Αντικειμενοστραφής Προγραμματισμός: Β' Μέρος Εργασίας - Χειμερινό εξάμηνο 23-24

Σταύρος Κώτσιλας 1115201700292

Οδηγίες για την εκτέλεση του προγράμματος βρίσκονται στο τέλος του αρχείου

Το πρόγραμμα έτρεξε επιτυχώς στα linux της σχολής. Δοκιμάστηκε για memory leaks με valgrind και είχε 0 ERRORS κατά την λήξη του

Το README χρησιμοποιείται συμπληρωματικά των comments που βρίσκονται στον κωδικά

Τα αρχεία του προγράμματος χωριστήκαν σε header files και .cpp files για καλύτερη απόκρυψη πληροφορίας και αναγνωσιμότητα*

Η ονομασία των συναρτήσεων έγινε με στόχο την άμεση κατανόηση των λειτουργιών τους.

Input Files

Τα δεδομένα στα αρχεία δεν είναι προσαρμοσμένα για μία ρεαλιστική προσομοίωση εξαμήνου, κάποιες μεταβλητές μπορεί να πάρουν και ακραίες τιμές (π.χ πολλά ects). Στο τεχνικό κομμάτι δεν επηρεάζεται η λειτουργικότητα των συναρτήσεων

Τα αρχεία τα όποια **φορτώνονται στην μνήμη του προγράμματος κατά την εκκίνησή του** είναι:

~courses.txt – περιέχει τα μαθήματα που προσφέρει το τμήμα στους φοιτητές (για να εγγραφούν) και στους καθηγητές (για να διδάξουν). Η πληροφορία είναι κωδικοποιημένη ως εξής (όπου "_" είναι space):

όνομαΜαθήματος_ects_υποχρεωτικό(0 ή 1)_εξάμηνο

~departmentInfo.txt – περιέχει της πληροφορίες για το πανεπιστημιακό τμήμα. Η κωδικοποίηση της πληροφορίας είναι:

Όνομα_εξάμηνα_ects_υποχρεωτικάΜαθήματα

~professors.txt – περιέχει τους καθηγητές του τμήματος:

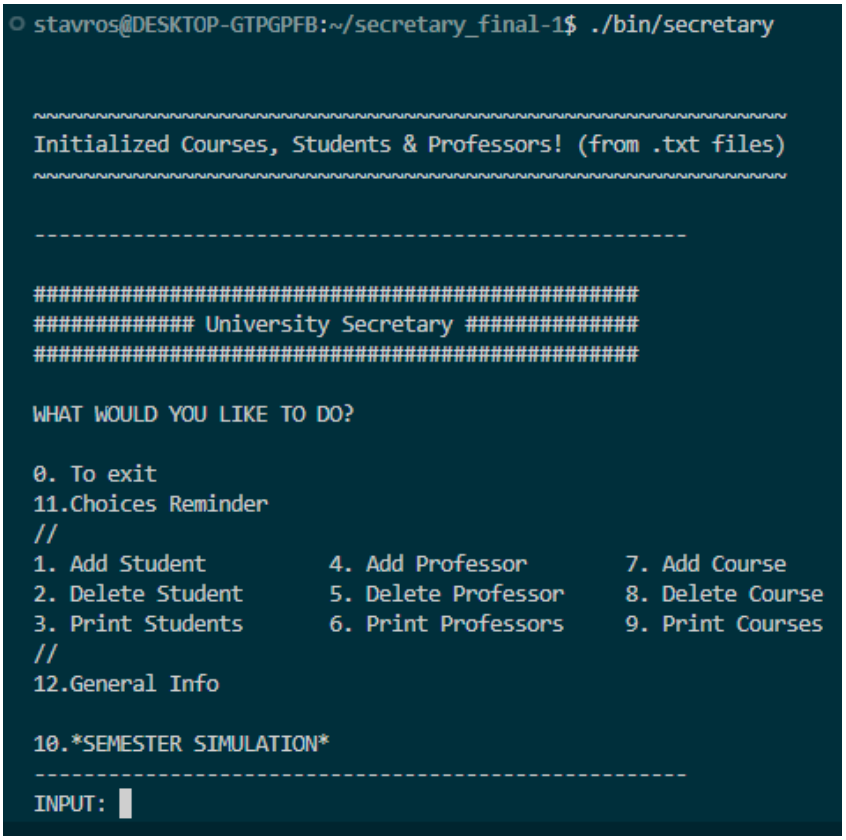
Όνομα_ContactInfo_id

~students.txt – ο κατάλογος με τους φοιτητές:

Όνομα_ContactInfo_id_ΣύνολοECTS_εξάμηνοΦοίτητσης_ΠερασμέναΥποχρεωτικά

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ:

-Κατά την έναρξη του προγράμματος τα αρχεία .txt φορτώνονται στην μνήμη του.
-Παρουσιάζεται το εξής μενού:



ΒΑΣΙΚΕΣ ΛΕΙΤΟΥΡΓΙΕΣ

~Η εντολή 0 τερματίζει το πρόγραμμα.

~Η εντολή 11 εκτυπώνει ξανά το μενού.

~Κατευθείαν μπορούν να εκτελεστούν οι εντολές **3,6,9,** όπως και η **12** οι οποίες θα **εκτυπώσουν στην κονσόλα** (από τα δεδομένα των .txt αρχείων) τους **φοιτητές**, τους **καθηγητές**, τα **μαθήματα** που προσφέρει το τμήμα, και **γενικές πληροφορίες** του τμήματος αντίστοιχα.

