



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Σχεδίαση και Επέκταση ενός Συστήματος Αυτόματης Αξιολόγησης Προγραμματιστικών Ασκήσεων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΝΤΩΝΙΟΣ ΑΓΓΕΛΑΚΗΣ

Επιβλέπων : Νικόλαος Παπασπύρου
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2018



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

Σχεδίαση και Επέκταση ενός Συστήματος Αυτόματης Αξιολόγησης Προγραμματιστικών Ασκήσεων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΝΤΩΝΙΟΣ ΑΓΓΕΛΑΚΗΣ

Επιβλέπων : Νικόλαος Παπασπύρου
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Μαρτίου 2018.

.....
Νικόλαος Παπασπύρου
Αν. Καθηγητής Ε.Μ.Π.

.....
Αριστείδης Παγουρτζής
Αν. Καθηγητής Ε.Μ.Π.

.....
Γεώργιος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2018

.....
Αντώνιος Αγγελάκης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Αντώνιος Αγγελάκης, 2018.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Τα συστήματα αυτόματης αξιολόγησης προγραμματιστικών ασκήσεων αποτελούν εδώ και αρκετά χρόνια απαραίτητο εργαλείο για τη διεξαγωγή ολυμπιάδων και διαγωνισμών πληροφορικής, ενώ παράλληλα διευκολύνουν σε μεγάλο βαθμό την διαδικασία αξιολόγησης εργασιών στον ακαδημαϊκό τομέα. Σκοπός της παρούσας διπλωματικής εργασίας είναι η μελέτη συστημάτων τέτοιου τύπου καθώς και η επέκταση του συστήματος αξιολόγησης Grader που χρησιμοποιείται από το Εργαστήριο Τεχνολογίας Λογισμικού του ΕΜΠ και τον Πανελλήνιο Διαγωνισμό Πληροφορικής.

Αρχικά, μελετούνται μερικά από τα σημαντικότερα συστήματα αυτόματης αξιολόγησης ελεύθερου λογισμικού, τα οποία κατέχουν διαπιστευτήρια από τις μεγαλύτερες διοργανώσεις προγραμματισμού. Έπειτα, παρουσιάζεται η σχεδίαση του Grader και ο τρόπος λειτουργίας του. Τέλος, αναλύονται οι επεκτάσεις που έγιναν στο Grader στο πλαίσιο της παρούσας εργασίας. Οι επεκτάσεις αυτές βελτιώνουν τη λειτουργικότητα του Grader, καθιστώντας το ένα πιο ευέλικτο και σταθερό σύστημα και αυτοματοποιώντας πολλές χρονοβόρες διαδικασίες.

Λέξεις κλειδιά

Συστήματα αξιολόγησης, Συστήματα διαχείρισης διαγωνισμών, CMS, Grader, PHP, Python, Ανάπτυξη Λογισμικού, Λογισμικό ανοιχτού κώδικα, Ελεύθερο λογισμικό.

Abstract

For several years now, automatic grading systems for programming exercises are considered an indispensable tool for conducting Olympiads in Informatics and programming competitions, while greatly facilitating the grading process in the academic field. The purpose of this diploma thesis is the study of such systems as well as the extension of Grader, the grading system used by the NTUA's Software Engineering Laboratory and the Panhellenic Informatics Competition.

Initially, we investigate some of the most renowned FOSS automatic grading systems which have been used from the greatest international competitions. Afterwards, we present Grader's design and mode of operation. Finally, we analyze the extensions implemented for Grader in the context of this thesis. These extensions improve the functionality of Grader, making it a more versatile and stable system and automating many time-consuming common tasks.

Key words

Grading Systems, Contest management systems, CMS, Grader, PHP, Python, Software development, Free and open source software.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα της διπλωματικής μου εργασίας, κ. Νίκο Παπασπύρου για την ευκαιρία που μου έδωσε να συνεργαστώ μαζί του, το χρόνο που μου αφιέρωσε και την υποστήριξη του.

Επιπλέον, ευχαριστώ πολύ την οικογένεια μου που στάθηκε δίπλα μου όλα αυτά τα χρόνια και τους φίλους και συμφοιτητές μου που με στήριξαν σε όλη τη διάρκεια των σπουδών μου.

Αντώνιος Αγγελάκης,
Αθήνα, 22η Μαρτίου 2018

Η εργασία αυτή είναι επίσης διαθέσιμη ως Τεχνική Αναφορά CSD-SW-TR-1-18, Εθνικό Μετσόβιο Πολυτεχνείο, Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών, Εργαστήριο Τεχνολογίας Λογισμικού, Μάρτιος 2018.

URL: <http://www.softlab.ntua.gr/techrep/>
FTP: <ftp://ftp.softlab.ntua.gr/pub/techrep/>

Περιεχόμενα

Περίληψη	5
Abstract	7
Ευχαριστίες	9
Περιεχόμενα	11
Κατάλογος πινάκων	13
Κατάλογος σχημάτων	15
1. Εισαγωγή	17
1.1 Σκοπός	17
1.2 Δομή Εργασίας	18
2. Συστήματα Αυτόματης Αξιολόγησης	19
2.1 CMS	19
2.1.1 Τεχνικά Χαρακτηριστικά	20
2.1.2 Εγκατάσταση και Χρήση	21
2.2 Mooshak 2.0	24
2.2.1 Τεχνικά Χαρακτηριστικά	24
2.2.2 Εγκατάσταση και Χρήση	25
2.3 CATS	27
2.3.1 Τεχνικά Χαρακτηριστικά	27
2.3.2 Εγκατάσταση και Χρήση	28
3. Περιγραφή Grader - Kewii	31
3.1 Στοιχεία και έννοιες του συστήματος	31
3.1.1 Προβλήματα	31
3.1.2 Διαγωνισμοί	31
3.1.3 Αρχεία Ελέγχου	32
3.1.4 Χρήστες/Διαγωνιζόμενοι	32
3.1.5 Υποβολές	32
3.1.6 Προγράμματα Αξιολόγησης	32
3.2 Σύστημα αξιολόγησης Kewii	33
3.3 Διαδικτυακή εφαρμογή Grader	34
3.3.1 Εμφάνιση διαθέσιμων διαγωνισμών και προβλημάτων	35
3.3.2 Εμφάνιση υποβολών προβλήματος και πηγαίου κώδικα	35
3.3.3 Δημιουργία και διαχείριση προβλημάτων και διαγωνισμών	35
3.3.4 Ενεργοποίηση διαγωνισμού και τελική αξιολόγηση	35

4. Προσθήκη Ομάδων Αρχείων Ελέγχου	37
4.1 Προσθήκη blue tag για μη απαραίτητα αρχεία ελέγχου	37
4.1.1 Κίνητρο	37
4.1.2 Υλοποίηση	38
4.2 Ομάδες αρχείων ελέγχου (testcase groups)	39
4.2.1 Κίνητρο	39
4.2.2 Προδιαγραφές	39
4.2.3 Υλοποίηση	40
5. Σχεδίαση για διαχωρισμό Προβλημάτων - Διαγωνισμών	45
5.1 Κίνητρο	45
5.2 Υλοποίηση	47
6. Λοιπές Προσθήκες	51
6.1 Προσθήκη γλώσσας προγραμματισμού Python	51
6.1.1 Προσθήκη γλωσσών στο Grader/Kewii	51
6.1.2 Επιλογή Python	52
6.2 Μαζικό ανέβασμα αρχείων ελέγχων/ομάδων	52
6.2.1 Κίνητρο	52
6.2.2 Υλοποίηση	53
6.2.3 Εργαλείο δημιουργίας	54
6.3 Αλλαγή MySQL connector	56
6.3.1 Συνοπτική περιγραφή επέκτασης PDO	57
7. Συμπεράσματα	59
7.1 Καταληκτικές Παρατηρήσεις	59
7.2 Μελλοντική Εργασία	59
Βιβλιογραφία	61

Κατάλογος πινάκων

3.1	Τύποι εκτέλεσης αρχείων ελέγχου	32
-----	---	----

Κατάλογος σχημάτων

2.1	Η αρχιτεκτονική του CMS	21
2.2	Σελίδα προβλήματος CMS	22
2.3	Σελίδα βαθμολογιών CMS	23
2.4	Σελίδα διαχείρισης διαγωνισμού CMS	23
2.5	Η αρχιτεκτονική του Mooshak	24
2.6	Σελίδα προβλήματος Mooshak	26
2.7	Σελίδα βαθμολογίας Mooshak	26
2.8	Διατύπωση προβλήματος CATS	29
2.9	Σελίδα διαγωνισμού και υποβολής CATS	29
3.1	Ροή Kewii	34
4.1	Προσθήκη blue tag	38
4.2	Βάση πριν τα testcase groups	40
4.3	Βάση μετά τα testcase groups	41
4.4	Επισκόπηση ομάδων αρχείων ελέγχου	42
4.5	Διαχείριση ομάδας αρχείων ελέγχου	42
4.6	Νέα σελίδα εμφάνισης υποβολών	43
4.7	Νέα σελίδα λεπτομέρειων υποβολής	43
5.1	Προβλήματα εκτός διαγωνισμών	46
5.2	Βάση πριν το διαχωρισμό	46
5.3	Βάση μετά το διαχωρισμό	48
5.4	Διαχείριση διαγωνισμών με αντιγραφή προβλημάτων	48
5.5	Εμφάνιση όλων των προβλημάτων	49
6.1	Προσθήκη επιλογής μαζικού ανεβάσματος αρχείων	55
6.2	Εκτέλεση διαδραστικού generator αρχείου descriptor.json	56
6.3	Σύγκριση των επεκτάσεων MySQL της PHP	57

Κεφάλαιο 1

Εισαγωγή

1.1 Σκοπός

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός και η υλοποίηση νέων δυνατοτήτων σε ένα σύστημα αυτόματης αξιολόγησης προγραμματιστικών ασκήσεων. Το σύστημα που τροποποιήθηκε χρησιμοποιείται τόσο από το Εργαστήριο Τεχνολογίας Λογισμικού ¹, για προγραμματιστικές ασκήσεις και εξετάσεις, όσο και από την Ελληνική Εταιρεία Επιστημόνων και Επαγγελματιών Πληροφορικής και Επικοινωνιών (ΕΠΥ) ² για τη διοργάνωση των Πανελλήνιων Διαγωνισμών Πληροφορικής.

