τη διακοπή εκτέλεσης (crash) που συνεπάγεται ολική αποτυχία του προγράμματος.
- Αγνοεί την επιστρεφόμενη τιμή
 - Κακό, αλλά δεν μπορούμε να του επιβάλλουμε να πράξει διαφορετικά.
- Ο μηχανισμός των εξαιρέσεων είναι προτιμότερος διότι μια εξαίρεση δεν περνάει απαρατήρητη.

257

Έγερση Εξαίρεσης

- Απαιτεί ειδική υποστήριξη από τη γλώσσα και το περιβάλλον εκτέλεσης.
- Δεν χρειάζεται να επιστρέφονται «ειδικές» τιμές που υποδηλώνουν εκδήλωση σφάλματος.
- Ο πελάτης δεν μπορεί να αγνοήσει το σφάλμα.
 - Αν ο πελάτης δεν κάνει κάτι, το πρόγραμμα διακόπτεται.
- Ενθαρρύνεται η χρήση ενεργειών αντιμετώπισης του σφάλματος και ανάνηψης.

258

Έγερση Εξαίρεσης στη Java

Τεκμηρίωση για την πιθανότητα έγερσης εξαίρεσης

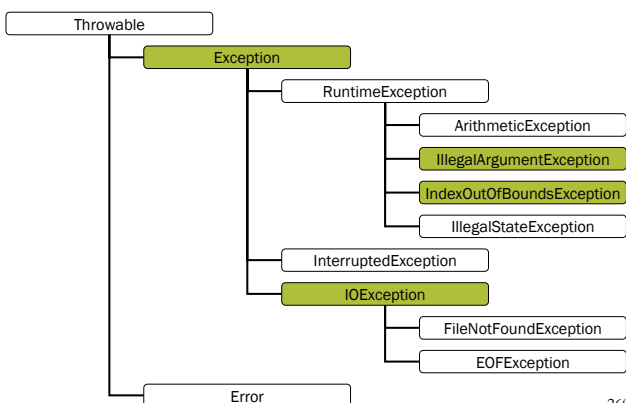
```
/**
 * Θέτει τον κώδικα ISBN για αυτό το βιβλίο.
 * @param code ο νέος κωδικός του βιβλίου
 * @throws Exception αν ο κωδικός δεν έχει
 * το σωστό μήκος.
 */
public void setISBNcode(String code)
{
    if (code.length != 10)
        throw new IllegalArgumentException (
            "Σφάλμα στο μήκος του ISBN");
    this.code = code;
}
```

Έγερση της εξαίρεσης

Τύπος εξαίρεσης

259

Τύποι Εξαιρέσεων στη Java



260

Αποτέλεσμα της Έγερσης μιας Εξαίρεσης

- Η μέθοδος μέσα στην οποία εγείρεται η εξαίρεση, τερματίζει άμεσα.
- Δεν επιστρέφεται επιστρεφόμενη τιμή.
- Η εκτέλεση δεν συνεχίζεται αυτόματα στην καλούσα μέθοδο, όπως αν συνέβαινε ο κανονικός τερματισμός της μεθόδου.
 - Σκόπιμα: για να μην μπορεί να συνεχίσει η εκτέλεση και η εξαίρεση να περάσει απαρατήρητη.
- Αν δεν 'συλληφθεί' η εξαίρεση, το πρόγραμμα τερματίζει ανεπιτυχώς.

261

Έλεγχος Παραμέτρων

```

/**
 * Επιστρέφει βιβλίο με συγκεκριμένο κωδικό.
 * @param bookCode String ο κωδικός του βιβλίου
 * @throws NullPointerException αν ο κωδικός είναι null
 * @throws IllegalArgumentException αν ο κωδικός είναι κενός
 */
public Book getBook(String bookCode) {
    if (bookCode == null)
        throw new NullPointerException("null στην getBook");

    if (bookCode.trim().length() == 0)
        throw new IllegalArgumentException(
            "Κενό κλειδί βιβλίου στην getBook");

    return (Book) this.get(bookCode);
}

```

262

Ματαίωση Κατασκευής Αντικειμένου

```

/**
 * Κατασκευαστής
 */
public Book (String title, String author, String isbn) {
    if (title == null) title = "";
    if (author == null) author = "";
    if (isbn == null) isbn = "";

    this.title = title.trim();
    this.author = author.trim();
    this.isbn = isbn.trim();

    if (this.title.length() == 0 ||
        this.author.length() == 0 ||
        this.isbn.length() == 0)
        throw new IllegalStateException(
            "Τίτλος, συγγραφέας ή ISBN ήταν κενά.");
}

```

263

Μη Χειριζόμενες Εξαιρέσεις

- Αν εγείρουμε μια εξαίρεση, και δεν έχουμε υλοποιήσει χειριστή, το πρόγραμμα θα διακοπεί με μήνυμα:

```

java.lang.IllegalArgumentException: Περίμενα αριθμό
at untitled8.Untitled1.main(Untitled1.java:5)
Exception in thread "main"

```

- Σε μερικές περιπτώσεις οι συνθήκες που προκαλούν την έγερση μιας εξαίρεσης μπορούν να αντιμετωπιστούν και το πρόγραμμα να προβεί σε διορθωτική ενέργεια, δίχως να απαιτείται ο τερματισμός του.

