

Χαροκόπειο Πανεπιστήμιο  
Τμήμα Πληροφορικής και Τηλεματικής

Αντικειμενοστρεφής Προγραμματισμός 2  
Θέμα εργασίας

**Τζουκ μποξ**

Τελευταία ενημέρωση: 27 Δεκεμβρίου 2022

## ΘΕΜΑ

Να σχεδιαστεί και να υλοποιηθεί από ομάδα δύο ή τριών φοιτητών μια εφαρμογή αναπαραγωγής αρχείων ήχου. Η γλώσσα υλοποίησης της εφαρμογής είναι η Java. Λεπτομερείς προδιαγραφές της εφαρμογής περιγράφονται στην ενότητα 1.

### Παραδοτέα, ημερομηνίες και βαθμολόγηση

Η εργασία θα παραδοθεί τμηματικά σε 2 μέρη. Τα ζητούμενα του κάθε μέρους περιγράφονται στην ενότητα 2.

| Μέρος             | Μονάδες | Bonus | Ημ/νία παράδοσης   |
|-------------------|---------|-------|--------------------|
| Μέρος 1           | 2.0     | —     | 18 Δεκεμβρίου 2022 |
| Μέρος 2           | 2.0     | —     | 20 Ιανουαρίου 2023 |
| Προαιρετικό μέρος | —       | 1.0   | 20 Ιανουαρίου 2023 |
| Συνολική εργασία  | 4.0     | 1.0   |                    |

Μετά την παράδοση και των δύο μερών θα ακολουθήσει δειγματοληπτική προφορική εξέταση.

Για τα διάφορα τμήματα της εργασίας πρέπει να παραδίδεται εμπρόθεσμα από κάθε ομάδα ο αντίστοιχος κώδικας σε ηλεκτρονική μορφή, καθώς και σαφείς οδηγίες για την παραγωγή ενός εκτελέσιμου προγράμματος και την επίδειξη της λειτουργίας του αντίστοιχου τμήματος. Καθυστερημένες ασκήσεις θα βαθμολογούνται με μικρότερο βαθμό, αντιστρόφως ανάλογα προς το χρόνο καθυστέρησης. Η μορφή και τα περιεχόμενα των παραδοτέων πρέπει να συμφωνούν με τις οδηγίες που δίνονται στην ενότητα 5.

## 1 Προδιαγραφές

### 1.1 Λίστα τραγουδιών

Η εφαρμογή αναπαράγει τα τραγούδια από μια λίστα τραγουδιών. Μια λίστα τραγουδιών δημιουργείται με δύο τρόπους:

1. Για την δημιουργία της λίστας ο χρήστης δίνει το όνομα ενός φακέλου μέσα στον οποίο το πρόγραμμα θα κοιτάξει για τραγούδια (υποστηριζόμενα αρχεία ήχου). Ο φάκελος μπορεί να περιέχει κι άλλα αρχεία που δεν είναι αρχεία ήχου. Αυτά τα αρχεία δεν θα πρέπει να συμπεριληφθούν στην λίστα που θα δημιουργηθεί. Η αναγνώριση των αρχείων ήχου γίνεται από την κατάληξή τους.
2. Ο δεύτερος τρόπος που μπορεί να δημιουργηθεί μια λίστα τραγουδιών είναι ο χρήστης να δώσει το όνομα ενός αρχείου m3u. Το αρχείο m3u είναι αρχείο χαρακτήρων και η

διάταξη του είναι πολύ απλή. Η γραμμογράφηση ενός τέτοιου αρχείου περιγράφεται στο σύνδεσμο: <http://forums.winamp.com/showthread.php?t=65772>.

## 1.2 Επιλογή τραγουδιού

- Ο χρήστης μπορεί να επιλέξει ένα τραγούδι από την λίστα για να αρχίσει η αναπαραγωγή του.
- Θα πρέπει να μπορεί να το σταματήσει πριν τελειώσει.
- Θα πρέπει να μπορεί να το διακόψει προσωρινά, και να το συνεχίσει από το σημείο που το διέκοψε ή από την αρχή.
- Θα πρέπει να μπορεί να επιλέξει να πάει στο επόμενο τραγούδι πριν τελειώσει το τραγούδι. Το επόμενο τραγούδι αποφασίζεται από μια στρατηγική που περιγράφεται με περισσότερες λεπτομέρειες στην ενότητα 1.2.1.