Το σύστημα αυτόματης αξιολόγησης (Grader) δέχεται τις υποβολές των διαγωνιζομένων σε συγκεκριμένα προβλήματα που ανήκουν σε διαγωνισμούς, ώστε να τις χαρακτηρίσει ενεργές ή όχι, αξιολογώντας το αποτέλεσμα και την απόδοση τους σε συγκεκριμένα αρχεία ελέγχου. Έπειτα, αφού κλείσουν οι υποβολές, επαναξιολογεί όλες τις ενεργές υποβολές αυτόματα, ώστε να παράξει τα τελικά αποτελέσματα.

Ο Grader, στην πρότερη κατάσταση του, επέτρεπε μόνο τη δημιουργία μεμονωμένων αρχείων ελέγχου και όχι συνδυαστικών ομάδων περιορίζοντας την ευελιξία των διαχειριστών και μην επιτρέποντας οποιαδήποτε ομαδοποίηση ή χαρακτηρισμό των αρχείων. Επιπροσθέτως, ήταν εμφανής η έλλειψη ενός τύπου εκτέλεσης αρχείου ελέγχου που να μην προκαθορίζει την αξιολόγηση μιας υποβολής ως θετική ή αρνητική ανεξάρτητα από την έκβαση του.

Η αρχική σχεδίαση του Grader είχε σκοπό τη δημιουργία ενός συστήματος αυτόματης αξιολόγησης για διαγωνισμούς πληροφορικής, για να χρησιμοποιηθεί κυρίως από την ΕΠΥ. Ως αποτέλεσμα, κάθε πρόβλημα αντιστοιχίζεται σε έναν μόνο διαγωνισμό και τόσο οι διαγωνιζόμενοι όσο και οι υποβολές τους συνδέονται με το πρόβλημα. Για τη χρήση του Grader σε εργασίες προγραμματισμού, θα μας ωφελούσε η ύπαρξη διαχωρισμού προβλήματος και υποβολών ώστε τα προβλήματα να μπορούν να επαναχρησιμοποιηθούν ευκολότερα, χωρίς να "κουβαλάνε" τις υποβολές που έχουν γίνει σε αυτά.

Επιπλέον, κρίθηκε σημαντικό να προστεθεί η Ρυθση στις διαθέσιμες γλώσσες υποβολής καθώς πρόκειται για μια από τις πλέον δημοφιλείς γλώσσες και χρησιμοποιείται ως εισαγωγική γλώσσα προγραμματισμού σε σπουδαία ακαδημαϊκά ιδρύματα, όπως είναι το MIT και το Stanford [Guo14]. Τέλος, ήταν απαραίτητο να γίνουν μικρές βελτιστοποιήσεις στη λογική του Grader, να προστεθούν μικρότερες δυνατότητες που επιδιώκουν τη βελτίωση της ευκολίας χρήσης για διαγωνιζόμενους και διαχειριστές και να αντικατασταθούν απαρχαιωμένα, χωρίς ενεργή ανάπτυξη εργαλεία/βιβλιοθήκες για την επίτευξη καλύτερης απόδοσης και ασφάλειας.

¹ <http://grader.softlab.ntua.gr>

² <http://hellenico.gr/grader>

1.2 Δομή Εργασίας

Η εργασία ακολουθεί την παρακάτω δομή:

- **Κεφάλαιο 2:** Συστήματα Αυτόματης Αξιολόγησης
Παρουσιάζουμε κάποια γνωστά συστήματα αυτόματης αξιολόγησης με παρόμοια λειτουργία και σκοπό με το Grader.
- **Κεφάλαιο 3:** Περιγραφή Grader - Kewii
Περιγράφονται τα βασικά δομικά στοιχεία και έννοιες του συστήματος που μελετούμε. Αναλύεται το σύστημα αξιολόγησης Kewii, η διαδικτυακή εφαρμογή Grader και η επικοινωνία μεταξύ τους.
- **Κεφάλαιο 4:** Προσθήκη Ομάδων Αρχείων Ελέγχου
Περιγράφεται αρχικά η προσθήκη του blue tag, ενός νέου τύπου εκτέλεσης αρχείων ελέγχου. Αναλύεται η σχεδιαστική λογική και η υλοποίηση της νέας δυνατότητας του Grader με σκοπό την ομαδοποίηση των αρχείων ελέγχου.
- **Κεφάλαιο 5:** Σχεδίαση για διαχωρισμό Προβλημάτων - Διαγωνισμών
Περιγράφεται η υλοποίηση της τροποποίησης του συστήματος για την βελτίωση της λειτουργίας του Grader στο πλαίσιο προγραμματιστικών ασκήσεων.
- **Κεφάλαιο 6:** Λοιπές Προσθήκες
Στο συγκεκριμένο κεφάλαιο παρατίθενται βελτιώσεις και προσθήκες μικρότερου μεγέθους: Προσθήκη της Python, μαζικό ανέβασμα αρχείων/ομάδων και αλλαγή επέκτασης MySQL.
- **Κεφάλαιο 7:** Συμπεράσματα
Στο τελευταίο κεφάλαιο παρουσιάζονται παρατηρήσεις σχετικά με τη διπλωματική και αναφέρονται ιδέες για περαιτέρω ανάπτυξη του συστήματος.

Κεφάλαιο 2

Συστήματα Αυτόματης Αξιολόγησης

Ο Grader έχει σκοπό τη διοργάνωση προγραμματιστικών διαγωνισμών ή εργασιών με αυτόματο τρόπο υποβολής, εκτέλεσης και αξιολόγησης των λύσεων. Το παρόν σύστημα είναι κλειστού κώδικα, όμως θα είχε σημασία να μελετήσουμε συστήματα με παρόμοια λειτουργία ώστε να δούμε ομοιότητες και διαφορές με το δικό μας.

Προτιμήθηκε να ελεγχθούν μόνο συστήματα ελεύθερου λογισμικού και ανοιχτού κώδικα διότι μας προσφέρουν σημαντικά πλεονεκτήματα. Αρχικά, μας επιτρέπουν να ερευνήσουμε τον τρόπο που είναι σχεδιασμένα και να πάρουμε ιδέες για τον Grader. Επιπλέον, πολλές φορές παρέχουν καλύτερη ασφάλεια, καθώς οποιοσδήποτε μπορεί να ελέγξει τον κώδικα για ευπάθειες. Φυσικά, το τελευταίο ισχύει υπό την προϋπόθεση ότι υπάρχει πρωτοβουλία για ανεξάρτητο έλεγχο της ασφάλειας (audit), αφού η απλή δημοσιοποίηση του κώδικα μπορεί να δίνει την ψευδαίσθηση (όπως αναφέρεται στο [Hans02]). Τέλος, η μελέτη των εν λόγω συστημάτων είναι απαραίτητη αφού, χάρη στην ελευθερία χρήσης που προσφέρουν, μπορούν να αντικαταστήσουν το Grader χωρίς μεγάλο κόστος (κυρίως αυτό της μετάβασης) σε περίπτωση που κριθούν ως ανώτερα.

Τα συστήματα που θα μελετηθούν είναι τα παρακάτω:

- CMS
- Mooshak 2.0
- CATS

Και τα 3 έχουν σχεδιαστεί με σκοπό τη διεξαγωγή προγραμματιστικών διαγωνισμών, ολυμπιάδων πληροφορικής, αλλά και χρήση για εργαστηριακές ασκήσεις και εργασίες.

2.1 CMS

Το πρώτο σύστημα που θα μελετήσουμε είναι το Contest Management System, CMS εν συντομία ¹. Πρόκειται για ένα καταναμημένο σύστημα διαχείρισης και διεξαγωγής διαγωνισμών το οποίο σχεδιάστηκε αρχικά για την Διεθνή Ολυμπιάδα Πληροφορικής του 2012. Αποτελείται από ένα πλήθος μικρο-υπηρεσιών που συνθέτουν το συνολικό σύστημα.

Είναι, πιθανότατα, το πιο ολοκληρωμένο σύστημα για διαγωνισμούς, δεδομένου του μεγάλου αριθμού διαπιστευτηρίων που κατέχει, συμπεριλαμβανομένης της χρήσης του σε όλες σχεδόν τις Διεθνείς Ολυμπιάδες από το 2012. Οι δημιουργοί του είχαν ως στόχο τη δημιουργία ενός συστήματος ασφαλούς, ανθεκτικού σε σφάλματα λογισμικού και υλικού, ανοιχτού, επεκτάσιμου και εύχρηστου.

Η ανάπτυξη του ξεκίνησε το 2010 και πλέον βασίζεται στην κοινότητα για συνεισφορά στην ανάπτυξη του, στην προσθήκη μεταφράσεων και στη βελτίωση του documentation. Η άδεια που χρησιμοποιεί είναι η AGPL-3+ (GNU Affero General Public License), η οποία επιτρέπει εμπορική χρήση, τροποποίηση και διανομή, με την προϋπόθεση να παραμείνει η ίδια άδεια και να δημοσιευτεί ο πηγαίος κώδικας.

