

## *Αναφορά Εργασίας*

### *Αντικειμενοστραφή-Προγραμματισμού II*

Η συγκεκριμένη εργασία, όντας γραμμένη σε γλώσσα Java, έχει ως στόχο τη δημιουργία ενός καλενταριού (calendar). Για τους σκοπούς της εργασίας, η υλοποίηση έχει γίνει σε 2 κομμάτια. Το 1<sup>ο</sup> κομμάτι (part 1), επικεντρώνεται στην αλληλεπίδραση με το χρήστη μέσω του terminal του συστήματός του και προϋποθέτει ο χρήστης να γνωρίζει λεπτομερώς τα βήματα που οφείλει να ακολουθήσει για την ομαλή λειτουργία του προγράμματος. Από την άλλη, το 2<sup>ο</sup> κομμάτι (part 2), δεν απαιτεί τίποτα από το χρήστη. Η εκτέλεσή του είναι απλή και ο χρήστης μπορεί πολύ εύκολα να καταλάβει τι ζητείται από εκείνον. Δίνεται μεγάλη έμφαση στην ρεαλιστική και άμεση συνεργασία με το χρήστη, μέσω ενός απλού γραφικού περιβάλλοντος διεπαφής (GUI). Πέρα από αυτά, η ουσιαστικότερη διαφορά μεταξύ των δύο κομματιών, είναι ότι το 1<sup>ο</sup> κομμάτι, παρόλης της απλής του υλοποίησης και εφαρμογής, δεν γίνεται εύκολα προσιτό σε ένα καθημερινό άνθρωπο. Αντιθέτως, το 2<sup>ο</sup> κομμάτι είναι εύκολα προσεγγίσιμο και ευνόητο από το καθένα μας, εφόσον μοιάζει με τις εφαρμογές που χρησιμοποιούμε σε καθημερινή βάση, απ' τη στιγμή που έχουμε πρόσβαση σε μια ηλεκτρονική συσκευή.

#### *1<sup>ο</sup> μέρος:*

Το 1<sup>ο</sup> κομμάτι αποτελείται από τέσσερις κλάσεις:

- ❖ Η πρώτη κλάση ονομαζόμενη *Main*, έχει ως κύριο ρόλο τον έλεγχο της ομαλής λειτουργίας του κώδικα καθώς και την επιβεβαίωση πως ο χρήστης έδωσε τα σωστά ορίσματα. Καλεί τις κατάλληλες μεθόδους από την επόμενη κλάση, *myCalendar*.
- ❖ Η κλάση *myCalendar* αναλαμβάνει όλες τις συναρτήσεις που χειρίζονται τις λειτουργίες του καλενταριού. Χειρίζεται τόσο τη δημιουργία, φόρτωση, χρήση, αποθήκευση και εμφάνιση συγκεκριμένων λεπτομερειών από το καλεντάρι, όσο και τον λεπτομερή έλεγχο εγκυρότητας αυτών.

Αναλυτικά, περιέχει τις μεθόδους:

- *loadCalendar*, η οποία φορτώνει το καλεντάρι και εμφανίζει κατάλληλο μήνυμα, ανάλογα με το αν η φόρτωση επέτυχε ή όχι.
- *newCalendar*, η οποία δημιουργεί ένα νέο καλεντάρι, στο οποίο προστίθενται κάποια βασικά χαρακτηριστικά.

