



Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Τεχνολογία Λογισμικού Εργασία Εξαμήνου

Ομάδα Υλοποίησης:

Χανός Στέφανος – 1573

Θέμα Υλοποίησης:

Λογισμικό δημιουργίας προγράμματος νοσηλευτικής μονάδας.

Κώδικας:

Αρχικά εισάγουμε τα ονόματα των νοσηλευτών σε ένα αρχείο **Staff.txt**. Ξεκινάμε με την υλοποίηση της κλάσης **Days.py** η οποία περιέχει ένα κονστράκτορα που θα ορίζει το όνομα κάθε οντότητας Days. Στην ίδια κλάση έχουμε και μια συνάρτηση **getName()** για να μας επιστρέφει το όνομα των ημερών όπου το χρειαστούμε.

Στην κλάση **Shift.py** έχουμε έναν κονστράκτορα για να αρχικοποιεί τον τύπο της βάρδιας που θα αντιστοιχεί σε κάθε νοσηλευτή και μια συνάρτηση **getType()** όπου θα μας επιστρέφει τον τύπο της βάρδιας όπου τον χρειαστούμε.

Στην κλάση **Executive.py** έχουμε δύο μεταβλητές που αντιστοιχούν στα στοιχεία του Διευθυντή μονάδας, έναν κονστράκτορα χωρίς ορίσματα για να μπορούμε να

δημιουργήσουμε μια οντότητα **Executive** , μια συνάρτηση **getUsername()** και μια **getPassword()** για να μπορούμε να τα αξιοποιούμε όπου θέλουμε και τέλος μια συνάρτηση **Approval()** η οποία θα δουλεύει ως boolean για να μας ενημερώνει αν το πρόγραμμα εγκρίνεται ή όχι.

Στην κλάση **Unit_Manager.py** έχουμε δύο μεταβλητές που αντιστοιχούν στα στοιχεία του Υπευθυνου μοναδας , έναν κονστράκτορα χωρίς ορίσματα για να μπορούμε να δημιουργήσουμε μια οντότητα **Unit_manager** , μια συνάρτηση **getUsername()** και μια **getPassword()** για να μπορούμε να τα αξιοποιούμε όπου θέλουμε. Έχουμε επίσης μια συνάρτηση **Admission()** όπου ενημερώνει τον χρήστη πως αμα θέλει να ενημερώσει την λίστα των νοσηλευτών θα πρέπει να πάει στο αρχείο **Staff.txt** και να κάνει τις απαραίτητες αλλαγές. Οι συναρτήσεις **getShiftType()** και **getDaysName()** μας επιστρέφουν την ημέρα και την βάρδια που του αντιστοιχούν κάθε φορά και ομοίως η **DaysOff()** και η **OffWork()** τα ρεπό και τις άδειες αντιστοιχα.

Στην κλάση **Nurses.py** οι συναρτήσεις **DaysOff()** και **OffWork()** μας επιστρέφουν τα ρεπό και τις άδειες που αντιστοιχούν σε κάθε νοσηλευτή της μονάδας.

Στην κλάση **Nursing_Staff.py** αρχικά κληρονομουντε οι κλάσεις **Executive.py** , **Unit_Manger.py** και **Nurses.py**. Έχουμε έναν κονστράκτορα χωρίς ορίσματα για να μπορούμε να δημιουργήσουμε μια οντότητα **Nursing_Staff**. Στην συνέχεια δημιουργούμε μια λίστα με όνομα **Name** μέσα στην οποία θα αποθηκευτούν τα ονόματα των νοσηλευτών. Για να γίνει αυτό διαβάζουμε το αρχείο **Staff.txt** και αντιστοιχούμε κάθε όνομα που έχει σε μια θέση της λίστας μας. Η συνάρτηση **getName()** επιστρέφει το όνομα της αντίστοιχης οντότητας. Τέλος η συνάρτηση **Preferences()** θα επιστρέφει τις μέρες-βάρδιες-άδειες τις οποίες θα προτιμούσαν κάποιοι νοσηλευτές να τους αντιστοιχούν.

Στην κλάση **Parameters.py** αρχικά κληρονομούνται οι κλάσεις **Days.py** , **Shift.py** και **Nursing_Staff.py**. Έχουμε έναν **super** κονστράκτορα για να μπορεί η κλάση μας να βλέπει τις μεταβλητές των κλάσεων που κληρονομεί. Τέλος η συνάρτηση **Available()** μας επιστρέφει τη διαθεσιμότητα των ημερών-βαρδιών που έχει επιλέξει ο κάθε νοσηλευτής.

Στην κλάση **Menu()** υπάρχει η συνάρτηση **Login()** μέσα στην οποία γίνεται ο έλεγχος της ορθότητας των στοιχείων που έδωσε ο χρήστης(δηλαδή αν αντιστοιχούν στον Unit_Manager ή στον Executive) για να μπορέσει να γίνει η είσοδος στο πρόγραμμα. Τέλος έχουμε τη συνάρτηση **Logout()** η οποία πραγματοποιεί έξοδο από το πρόγραμμα.

Στην κλάση **Unit()** υπάρχουν τα προηγούμενα προγράμματα βαρδιών που έχουν δημιουργηθεί.

Στην κλάση **Programm()** αρχικά κληρονομούνται οι κλάσεις **Days.py** , **Shift.py** **Nursing_Staff.py** , **Menu.py** , **Nurses.py** , **Executive.py** , **Unit_Manger.py** , **Parameters.py** , **Programm.py** . Έχουμε έναν κονστράκτορα χωρίς ορίσματα. Στην συνέχεια μέσα στην main δημιουργούμε 7 οντότητες της κλάσης **Days()** μια για κάθε μέρα της εβδομάδας , 3 οντότητες της κλάσης **Shift()** μια για κάθε τύπο βάρδιας. Αποθηκεύουμε τα ονόματα των νοσηλευτών σε μεταβλητές και δημιουργούμε μια οντότητα της κλάσης Executive και Unit_Manger. Μετά υπάρχει μια ατέρμονη λούπα όπου αναλογα τι θα επιλέξει ο χρήστης το πρόγραμμα θα εκτελεί διαφορετική λειτουργία.

Κατανοούμε πως η υλοποίηση του κώδικα μας δεν πληροί όλες τις προϋποθέσεις που πρώτοι εμείς βάλαμε , αλλά ο περιορισμένος χρόνος που είχαμε και οι υποχρεώσεις άλλων μαθημάτων δεν μας επέτρεψαν να αφιερώσουμε τον απαραίτητο χρόνο για την υλοποίηση ενός πλήρως λειτουργικού λογισμικού.