¹ <https://cms-dev.github.io/>

2.1.1 Τεχνικά Χαρακτηριστικά

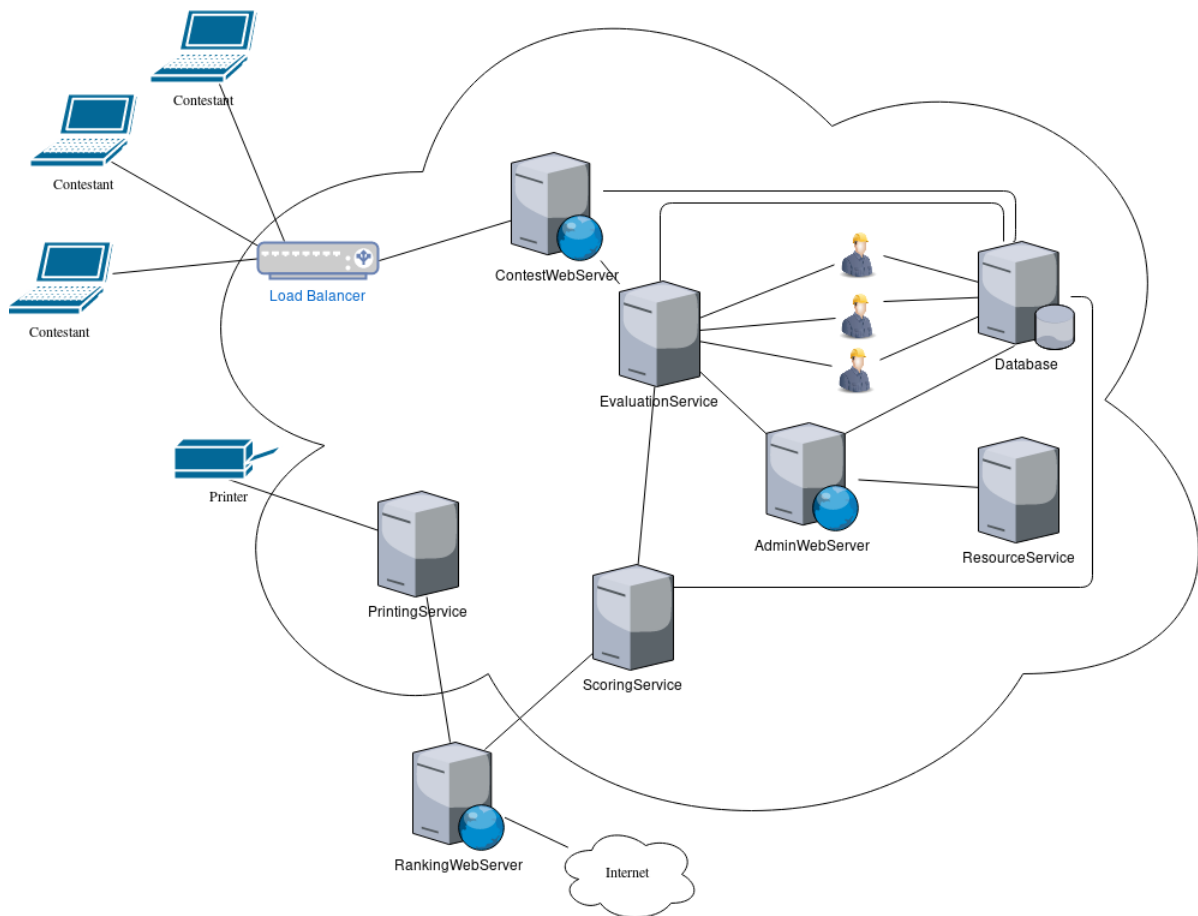
Το CMS είναι γραμμένο σε Python και αποτελείται από πολλά μικρά κομμάτια που αναλαμβάνουν μια ξεχωριστή λειτουργία. Αυτά μπορούν να έχουν εγκατασταθεί σε διαφορετικούς εξυπηρετητές, ακόμα και σε απομακρυσμένα συστήματα. Οι υπηρεσίες-κομμάτια, από τα οποία αποτελείται είναι τα παρακάτω:

- **LogService**
Συλλέγει σε ένα μέρος όλα τα μηνύματα καταγραφής.
- **ResourceService**
Λαμβάνει δεδομένα για όλες τις υπηρεσίες που τρέχουν και μπορεί να τις ξεκινήσει ταυτόχρονα.
- **Checker**
Ελέγχει την κατάσταση των υπηρεσιών.
- **EvaluationService**
Οργανώνει την ουρά υποβολών και διανέμει τις εργασίες στους Workers.
- **Worker** Εκτελεί τις εργασίες σε ένα αποκλεισμένο περιβάλλον.
- **ScoringService**
Συλλέγει τα αποτελέσματα των εκτελέσεων και τα αξιολογεί για να παράξει τις βαθμολογίες.
- **ProxyService**
Στέλνει τις βαθμολογίες στον εξυπηρετητή Αποτελεσμάτων. [MAGG14]
- **PrintingService**
Αναλαμβάνει την εκτύπωση των εγγράφων.
- **ContestWebServer**
Ο εξυπηρετητής της ιστοσελίδας του διαγωνισμού, με τον οποίο αλληλεπιδρούν οι διαγωνιζόμενοι.
- **AdminWebServer**
Ο εξυπηρετητής της ιστοσελίδας διαχείρισης του διαγωνισμού.
- **RankingWebServer**
Ο εξυπηρετητής της ιστοσελίδας εμφάνισης των αποτελεσμάτων.

Όλες οι παραπάνω λειτουργίες αλληλεπιδρούν μεταξύ τους με τον τρόπο που φαίνεται στο σχήμα 2.1.

Ο κύκλος διεξαγωγής ενός διαγωνισμού είναι γνώριμος. Αφού στηθούν όλες οι απαραίτητες υπηρεσίες, οι διαχειριστές μπορούν να δημιουργήσουν το διαγωνισμό μέσω του AdminWebServer, μαζί με τα προβλήματα και τα τεστ που περιέχει. Έπειτα, κάθε διαγωνιζόμενος αλληλεπιδρά με τη σελίδα υποβάλλοντας τις λύσεις του. Αυτές αποθηκεύονται στο δίσκο μέχρι να δοθούν σε κάποιο Worker από το EvaluationService και να εκτελεστούν στο ασφαλές περιβάλλον. Σε εκείνο το σημείο αναλαμβάνει τα ηνία το ScoringService που αξιολογεί τις εκτελέσεις και στέλνει τα αποτελέσματα στο RankingWebServer, ώστε να ανανεωθεί με την τελευταία εικόνα.

Η βάση που χρησιμοποιεί το CMS είναι PostgreSQL και η λειτουργία του βασίζεται σε αποκλειστικά χαρακτηριστικά της (LO, Large Object), οπότε μόνο αυτή υποστηρίζεται. Οι γλώσσες προγραμματισμού που υποστηρίζει η ευσταθής έκδοση του είναι οι: C, C++, PHP, Pascal, Java και Python2. Στην τελευταία έκδοση υπό ανάπτυξη (development) υποστηρίζονται επίσης C#, Haskell, Python3 και Rust. Για την επέκτασή του με παραπάνω γλώσσες αρκεί η προσθήκη ενός αρχείου προδιαγραφών για κάθε νέα γλώσσα.



Σχήμα 2.1: Οι κυριότερες υπηρεσίες του CMS και οι σχέσεις μεταξύ τους. Βασισμένο στο σχήμα 1 του [Magg12].

2.1.2 Εγκατάσταση και Χρήση

Η εγκατάσταση προϋποθέτει την ύπαρξη πλήθους εργαλείων για την υποστήριξη των γλωσσών και του συστήματος. Όπως έχει αναφερθεί, το CMS αποτελείται από πολλές διαφορετικές υπηρεσίες, οι οποίες πρέπει να τρέχουν σε ένα ή παραπάνω μηχανήματα για την πλήρη λειτουργία. Χάρη στην ύπαρξη του ResourceService όλες οι υπηρεσίες ενός μηχανήματος μπορούν να εκκινήσουν ταυτόχρονα ώστε να μειωθεί η χρονοβόρα εργασία.

Για να τρέξει, επίσης, το σύστημα θα πρέπει να γίνουν οι απαραίτητες ρυθμίσεις στην βάση αλλά και στο CMS. Μόλις ολοκληρωθεί κι αυτό το βήμα, οι διαχειριστές μπορούν να δημιουργήσουν διαγωνισμούς και προβλήματα με τρεις τρόπους: μέσω της διεπαφής Admin, κατευθείαν από το σύστημα αρχείων τους περιγράφοντας με αρχείο προδιαγραφών τη μορφή τους ή με απευθείας εισαγωγή προηγούμενου διαγωνισμού που εξάχθηκε από CMS.

Οι διαγωνιζόμενοι συμμετέχουν στο διαγωνισμό μέσω της σελίδας του ContestWebServer. Εκεί βλέπουν για κάθε πρόβλημα την περιγραφή του και όλα τα σχετικά με αυτό στοιχεία και μπορούν να υποβάλλουν τις λύσεις τους. Ένα ενδιαφέρον χαρακτηριστικό είναι ότι οι διαγωνιζόμενοι μπορούν να δημιουργούν δικά τους αρχεία ελέγχου, στα οποία θα ελεγχθεί η λύση τους. Επιπλέον, υπάρχει η δυνατότητα χρήσης tokens, τα οποία μοιράζονται στους διαγωνιζόμενους και περιορίζουν τις φορές που μπορούν να δουν τα αναλυτικά αποτελέσματα της υποβολής τους σε φανερά και κρυφά αρχεία ελέγχου.

My Contest
localhost:8888/tasks/compression/description
Logged in as Mario Rossi (m.rossi) Logout Automatic

Server time: 09:08:43
Time left: 02:51:16

Overview
Communication
MYTASK
Statement
Submissions
MYOTHERTASK
Statement
Submissions
YOUGETIT
Statement
Submissions
Documentation
Testing

Contest Management System is released under the GNU Affero General Public License .

A very hard example task (mytask) description

Statement

Download task statement

Some details

Type	Two steps	
Time limit	1 second	
Memory limit	256 MiB	
Compilation commands	C	<code>/usr/bin/gcc -DEVAL -static -O2 -std=c11 -o manager mytask_send.c mytask_recv.c manager.c -lm</code>
	C++	<code>/usr/bin/g++ -DEVAL -static -O2 -std=c++11 -o manager mytask_send.cpp mytask_recv.cpp manager.cpp</code>
Tokens	You have an infinite number of tokens.	