- *calendarHandler*, η οποία καλείται για τη προσθήκη νέων ραντεβού και εργασιών στο καλεντάρι, καθώς και παίζει το ρόλο καθοδήγησης του χρήστη στις κινήσεις που πρέπει να λάβει για την επιτυχή δημιουργία και προσθήκη αυτών. Για να το επιτυγχάνει αυτό, καλεί όπου πρέπει, τις μεθόδους *newAppointmentToCalendar* και *newWorkToCalendar*, των κλάσεων *Appointment* και *Work* αντίστοιχα.
- *saveCalendar*, η οποία αποθηκεύει στο καλεντάρι τα καινούργια ραντεβού και εργασίες, ελέγχοντας πάντα, ότι η όλη διαδικασία ολοκληρώθηκε χωρίς κανένα πρόβλημα.
- *calendarLists*, ο ρόλος της οποίας είναι να εμφανίζει τα ραντεβού ή τις εργασίες από το καλεντάρι ανάλογα με το όρισμα που έδωσε ο χρήστης κατά την αρχή εκτέλεσης του προγράμματος (*day/week/month/pastday/pastweek/pastmonth/todo/due*). Για παράδειγμα, στη περίπτωση του *day*, θα εμφανιστούν τα ραντεβού και εργασίες μέχρι το τέλος της ημέρας, ενώ στο *pastday*, από την αρχή της ημέρας μέχρι την τρέχον ώρα του χρήστη κ.ο.κ. Στο *todo* θα εκτυπωθούν οι εργασίες που δεν έχουν ολοκληρωθεί και των οποίων η προθεσμία δεν έχει περάσει, ενώ στο *due* οι εργασίες που δεν έχουν ολοκληρωθεί και των οποίων η προθεσμία έχει περάσει.

Επιπλέον, υπάρχουν οι συναρτήσεις *getDateTime* και *dateTimeFormatChecker*, οι οποίες ασχολούνται κυρίως με την ώρα των ραντεβού και εργασιών, καθώς και το σωστό χειρισμό της.

- ❖ Η κλάση *Appointment* περιλαμβάνει τη μέθοδο *newAppointmentToCalendar*, η οποία καλείται για τη δημιουργία ραντεβού και εμφανίζει τα κατάλληλα μηνύματα για την ορθή καθοδήγηση του χρήστη, όπως και ελέγχει την ομαλή λειτουργία του προγράμματος.
- ❖ Η κλάση *Work*, παρόμοια με την *Appointment*, περιλαμβάνει μια αντίστοιχη μέθοδο *newWorkToCalendar*, που καλείται για τη δημιουργία εργασίας και εμφανίζει μήνυμα όπου χρειάζεται για να βοηθήσει το χρήστη, καθώς και επιβεβαιώνει τη κανονική λειτουργία του προγράμματος.

## 2<sup>ο</sup> μέρος:

Στο 2<sup>ο</sup> μέρος του προγράμματος έχουν γίνει σημαντικές αλλαγές, σε σχέση με το 1<sup>ο</sup> μέρος. Αρχικά, στο 1<sup>ο</sup> μέρος χρησιμοποιούταν η προτεινόμενη -από τη βιβλιοθήκη ical4j- μέθοδος για την αποθήκευση ημερολογίων. Εκτός από τη μέθοδο της αποθήκευσης στο 1<sup>ο</sup> μέρος, γινόταν και ένα φιλτράρισμα των χαρακτηριστικών (*properties*) που το πρόγραμμα δεν αναγνώριζε (πχ. *LOCATION*, *URL*, *PRIORITY*). Στο 2<sup>ο</sup> μέρος, ωστόσο, προχωρήσαμε σε μια διαφορετική προσέγγιση, μετατρέποντας το αντικείμενο *Calendar* σε μια συμβολοσειρά (*string*) και αποθηκεύοντας αυτήν τη συμβολοσειρά *byte-by-byte*. Αυτή η αλλαγή έγινε για να μειωθεί και να απλοποιηθεί ο κώδικας.

Ένα σημαντικό μέρος του 2<sup>ου</sup> κομματιού, υλοποιήθηκε βάσει το 1<sup>ο</sup> κομμάτι, με κάποιες ουσιαστικές αλλαγές. Πιο συγκεκριμένα, οι κλάσεις *Appointment* και *Work* διαγράφηκαν πλήρως, καθώς ο κώδικάς τους έπρεπε να αλλοιωθεί αρκετά, εφόσον όλα τα μηνύματα που προηγούμενος εμφανίζονταν στο χρήστη μέσω του *terminal* του, πια εκτυπώνονται με ευανάγνωστο και όμορφο τρόπο στην οθόνη, καθώς έγινε λεπτομερής χρήση παραθυρικών εφαρμογών. Αναλυτικότερα, επειδή ο πηγαίος κώδικας των *Work* και *Appointment* θα άλλαζε σημαντικά αν παρέμεναν σε ξεχωριστές κλάσεις, αποφασίστηκε να γίνει η ενσωμάτωσή τους στη κλάση *NewEventFrame*. Ένα μεγάλο κομμάτι του κώδικα που είχε να κάνει με την ενημέρωση του χρήστη μέσω μηνυμάτων, μετατράπηκε σε εμφάνιση παραθυρικών εφαρμογών, προσφέροντας έτσι στο χρήστη, ένα γραφικό και ευνόητο περιβάλλον. Για το σκοπό αυτό, τροποποιήθηκαν οι κλάσεις *Main* και *myCalendar*, καθώς και δημιουργήθηκαν οι κλάσεις *CalendarFrames*, *NewEventFrame*, *SortedEventsFrame*, *EditEventFrame* και *EditStatusFrame*.

