

Ηλεκτρονική εξέταση στον Προγραμματισμό Υπολογιστών με C++

Επαναληπτική Εξεταστική Σεπτεμβρίου 2021

Εκφώνηση

Στο διαγώνισμα, καλείστε να υλοποιήσετε μια κλάση η οποία μοντελοποιεί έναν πίνακα 2 διαστάσεων και υλοποιεί ορισμένες πράξεις πάνω σε αυτόν. Η (template) κλάση αυτή πρέπει:

- Να έχει το όνομα **matrix**
- Να ανήκει σε ένα namespace με όνομα **math**
- Να περιέχεται σε ένα αρχείο που θα δημιουργήσετε με όνομα **matrix.h** (δήλωση και υλοποίηση).

Την κλάση τη χρησιμοποιείτε για τη σωστή μεταγλώττιση και λειτουργία των τεσσάρων συναρτήσεων που ελέγχουν τη λειτουργία της κλάσης και καλούνται από τη `main` στο αρχείο `main.cpp`. Πρέπει να διαβάσετε τον κώδικα των `Test` και **να συμπεράνετε ποιες μέθοδοι και τελεστές λείπουν από την `matrix`**, ώστε να τις δηλώσετε και υλοποιήσετε σωστά μέσα στην κλάση στο αρχείο `matrix.h`. Στον κώδικα των `test`, επισημαίνονται για βοήθεια οι σχετικές γραμμές με το σχόλιο:

```
// TODO
```

(θα σας τις επισημάνει και ο compiler βέβαια έτσι κι αλλιώς)

Στον κώδικα που σας δίνεται, υπάρχουν ακόμα τα ακόλουθα αρχεία:

- `main.cpp`: Το κύριο αρχείο που παράγει την εφαρμογή. Εκεί βρίσκονται α) η συνάρτηση `main` της εφαρμογής και β) οι δηλώσεις και υλοποιήσεις των τεσσάρων συναρτήσεων που καλούνται από τη συνάρτηση `main` και πρέπει να τρέχουν σωστά. Μέσα σε αυτές τις συναρτήσεις θα βρείτε ποιες μέθοδοι καλούνται για την κλάση `matrix` που θα φτιάξετε και άρα πρέπει να υλοποιηθούν.
- `tests.h`: περιέχει κώδικα για τον έλεγχο της κλάσης ο οποίος καλείται από τις τέσσερις συναρτήσεις ελέγχου στη `main`. Δεν χρειάζεται να ασχοληθείτε με αυτό το αρχείο.
- `matrix_printer.h`: Περιέχει μια βοηθητική κλάση εκτύπωσης των πινάκων και τη χρησιμοποιούν τα διάφορα τεστ. Δεν χρειάζεται να ασχοληθείτε με αυτό το αρχείο.
- `Matrix_exceptions.h`: Δήλωση και υλοποίηση 2 εξειδικευμένων `exceptions` που μπορείτε να χρησιμοποιήσετε κατά την υλοποίηση μεθόδων της `matrix` (βλ. προαιρετικό ζητούμενο παρακάτω).

Προσοχή: **ΔΕΝ ΠΡΕΠΕΙ ΝΑ ΜΕΤΑΒΑΛΕΤΕ ΚΑΝΕΝΑ ΑΡΧΕΙΟ ΑΠΟ ΑΥΤΑ ΠΟΥ ΣΑΣ ΔΙΝΟΝΤΑΙ**. Γράφετε το κώδικά σας στο αρχείο `matrix.h` και κάνετε `compile` το πρόγραμμα έτσι ώστε ο κώδικας που υπάρχει μέσα στις 4 συναρτήσεις που καλούνται από τη `main` να δουλεύει. Κατά την ανάπτυξη του κώδικά σας, μπορείτε να βάλετε σε σχόλια όποια από αυτές θέλετε, μέχρι να είναι σε θέση να εκτελεστούν. Προφανώς, τοπικά μπορείτε να πειραματιστείτε με τον κώδικά σας όπως επιθυμείτε, αλλά για τη βαθμολόγηση δε θα ληφθεί υπόψη καμία παρέμβαση στα περιεχόμενα των αρχείων που σας δίνονται έτοιμα. Αν κάποια από τις συναρτήσεις ελέγχου δε δουλεύει όταν ολοκληρώσετε τον κώδικά σας, αφήστε τη σε σχόλια, καθώς θα χρειαστεί να καταγράψετε την έξοδο της εκτέλεσης του προγράμματος.