Attachments

mytask_recv.c C source code 646 B	mytask_send.c C source code 505 B	grader.c C source code 1.97 KIB
mytask_recv.cpp C++ source code 646 B	mytask_send.cpp C++ source code 505 B	grader.cpp C++ source code 1.97 KIB

Σχήμα 2.2: Η σελίδα ενός προβλήματος, όπως τη βλέπει ένας διαγωνιζόμενος. Διακρίνονται τα στοιχεία του προβλήματος και όλα τα επισυναπτόμενα. Πηγή: <https://cms-dev.github.io/screenshots.html>

Ακολουθούν φωτογραφίες από τις διεπαφές του διαγωνισμού, του Admin και της σελίδας των αποτελεσμάτων.

your logo here

Day 2
2:29:41

✓	Rank	First Name	Last Name	Team	tes...	lin...	ca...	Day 1	co...	con...	bri...	Day 2	Global
	1	Velva	Melodee		100	100	100	300	51.62	100	100	251.62	551.62
	2	Ling	Else		100	100	100	300	50.09	100	85	235.09	535.09
	3	Hassie	Alanna		100	100	100	300	0	31.22	85	116.22	416.22
	4	Weldon	Luther		100	100	100	300	3.14	15.24	85	103.38	403.38
	5	Ling	Michele		100	70	100	270	49.32	0	70	119.32	389.32
	6	Aurea	Martha		70	100	100	270	51.64	15.24	30	96.88	366.88
	7	Marcellus	Millie		100	100	100	300	6.91	15.24	30	52.15	352.15
	8	Miss	Nam		100	100	100	300	0	18.03	30	48.03	348.03
	9	Lorette	Delphine		100	100	100	300	46.48	0	0	46.48	346.48
	10	Emelda	Weldon		100	100	100	300	12.62	24.07	0	36.69	336.69
	11	Lorette	Ciprietti		100	100	100	300	13.63	0	0	13.63	313.63
	12	Lakendra	Naoma		100	100	100	300	4.75	8.11	0	12.86	312.86
	13	Shantae	Nam		100	100	100	300	0	0	0	0	300
	13	Launa	Aurea		100	100	100	300	0	0	0	0	300
	15	Deonna	Marcelino		20	100	100	220	0	16.28	0	16.28	236.28
	16	Geralyn	Ehtel		100	70	5	175	3.9	29.92	0	33.82	208.82
	17	Delphine	Luther		20	70	5	95	13.16	0	85	98.16	193.16
	18	Alanna	Sherilyn		20	100	5	125	0	29.92	0	29.92	154.92
	19	Miss	Johnnie		50	70	5	125	6.8	17.44	0	24.24	149.24
	20	Luther	Coy		20	70	0	90	49.32	0	0	49.32	139.32
	21	Lorette	Luther		20	100	0	120	0	0.06	0	0.06	120.06

Σχήμα 2.3: Η σελίδα της βαθμολογίας, με τη συνολική κατάταξη και ανά διαγωνιζόμενο σε όλα τα προβλήματα. Πηγή: <https://cms-dev.github.io/screenshots.html>

Administration
day2

- Overview
- Resource usage
- ***
- General
- Ranking
- Submissions
- Users
- Tasks
- Announcements
- Questions
- ***

Time left: 1:05:43

CMS codebase is released under the GNU Affero General Public License.

Overview

Submissions status

Status	Number
Scored	278
Compilation failed	16
Evaluating...	1
Cannot evaluate (please check)	1
Invalid (please check)	-1

Queue status

Id	Job	Priority	Since
1	Evaluating the result of submission_2058 on dataset_15 [10 time(s) in queue] [testcase: '018']	2	4 second(s)

Workers status

Shard	Connected	Current job	Since	Action
0	Yes	Evaluating the result of submission_2058 on dataset_15 [testcase: '014']	0 second(s)	Disable
1	Yes	Evaluating the result of submission_2058 on dataset_15 [testcase: '015']	0 second(s)	Disable
2	Yes	Evaluating the result of submission_2058 on dataset_15 [testcase: '016']	0 second(s)	Disable
3	Yes	Evaluating the result of submission_2058 on dataset_15 [testcase: '017']	0 second(s)	Disable
4	No	N/A	N/A	Disable
5	No	N/A	N/A	Disable
6	No	N/A	N/A	Disable

Σχήμα 2.4: Η σελίδα της διαχείρισης ενός διαγωνισμού. Διακρίνονται συνολικά στατιστικά για τις υποβολές, η κατάσταση της ουράς και των Workers. Πηγή: <https://cms-dev.github.io/screenshots.html>

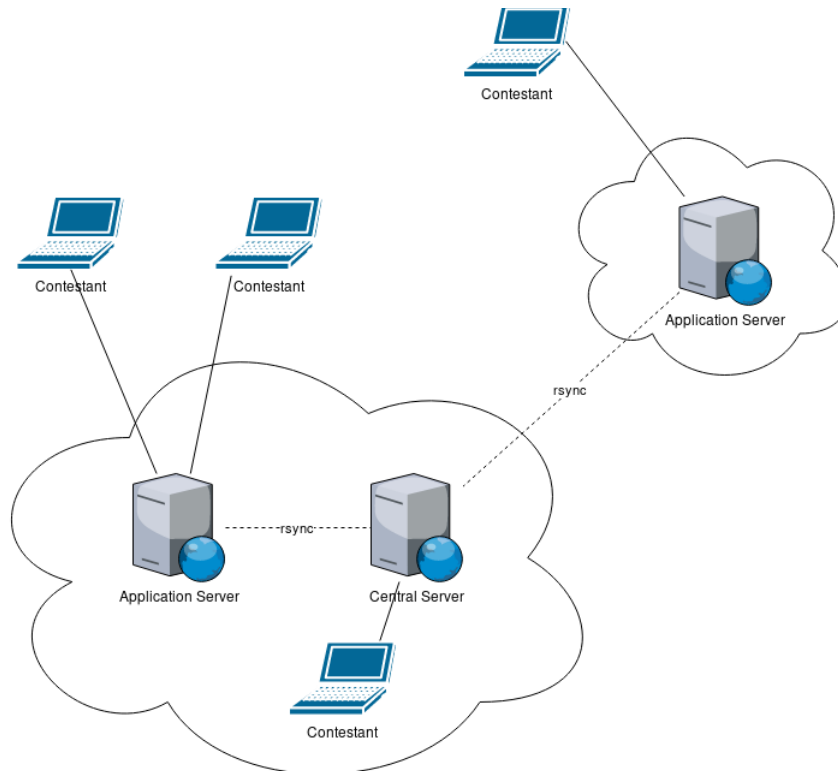
2.2 Mooshak 2.0

Το Mooshak 2.0² είναι κι αυτό ένα σύστημα διαχείρισης διαγωνισμών με αυτόματη αξιολόγηση για τις υποβολές. Αποτελεί τη νεότερη υλοποίηση του Mooshak, με μεταφορά του κώδικά από C++ και Tcl σε Java με χρήση της εργαλειοθήκης Google Web Toolkit. Η αρχική έκδοση του Mooshak δημιουργήθηκε το 2000 και βασίζεται σε ένα παλαιότερο διαδικτυακό σύστημα εκμάθησης, το Ganesh. Ο κώδικας του διατίθεται επίσης ελεύθερα με χρήση της άδειας Artistic License, η οποία δεν προϋποθέτει την διατήρηση άδειας ελεύθερου λογισμικού σε μελλοντική τροποποίηση και διανομή.

Οι βασικές του προδιαγραφές βασίζονται στους κανόνες των διαγωνισμών ICPC, αλλά υποστηρίζει και άλλες μορφές διαγωνισμών, όπως είναι οι Ολυμπιάδες πληροφορικής. Έχει χρησιμοποιηθεί σε πληθώρα τοπικών και διεθνών διαγωνισμών αλλά και σταθερά στο πλαίσιο εκμάθησης και αξιολόγησης φοιτητών [Leal08].

2.2.1 Τεχνικά Χαρακτηριστικά

Το Mooshak 2.0 βασίζεται σε μια κοινή αρχιτεκτονική εξυπηρετητή και πελατών με τη διαφορά ότι υποστηρίζει τη δημιουργία πρόσθετων κόμβων. Οι κόμβοι αυτοί έχουν σκοπό τη διατήρηση της σταθερότητας και απαιτούν συνεχή συγχρονισμό με τον κεντρικό εξυπηρετητή ώστε να αναλαμβάνουν μέρος του δικτυακού φόρτου αλλά και να αποτελούν εφεδρικές λύσεις σε περίπτωση βλάβης των υπολοίπων κόμβων.



Σχήμα 2.5: Η αρχιτεκτονική του Mooshak με τρεις εξυπηρετητές, όπου οι δύο βρίσκονται στο ίδιο τοπικό δίκτυο και όλοι συγχρονίζουν τα δεδομένα τους. Βασισμένο στο σχήμα 4 του [Leal03].

Αντίθετα με άλλα συστήματα, το Mooshak δε χρησιμοποιεί βάση και περιορίζεται στην αποθήκευση όλων των δεδομένων του στο σύστημα αρχείων. Οι γλώσσες προγραμματισμού που υποστηρίζει περιλαμβάνουν τις C, C++, Java, Pascal, Perl, Python, Haskell, Haskell και Prolog, ενώ η επέκταση του ώστε να υποστηρίξει πρόσθετες γλώσσες δεν αποτελεί δύσκολη διαδικασία [Ribe08].

Το σύστημα είναι επίσης παραμετροποιήσιμο ως προς τον τρόπο αξιολόγησης καθώς έχει σχεδιαστεί για πολλούς διαφορετικούς τύπους διαγωνισμών. Χαρακτηριστικό είναι ότι στη διαδικασία της υποβολής, πριν από τα στάδια της μεταγλώττισης, της εκτέλεσης και της αξιολόγησης, δίνεται η δυνατότητα προσθήκης μιας λειτουργίας στατικής ανάλυσης του πηγαίου κώδικα της υποβολής και αξιολόγησης του με χρήση κριτηρίων ορισμένων από τους διαχειριστές του διαγωνισμού. Αυτό επιτρέπει τη χρήση του Mooshak, παραδείγματος χάρη για ένα διαγωνισμό τύπου code golf, δηλαδή επίτευξης της λύσης με το λιγότερο δυνατό κώδικα.

2.2.2 Εγκατάσταση και Χρήση

Η εγκατάσταση του Mooshak 2 δεν έχει πολλές απαιτήσεις. Συγκεκριμένα, χρειάζεται μόνο το περιβάλλον της Java και το λογισμικό του εξυπηρετητή. Μόλις γίνει αυτό, μπορούν να στηθούν επίσης επιπλέον κόμβοι εάν είναι επιθυμητό, τοπικά ή απομακρυσμένα. Η δημιουργία των διαγωνισμών και των προβλημάτων γίνεται μέσω της διαδικτυακής διεπαφής του Διαχειριστή.

Οι διαγωνιζόμενοι, μπαίνοντας στην ιστοσελίδα του Mooshak, μπορούν να διαβάσουν τις περιγραφές των προβλημάτων και να πραγματοποιήσουν τις υποβολές τους είτε ανεβάζοντας το πρόγραμμα τους, είτε χρησιμοποιώντας τον επεξεργαστή κειμένου της ιστοσελίδας ώστε να το συντάξουν επιτόπου.

Οι δημιουργοί του Mooshak παραθέτουν στη σελίδα του μόνιμα περιβάλλοντα δοκιμών, στα οποία μπορεί ο οποιοσδήποτε να χρησιμοποιήσει ένα ζωντανό Mooshak σύστημα είτε για να πραγματοποιήσει υποβολές, είτε για να δημιουργήσει νέους διαγωνισμούς και προβλήματα.

J - Financial Risk

$$\frac{\sum_c \int_{t_0}^{t_1} p(D_c(t) - L_c)}{\sum_c \int_{t_0}^{t_1} D_c(t)}$$

You were asked to write a program to compute the percentage of uncovered risk in the last fiscal year, using the formula above and the data provided by a listing of the ACME's ERP.