264

Δήλωση Χειριζόμενων Εξαιρέσεων

- Αν μια μέθοδος εγείρει μια εξαίρεση που σκοπεύουμε να την αντιμετωπίσουμε (χειριστούμε), πρέπει να το δηλώσει στην επικεφαλίδα της με τη λέξη **throws**.

```

public void saveTo(String fname) throws IOException {
    ...
}

```

- Μην ξεχνάτε και την ετικέτα **@throws** στην javadoc τεκμηρίωση.

265

Χειρισμός Εξαιρέσεων

- Εντολές που μπορεί να εγείρουν εξαίρεση τις περικλείουμε σε ένα σώμα **try**.

```

try {
    addressbook.saveTo(filename);
}
catch (IOException e) {
    System.out.println(
        "Δεν είναι δυνατή αποθήκευση στο " + filename);
}

```

- Αν δεν εγερθεί εξαίρεση εκτελούνται οι εντολές στο σώμα **try**.
- Αν εγερθεί εξαίρεση, εκτελούνται οι εντολές του σώματος **catch**.

266

Χειρισμός Πολλαπλών Εξαιρέσεων

```

try {
    ...
    ref.process();
    ...
}
catch (EOFException e) {
    // Take action on an end-of-file exception.
    ...
}
catch (FileNotFoundException e) {
    // Take action on a file-not-found exception.
    ...
}

```

267

Ανάνηψη από Εξαιρέση

```
// Try to save the address book.
boolean successful = false;
int attempts = 0;
do {
    try {
        addressbook.saveTo(filename);
        successful = true;
    }
    catch (IOException e) {
        System.out.println("Unable to save to " + filename);
        attempts++;
        if (attempts < MAX_ATTEMPTS)
            filename = an alternative file name;
    }
} while (!successful && attempts < MAX_ATTEMPTS);
if (!successful) {
    Report the problem and give up;
}
```

268

Ισχυρισμοί και Εξαιρέσεις

- Η έγερση μιας εξαιρέσης είτε θα τερματίσει το πρόγραμμα, είτε θα προκαλέσει την εκτέλεση ενός χειριστή.
- Με τους ισχυρισμούς μπορούμε να εγείρουμε εξαιρέση `AssertionError`.
- Οι ισχυρισμοί χρησιμοποιούνται κατά τη φάση της ανάπτυξης (αποσφαλμάτωση).

Υπενθύμηση

269

Είσοδος / Έξοδος

- Η έξοδος, αλλά κυρίως η είσοδος μπορεί να παράγει σφάλματα, καθώς έχει να κάνει με αλληλεπίδραση με το εξωτερικό περιβάλλον.
- Χρησιμοποιούμε το πακέτο `java.io`
- Σχετική εξαιρέση `java.io.IOException`
- Εισαγωγικό μάθημα
<http://java.sun.com/docs/books/tutorial/essential/io/index.html>

270

Έξοδος Κειμένου

- Χρησιμοποιούμε την κλάση `FileWriter`

```
import java.io.*;
...
try {
    FileWriter writer = new FileWriter("name of file");
    while (there is more text to write) {
        ...
        writer.write(next piece of text);
        ...
    }
    writer.close();
}
catch(IOException e) {
    something went wrong with accessing the file
}
```

271

Είσοδος Κειμένου

- Χρησιμοποιούμε την κλάση `BufferedReader` για ανάγνωση ανά γραμμή και όχι ανά χαρακτήρα.

```
try {
    BufferedReader reader = new BufferedReader(new FileReader("fname"));
    String line = reader.readLine();
    while (line != null) {
        do something with line
        line = reader.readLine();
    }
    reader.close();
}
catch (FileNotFoundException e) {
    the specified file could not be found
}
catch (IOException e) {
    something went wrong with reading or closing
}
```

272

Όροι Ενότητας 10

- Εξαιρέσεις
- Έγερση εξαιρέσης
- Χειρισμός εξαιρέσης
- Τύποι εξαιρέσεων
- Είσοδος κειμένου
- Έξοδος κειμένου

Τεκμήριο Ενότητας

- `throw`
- `throws`
- `try-catch`

273