

ΑΚΑΔ. ΕΤΟΣ 2021-2022, ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ

# ΔΙΚΤΥΑΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ (Ζυγοί ΑΜ)

## Εργαστήριο

### ΤΕΛΙΚΗ ΕΡΓΑΣΙΑ (ΕΞΕΤΑΣΗ )

για τα τμήματα

ΔΠΖ01, ΔΠΖ02, ΔΠΖ03, ΔΠΖ04 και ΔΠΖ05

(Διδάσκων: Α. Γεωργούλας)

#### Εισαγωγή

Μια τράπεζα διατηρεί μια σχεσιακή βάση δεδομένων, στην οποία αποθηκεύονται στοιχεία **τραπεζικών λογαριασμών** όπως:

- Κωδικός λογαριασμού (Account ID),
- Στοιχεία του πελάτη (όνομα, στοιχεία επικοινωνίας, κλπ.),
- Τρέχον υπόλοιπο λογαριασμού,
- Καθώς και οποιαδήποτε άλλα στοιχεία κρίνετε εσείς απαραίτητα για την υλοποίηση των λειτουργιών που περιγράφονται παρακάτω.

Για λόγους απλότητας, υποθέτουμε ότι όλες οι παραπάνω πληροφορίες περιλαμβάνονται σε **ΕΝΑΝ** μοναδικό πίνακα της ΒΔ.

#### Αντικείμενο εργασίας

Αντικείμενο της εργασίας είναι η ανάπτυξη μιας απλοποιημένης εφαρμογής απομακρυσμένης διαχείρισης στους τραπεζικούς λογαριασμούς που περιεγράφηκαν στην εισαγωγή, με σκοπό την εκτέλεση **οχτώ συγκεκριμένων λειτουργιών**. Οι λειτουργίες που θα πρέπει να προσφέρει η εφαρμογή είναι οι εξής:

1. **Δημιουργία λογαριασμού:** η λειτουργία αυτή θα επιτρέπει στον χρήστη να δημιουργεί έναν λογαριασμό παρέχοντας όλα τα στοιχεία που απαιτούνται. Εφόσον η συναλλαγή ολοκληρωθεί επιτυχώς θα επιστρέφεται η ένδειξη "*Account Created Successfully*".

2. **Κατάθεση** ποσού: η λειτουργία αυτή θα επιτρέπει στον χρήστη να κάνει κατάθεση ενός χρηματικού ποσού σε έναν λογαριασμό (βάσει Account ID). Εφόσον η κατάθεση ολοκληρωθεί επιτυχώς θα επιστρέφεται η ένδειξη "*Successful Deposit*". Εάν η κατάθεση αποτύχει, θα επιστρέφεται η ένδειξη "*Deposit Error*".
3. **Ανάληψη** ποσού: η λειτουργία αυτή θα επιτρέπει στον χρήστη να κάνει ανάληψη ενός χρηματικού ποσού από έναν λογαριασμό (βάσει Account ID). Εφόσον η ανάληψη ολοκληρωθεί επιτυχώς θα επιστρέφεται η ένδειξη "*Successful Withdrawal*". Εάν το υπόλοιπο δεν επαρκεί για τη συναλλαγή θα πρέπει να επιστρέφεται η ένδειξη "*Insufficient Balance*".
4. **Μεταφορά** σε άλλο λογαριασμό: η λειτουργία αυτή θα επιτρέπει στον χρήστη να μεταφέρει ένα χρηματικό ποσό από έναν λογαριασμό σε κάποιον άλλο (βάσει Account ID). Εφόσον η μεταφορά ολοκληρωθεί επιτυχώς θα πρέπει να επιστρέφεται η ένδειξη "*Successful transfer*". Εάν το υπόλοιπο δεν επαρκεί για τη συναλλαγή θα πρέπει να επιστρέφεται η ένδειξη "*Insufficient Balance*".
5. **Απενεργοποίηση/Ενεργοποίηση** λογαριασμού: η λειτουργία αυτή θα επιτρέπει στον χρήστη να απενεργοποιεί έναν λογαριασμό. Ένας απενεργοποιημένος λογαριασμός συνεχίζει να υπάρχει (και να διατηρεί το υπόλοιπο του) αλλά **δεν θα μπορεί να κάνει συναλλαγές**. Αντίστοιχα, ο χρήστης θα μπορεί να ενεργοποιεί εκ νέου έναν απενεργοποιημένο λογαριασμό. Εφόσον η συναλλαγή ολοκληρωθεί επιτυχώς θα επιστρέφεται η ένδειξη "*Account Activated*" ή "*Account Deactivated*".
6. **Προβολή στοιχείων**: η λειτουργία αυτή θα επιτρέπει στον χρήστη να βλέπει όλα τα στοιχεία ενός λογαριασμού (βάσει Account ID).
7. **Προβολή όλων**: η λειτουργία αυτή θα επιτρέπει στον χρήστη να βλέπει όλα τα στοιχεία ΟΛΩΝ των λογαριασμών.
8. **Διαγραφή** λογαριασμού: η λειτουργία αυτή θα επιτρέπει στον χρήστη να διαγράφει οριστικά έναν λογαριασμό.