Input description

The listing you will use as input starts with the number of clients with sales in the last fiscal year. For each client the first line has the debt limit and the number of sales. In the following lines, one for each sale, there is the sale's value, the day of the year of the invoice and the day of the year of the receipt.

Output description

Your program must write the percentage of uncovered risk, truncated to 2 fractional digits.

Sample input

```
1
40000 3
35000 30 60
15000 40 70
40000 90 120
```

Sample output

```
7.40%
2001-09-24
```

Σχήμα 2.6: Η σελίδα παρουσίασης ενός προβλήματος για τους διαγωνιζόμενους στο σύστημα Mooshak.

#	Flag	Group	Team	?	A	B	J	P	Q	R	Solved	Points
1		myGroup	team				223 13:57 (3)		143 12:52 (0)		2	002 03:49
2		DFLT	TULIO ROBERTO NASCIMENTO DE SOUSA								0	00:00:00
3		DFLT	caha								0	00:00:00
4		DFLT	RENATO PINTO DE ANDRADE FILHO								0	00:00:00
5		DFLT	ALEFF VINICIUS ARAUJO GOMES PASSOS								0	00:00:00
6		DFLT	DANILO MEDEIROS DE ALMEIDA CARDINS								0	00:00:00
7		DFLT	BARBARA NICOLY MENEZES DE OLIVEIRA								0	00:00:00
8		DFLT	MATHEUS LEOR LOPES DE LIMA MARROCOS								0	00:00:00
9		DFLT	EDUARDO SILVA FERNANDES								0	00:00:00
10		DFLT	ENIO MATHEUS ANDRADE RANGEL								0	00:00:00
11		DFLT	team1				----- (2)				0	00:00:00
12		DFLT	RODRIGO RAYNTON								0	00:00:00

Σχήμα 2.7: Η σελίδα βαθμολογίας όλων των διαγωνιζόμενων ομάδων σε ένα διαγωνισμό.

2.3 CATS

Το CATS ³ είναι το τρίτο σύστημα που θα αναλυθεί. Αφορά και αυτό τη διεξαγωγή και τον έλεγχο προγραμματιστικών διαγωνισμών και συντηρείται από τον Alexander Klenin του Far Eastern Federal University [M04]. Χρησιμοποιείται τόσο για μεγάλες διοργανώσεις, όπως είναι το ICPC Ρωσίας και Άπω Ανατολής, καθώς και για πλήθος ακαδημαϊκών μαθημάτων και διαγωνισμών. Κατέχει άδεια GPL 2.0 επιτρέποντας την ελεύθερη χρήση, τροποποίηση και διανομή του.

Έχει αρκετές δυνατότητες που το διακρίνουν ανάμεσα στα υπόλοιπα συστήματα συμπεριλαμβανομένων των παρακάτω:

- Μεγάλος αριθμός προγραμματιστικών γλωσσών και μεταγλωττιστών
- Αυτόματος ορισμός διαγωνισμών, προβλημάτων και αρχείων ελέγχου με χρήση ενός zip αρχείου με μια περιγραφή σε XML
- Έτοιμα πρόσθετα (modules) για την αξιολόγηση των υποβολών, π.χ. για αυτόματη δημιουργία αρχείων ελέγχου
- Περιορισμός πρόσβασης ανάλογα με τη διεύθυνση IP των χρηστών
- Αυτόματος έλεγχος για αντιγραφή κώδικα μεταξύ των διαγωνιζομένων

2.3.1 Τεχνικά Χαρακτηριστικά

Το CATS είναι υλοποιημένο σε Perl και χρησιμοποιεί βάση δεδομένων Oracle. Το σύστημα αποτελείται από τον εξυπηρετητή που "σηκώνει" την ιστοσελίδα του διαγωνισμού για διαγωνιζόμενους και διαχειριστές και από τον εξυπηρετητή των αξιολογήσεων, ο οποίος αναλαμβάνει τη δημιουργία των δυνητικά επικίνδυνων εργασιών που τρέχουν σε ένα περιορισμένο περιβάλλον με τη χρήση ενός πρόσθετου που λέγεται spawner.

Η εφαρμογή αξιολόγησης εμπεριέχει διεπαφή μέσω της γραμμής εντολών για τη γρήγορη δημιουργία και αξιολόγηση προβλημάτων από τους διαχειριστές. Αυτή διατίθεται και ανεξάρτητα από το υπόλοιπο πρόγραμμα προς αντικατάσταση αντίστοιχων διαδικτυακών εργαλείων όπως είναι το Polygon ⁴. Η δημιουργία των προβλημάτων γίνεται με το ανέβασμα ενός συμπιεσμένου αρχείου το οποίο περιέχει μια XML περιγραφή και τα απαραίτητα αρχεία ελέγχου.

```
<?xml version="1.0" encoding="utf8"?>
<CATS version="1.4">
<Problem title="Titlos" lang="en" tlimit="2" mlimit="64"
  author="Admin" inputFile="titlos.in" outputFile="titlos.out">

<Keyword code="alg.geometry"/>

<ProblemStatement>
  Problem statement and description.
</ProblemStatement>

<InputFormat>
  Description of input format.
</InputFormat>
```

³ imcs.dvfu.ru/cats?lang=en

⁴ <https://polygon.codeforces.com/>

```

<OutputFormat>
  Description of output format.
</OutputFormat>

<ProblemConstraints>
  $1 \le m \le 100$, $-100 \le x_i \le 100$, 0 &lt; y_i \le 100$
</ProblemConstraints>

<Sample rank="1">
<SampleIn>
  3
  1 2 3
  4 5 6
  7 8 9
</SampleIn>
<SampleOut>
  6
  15
  24
</SampleOut>
</Sample>

<Solution name="sol" src="titlos_al.dpr" />
<Import guid="std.testlib.1.1" type="checker" />
<Checker name="ch" src="check.dpr" style="testlib"/>

<TestRange from="1" to="42">
  <In src="tests/%0n"/>
  <Out use="sol"/>
</TestRange>

</Problem>
</CATS>

```

Listing 2.1: Παράδειγμα xml προβλήματος CATS. Το CATS δημιουργεί τη σελίδα του προβλήματος και τα χαρακτηριστικά του σύμφωνα με το συγκεκριμένο αρχείο ενώ διαθέτει δυνατότητα παρουσίασης LaTeX τύπων όπως αυτός στο ProblemConstraints.

Οι γλώσσες προγραμματισμού που υποστηρίζονται είναι οι C, C++, delphi, VB, Java, C#, Perl, Python, Ruby, PHP, Erlang, Javascript και SQL.

2.3.2 Εγκατάσταση και Χρήση

Η εγκατάσταση του CATS είναι αρκετά εύκολη, καθώς έχει πολύ λίγες εξαρτήσεις και διαθέτει έτοιμα scripts για το deployment και την αρχική παραμετροποίηση. Αφού τρέξουν αυτά και ρυθμιστεί η σύνδεση με τη βάση, το σύστημα είναι έτοιμο να χρησιμοποιηθεί.

Η διαδικασία εμφάνισης των ορισμών των προβλημάτων, υποβολής των λύσεων και διαχείρισης των διαγωνισμών από τους διαχειριστές είναι παρόμοια με τα προηγούμενα συστήματα και παρουσιάζεται με φωτογραφίες παρακάτω. Άξια αναφοράς είναι η δυνατότητα του CATS για διαχείριση διαγωνισμών, αξιολόγηση και παρουσίαση των αποτελεσμάτων απ' ευθείας μέσω της γραμμής εντολών μέσω ενός API που προσφέρεται.

0 - a, 1 - ab

Author: algolist
Input file: input.txt
Output file: output.txt

Time Limit: 4 sec
Memory Limit: 64 MB

Condition

Given the string s , consisting of N characters 0 or 1, as well as the string t , consisting of M characters a or b, Above the string s are allowed to perform the following actions:

- substitute the symbol 0 for a string of one or more characters a
- replace symbol 1 by a string of one or more characters a, or a string of one or more characters b

It is required to determine whether it is possible to convert the string s in line t with the help of these actions.

Input File Format

The first line of the input file contains the numbers N M .

The second line of the input file contains the string s .

The third line of the input file contains the string t .

Output file format

The output file should contain a single character Y or of N.

Restrictions

$1 \leq N, M \leq 10000$

Sample test cases

No.	Input file (input.txt)	Output file (output.txt)
1	3 3 101 bab	Y
2	4 3 1001 bab	N

Σχήμα 2.8: Η σελίδα του προβλήματος στο CATS. Δεν περιέχει τίποτα παραπάνω από τη διατύπωση, αφού η υποβολή γίνεται από τη σελίδα του διαγωνισμού. Το σύστημα του CATS επιτρέπει και τη χρήση LaTeX όπως φαίνεται στο Restrictions.

The screenshot shows the CATS Archive website interface. At the top, there is a navigation bar with links for 'logout', 'contests', 'problems', 'console', 'modules', 'compilers', and 'keywords'. Below this is a search bar and a 'show' dropdown menu. The main content area displays a table of problems with columns for 'Problem', 'Contest', and 'OK / WA / TL'. The first problem, '0 - a, 1 - ab', is highlighted. Below the table, there is a 'Solution file:' section with a 'Choose File' button and a 'No file chosen' message. There is also a 'Dev. environment:' dropdown menu set to 'by file extension' and a 'Solution text:' text area. A 'submit' button is located at the bottom left of the page.

Σχήμα 2.9: Η σελίδα του διαγωνισμού που περιέχει όλα τα προβλήματα του. Επιλέγοντας ένα πρόβλημα μπορούμε να υποβάλουμε τη λύση στο κάτω μέρος της σελίδας.

Κεφάλαιο 3

Περιγραφή Grader - Kewii

Το σύστημα αποτελείται από το το σύστημα αξιολόγησης Kewii, το backend της εφαρμογής μας που λειτουργεί ως δαίμονας (πρόγραμμα που τρέχει συνεχώς) στον εξυπηρετητή με σκοπό την μεταγλώττιση και αξιολόγηση των υποβολών που λαμβάνει, και από τη διαδικτυακή εφαρμογή Grader, η οποία αναλαμβάνει την αλληλεπίδραση με χρήστες και διαχειριστές, την (έμμεση) επικοινωνία με τον Kewii και τη συνολική υλοποίηση της λογικής του συστήματος όσον αφορά στον τρόπο λειτουργίας των επιμέρους στοιχείων του και τον τρόπο αξιολόγησης. Ακολουθεί μια πρώτη ανάλυση των δομικών στοιχείων και των αλληλεπιδράσεων τους, πριν αναλυθούν Kewii και Grader.