Τροποποιήσεις:

### ❖ Κλάση *Main*

Προηγούμενος, η δουλειά της *Main* ήταν να ελέγχει ότι το πρόγραμμα λειτουργεί σωστά και να καλεί άλλες κλάσεις. Μετά από τις τροποποιήσεις που έγιναν όμως, ο ρόλος της *Main* έγινε να καλεί απλούστατα τη κλάση *CalendarFrames*, η υλοποίηση της οποίας αναλύεται παρακάτω.

## ❖ Κλάση *myCalendar*

Η μεγαλύτερης σημασίας τροποποίηση που έγινε σε αυτή τη κλάση, είναι η δημιουργία της μεθόδου *notificationCheck*, η οποία ενημερώνει το χρήστη για οποιοδήποτε γεγονός πρόκειται να λάβει χώρα σε λιγότερο από 10 λεπτά. Επιπρόσθετα, έγινε αφαίρεση των *saveCalendar*, *calendarHandler*, *newCalendar* και *dateTimeFormatChecker*, ο πηγαίος κώδικας των οποίων χρησιμοποιήθηκε περιστασιακά στις υπόλοιπες κλάσεις, όπως για παράδειγμα η *saveCalendar* τροποποιήθηκε και προστέθηκε στο τέλος της *CalendarFrames*.

Καινούργιες κλάσεις:

## ❖ *CalendarFrames*

Η κλάση *CalendarFrames* περιέχει τη μέθοδο *actionPerformed*, η οποία (μέσω του *constructor*) εμφανίζει στο χρήστη το αρχικό παράθυρο, καλωσορίζοντάς τον και καθοδηγώντας τον στις διαθέσιμες επιλογές που μπορεί να λάβει μέσω ενός μενού. Το παράθυρο αυτό έχει διάφορες χρήσεις, καθώς παραμένει πάντα ανοιχτό και δίνει στο χρήστη τις εξής επιλογές:

### ➤ *Create*

Στη περίπτωση που ο χρήστης θέλει να δημιουργήσει νέο καλεντάρι, διαλέγει αυτή την επιλογή. Εφόσον ο χρήστης έχει επιλέξει να δημιουργήσει νέο καλεντάρι, βρίσκονται μπροστά του τέσσερα κουμπιά:

- ✚ *NewEvent*, όπου εμφανίζει καινούργιο παράθυρο, δίνοντας στο χρήστη τη δυνατότητα δημιουργίας ραντεβού ή εργασίας.
- ✚ *EditEvent*, όπου εμφανίζονται στο χρήστη τα γεγονότα που υπάρχουν στο καλεντάρι, επιτρέποντάς του να τροποποιήσει τα χαρακτηριστικά τους.
- ✚ *EditStatus*, το οποίο εμφανίζει στο χρήστη τις εργασίες που εμπεριέχει το καλεντάρι, επιτρέποντάς του να αλλάξει τη κατάστασή τους από ολοκληρωμένα (*complete*), σε μη ολοκληρωμένα (*incomplete*).
- ✚ *Events*, όπου εμφανίζει όλα τα γεγονότα που υπάρχουν στο καλεντάρι, σε μια συγκεκριμένη χρονική περίοδο (πχ. *day*)