**Σημ. 1:** Για λόγους **απλότητας**, κάνουμε την παραδοχή ότι ο χρήστης μπορεί να εκτελέσει τις παραπάνω ενέργειες σε οποιονδήποτε λογαριασμό απλά γνωρίζοντας το **Account ID** (χωρίς δηλαδή να απαιτείται κάποιου είδους αυθεντικοποίηση).

**Σημ. 2:** Εάν ένας λογαριασμός είναι **ανύπαρκτος** (π.χ. δώσουμε λάθος **Account ID**) ή **απενεργοποιημένος** τότε η οποιαδήποτε συναλλαγή θα πρέπει να απορρίπτεται και να

επιστρέφεται η ένδειξη "Transaction Error – Account not found" ή "Transaction Error – Account is not active".

## Επιλογές υλοποίησης

Για την υλοποίηση της εφαρμογής **έχετε τρεις** (βαθμολογικά ισοδύναμες) **επιλογές**:

### A. Να υλοποιήσετε την εφαρμογή ως **Web Application** με τη χρήση **Servlets**.

Στην περίπτωση αυτή θα πρέπει υλοποιήσετε απλές HTML ιστοσελίδες και τα κατάλληλα servlets που να υλοποιούν τις αντίστοιχες λειτουργίες. Θα πρέπει να αποφασίσετε μόνοι σας πόσες ιστοσελίδες θα χρειαστείτε (στατικές και δυναμικές), πως θα παρουσιάζουν τα δεδομένα/αποτελέσματα καθώς και τη ροή πλοήγησης μεταξύ των αυτών. Προτείνεται η έναρξη να γίνεται από μια **αρχική σελίδα**, η οποία θα προβάλει (με τη μορφή links) τις διάφορες ενέργειες που μπορεί να κάνει ο χρήστης.

### B. Να υλοποιήσετε την εφαρμογή ως **Java Client-Server Application** με τη χρήση **Sockets**.

Στην περίπτωση αυτή θα πρέπει να υλοποιήσετε τον **server** και τον **client**. Η εφαρμογή client μπορεί να αναπτυχθεί είτε σαν απλή **console application** ή να προσθέσετε και **graphical user interface** εφόσον το επιθυμείτε. Ο **Server** θα πρέπει να εκτελείται συνεχώς (continuous server). Για λόγους απλότητας, κάνουμε την παραδοχή ότι ο server μπορεί να εξυπηρετεί **έναν client** κάθε χρονική στιγμή, οπότε δεν χρειάζεται να υλοποιήσετε multithreaded server. Θα πρέπει να σχεδιάσετε **το δικό σας πρωτόκολλο επικοινωνίας** μεταξύ του **client** και του **server**. Για λόγους απλότητας, θα είναι αποδεκτή η λύση όπου οι πληροφορίες θα ανταλλάσσονται **όλες μαζί** (π.χ. ο client θα στέλνει ένα String που μπορεί να περιλαμβάνει π.χ. την εντολή, τον κωδικό λογαριασμού και τα στοιχεία που θέλουμε να στείλουμε). Αντίστοιχα, ο server θα πρέπει να "τεμαχίζει" κατάλληλα το string αυτό και να εξάγει τις πληροφορίες που χρειάζεται. Επίσης, για λόγους διευκόλυνσης υλοποίησης της λειτουργίας #7, θα είναι αποδεκτό τα στοιχεία όλων των λογαριασμών να επιστρέφονται σαν ένα ενιαίο string.

### C. Να υλοποιήσετε τις υπηρεσίες σαν **RESTful webservice**.

Στην περίπτωση αυτή θα πρέπει να σχεδιάσετε μόνοι σας τα **endpoints** (URIs) και να αποφασίσετε τις μεθόδους (GET, POST, κλπ.) που είναι κατάλληλες για κάθε μια από τις

υπηρεσίες. Επίσης, θα πρέπει να αποφασίσετε με ποιο τρόπο θα περνάτε τις παραμέτρους που απαιτούνται σε κάθε περίπτωση. Οι υπηρεσίες που επιστρέφουν στοιχεία θα πρέπει να παρέχονται με δύο επιλογές, σε μορφή XML και σε μορφή JSON. **Δεν απαιτείται** να αναπτύξετε κάποια ξεχωριστή εφαρμογή να καταναλώνει αυτές τις υπηρεσίες. Θα πρέπει απλά να προσθέσετε μια **index page** όπου να περιγράφονται τα **endpoints** που μπορεί κάποιος να χρησιμοποιήσει.