3.1 Στοιχεία και έννοιες του συστήματος

Στην ενότητα αυτή θα περιγραφούν συνοπτικά τα επιμέρους στοιχεία του συστήματος μας. Η έννοια και η λειτουργία τους είναι απαραίτητη για να γίνει αντιληπτός ο διαχωρισμός μεταξύ Kewii και Grader, δηλαδή backend και frontend.

3.1.1 Προβλήματα





Προβλήματα είναι οι ανεξάρτητες ασκήσεις που τίθενται προς επίλυση στους διαγωνιζόμενους/χρήστες. Κάθε πρόβλημα έχει χρονικά όρια εκτέλεσης και όρια μνήμης, όπως και ιδιότητες για τον τρόπο εκτέλεσης και αξιολόγησης. Η αξιολόγηση του γίνεται πάνω σε συγκεκριμένα αρχεία εισόδου και εξόδου, τα αρχεία ελέγχου. Προαιρετικά, ένα πρόβλημα μπορεί, επιπλέον, να διαθέτει ένα πρόγραμμα αξιολόγησης των υποβολών.

Τα προβλήματα έχουν νόημα αποκλειστικά μέσα σε διαγωνισμούς, καθώς η τελική αξιολόγηση αφορά τους διαγωνισμούς και το σύνολο των προβλημάτων που περιέχουν. Η αρχική σχεδίαση του συστήματος είχε ως στόχο τη χρήση του σε διαγωνισμούς πληροφορικής, οπότε τα προβλήματα είναι συνδεδεμένα στενά τόσο με τον διαγωνισμό που ανήκουν όσο και με τις υποβολές που έχουν γίνει σε αυτά. Κάθε πρόβλημα μπορεί να ανήκει μόνο σε ένα διαγωνισμό.

Σε επόμενο κεφάλαιο θα διερευνηθεί και θα περιγραφεί η υλοποίηση ενός εναλλακτικού σχεδιασμού που επιτρέπει την χρήση των προβλημάτων σε πολλαπλούς διαγωνισμούς και την αλλαγή της σύνδεσης των υποβολών από το πρόβλημα προς τους διαγωνισμούς.

3.1.2 Διαγωνισμοί

Οι διαγωνισμοί αντιστοιχούν σε διοργανώσεις πληροφορικής, εξετάσεις ή σειρές ασκήσεων. Εμπεριέχουν προβλήματα και είναι ενεργοί/ορατοί σε ένα χρονικό διάστημα όπου οι διαγωνιζόμενοι μπορούν να υποβάλλουν τις λύσεις τους. Μόλις ολοκληρωθεί η διεξαγωγή τους, ο διαχειριστής μπορεί να εκκινήσει την τελική αξιολόγηση, κατά την οποία βαθμολογούνται οι ενεργές υποβολές των διαγωνιζόμενων σε όλα τα προβλήματα του διαγωνισμού. Οι διαγωνισμοί μπορούν να είναι ορατοί σε όλους ή μόνο σε επιλεγμένους χρήστες.

Τύπος εκτέλεσης	Λειτουργία
Πορτοκαλί 	Το αρχείο ελέγχου χρησιμοποιείται σε κάθε υποβολή για το συγκεκριμένο πρόβλημα αλλά δεν είναι ορατό στους χρήστες.
Κίτρινο 	Το αρχείο ελέγχου χρησιμοποιείται σε κάθε υποβολή για το συγκεκριμένο πρόβλημα και είναι ορατό στους χρήστες.
Πράσινο 	Το αρχείο ελέγχου χρησιμοποιείται μόνο κατά την τελική αξιολόγηση.
Μωβ 	Το αρχείο ελέγχου δεν χρησιμοποιείται.

Πίνακας 3.1: Τύποι εκτέλεσης αρχείων ελέγχου

3.1.3 Αρχεία Ελέγχου

Τα αρχεία ελέγχου είναι ζευγάρια αρχείων εισόδου και εξόδου και ανήκουν σε προβλήματα. Μια σωστή υποβολή/λύση θα πρέπει για κάθε αρχείο εισόδου που δέχεται, να παράγει έξοδο ίδια με αυτή του αντίστοιχου αρχείου εξόδου. Κάθε αρχείο ελέγχου χαρακτηρίζεται από τους πόντους που αξίζει και από τον τύπο εκτέλεσης του, που σχετίζεται με το πότε εκτελείται και αν είναι ορατό στους χρήστες. Οι τύποι εκτέλεσης αντιστοιχούν σε χρώματα (tags) και παρουσιάζονται στον Πίνακα 3.1.

Σε επόμενο κεφάλαιο θα δημιουργηθεί ένας νέος τύπος εκτέλεσης. Επιπλέον, θα αλλάξει ριζικά ο τρόπος αξιολόγησης καθώς θα εισαχθεί η έννοια των ομάδων αρχείων ελέγχου και τα αρχεία θα αξιολογούνται μόνο μέσα σε αυτές.

3.1.4 Χρήστες/Διαγωνιζόμενοι

Οι χρήστες ή διαγωνιζόμενοι είναι τα άτομα που αλληλεπιδρούν με το σύστημα με σκοπό την υποβολή λύσεων στους διαγωνισμούς που συμμετέχουν. Οι διαχειριστές αποτελούν κι αυτοί χρήστες, με τη διαφορά ότι έχουν δικαίωμα πρόσβασης στις σελίδες διαχείρισης διαγωνισμών και προβλημάτων. Οι τελευταίοι έχουν επίσης δικαίωμα να εισάγουν νέους χρήστες από ένα αρχείο, να ορίσουν την ορατότητα των διαγωνισμών για συγκεκριμένους διαγωνιζόμενους και να απαντήσουν σε μηνύματα αυτών.

3.1.5 Υποβολές

Η υποβολή μιας λύσης προϋποθέτει το "ανέβασμα" του κώδικα του διαγωνιζόμενου για να γίνει η μεταγλώττιση του στον εξυπηρετητή, να εκτελεστεί από τον Kewii για κάθε αρχείο ελέγχου που αντιστοιχεί στο είδος της υποβολής και να αξιολογηθεί η ορθότητα της εξόδου. Οι υποβολές χωρίζονται σε δύο κύριες κατηγορίες, την κανονική και την τελική. Ουσιαστικά, όλες οι υποβολές των διαγωνιζόμενων είναι κανονικές και εφόσον υπάρχει, για έναν διαγωνιζόμενο, τουλάχιστον μία υποβολή που περνάει όλα τα αρχεία ελέγχου στα οποία αξιολογήθηκε, αυτή θεωρείται ενεργή για αυτόν. Όταν ολοκληρωθεί η περίοδος ανοιχτών υποβολών ενός διαγωνισμού, όλες οι ενεργές υποβολές για κάθε πρόβλημα του διαγωνισμού επαναυποβάλλονται στον Kewii για την τελική αξιολόγηση τους.

3.1.6 Προγράμματα Αξιολόγησης

Οι υποβολές σε κάθε πρόβλημα αξιολογούνται συνήθως με σύγκριση της εξόδου του υποβληθέντος προγράμματος με αυτήν του αρχείου ελέγχου για την αντίστοιχη είσοδο. Εξαίρεση σε αυτό αποτελεί η δυνατότητα που παρέχει ο Kewii για ορισμό ενός ειδικού προγράμματος αξιολόγησης στο πρόβλημα το οποίο αναλαμβάνει να συγκρίνει την αναμενόμενη έξοδο με αυτήν που παρήγαγε το υποβληθέν πρόγραμμα και να βαθμολογεί στην κλίμακα $[0, 1]$. Το πρόγραμμα αξιολόγησης είναι και αυτό αποθηκευμένο στον εξυπηρετητή και μεταγλωττίζεται και εκτελείται κατά τη διαδικασία αξιολόγησης.

3.2 Σύστημα αξιολόγησης Kewii

Ο Kewii είναι μια εφαρμογή γραμμένη στη γλώσσα C και τρέχει στον εξυπηρετητή ελέγχοντας διαρκώς για την ύπαρξη νέων υποβολών. Για κάθε νέα υποβολή που εντοπίζει, βρίσκει τον πηγαίο κώδικα που έχει αποθηκευτεί από τον Grader μαζί με συγκεκριμένα μεταδεδομένα, μεταγλωττίζει τον κώδικα δημιουργώντας το εκτελέσιμο αρχείο και το εκτελεί χρησιμοποιώντας τα απαραίτητα μέτρα ασφαλείας ώστε να συγκρίνει την έξοδο για κάθε αρχείο ελέγχου με την σωστή. Μόλις τελειώσει η εκτέλεση ή ξεπεραστούν τα όρια της, ενημερώνει τη βάση δεδομένων, με χρήση ενός PHP script, για την έκβαση της εκτέλεσης και ειδοποιεί το Grader καλώντας ένα μοναδικό για κάθε υποβολή σύνδεσμο (callback) ώστε αυτός να αναλάβει την ανάλυση των αποτελεσμάτων.

