

Το παρόν README, αφορά την Άσκηση 1, της Εργασίας 2.
Κληρονομικότητα στη c++

Η άσκηση χωρίζεται σε 3 αρχεία:

- main.cpp
Το αρχείο αυτό αποτελείται από τη main συνάρτηση μέσα στην οποία γίνεται η κλήση των διαφόρων συναρτήσεων με σκοπό την πραγματοποίηση του ζητούμενου της άσκησης.
- ask3.cpp
Το αρχείο αυτό έχει τις υλοποιήσεις των διαφόρων συναρτήσεων των κλάσεων καθώς και κάποιες global συναρτήσεις που χρησιμοποιούνται από τη main
- class.h
Το αρχείο αυτό είναι ένα header file που περιέχει τις δηλώσεις των κλάσεων καθώς και τις υλοποιήσεις των constructor & destructor τους.

->class.h / ask3.cpp

class Person:

Η abstract υπερκλάση Person αναφέρεται σε έναν άνθρωπο.

Το protected section αποτελείται από ένα string name που είναι το όνομα του και 3 integers το "fl" (~floor), "cl" (~class), fatigue (~πόντοι κόπωσης).

Στο public section βρίσκονται μόνο οι διάφορες συναρτήσεις που επεξεργάζονται τα παραπάνω members.

- Person(string nam, int f, int c)
Είναι ο constructor της κλάσης και χρησιμοποιείται initializer list για να αρχικοποιηθούν τα member της κλάσης. Name = nam, fl = f, cl = c και το fatigue παίρνει την τιμή 0.
- ~Person()
Είναι ο destructor της κλάσης. Στο σώμα του έχει μόνο ένα μήνυμα εκτύπωσης, μέσα σε σχόλια.
- virtual void print() = 0
Pure virtual συνάρτηση. Κάνει την κλάση abstract.
- void printNAME()
Η συνάρτηση printNAME εκτυπώνει το όνομα του ανθρώπου (χωρίς αλλαγή γραμμής).
- void getF_CL(int &f, int &c)
Η συνάρτηση getF_CL επιστρέφει στα ορίσματα της, τον όροφο (f~fl) και την τάξη (c~cl) που ανήκει ο άνθρωπος.
- void setFATIGUE(int N, int L)
Η συνάρτηση setFATIGUE, υπολογίζει το member "fatigue" της Person.
- int getFATIGUE()
Η συνάρτηση getFATIGUE επιστρέφει το member "fatigue" της Person.

class Student:

Η κλάση Student αναφέρεται σε έναν μαθητή και είναι υποκλάση της Person.

Το private section αποτελείται από ένα string "at" που υποδεικνύει που βρίσκεται ο μαθητής και από ένα string "type" που δηλώνει αν είναι Junior or Senior.

Στο public section βρίσκονται μόνο οι διάφορες συναρτήσεις που επεξεργάζονται τα παραπάνω members.

- Student(string nam, int f, int c)
Είναι ο constructor της κλάσης, που καλεί τον constructor της Person και αρχικοποιεί το member "at" σε "OFF" (~off school) και το "type" σε "Junior" or "Senior" ανάλογα με τον αν η τάξη του είναι <=3 ή όχι αντίστοιχα.
- ~Student()
Είναι ο destructor της κλάσης, που καλεί τον destructor της Person. Στο σώμα υπάρχει μόνο ενημερωτικό μήνυμα ότι ο μαθητής καταστράφηκε.
- virtual void print()
Η print είναι η συνάρτηση που εκτυπώνει τα στοιχεία του μαθητή (όνομα, όροφο, τάξη).
- string getAT()
Η getAT επιστρέφει στο όνομα της το member "at" που δείχνει που βρίσκεται ο μαθητής.
- void setAT(string there)
Η setAT δέχεται ως όρισμα ένα string, το οποίο ανατίθεται στο member "at" της κλάσης που δείχνει που βρίσκεται ο μαθητής.

class Teacher:

Η κλάση Teacher αναφέρεται σε έναν δάσκαλο και είναι υποκλάση της Person.

Το private section αποτελείται από έναν integer "flag" που δείχνει αν ο καθηγητής βρίσκεται εντός ή εκτός τάξης.