Αν επιλεγεί κάποιο άλλο αρχείο για αναπαραγωγή ενώ παίζει ήδη κάποιο τραγούδι τότε το τραγούδι που παίζει σταματάει και αρχίζει το επιλεγμένο.

### 1.2.1 Στρατηγική επιλογής επόμενου τραγουδιού

Η εφαρμογή πρέπει να υποστηρίζει τουλάχιστον τρεις στρατηγικές επιλογής του επόμενου τραγουδιού.

1. Το επόμενο τραγούδι που εμφανίζεται στην λίστα μετά το τραγούδι που μόλις τελείωσε.
2. Τυχαία επιλογή από τα τραγούδια που δεν έχουν παίζει μέχρι τώρα.
3. Το ίδιο τραγούδι μέχρι να το αλλάξει ο χρήστης.

Ο χρήστης θα πρέπει να μπορεί να αλλάξει την στρατηγική δυναμικά κατά την διάρκεια της εφαρμογής.

## 1.3 Αναπαραγωγή ήχου

Τα αρχεία ήχου που θα υποστηρίζει η εφαρμογή είναι τύπου mp3. Για να μπορέσει να αναπαράγει τα τραγούδια στα ηχεία του υπολογιστή θα πρέπει να χρησιμοποιήσει μια βοηθητική βιβλιοθήκη που δίνεται στα πλαίσια της εργασίας. Η πρόσβαση στην βιβλιοθήκη γίνεται μέσω μιας ορισμένης διεπαφής που περιγράφεται στην ενότητα 3.

## 1.4 Διεπαφή χρήστη

Ο χρήστης διαχειρίζεται την εφαρμογή μέσω παραθυρικής διεπαφής. Κατ' ελάχιστο, η εφαρμογή θα πρέπει να παρέχει:

- μια λίστα με όλα τα διαθέσιμα προς επιλογή τραγούδια και την δυνατότητα της επιλογής ενός στοιχείου από την λίστα,
- κουμπιά για την έναρξη της αναπαραγωγής, τον τερματισμό και την προσωρινή διακοπή και την συνέχιση του επιλεγμένου τραγουδιού,
- κουμπί για την μετάβαση στο επόμενο τραγούδι σύμφωνα με την στρατηγική επιλογής
- ετικέτες με λεκτικά για την κατάσταση της εφαρμογής, πχ. αν αναπαράγει κάποιο τραγούδι και τα στοιχεία του, αν δεν αναπαράγει κάτι και περιμένει κάποια επιλογή, κτλ.
- τρόπος επιλογής του ονόματος του φακέλου με τα τραγούδια προς αναπαραγωγή,
- τρόπος επιλογής του αρχείου m3u με τα τραγούδια προς αναπαραγωγή,
- επιλογή για τερματισμό του προγράμματος.

### 1.4.1 Καταγραφή γεγονότων

Πέρα από την παραθυρική διεπαφή, η εφαρμογή πρέπει να εκτυπώνει στην κονσόλα σημαντικά γεγονότα για την κατάσταση της. Τέτοια γεγονότα περιλαμβάνουν όλα τα σφάλματα κατά την διάρκεια εκτέλεσης της εφαρμογής, βασικές αποφάσεις της εφαρμογής, όπως πότε ξεκινάει ή σταματάει μια αναπαραγωγή, ποια αρχεία αγνοήθηκαν γιατί δεν αναγνωρίστηκαν σαν αρχεία ήχου κτλ.

## 2 Μέρη εργασίας

### 2.1 Μέρος 1

Το πρώτο μέρος δεν περιέχει παραθυρική εφαρμογή. Η εφαρμογή είναι μέσω της κονσόλας.