- *Load*  
Όταν ο χρήστης επιθυμεί να φορτώσει ένα ήδη υπάρχων καλεντάρι, τότε θα πατήσει το κουμπλί “*Load*”. Αυτό θα εμφανίσει καινούργιο παράθυρο για τη φόρτωση καλενταριού και αφού φορτωθεί, ο χρήστης μπορεί να επιλέξει να κάνει τις ίδιες ακριβώς πράξεις, όπως στη περίπτωση “*Create*”.
- *Save*  
Κατά το πάτημα αυτού του κουμπιού, εμφανίζεται στο χρήστη καινούργιο παράθυρο, όπου μπορεί να αποθηκεύσει, όπου αυτός επιθυμεί, το καλεντάρι ή τις τροποποιήσεις που έγιναν σε αυτό.
- *Exit*  
Ο χρήστης θα επιλέξει το συγκεκριμένο κουμπί, όταν είναι ικανοποιημένος με τις τροποποιήσεις ή αλλαγές που έχει κάνει και επιθυμεί να αποχωρήσει από το πρόγραμμα.

#### ❖ *NewEventFrame*

Η *NewEventFrame*, χρησιμοποιείται όταν ο χρήστης έχει επιλέξει το “*NewEvent*”, έχει δηλαδή ως στόχο, να δημιουργήσει καινούργιο γεγονός (ραντεβού/εργασία). Πιο συγκεκριμένα, εμφανίζονται 2 *JRadio Buttons* (*Appointment* και *Work*). Αφού ο χρήστης επιλέξει μία από τις δύο επιλογές, εμφανίζεται ένα νέο παράθυρο για την εισαγωγή των στοιχείων του καινούργιου γεγονότος. Τέλος, με το κουμπί “*Add Event*” ελέγχονται τα *format* των *Start Date* και *Duration* (αν είναι *Appointment*), καθώς και τα *Deadline* και *Status* (αν είναι *Work*). Αν τα *format* των παραπάνω είναι σωστά τότε προστίθεται το event στο *Calendar*, αλλιώς εμφανίζεται ένα παράθυρο που ενημερώνει τον χρήστη με τα αποδεκτά -από το πρόγραμμα- *format*.

#### ❖ *SortedEventsFrame*

Γίνεται χρήση της *SortedEventsFrame*, στη περίπτωση που ο χρήστης επιθυμεί να δει τα γεγονότα που περιέχει ένα συγκεκριμένο καλεντάρι σε κάποια χρονική περίοδο. Συγκεκριμένα, μέσω ενός *JCombo Box* ο χρήστης, μπορεί να επιλέξει από μία πληθώρα λιστών (βλ. *CalendarLists*). Αφού ο χρήστης επιλέξει μια λίστα τότε εμφανίζονται ταξινομημένα -από την ώρα- γεγονότα με συγκεκριμένες πληροφορίες για αυτά τα γεγονότα, όπως αν είναι *Appointment* η *Work*, ο τίτλος, η περιγραφή του γεγονότος, το *Start Date* και *Duration* (αν υπάρχει και είναι *Appointment*), αλλιώς εμφανίζεται το *Due* και *Status* αν είναι *Work*.

#### ❖ *EditEventFrame*

Η *EditEventFrame*, χρησιμοποιείται όταν ο χρήστης έχει επιλέξει το “*EditEvent*”, έχει δηλαδή σκοπό, να τροποποιήσει τα χαρακτηριστικά ενός γεγονότος (ραντεβού/εργασία). Συγκεκριμένα το κάθε event ακολουθείται από ένα *JButton* και εφόσον ο χρήστης το πατήσει, θα ανοίξει ένα νέο παράθυρο και θα μπορεί να εισάγει τις νέες πληροφορίες για το event που επέλεξε. Με το κουμπί *Done* κλείνει το παράθυρο.

#### ❖ *EditStatusFrame*

Γίνεται χρήση της *EditStatusFrame*, όταν επιλεγθεί το “*EditStatus*”, θέλει δηλαδή ο χρήστης, να τροποποιήσει τη κατάσταση μιας εργασίας (βλ. *EditStatus*). Ο χρήστης έχει την δυνατότητα να αλλάξει άμεσα το *Status* μιας εργασίας, μέσω 2 *JRadio Buttons* (Complete αν η εργασία έχει τελειώσει αλλιώς *Incomplete*). Τέλος με το κουμπί *Done* κλείνει το παράθυρο της εξεργασίας του *Status*.