Στο public section βρίσκονται μόνο οι διάφορες συναρτήσεις που επεξεργάζονται τα παραπάνω members.

- Teacher(string nam, int f, int c)
Είναι ο constructor της κλάσης, που καλεί τον constructor της Person και αρχικοποιεί το member "flag" σε 0, δηλαδή ο καθηγητής βρίσκεται εκτός τάξης.
- ~Teacher()
Είναι ο destructor της κλάσης, που καλεί τον destructor της Person. Στο σώμα υπάρχει μόνο ενημερωτικό μήνυμα ότι ο μαθητής καταστράφηκε.
- virtual void print()
Η print είναι η συνάρτηση που εκτυπώνει τα στοιχεία του καθηγητή (όνομα, όροφο, τάξη).
- int getFLAG()
Η getFLAG επιστρέφει την τιμή του member "flag" που δείχνει αν ο καθηγητής είναι εντός ή εκτός τάξης.
- void setFLAG()
Η setFLAG θέτει την τιμή του member "flag" σε 1, δηλαδή ο καθηγητής βρίσκεται εντός τάξης.

class Place:

Η υπερκλάση Place αναφέρεται σε έναν χώρο.

Αποτελείται μόνο από public section.

- Place()
Είναι ο constructor της κλάσης που περιέχει μόνο ενημερωτικό μήνυμα μέσα σε σχόλια.
- ~Place()
Είναι ο destructor της κλάσης που περιέχει μόνο ενημερωτικό μήνυμα μέσα σε σχόλια.

class Schoolyard:

Η κλάση Schoolyard αναφέρεται στο προαύλιο και είναι υποκλάση του Place.

Αποτελείται μόνο από public section.

- Schoolyard()
- Είναι ο constructor της κλάσης που περιέχει μόνο ενημερωτικό μήνυμα μέσα σε σχόλια.
- ~Schoolyard()
Είναι ο destructor της κλάσης που περιέχει μόνο ενημερωτικό μήνυμα μέσα σε σχόλια.
- void enter (Student* st);
Η enter θεωρητικά εισάγει έναν μαθητή στο προαύλιο. Στο σώμα της εκτυπώνεται μόνο ενημερωτικό μήνυμα ότι εισήχθη.
- void exit (Student* st);
Η enter θεωρητικά εξάγει έναν μαθητή από το προαύλιο. Στο σώμα της εκτυπώνεται μόνο ενημερωτικό μήνυμα ότι εξήχθη.

class Stairs:

Η κλάση Stairs αναφέρεται στο κλιμακοστάσιο και είναι υποκλάση του Place.

Αποτελείται μόνο από public section.

Τόσο οι constructors/destructors, όσο και οι enter/exit έχουν τις αντίστοιχες λειτουργίες με αυτές της Schoolyard.

- Stairs()
- ~Stairs()
- void enter(Student* st)
- void exit (Student* st)

class Corridor:

Η κλάση Corridor αναφέρεται στο διάδρομο και είναι υποκλάση του Place.

Αποτελείται μόνο από public section.

Τόσο οι constructors/destructors, όσο και οι enter/exit έχουν τις αντίστοιχες λειτουργίες με αυτές της Schoolyard.

- Corridor()
- ~Corridor()
- void enter(Student* st)
- void exit (Student* st)

class Class:

Η κλάση Class αναφέρεται στην τάξη και είναι υποκλάση του Place.

Το private section αποτελείται από 2 integers ppl (~people, πόσοι υπάρχουν μέσα στην τάξη) και capacity (πόσους χωράει η τάξη) καθώς και από 2 πίνακες, εκ των οποίων ο ένας περιέχει pointers που δείχνουν σε Student και ο άλλος pointers που δείχνουν σε Teacher.

Στο public section βρίσκονται μόνο οι διάφορες συναρτήσεις που επεξεργάζονται τα παραπάνω members.

- Class(int cap)

Είναι ο constructor της κλάσης. Το όρισμα cap αρχικοποιεί το member "capacity" και το "ppl" παίρνει την τιμή 0. Επίσης δημιουργείται ένας πίνακας που περιέχει pointers που δείχνουν σε Student, θέσεων cap και ο pointer που δείχνει σε Teacher γίνεται nullptr.

- ~Class

Είναι ο destructor της κλάσης. Κάνει delete τόσο τον πίνακα από pointers όσο και τον pointer σε Teacher.