Το πρόγραμμα δέχεται μέχρι 2 ορίσματα. Το πρώτο όρισμα είναι είτε αρχείο ήχου είτε αρχείο m3u. Το δεύτερο όρισμα είναι μια από τις στρατηγικές επιλογής random, order ή loop.

Για παράδειγμα, αν η εφαρμογή είναι στο jukebox.jar η παρακάτω εντολή

```
java -jar jukebox.jar parakia.mp3
```

θα αναπαράγει το τραγούδι `parakia.mp3` μέχρι να τελειώσει και θα τερματίσει. Ο χρήστης δεν έχει την δυνατότητα να το διακόψει και να το συνεχίσει και ο μόνος τρόπος να αποφύγει να ακούσει όλο το τραγούδι είναι τερματίζοντας την εφαρμογή. Η εντολή

```
java -jar jukebox.jar parakia.mp3 loop
```

θα κάνει το ίδιο με την προηγούμενη εντολή αλλά όταν τελειώσει το τραγούδι θα ξεκινήσει πάλι από την αρχή.

Η εντολή

```
java -jar jukebox.jar best.m3u random
```

διαβάσει τα τραγούδια που υπάρχουν στην λίστα `best.m3u` και τα αναπαράγει σύμφωνα με την στρατηγική `random`. Αν δεν δίνεται στρατηγική τότε εκτελούνται τα τραγούδια με την σειρά που γράφονται στην λίστα. Το ίδιο συμβαίνει και αν η στρατηγική επιλογής είναι `order`. Ο χρήστης θα πρέπει να ακούσει όλο το τραγούδι και όλη την λίστα των τραγουδιών εκτός και αν τερματίσει το πρόγραμμα. Το πρόγραμμα εκτυπώνει στην κονσόλα ποιο αρχείο αρχίζει και ποιο αρχείο μόλις τελείωσε (δείτε την ενότητα 1.4.1).

## 2.2 Μέρος 2

Το δεύτερο μέρος περιέχει την λειτουργικότητα που περιγράφεται στην ενότητα 1.

## 2.3 Προαιρετικό μέρος

Προαιρετικά, μπορείτε να προσθέσετε την εξής λειτουργικότητα. Τα περισσότερα `mp3` περιέχουν μεταδεδομένα\* όπως το όνομα του τραγουδιού, τους τραγουδιστές, το είδος μουσικής κτλ. Αν υπάρχουν αυτά τα δεδομένα στο αρχείο τότε να εμφανίζονται στην λίστα των τραγουδιών ώστε ο χρήστης να έχει περισσότερες πληροφορίες για τα τραγούδια. Για το διάβασμα των μεταδεδομένων μπορείτε να χρησιμοποιήσετε εξωτερική βιβλιοθήκη.

## 3 Βοηθητικές βιβλιοθήκες

Για την αναπαραγωγή των αρχείων `mp3` θα πρέπει να χρησιμοποιήσετε την βιβλιοθήκη `musicplayer`<sup>†</sup> που σας παρέχεται στα πλαίσια της εργασίας. Για να συμπεριλάβετε την βιβλιοθήκη στην εφαρμογή σας θα πρέπει να ενημερώσετε το `pom.xml` του `maven project` σας με τις παρακάτω πληροφορίες.

Να προσθέσετε στα `<dependencies>` το παρακάτω

```
<dependency>
  <groupId>gr.hua.dit.oop2</groupId>
```

\* <https://id3.org/id3v2-00>

† <https://gitlab.com/acharal/maven/-/packages/10619464>

```

<artifactId>musicplayer</artifactId>
<version>2.0.0</version>
</dependency>

```

και σαν ξεχωριστά μπλοκ <repositories> στο pom.xml τις πληροφορίες από που θα κατεβάσει το maven την βιβλιοθήκη.

```

<repositories>
  <repository>
    <id>gitlab-maven</id>
    <url>https://gitlab.com/api/v4/projects/41008035/packages/maven</url>
  </repository>
</repositories>

```

Η τελευταία έκδοση της βιβλιοθήκης είναι η 2.0.0.