~Μπορούν να **δημιουργηθούν** νέοι φοιτητές, καθηγητές και μαθήματα με τις εντολές **1, 2** και **3** αντίστοιχα (και θα εμφανιστούν με τις εντολές **3, 6** και **9**). Τα αρχεία ενημερώνονται κατευθείαν σε **"new_students.txt"**, **"new_professors.txt"** και **"new_courses.txt"** στο φάκελο **"OUTPUT/UPDATED_TXT"**. Το ίδιο ισχύει και για στην **αφαίρεση** τους. Οι εντολές (add/delete) κάνουν **overwrite τα αρχεία "new_students.txt", "new_professors.txt"** και **"new_courses.txt"** (αν γίνει κάποια αλλαγή των αρχικών δεδομένων).

~Έχει υλοποιηθεί η **κλάση Course** με τις μεθόδους και χαρακτηριστικά που ζητούνται. Δίνεται η δυνατότητα μεταφοράς ενός μαθήματος σε άλλο εξάμηνο και παρουσιάζεται παράδειγμα στο **"main.cpp"** αρχείο.

ΠΡΟΣΟΜΟΙΩΣΗ ΑΚΑΔΗΜΑΪΚΟΥ ΕΞΑΜΗΝΟΥ

Οι αλλαγές από τις βασικές λειτουργίες (add/delete) μεταφέρονται ως δεδομένα στην προσομοίωση.

***Το πρόγραμμα δεν υποστηρίζει επαναλαμβανόμενες εκτελέσεις της προσομοίωσης. ***

Μετά την προσομοίωση, δημιουργείται ένας καινούργιος κατάλογος φοιτητών *"OUTPUT/UPDATED_TXT/UPDATED_STUDENT_INFO.txt"* όπου έχουν ενημερωθεί τα ects και τα μαθήματα κορμού (αν έχουν προβιβάσιμο βαθμό)

***Η βαθμολόγηση γίνεται με την παραγωγή ψευδοτυχαίων αριθμών, για χάρη ευκολίας και επίδειξης των *λειτουργιών* των άλλων συναρτήσεων ***

~Στο πρόγραμμα η έννοια του ακαδημαϊκού εξαμήνου υλοποιήθηκε υπό την μορφή ***SEMESTER SIMULATION***, μίας σειράς συναρτήσεων την κλάσης Secretary όπου ανταποκρίνονται στα ζητούμενα της εργασίας:

~Ορίζονται καθηγητές μαθημάτων:

```
void Secretary::AssignCoursesToProfessors()
```

~Εγγράφονται φοιτητές σε μαθήματα (Του εξαμήνου τους και παλαιότερα):

```
void Secretary::EnrollStudentsToCourses()
```

~Βαθμολόγηση των φοιτητών στα μαθήματα αυτά:

```
void Secretary::GradeStudents()
```

~Αποθήκευση των φοιτητών που πέρασαν ένα συγκεκριμένο μάθημα στο αρχείο **"OUTPUT/Students_Passed_" + courseName + ".txt:**

```
void Secretary::StudentsPassedCourse(const string& courseName)
```

~Εκτύπωση στατιστικών των μαθημάτων των καθηγητών στο αρχείο **"OUTPUT/Professor_Statistics.txt":**

```
void Secretary::ProfessorStatistics()
```

~Αναλυτική βαθμολογία των φοιτητών για το εξάμηνο στο φάκελο **“OUTPUT/STUDENT_REPORTS”:**

```
void Secretary::FileStudentsGrades()
```

~Αποθήκευση των φοιτητών που μπορούν να πάρουν πτυχίο (συμπληρωμένα έτη σπουδών, έχει περάσει υποχρεωτικά μαθήματα, επαρκή ects) στο αρχείο **"OUTPUT/Eligible_For_Degree.txt":**

```
void Secretary::EligibleForDegree()
```

Οποιεσδήποτε άλλες συναρτήσεις υπάρχουν στο πεδίο **#SIMULATION FUNCTIONS#** του αρχείου **"secretary.cpp"** είναι συμπληρωματικές και βοηθούν στην λειτουργία των από πάνω.

Η προσομοίωση εκτελεί τις παρακάτω μεθόδους:

```
//----------------------------------------
        case 10:
            secretary.AssignCoursesToProfessors();
            //secretary.PrintProfessorsCourses();

            secretary.EnrollStudentsToCourses();
            //secretary.PrintEnrolledStudens();

            secretary.GradeStudents();
            //secretary.PrintGradedStudents();
            secretary.FileStudentsGrades();
            secretary.StudentsPassedCourse("Game_Development");

            secretary.ProfessorStatistics();
            secretary.WriteUpdatedStudentInfo();
            secretary.EligibleForDegree();
            break;
```

ΕΠΕΞΗΓΗΣΗ ΚΛΑΣΕΩΝ

Για όλες τις κλάσεις έχουν υλοποιηθεί accessors και mutators για τα χαρακτηριστικά τους

~Κλάση **Department** με παράγωγη κλάση (κληρονομικότητα) την **Secretary.**

~Κλάση **Person** με παράγωγες τις **Student** και **Professor.**

~Κλάση **Course** με παράγωγη την **Report** (Χαρακτηριστικά της Report είναι όλα αυτά της Course με επιπλέον **τον βαθμό** του μαθήματος του φοιτητή).

COMPILE & RUN

Προσοχή: για την σωστή λειτουργία του προγράμματος πρέπει να ύπαρχει τα directories OUTPUT/, OUTPUT/STUDENTS_REPORTS και OUTPUT/UPDATED_TXT

>make
>./bin/secretary
>valgrind –leak-check=full ./bin/secretary

>make clean