- void print()

Η print εκτυπώνει το δάσκαλο της τάξης και στη συνέχεια τους μαθητές που βρίσκονται σε αυτή. Τόσο στο δάσκαλος, όσο και σε κάθε μαθητή εκτυπώνονται οι πόντοι κούρασης τους μέσα σε παρενθέσεις.

- void enter(Student* st, int c)

Η enter εισάγει ουσιαστικά ένα μαθητή στην τάξη του (τον προσθέτει στον πίνακα των μαθητών), εμφανίζοντας ενημερωτικό μήνυμα.

- void place(Teacher* tchr)

Η place εισάγει ουσιαστικά έναν καθηγητή στην τάξη του (ανάθεση ορίσματος στο member teacher), εμφανίζοντας ενημερωτικό μήνυμα.

- void operate(int N, int L, int Lt)

Η operate υπολογίζει και αναθέτει τους πόντους κούρασης των καθηγητών και των μαθητών. Το "N" είναι οι ώρες που λειτουργεί το σχολείο, το "L" ο δείκτης κούρασης του μαθητή και το "Lt" ο δείκτης κούρασης του καθηγητή.

class Floor:

Η κλάση Floor αναφέρεται στον όροφο.

Το private section αποτελείται από ένα πίνακα 6 θέσεων που περιέχουν pointers που δείχνουν σε Class, καθώς και έναν pointer που δείχνει σε Corridor.

Στο public section βρίσκονται μόνο οι διάφορες συναρτήσεις που επεξεργάζονται τα παραπάνω members.

- Floor(int CL)
Είναι ο constructor της κλάσης. Καλεί τον constructor της Class στην οποία και περνάει το όρισμα CL. Επίσης, κάνει memory allocation τόσο για τον πίνακα των Class όσο και για το Corridor.
- ~Floor()
Είναι ο destructor της κλάσης. Κάνει delete τους δείκτες.
- void print()
Η print του Floor καλεί για κάθε τάξη την print του Class.
- void enter(Student* st, int f, int c)
Η enter εισάγει ουσιαστικά έναν μαθητή στο όροφο του, καλώντας παράλληλα την enter του Corridor.
- void place(Teacher* tchr, int c)
Η place εισάγει ουσιαστικά τον καθηγητή στον όροφο του και βάζοντας τον στην τάξη του μέσω της place του Class.
- void operate(int N, int Lj, int Ls, int Lt)
Η operate καλεί την operate της κάθε Class. Το "N" είναι οι ώρες λειτουργίας του ορόφου, το "Lj" ο δείκτης κούρασης των Junior μαθητών, το "Ls" των Senior και το "Lt" των καθηγητών.

class School

Η κλάση School αναφέρεται στο σχολείο.

Το private section αποτελείται από 2 integers μετρητές μαθητών (STrppl) και καθηγητών (TCHRppl), έναν pointer που δείχνει σε Schoolyard, έναν που δείχνει σε Stairs και έναν πίνακα 3 θέσεων που περιέχει pointers που δείχνουν σε Floor.

Στο public section βρίσκονται μόνο οι διάφορες συναρτήσεις που επεξεργάζονται τα παραπάνω members.

- School(int CL)
Είναι ο constructor της κλάσης. Αρχικοποιεί το STrppl & TCHRppl με την τιμή "0". Επίσης γίνεται memory allocation για το Schoolyard, Stairs και τον πίνακα 3 θέσεων που περιέχει pointers σε Floor. Το όρισμα περνιέται ως όρισμα στον constructor του Floor. Τέλος εκτυπώνεται ενημερωτικό μήνυμα για τη δημιουργία του σχολείου.
- ~School()
Είναι ο destructor της κλάσης. Κάνει delete τους διάφορους pointers και εκτυπώνει ενημερωτικό μήνυμα καταστροφής του σχολείου.
- void print()
Η print εκτυπώνει τους καθηγητές και τους μαθητές που βρίσκονται σε κάθε τάξη, κάθε ορόφου, καθώς και τους πόντους κούρασής τους.
- void enter(Student* st)
Η enter εισάγει έναν μαθητή στο σχολείο και τον προωθεί μέχρι την τάξη του.
- void place(Teacher* tchr)
Η place εισάγει έναν καθηγητή στο σχολείο και τον προωθεί μέχρι την τάξη του.
- void operate (int N, int Lj, int Ls, int Lt)
Η operate προσομοιώνει τις ώρες λειτουργίας του σχολείου, υπολογίζοντας τους πόντους κούρασης των μαθητών και των καθηγητών.

