

### Προγραμματιστικές Ασκήσεις, Εργαστήριο 4

Στην άσκηση αυτή θα επεκτείνετε τα προγράμματα που κάνατε για το Εργαστήριο 3. Στο εcourse υπάρχει η λύση του Τρίτου Εργαστηρίου, για κάθε μέρος χωριστά. Μπορείτε να κατεβάσετε τον κώδικα και να τον τροποποιήσετε κατάλληλα, είτε να τροποποιήσετε τον δικό σας κώδικα από το Τρίτο Εργαστήριο. Συγκεκριμένα, ο κώδικας που χρειάζεστε από το 3 είναι η κλάση **BankAccount** (την οποία θα επεκτείνετε στο 2 μέρος), ενώ η **main** στην **TestBankAccount** θα εκτελεί πλέον μόνο όσα σας ζητούνται παρακάτω (μέρος 3).

Η άσκηση έχει τα εξής μέρη

1. Δημιουργήστε την κλάση **BankCheck** η οποία κρατάει πληροφορία για μία επιταγή. Η κλάση θα έχει δύο πεδία, ένα String που κρατάει το όνομα του αποδέκτη της επιταγής (payee) και έναν πραγματικό αριθμό για το ποσό της επιταγής. Και τα δύο πεδία αρχικοποιούνται στον constructor.

Η κλάση έχει επίσης τρεις μεθόδους:

- a) Μεθόδους πρόσβασης (**accessor** methods) και για τα δύο πεδία
- b) Μέθοδο μετάλλαξης (**mutator** method) για το πεδίο του ονόματος.
- c) Την μέθοδο **toString** η οποία επιστρέφει την String αναπαράσταση της επιταγής. Για μία επιταγή με αποδέκτη τον Panayiotis για το ποσό των 200 ευρώ θα πρέπει να είναι το String: "[Check to Panayiotis: 200€]"
- d) Την μέθοδο **equals** η οποία ελέγχει για ισότητα. Δύο επιταγές είναι ίδιες αν έχουν τον ίδιο αποδέκτη και το ίδιο ποσό.

2. Τροποποιήστε την κλάση BankAccount προσθέτοντας τις εξής μεθόδους:

- a) Την μέθοδο **toString** η οποία επιστρέφει την String αναπαράσταση του λογαριασμού. Για ένα λογαριασμό με όνομα κάτοχου Panayiotis, θα πρέπει να είναι το String: "Panayiotis account"
- b) **Υπερφορτώστε** την μέθοδο **deposit** ώστε να παίρνει σαν όρισμα μια επιταγή (δηλαδή, ένα αντικείμενο BankCheck), και εφόσον το όνομα του αποδέκτη της επιταγής είναι ίδιο με αυτό του κατόχου του λογαριασμού καταθέτει το ποσό στον λογαριασμό. Η μέθοδος επιστρέφει μια Boolean τιμή αν η κατάθεση ήταν επιτυχής ή όχι.

3. Η κλάση **TestBankAccount** έχει την **main** μέθοδο. Στην **main** θα υλοποιήσετε τα εξής βήματα:

- a) Δημιουργήστε ένα τραπεζικό λογαριασμό (BankAccount) με όνομα το όνομα σας.
- b) Καταθέσετε 1000 ευρώ στον λογαριασμό.
- c) Ζητήστε να σας δώσει ο χρήστης από την είσοδο ένα όνομα και δημιουργήστε μια επιταγή (BankCheck) για το ποσό των 200 ευρώ σε αυτό το όνομα.
- d) Στην συνέχεια προσπαθήστε να καταθέσετε την επιταγή στον λογαριασμό σας. Όσο η κατάθεση δεν είναι επιτυχής, ζητήστε από τον χρήστη να σας δώσει νέο όνομα, αλλάξτε το όνομα στην επιταγή και προσπαθήστε ξανά.
- e) Τυπώστε ότι έγινε επιτυχής κατάθεση, τυπώνοντας τις πληροφορίες της επιταγής και του λογαριασμού, χρησιμοποιώντας τις αντίστοιχες toString μεθόδους.
- f) Δημιουργήστε μια δεύτερη επιταγή στο όνομα σας για το ποσό των 200 ευρώ.
- g) Ελέγξτε αν η δεύτερη επιταγή είναι ίδια με την πρώτη και αν ναι, τυπώστε ένα μήνυμα ότι η επιταγή έχει ήδη κατατεθεί. Αλλιώς καταθέστε την επιταγή.
- h) Τυπώστε την κατάσταση του λογαριασμού.

Τα χρώματα παραπάνω ομαδοποιούν τις εντολές ανάλογα με την υλοποίηση που χρειάζεται να έχετε κάνει:

- Με **πράσινο**, οι εντολές που τρέχουν με την υπάρχουσα υλοποίηση που σας δίνεται.
- Με **κόκκινο**, οι εντολές που χρειάζονται υλοποίηση στην BankCheck.
- Με **μπλε**, οι εντολές που χρειάζονται υλοποίηση μεθόδων στην BankAccount.

Συνίσταται να χτίσετε την main βήμα βήμα. Κάθε φορά που υλοποιείτε μια μέθοδο γράψτε και τις αντίστοιχες εντολές στη main και κάνετε τεστ.

Η έξοδος σας θα πρέπει να είναι παρόμοια (διαφορετικό όνομα) με την παρακάτω.

```
>java TestBankAccount
Give the payee's name for the check: Nikos
Cannot deposit check. Please give the correct payee's name: Giannis
Cannot deposit check. Please give the correct payee's name: Panayiotis
Deposited check [Check to Panayiotis: 200.0E] to account: Panayiotis account

Check [Check to Panayiotis: 200.0E] has already been deposited

Panayiotis account:
The balance in the account is 1200.0 euros
```

### ΥΠΟΔΕΙΞΕΙΣ:

- Οι μέθοδοι **toString** και **equals** ορίζονται με συγκεκριμένο τρόπο που έχουμε περιγράψει στις διαφάνειες του μαθήματος.
- Η υπερφόρτωση είναι όταν ορίζουμε ξανά μια μέθοδο με το ίδιο όνομα, αλλά με διαφορετικά ορίσματα.
- Όταν λέμε «*Το Χ αρχικοποιείται στον constructor*» (π.χ., εδώ το Χ είναι το όνομα και το ποσό για την επιταγή), αυτό σημαίνει ότι ο constructor παίρνει σαν όρισμα τιμή για το Χ.
- Κάθε φορά που υλοποιείτε τις απαραίτητες μεθόδους για ένα βήμα, βάλετε τις αντίστοιχες εντολές στη main, κάντε compile και τρέξτε το πρόγραμμα σας.

### ΓΕΝΙΚΕΣ ΟΔΗΓΙΕΣ

Ο κώδικας πρέπει να υπάρχει στον λογαριασμό σας CSE, δηλαδή στα σύστημα αρχείων των υπολογιστών του εργαστηρίου του Τμήματος. Προτεινόμενη δομή αρχείων/καταλόγων: ~home/JavaLabs/lab<αριθμός εργαστηρίου>. Φροντίστε λοιπόν να μεταφέρετε έγκαιρα τα αρχεία που δουλέψατε στο σπίτι σύμφωνα με τις Τεχνικές Οδηγίες που αναγράφονται στο ecourse του μαθήματος.