Για την ανάπτυξη του κώδικα μπορείτε προφανώς να χρησιμοποιήσετε οποιονδήποτε compiler C++ (έκδοση γλώσσας 11 και μετά), IDE και λειτουργικό σύστημα.

Ζητούμενα

Ζητούμενο 1. Υλοποιήστε και ανεβάστε το αρχείο `matrix.h` ΜΟΝΟ, ως ένα από τα 2 αρχεία που σας ζητούνται. ΜΗΝ ανεβάσετε κάποιο zip αρχείο με όλο τον κώδικά σας: Δε θα βαθμολογηθεί.

Ζητούμενο 2. Ανεβάστε ένα αρχείο κειμένου που να περιέχει την έξοδο του προγράμματός σας. Μην πάρετε screenshot, καθώς ενδέχεται να μη χωράει όλη η εκτύπωση στην οθόνη ή στο παράθυρο της κονσόλας χωρίς scrolling. Π.χ. από

κονσόλα Windows, pop-up menu από τη system bar του παραθύρου: Edit->Mark και μετά click-drag σε όλο το κείμενο προς επιλογή. Καταχωρείται το αντιγραμμένο κείμενο με δεξί click.

Προαιρετικό ζητούμενο. Μπορείτε να χρησιμοποιήσετε τα 2 exceptions που σας δίνονται ήδη υλοποιημένα στο αρχείο matrix_exceptions για να αναχαιτίσετε άτοπες καταστάσεις κατά την κλήση μεθόδων της matrix. Η μία εξαίρεση αφορά στην ασυμβατότητα διαστάσεων μεταξύ πινάκων και η άλλη στην πρόσβαση στοιχείων εκτός ορίων του πίνακα. Η χρήση εξαιρέσεων προσθέτει μέχρι άλλο ένα 0.5 στο βαθμό, με μέγιστο το 6 πάντα.

Σχόλια

Οι αρίθμηση των στοιχείων του πίνακα είναι zero-based. Π.χ., το πάνω αριστερό στοιχείο του πίνακα A είναι το $a_{0,0}$.

Οι συναρτήσεις ελέγχου που καλούνται από τη main σας, κάνουν κάποιο βασικό validation των λειτουργιών που υλοποιείτε, οπότε πρέπει να ελέγχετε από την έξοδο στην κονσόλα της εφαρμογής σας αν τα τεστ που αναφέρει ότι εκτελεί, ολοκληρώνονται επιτυχώς [PASS] ή όχι [FAIL]. Κάτι που φαίνεται να περνάει τον έλεγχο δε σημαίνει ασφαλώς ότι δουλεύει απαραίτητα σωστά. Η βαθμολόγηση θα γίνει με βάση την υλοποίησή σας και όχι με βάση τα PASS/FAIL.

Η σειρά κλήσης των συναρτήσεων ελέγχου από τη main είναι καθορισμένη έτσι ώστε τα επόμενα τεστ να εξαρτώνται από την ορθή υλοποίηση των μεθόδων της matrix που ζητήθηκαν στις προηγούμενες συναρτήσεις ελέγχου. Καλό είναι να ακολουθήσετε την ίδια σειρά κι εσείς κατά την υλοποίηση της matrix. Για ευκολία, στην αρχή του αρχείου main.cpp σας έχουν δοθεί 4 defines για να ενεργοποιείτε και απενεργοποιείτε γρήγορα τα σχετικά tests:

```
#define ENABLE_TEST1
#define ENABLE_TEST2
#define ENABLE_TEST3
#define ENABLE_TEST4
```

Βαθμολογία		Βαθμός
Ορθή δήλωση κλάσης matrix		0.5
Συνάρτηση ελέγχου	Λειτουργικότητα που ελέγχεται	
test_matrix_creation	constructor, destructor, ερώτηση διαστάσεων πίνακα.	1.0
test_element_queries	Ανάκτηση, ενημέρωση μεμονωμένων στοιχείων πίνακα, καταμέτρηση συγκεκριμένων στοιχείων πίνακα, ανάκτηση υποπίνακα.	2.0
test_copy	Copy constructor, copy assignment	1.0
test_matrix_operations	Υλοποίηση βασικών πράξεων μεταξύ πινάκων	1.5
Χρήση exceptions στην υλοποίηση της matrix (βλ. προαιρετικό ζητούμενο)		+ 0.5