!SOS!

Υπάρχουν και 2 συναρτήσεις που δεν ανήκουν σε κάποια κλάση.

- 1) Student** STdata(int CL)
- 2) Teacher** TCHRdate()

Οι συναρτήσεις αυτές καλούνται από τη main και έχουν τις εξής λειτουργίες.

- 1) Η STdata επιστρέφει στο όνομα της έναν πίνακα που περιέχει pointers που δείχνουν σε Student. Ουσιαστικά αυτός ο πίνακας είναι ο πίνακας των μαθητών. Στο σώμα της, διαβάζονται δεδομένα από το αρχείο "studentFULL.txt" το οποίο έχει το format: Name-FloorNumber-ClassNumber, που είναι τα στοιχεία ενός μαθητή.
- 2) Η TCHRdata επιστρέφει στο όνομα της έναν πίνακα που περιέχει pointers που δείχνουν σε Teacher. Ουσιαστικά αυτός ο πίνακας είναι ο πίνακας των καθηγητών. Στο σώμα της, διαβάζονται δεδομένα από το αρχείο "teacherFULL.txt" το οποίο έχει το format: Name-FloorNumber-ClassNumber, που είναι τα στοιχεία ενός καθηγητή.

****Είναι σημαντικό να ακολουθηθεί το συγκεκριμένο format δεδομένων. Τα στοιχεία χωρίζονται με παύλα "-" και χωρίς κενά.**

->main.cpp

Το αρχείο περιέχει τη main συνάρτηση. Χρησιμοποιούνται τα ορίσματα που περνάει ο χρήστης στο command line (*). Αρχικά, διαβάζονται και αποθηκεύονται τα δεδομένα. Στη συνέχεια αρχικοποιείται το "σχολείο" και στη συνέχεια επιλέγεται τυχαία μέσω της rand (που παίρνει ως seed τον τρέχοντα χρόνο), αν θα γίνει εισαγωγή μαθητή (enter) η καθηγητή (place). Επίσης καλείται η operate που υπολογίζει τους πόντους κούρασης των μαθητών & καθηγητών και εκτυπώνονται όσοι βρίσκονται μέσα στο σχολείο. Τέλος, γίνεται delete των πινάκων που περιέχουν τους μαθητές & καθηγητές.

(*)

Ο χρήστης μέσω του command line δίνει 5 ορίσματα. Η σειρά είναι:

1. Πόσοι χωράνε σε κάθε τάξη (CL)
2. Δείκτης κούρασης Junior μαθητών (Lj)
3. Δείκτης κούρασης Senior μαθητών (Ls)
4. Δείκτης κούρασης καθηγητών (Lt)
5. Όρες λειτουργίας σχολείου (N)

Το (1), δηλαδή το πρώτο όρισμα, πρέπει να είναι ο αριθμός 2, καθώς τα .txt αρχεία είναι φτιαγμένα έτσι ώστε να χωράνε ακριβώς 2 μαθητές σε κάθε τάξη!

Τα υπόλοιπα ορίσματα μπορούν να πάρουν οποιονδήποτε άλλον ΑΚΕΡΑΙΟ αριθμό.

->makefile

Υπάρχει ένα makefile που κάνει τις εξής εντολές:

- make
 - Δημιουργεί το εκτελέσιμο
 - make clean
 - Σβήνει το εκτελέσιμο
-

ΣΗΜΕΙΩΣΕΙΣ

Για να τρέξετε το πρόγραμμα αρκεί να μπείτε στον φάκελο που περιέχει όλα τα αρχεία που προαναφέρθηκαν (main.cpp, ask3.cpp, class.h, studentsFULL.txt, teachersFULL.txt, makefile) και να γράψετε το εξής:

- make
- ./ask3 2 Lj Ls Lt N

[όπου Lj, Ls, Lt, N οι ακέραιοι αριθμοί που περνάει ο χρήστης ως ορίσματα -> βλ (*)]