Μπορείτε να επιλέξετε μόνο **ENAN** από τους τρεις παραπάνω τρόπους υλοποίησης. Σε κάθε περίπτωση, θα πρέπει πρώτα **να έχετε κατασκευάσει** την απαραίτητη **ΒΔ** ώστε να μπορέσετε να **επιδείξετε** τη λειτουργία των υπηρεσιών που έχετε υλοποιήσει.

### Εκτέλεση της εφαρμογής & επίδειξη ορθής εκτέλεσης

Με την ολοκλήρωση της ανάπτυξης **θα πρέπει να τρέξετε** την εφαρμογή **στον υπολογιστή σας** και να δοκιμάσετε όλες τις λειτουργίες που έχετε αναπτύξει. Η επίδειξη της ορθής εκτέλεσης της εφαρμογής σημαίνει ότι θα πρέπει να εκτελέσετε όλες τις λειτουργίες που υποστηρίζει η εφαρμογή σας και να λάβετε κατάλληλα (\*ευκρινή\*) **screenshots** που να **αποδεικνύουν** την ορθή εκτέλεση των λειτουργιών (π.χ. πληροφορίες που λαμβάνονται πίσω, αλλαγές που συμβαίνουν στη ΒΔ, κλπ.).

Τα αποδεικτικά αυτά θα πρέπει να συμπεριληφθούν στο συνοδευτικό **report** (αρχείο PDF) μαζί με τα δικά σας **σχόλια** σχετικά με την υλοποίησή σας. Το αρχείο αυτό (report) θα το **συνυποβάλλετε** μαζί με τον πηγαίο κώδικα της εφαρμογής.

Η **δοκιμή** (εκτέλεση) της εφαρμογής μπορεί να γίνει με οποιονδήποτε από τους τρόπους που έχουν επιδειχθεί μέσα στο εργαστήριο, ανάλογα με τη λύση που έχετε επιλέξει δηλαδή:

- Εάν επιλέξετε τη λύση με **Sockets**, θα πρέπει να εκτελέσετε την εφαρμογή (και client και server) και να δείξετε με screenshots τα αποτελέσματα στο **standard output** (ή στο GUI εφόσον έχετε αναπτύξει).
- Εάν επιλέξετε τη λύση με **Servlets**, θα πρέπει να εκτελέσετε την εφαρμογή μέσω **Web Browser** και να επιδείξετε τις διάφορες λειτουργίες με κατάλληλα screenshots.
- Εάν επιλέξετε τη λύση με **RESTful Web Services**, θα πρέπει να εκτελέσετε την εφαρμογή μέσω **Web Browser** ή **Postman** και να επιδείξετε τις διάφορες λειτουργίες

με κατάλληλα screenshots. **Δεν απαιτείται** να αναπτύξετε κάποιο UI που να καταναλώνει αυτές τις υπηρεσίες παρά μόνο να δείξετε τα αποτελέσματα των κλήσεων κάθε endpoint.

Σε κάθε περίπτωση, θα πρέπει να λαμβάνετε screenshots **και από τη ΒΔ** όπου θα φαίνεται το αποτέλεσμα της κάθε ενέργειας.

## Γενικές Υποδείξεις

Σχετικά με τη **Βάση Δεδομένων**:

- Μπορείτε να χρησιμοποιήσετε τον **Database Server** της επιλογής σας (π.χ. mysql,xampp).
- Για την πρόσβαση στη ΒΔ **θα πρέπει** να δημιουργήσετε έναν **χρήστη** (εκτός του root), ο οποίος **θα έχει σαν username το επώνυμό σας** και θα έχει δικαιώματα διαχείρισης στο domain.
- Θα πρέπει να αποφασίσετε **μόνοι σας τα πεδία και τη δομή** του πίνακα όπου θα αποθηκεύονται οι πληροφορίες των λογαριασμών.
- Στο συνοδευτικό **report** που θα παραδώσετε, θα πρέπει να περιγράψετε συνοπτικά (λεκτικά και με κατάλληλα screenshots) τις ενέργειες που κάνατε για την κατασκευή της ΒΔ ή ακόμα καλύτερα να συμπεριλάβετε στην υποβολή σας το **sql script** που χρησιμοποιήσατε για την κατασκευή της ΒΔ.