### 3.1 Η διεπαφή της βιβλιοθήκης

Η βασική διεπαφή της βιβλιοθήκης είναι η Player η οποία έχει τον παρακάτω ορισμό.

```

interface Player {

    void play(InputStream i) throws PlayerException;

    void startPlaying(InputStream i) throws PlayerException;

    Status getStatus();

    void pause();

    void resume();

    void stop();

    void close();

    void addPlayerListener(PlayerListener listener);

    void removePlayerListener(PlayerListener listener);

    void addProgressListener(ProgressListener listener);

    void removeProgressListener(ProgressListener listener);
}

```

Η Player μπορεί να αναπαράγει με δυο τρόπους ένα τραγούδι.

- Με την play όπου δέχεται ένα InputStream που περιέχει ένα τραγούδι mp3 και το αναπαράγει μέχρι το τέλος.
- Με την startPlaying όπου ξεκινάει η αναπαραγωγή και επιστρέφει αμέσως. Σε εκείνη την περίπτωση μπορούν να καλεστούν οι υπόλοιπες μέθοδοι, pause, resume και stop για να διαχειριστούν την αναπαραγωγή του τραγουδιού.

Και στις 2 περιπτώσεις πριν τερματιστεί η εφαρμογή θα πρέπει να καλεστεί η μέθοδος `close` για να τερματίσει τον `Player` και να αποδεσμεύσει τους πόρους του.

Η κατάσταση του `Player` μπορεί να εξεταστεί με την μέθοδο `getStatus`. Οι πιθανές καταστάσεις του `Player` μπορεί να είναι είτε `Status.IDLE` είτε `Status.PLAYING` είτε `Status.PAUSED`. Για την αλλαγή των καταστάσεων και για την προόδου της αναπαραγωγής ενημερώνονται και οι “ακροατές” `PlayerListener` και `ProgressListener` του αντικειμένου `Player`.

Το παρακάτω πρόγραμμα είναι ένα παράδειγμα χρήσης της βιβλιοθήκης.

```
import java.io.InputStream;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import gr.hua.dit.oop2.musicplayer.Player;
import gr.hua.dit.oop2.musicplayer.PlayerException;
import gr.hua.dit.oop2.musicplayer.PlayerFactory;

public class JukeBoxMain {
    public static void main(String[] args) {
        Player p = PlayerFactory.getPlayer();

        try {
            InputStream song = new FileInputStream("papakia.mp3");
            p.play(song);
        } catch (FileNotFoundException e) {
            System.err.println("File papakia.mp3 not found");
        } catch (PlayerException e) {
            System.err.println("Something's wrong with the player: " + e.getMessage());
        } finally {
            if (p != null)
                p.close();
        }
    }
}
```

### 3.2 Συμβάντα και ακροατές της `Player`

Η βιβλιοθήκη ορίζει δύο συμβάντα που μπορούν να γεννηθούν από μια κλάση `Player`.

- `PlayerEvent` που αναπαριστά μια αλλαγή στην κατάσταση του `Player`.
- `ProgressEvent` που αναπαριστά μια ενημέρωση στην πρόοδο αναπαραγωγής του αρχείου ήχου.

Οι αντίστοιχοι ακροατές ορίζονται από τις διεπαφές `PlayerListener` και `ProgressListener` με τους παρακάτω ορισμούς:

```
public interface PlayerListener {
    void statusUpdated(PlayerEvent e);
}
```

```
}
```

```
public interface ProgressListener {
    void progress(ProgressEvent e);
}
```