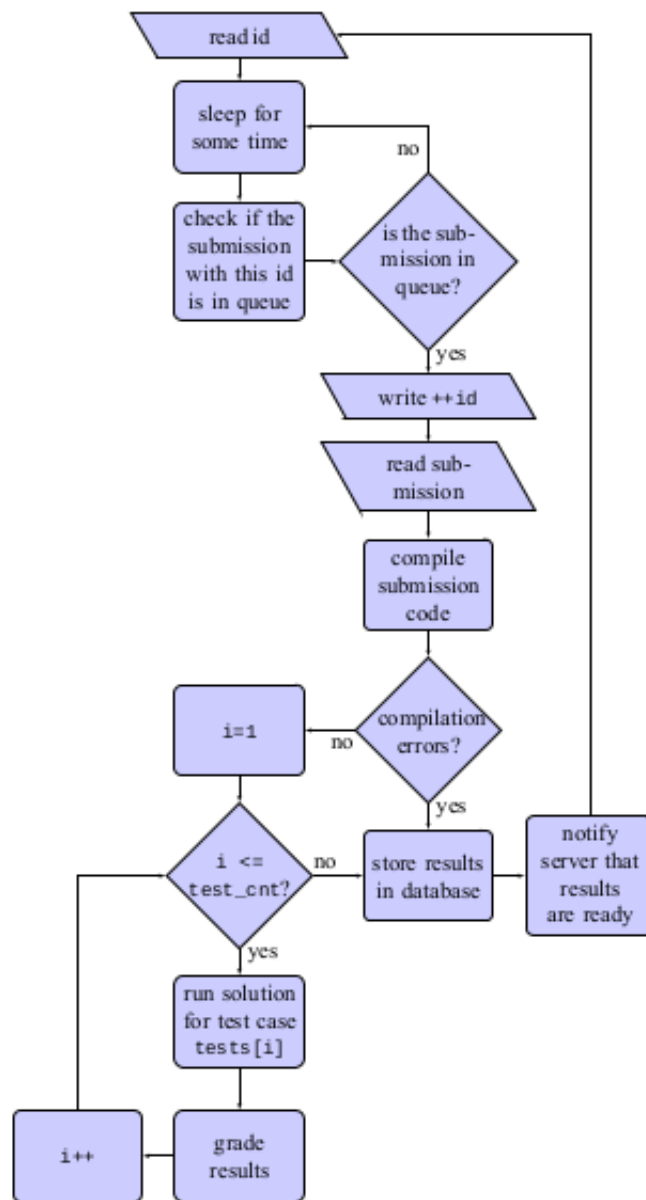
Κάθε υποβολή έχει έναν μοναδικό κωδικό, ο οποίος χρησιμοποιείται για την αλληλεπίδραση με το Grader κατά το callback, όπως και έναν αύξοντα αριθμό που χρησιμεύει στον Kewii για την διατήρηση της κατάστασης των εκτελέσεων. Τα μεταδεδομένα που εμπεριέχονται σε κάθε υποβολή και είναι απαραίτητα για την αξιολόγηση της είναι τα παρακάτω:

- Όνομα του προβλήματος
- Γλώσσα υποβολής
- Αρχεία ελέγχου που θα χρησιμοποιηθούν
- Όριο μνήμης και χρόνου εκτέλεσης
- Είδος εκτέλεσης
- Τρόπος αξιολόγησης

Κάθε πρόβλημα πρέπει να έχει μοναδικό όνομα και είναι απαραίτητο ώστε να επιλεγθούν τα σωστά αρχεία ελέγχου. Οι γλώσσες υποβολής που υποστηρίζονται είναι: C, C++, Pascal, Pazcal, F#, OCaml, SML, Java, Fortran και Haskell. Το είδος εκτέλεσης μπορεί να είναι batch ή interactive/partial. Στην πρώτη περίπτωση τα προγράμματα που υποβάλλονται αποτελούν ανεξάρτητες λύσεις ενώ στην δεύτερη ο κωδικός αλληλεπιδρά με συγκεκριμένες βιβλιοθήκες ή εντάσσεται σε έναν κοινό κορμό που έχει τεθεί για το συγκεκριμένο πρόβλημα. Ο τρόπος αξιολόγησης μπορεί να είναι κανονικός ή ειδικός. Στον κανονικό ελέγχεται η ομοιότητα της εξόδου του προγράμματος της υποβολής με την ορθή απάντηση, ενώ στον ειδική αξιολόγηση γίνεται χρήση ειδικού προγράμματος αξιολόγησης για την βαθμολόγηση της λύσης επιτρέποντας έτσι την ύπαρξη προβλημάτων βελτιστοποίησης.

Οι υποβολές που στέλνονται στον Kewii αποθηκεύονται σε μια ουρά σύμφωνα με τον αύξοντα αριθμό που παίρνουν και ο Kewii γνωρίζει κάθε στιγμή ποιος θα είναι ο επόμενος αριθμός υποβολής που θα αξιολογήσει. Το πρόγραμμα "κοιμάται" έως ότου βρει μια νέα υποβολή ελέγχοντας για αρχεία με τον συγκεκριμένο αριθμό. Επίσης, αν η διαδικασία της αξιολόγησης διακοπεί ενδιάμεσα, μπορεί να συνεχίσει από το σημείο που σταμάτησε. Η ροή λειτουργίας του παρουσιάζεται στο σχήμα 3.1.

Γενικά, η λειτουργία του Kewii περιορίζεται στο να δέχεται υποβολές για προβλήματα με συγκεκριμένα αρχεία ελέγχου. Αξιολογεί κάθε υποβολή με 0 - λάθος και 1 - σωστό για κάθε αρχείο ή αναθέτει την αξιολόγηση στο ειδικό πρόγραμμα που θα επιστρέψει μια τιμή ανάμεσα στα 0 και 1. Τότε, η "ευθύνη" του τελειώνει, δηλαδή δεν εμπλέκεται στη συνολική αξιολόγηση των διαγωνισμών και των προβλημάτων, τον ορισμό υποβολών ως ενεργών, ή με το ποια αρχεία ελέγχου θα επιλεγούν για κάθε υποβολή. Η αποστασιοποίηση του από τα παραπάνω, το γεγονός δηλαδή ότι δρα ως ένα μαύρο κουτί για την εκτέλεση των υποβολών, είναι πολύ σημαντικό διότι θα μας επιτρέψουν να επανασχεδιάσουμε τον αλγόριθμο της υποβολής, δημιουργώντας ομάδες αρχείων ελέγχου, όπως και άλλα κομμάτια του Grader χωρίς να χρειαστεί να τροποποιήσουμε τον Kewii.



Σχήμα 3.1: Το διάγραμμα ροής που ακολουθείται σε κάθε νέα υποβολή στον Kewii, όπως παρουσιάζεται στο [Τσ15].

3.3 Διαδικτυακή εφαρμογή Grader

Ο Grader είναι μια web εφαρμογή υλοποιημένη σε PHP και HTML/CSS/JS, η οποία αποτελεί το frontend και μέρος του backend του συστήματος μας. Ως frontend εφαρμογή αναλαμβάνει να παρουσιάσει σε διαγωνιζόμενους και διαχειριστές τις σελίδες διαχείρισης, υποβολών και αποτελεσμάτων, ενώ ως backend αναλαμβάνει την υλοποίηση της λογικής του συστήματος, των διάφορων αλγορίθμων και ροών και την επικοινωνία με τη βάση δεδομένων και τον Kewii.

Τα συνήθη σενάρια χρήσης της εφαρμογής αναφέρονται παρακάτω.

3.3.1 Εμφάνιση διαθέσιμων διαγωνισμών και προβλημάτων

Τόσο οι διαγωνιζόμενοι, όσο και οι διαχειριστές έχουν τη δυνατότητα να βλέπουν τους ενεργούς διαγωνισμούς και τα προβλήματα που περιέχονται σε αυτούς. Η σελίδα κάθε διαγωνισμού ενημερώνει το χρήστη για τη διάρκεια του καθώς και για το ποια προβλήματα περιέχει. Μεταβαίνοντας σε κάποιο πρόβλημα, ο χρήστης βλέπει την ακριβή περιγραφή του και έχει τη δυνατότητα να υποβάλλει τη λύση του.

3.3.2 Εμφάνιση υποβολών προβλήματος και πηγαίου κώδικα

Κάθε χρήστης μπορεί μέσα από τη σελίδα του διαγωνισμού να περάσει στις υποβολές για καθένα από τα προβλήματα του. Σε αυτή τη σελίδα υπάρχουν όλες οι υποβολές που έχει κάνει για αυτό το πρόβλημα, ενώ υπάρχει χρωματική διάκριση ανάμεσα σε σωστές και λανθασμένες υποβολές, καθώς και για την ενεργή υποβολή. Ως ενεργή υποβολή τίθεται η τελευταία σωστή, αν και υπάρχει η δυνατότητα στην ίδια σελίδα να επιλεγεί μια διαφορετική σωστή υποβολή ως ενεργή. Επιλέγοντας μια υποβολή, ο χρήστης μεταφέρεται στη σελίδα λεπτομερειών της υποβολής του, όπου εμφανίζονται και τα ακριβή αποτελέσματα της λύσης του για κάθε αρχείο ελέγχου. Για τα αρχεία ελέγχου που το επιτρέπουν, υπάρχει η σελίδα προεπισκόπησης. Παρόμοια δυνατότητα δίνεται και για τον πηγαίο κώδικα της υποβολής, αν ο χρήστης θέλει να τον δει στην εφαρμογή.

3.3.3 Δημιουργία και διαχείριση προβλημάτων και διαγωνισμών

Η δημιουργία και η διαχείριση διαγωνισμών γίνεται από τη σελίδα της διαχείρισης που δεν είναι προσβάσιμη στους κανονικούς χρήστες. Σε αυτή τη σελίδα, ο διαχειριστής μπορεί να δει όλους τους διαγωνισμούς και τα προβλήματα που έχουν δημιουργηθεί. Οι διαγωνισμοί παρουσιάζονται ταξινομημένοι κατά χρονολογική σειρά δημιουργίας μαζί με τα προβλήματα που περιέχονται στον καθένα. Δίνονται επιλογές για επεξεργασία προβλημάτων και διαγωνισμών, καθώς και δημιουργία νέων. Αφού δημιουργηθεί ένα πρόβλημα, αρχικά δεν ανήκει σε κάποιον διαγωνισμό έως ότου επιλεγεί κάποιος από το πλευρικό μενού μετακίνησης του προβλήματος. Διπλά σε κάθε πρόβλημα υπάρχουν, επιπλέον, στοιχεία για τα αρχεία ελέγχου που διαθέτει και τις συνολικές λύσεις που έχουν υποβληθεί.

Εξαιτίας της άμεσης σύνδεσης προβλημάτων και υποβολών, τα προβλήματα διατηρούν τις υποβολές που έχουν γίνει σε αυτά κατά τους σε νέους διαγωνισμούς. Επιπροσθέτως, όπως είναι προφανές, ο διαγωνισμός στον οποίο άνηκε αρχικά το πρόβλημα φαίνεται κενός και ουσιαστικά χάνει και την ιστορικότητα του. Ο συγκεκριμένος τρόπος λειτουργίας αποτελεί απόρροια της αρχικής σχεδίασης του Grader για διοργανώσεις και όχι για ακαδημαϊκούς σκοπούς, με αποτέλεσμα να μην έχει προβλεφθεί η δημιουργία διαγωνισμών-σειρών ασκήσεων με επαναχρησιμοποίηση προβλημάτων.