Σχετικά με την **ανάπτυξη** και **εκτέλεση** της εφαρμογής:

Για την ανάπτυξη του κώδικα μπορείτε να χρησιμοποιήσετε το **IDE/Editor** της επιλογής σας. Εφόσον επιλέξετε τη λύση με **Servlets** ή **WebServices** μπορείτε να κάνετε **deploy** τις υπηρεσίες σε οποιονδήποτε από τους δυο servers (GlassFish, Tomcat). Στο συνοδευτικό report που θα παραδώσετε, θα πρέπει να περιγράψετε συνοπτικά (λεκτικά και με κατάλληλα screenshots) τις ενέργειες που κάνατε για το deployment της υπηρεσίας.

## Υποβολή

Η εργασία θα πρέπει να υποβληθεί **ΑΠΟΚΛΕΙΣΤΙΚΑ** στο **e-class** (σε κατάλληλο χώρο που θα δημιουργηθεί ανά τμήμα) όπου θα φαίνεται και η **προθεσμία παράδοσης**. Πέραν της συγκεκριμένης προθεσμίας καμία εργασία **δε θα γίνει αποδεκτή**. Επίσης, **δε θα γίνει αποδεκτή** οποιαδήποτε εργασία υποβληθεί **με άλλο τρόπο** (email, κλπ.).

Επειδή το e-class επιτρέπει το ανέβασμα μόνο ενός αρχείου για κάθε χρήστη, όταν ολοκληρώσετε την άσκηση, θα πρέπει να συμπίεσετε τα αρχεία σε **ΕΝΑ τελικό zip** το οποίο και θα υποβάλετε. Το αρχείο zip θα περιέχει **ΑΠΑΡΑΙΤΗΤΑ** τα παρακάτω αρχεία:

- Τον κώδικα της εφαρμογής (δηλαδή ολόκληρο το project του **NetBeans** ή **IntelliJ**)
- Το συνοδευτικό αρχείο **PDF** όπου θα συμπεριλάβετε τα σχόλια σας και τα αποδεικτικά screenshots.

### Βαθμολογικά κριτήρια

Θα βαθμολογηθούν **θετικά** τα εξής:

- Η ορθή επίλυση του συνολικού προβλήματος και η ανάπτυξη κάθε ξεχωριστής λειτουργικότητας, σύμφωνα με την εκφώνηση.
- Η επιτυχημένη επίδειξη των λειτουργιών με κατάλληλα αποδεικτικά (screenshots στο pdf).
- Η καλή σχεδίαση των υπηρεσιών σύμφωνα με τις τεχνολογίες και τεχνικές που διδάχθηκαν στο μάθημα.
- Η πρωτοτυπία της λύσης (σε επίπεδο σχεδίασης και υλοποίησης).
- Η ποιότητα του κώδικα και η χρήση καλών προγραμματιστικών τεχνικών (αντικειμενοστραφής λύση, σωστή ονοματολογία κλάσεων, μεθόδων και μεταβλητών, σύνταξη, σωστή χρήση μεθόδων, πέρασμα παραμέτρων, αρχές κατασκευής κλάσεων, κλπ.).
- Η ύπαρξη επαρκών σχολίων μέσα στον κώδικα που να εξηγεί κρίσιμες λειτουργίες.

Θα βαθμολογηθούν **αρνητικά** τα εξής:

- Η μη υποβολή του συνοδευτικού αρχείου PDF με τα αποδεικτικά εκτέλεσης ή η υποβολή ελλιπών αποδεικτικών.
- Η λανθασμένη ή ελλιπής υλοποίηση των ζητούμενων λειτουργιών
- Η ύπαρξη σφαλμάτων μέσα στον κώδικα.
- Ο δυσανάγνωστος και δυσνόητος κώδικας (π.χ. έλλειψη indentation, κακή ονοματολογία, κακός διαχωρισμός σε μεθόδους, κλπ.).
- Η ύπαρξη άσχετων γραμμών κώδικα μέσα στον κώδικα.

- Η ύπαρξη **περίτεχνου/πολύπλοκου** κώδικα που δεν αιτιολογείται και δεν επεξηγείται επαρκώς με σχόλια.

**Σημ.:** Κατά την εκπόνηση των εργασιών θα πρέπει να **τηρήσετε αυστηρά** τους κανόνες **ακαδημαϊκής δεοντολογίας** και να αποφύγετε τη χρήση αθέμιτων πρακτικών. Όλες οι εργασίες θα υποβληθούν σε έλεγχο με κατάλληλο λογισμικό ανίχνευσης. Εάν υπάρξουν ενδείξεις αθέμιτων πρακτικών (π.χ. αντιγραφής) η εργασία θα μηδενίζεται με συνοπτικές διαδικασίες.