Ακολουθεί ένα παράδειγμα χρήσης του συστήματος με τους ακροατές.

```
Player p = PlayerFactory.getPlayer();
p.addPlayerListener(new PlayerListener() {
    public void statusUpdated(PlayerEvent e) {
        logger.info("Status changed to " + e.getStatus());
    }
});
p.addProgressListener(new ProgressListener() {
    public void progress(ProgressEvent e) {
        float seconds = e.getMicroseconds() * 1.0f / 1000000;
        logger.fine("Seconds: " + seconds);
    }
});
try {
    InputStream song = new FileInputStream("rudolph.mp3");
    p.startPlaying(song);
    Thread.sleep(10000);
    p.stop();
    InputStream lastSong = new FileInputStream("radetzky.mp3");
    p.startPlaying(lastSong);
    p.addPlayerListener(new PlayerListener() {
        public void statusUpdated(PlayerEvent e) {
            if (e.getStatus() == Player.Status.IDLE) {
                e.getSource().close();
            }
        }
    });
} catch (Exception e) {
    logger.severe(e.getMessage());
}
```

Το παράδειγμα αναπαράγει δύο τραγούδια. Στο πρώτο αναπαράγει τα πρώτα 10000 χιλιοστά του δευτερολέπτου (milliseconds), μετά το σταματάει και αρχίζει το τελευταίο τραγούδι. Όταν το τραγούδι ολοκληρωθεί, δηλαδή η κατάσταση του Player γίνει IDLE κλείνει τον Player και το πρόγραμμα τερματίζει. Κατά την διάρκεια της αναπαραγωγής των δυο τραγουδιών εκτυπώνονται μηνύματα στην κονσόλα.

## 4 Οδηγίες για την ανάπτυξη της εφαρμογής

- Επιλέξτε ομάδα και αρχίστε όσο το δυνατόν πιο γρήγορα. Η εργασία δεν είναι ατομική και δεν είναι αποδεχτό να έχει δουλέψει μόνο ένα μέλος της ομάδας.
- Δεν έχει νόημα να δουλεύετε όλοι στον ίδιο υπολογιστή.



Χωρίστε την εφαρμογή σε υποπροβλήματα και μοιραστείτε τον φόρτο εργασίας.

- Αποφασίστε τις διεπαφές που θα χρησιμοποιήσετε ώστε να απομονώσετε τα επιμέρους κομμάτια.
- Όταν αναπτύσσετε ένα κομμάτι να σκέφτεστε ότι οι προδιαγραφές μπορούν να αλλάξουν (γιατί μπορεί να αλλάξουν στην εργασία) και πως θα μπορούσατε να ικανοποιήσετε εύκολα καινούργιες απαιτήσεις.
- Αιτιολογήστε τις αποφάσεις που πήρατε κατά την διάρκεια της ανάπτυξης.

## 5 Οδηγίες για την παράδοση

Για το πακετάρισμα, την μεταγλώττιση και την χρήση των αναγκαίων βιβλιοθηκών θα πρέπει να χρησιμοποιηθεί το εργαλείο `maven`. Μπορείτε να χρησιμοποιήσετε οποιαδήποτε εφαρμογή IDE θέλετε αλλά η εφαρμογή σας θα πρέπει να μεταγλωττίζεται με το `maven` χωρίς την βοήθεια του IDE.

Για την συνεργασία μεταξύ των μελών της ομάδας προτείνεται η χρήση `git`<sup>‡</sup>.

### 5.1 Τεκμηρίωση κώδικα και τελική αναφορά

Επιπλέον, θα πρέπει να παρέχετε με την παράδοση ένα έγγραφο με τις οδηγίες εκτέλεσης και λεπτομέρειες που πιστεύετε ότι θα φανούν χρήσιμες.

Ενδέχεται να υπάρχει προφορική εξέταση σε μερικές ομάδες. Όσο καλύτερα τεκμηριωμένος είναι ο κώδικας και οι αποφάσεις που πήρατε τόσο λιγότερες είναι οι πιθανότητες για επιλογή της ομάδας σας.

### 5.2 Περί λογοκλοπής

Η αντιγραφή και η λογοκλοπή απαγορεύονται. Στην ακαδημαϊκή κοινότητα της οποίας είστε μέλη θεωρείτε σοβαρό παράπτωμα. Γι αυτό, αν χρησιμοποιήσετε κώδικα από εξωτερικές πηγές ή βιβλιοθήκες θα πρέπει να τις αναφέρετε ρητά.

---

<sup>‡</sup> <https://git-scm.com/>