Μια ακόμα σελίδα που είναι προσβάσιμη από την κεντρική της διαχείρισης προβλημάτων και διαγωνισμών είναι αυτή των αρχείων ελέγχου του προβλήματος. Σε αυτήν μπορεί ο διαχειριστής να ανεβάσει καινούρια αρχεία ελέγχου, καθώς και να αλλάξει τον τύπο εκτέλεσης και την βαθμολογία τους. Η συγκεκριμένη σελίδα θα μας απασχολήσει αρκετά στα επόμενα κεφάλαια, δεδομένου ότι θα τροποποιηθεί τόσο για τις ομάδες αρχείων ελέγχου όσο και για την προσθήκη μαζικής δημιουργίας αρχείων και ομάδων.

3.3.4 Ενεργοποίηση διαγωνισμού και τελική αξιολόγηση


Ένα πολύ συνηθισμένο σενάριο χρήσης για έναν διαχειριστή είναι αυτό κατά το οποίο, αφού έχει δημιουργήσει ένα νέο διαγωνισμό και τα προβλήματα του, πρέπει να το δημοσιοποιήσει στους χρήστες. Μέσα από τη σελίδα διαχείρισης των διαγωνισμών, έχει τη δυνατότητα αρχικά να επιλέξει τους χρήστες που επιτρέπεται να συμμετέχουν στο διαγωνισμό. Έπειτα, έχοντας επιλέξει την επιθυμητή ημερομηνία έναρξης και λήξης του διαγωνισμού κατά τη δημιουργία του, μπορεί να επιλέξει την ενεργοποίηση αυτού με το αντίστοιχο πλήκτρο στη σελίδα διαχείρισης. Τότε ο διαγωνισμός και τα

προβλήματα του γίνονται ορατά στους επιλεγμένους χρήστες και οι τελευταίοι μπορούν να ξεκινήσουν να κάνουν υποβολές.

Όταν λήξει ή απενεργοποιηθεί ο διαγωνισμός, ο διαχειριστής μπορεί να εκκινήσει την τελική αξιολόγηση, κατά την οποία επιλέγονται όλες οι ενεργές υποβολές για κάθε πρόβλημα του διαγωνισμού και αυτές αποστέλλονται στον Kewii για να αξιολογηθούν εκ νέου συμπεριλαμβάνοντας αυτή τη φορά τα αρχεία ελέγχου που είναι αποκλειστικά για την τελική αξιολόγηση (πράσινα). Μόλις ολοκληρωθεί η διαδικασία, εμφανίζονται στη σελίδα τα τελικά αποτελέσματα, τα οποία μπορούν να δημοσιοποιηθούν σαν δημοσιεύσεις στην αρχική σελίδα του Grader. Η βαθμολογία υπολογίζεται από το άθροισμα των ενεργών υποβολών του κάθε διαγωνιζόμενου στα προβλήματα που περιέχει ο διαγωνισμός. Η βαθμολογία του προβλήματος είναι το άθροισμα των πόντων των αρχείων ελέγχου της τελικής αξιολόγησης.

Κεφάλαιο 4

Προσθήκη Ομάδων Αρχείων Ελέγχου

Η μεγαλύτερη σε πολυπλοκότητα αλλαγή λειτουργικότητας του Grader ήταν η τροποποίηση του τρόπου αξιολόγησης των υποβολών με την εισαγωγή ομάδων αρχείων ελέγχου (testcase groups) για ομαδοποίηση των τελευταίων, καθώς και η δημιουργία ενός νέου τύπου εκτέλεσης των αρχείων. Η υλοποίηση του νέου τύπου εκτέλεσης, που θα αντιστοιχεί στο μπλε χρώμα (blue tag ) κατά την αντιστοίχιση που παρουσιάστηκε στον πίνακα 3.1, ήταν μια χρήσιμη εισαγωγή στη λειτουργία και στον κώδικα του Grader αλλά και τη βάση δεδομένων. Στο παρόν κεφάλαιο θα περιγραφεί πρώτα η μικρή αυτή προσθήκη και έπειτα η λογική και η υλοποίηση της προσθήκης των testcase groups.

4.1 Προσθήκη blue tag για μη απαραίτητα αρχεία ελέγχου

4.1.1 Κίνητρο

Ένα αρχείο ελέγχου χαρακτηρισμένο με blue tag ελέγχεται κανονικά σε κάθε αξιολόγηση αλλά το αποτέλεσμα της εκτέλεσης του δεν επηρεάζει την ορθότητα της υποβολής. Όπως τα "κίτρινα" αρχεία ελέγχου, τα "μπλε" διατηρούν κρυφή την είσοδο τους από τους διαγωνιζόμενους. Ο συγκεκριμένος τύπος εκτέλεσης είχε υλοποιηθεί πρώτα (πριν την παρούσα εργασία), στο branch του Grader που χρησιμοποιεί το hellenico.gr, το οποίο όμως διαφέρει αρκετά από αυτό του softlab, γεγονός που δεν επέτρεπε το απλό merge του σχετικού κώδικα.

Η ύπαρξη αρχείων ελέγχου που εξετάζονται στις κανονικές υποβολές αλλά δε κρίνουν την έκβαση τους προσφέρει το κύριο πλεονέκτημα ότι επιτρέπει την εισαγωγή δυσκολότερων αρχείων ελέγχου εκτός τελικής αξιολόγησης. Ας θυμηθούμε τη λειτουργία του Grader όσον αφορά στις αρχικές υποβολές και την τελική αξιολόγηση: Κατά τη διεξαγωγή του διαγωνισμού, κάθε πρόβλημα είναι ανοιχτό σε υποβολές. Μια υποβολή θεωρείται ορθή μόνο αν όλα τα αρχεία ελέγχου που έτρεξαν είναι σωστά. Για να πάρει βαθμολογία για το πρόβλημα, ο διαγωνιζόμενος πρέπει να έχει τουλάχιστον μια ορθή υποβολή σε αυτό (την ενεργή).

Αυτό δημιουργεί το πρόβλημα ότι αρχεία ελέγχου με μη εμφανείς δυσκολίες του αλγόριθμου (corner cases) ή αρχεία με μεγάλο μέγεθος εισόδου, αποτελούν ρίσκο όσον αφορά τον χαρακτηρισμό τους ως "κίτρινα" ή "πορτοκαλί", δηλαδή αρχεία ελέγχου που τρέχουν σε κανονικές και τελικές υποβολές. Όσοι διαγωνιζόμενοι δεν καταφέρουν να υποβάλουν λύση που να αξιολογηθεί σωστά σε όλα τα αρχεία ελέγχου δεν καταφέρνουν να βαθμολογηθούν καθόλου, πιθανόν άδικα. Ως φυσικό επακόλουθο, τέτοιου τύπου αρχεία ελέγχου μπορούν να χαρακτηριστούν μόνο ως πράσινα, δηλαδή να χρησιμοποιούνται αποκλειστικά στην τελική αξιολόγηση.

Τα μπλε αρχεία ελέγχου προσφέρουν μια λύση στο πρόβλημα, καθώς επιτρέπουν ουσιαστικά μια υποβολή να χαρακτηριστεί ορθή/ενεργή χωρίς να έχει "περάσει" όλα τα αρχεία ελέγχου. Ακόμα κι αν ο διαγωνιζόμενος δεν καταφέρει να βελτιώσει την υλοποίηση του έτσι ώστε να ικανοποιεί όλα τα αρχεία ελέγχου, θα διαθέτει μια ενεργή υποβολή και συνεπώς θα βαθμολογηθεί. Παράλληλα, θα γνωρίζει ότι έχει αποτύχει σε τουλάχιστον μία περίπτωση και θα έχει κίνητρο να συνεχίσει τις υποβολές ώστε να μη χάσει τη δυνατότητα να πάρει πλήρη βαθμολογία.

Αρχική Διαγωνισμοί Διαχείριση Επικοινωνία

Όρα Συστήματος
 Τετάρτη, 14 Μαρτίου 2018 - 19:04:44

trena - Αρχεία ελέγχου

Καλωσήρθες admin
 Όνομα: admin
 Επίθετο: admin
 Έξοδος

Αρχεία Ελέγχου

1ο αρχείο ελέγχου		3 βαθμοί
2ο αρχείο ελέγχου		1 βαθμοί
3ο αρχείο ελέγχου		1 βαθμοί

Συνολική βαθμολογία προβλήματος: 5 βαθμοί

- = Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου προκειμένου να γίνουν δεκτές.
- = Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου προκειμένου να γίνουν δεκτές και οι χρήστες θα μπορούν να δουν το συγκεκριμένο αρχείο ελέγχου.
- = Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου την ώρα της υποβολής και ο χρήστης θα μπορεί να δει το αποτέλεσμα, αλλά όχι το αρχείο ελέγχου. Το αποτέλεσμα της εκτέλεσης δεν επηρεάζει το αν θα γίνει δεκτή η υποβολή (για προβλήματα με partial feedback).
- = Το αρχείο ελέγχου θα χρησιμοποιηθεί μόνο στην τελική αξιολόγηση.
- = Το αρχείο ελέγχου δεν θα χρησιμοποιηθεί για την αξιολόγηση.

Νέο αρχείο ελέγχου

Αρχείο εισόδου: No file selected.

Αρχείο εξόδου: No file selected.

Σχήμα 4.1: Η σελίδα διαχείρισης αρχείων ελέγχου μετά την προσθήκη του blue tag.

4.1.2 Υλοποίηση

Η υλοποίηση δεν είχε ιδιαίτερες δυσκολίες αλλά απαιτούσε την κατανόηση της αρχιτεκτονικής του Grader και του Kewii. Αλλαγές απαιτούνταν τόσο στο frontend κομμάτι του Grader, όσο και στον κώδικα κατά την υποβολή και κατά το callback, που τρέχει μόλις ολοκληρωθεί η αξιολόγηση μιας υποβολής. Ο Kewii δε χρειάστηκε τροποποιήσεις, διότι όπως έχει περιγραφεί σε προηγούμενο κεφάλαιο, σε κάθε υποβολή λαμβάνει απλά τη λίστα με τα αρχεία ελέγχου που πρέπει να χρησιμοποιήσει και επιστρέφει την έκβαση τους. Η βάση δεδομένων επίσης δεν υποβλήθηκε σε αλλαγές, καθώς η μόνη αλλαγή που την επηρεάζει είναι η προσθήκη μιας πιθανής τιμής στο πεδίο του τύπου εκτέλεσης του πίνακα των αρχείων ελέγχου.

Οι αλλαγές που αφορούν στο frontend κομμάτι έγιναν κυρίως στη σελίδα της διαχείρισης αρχείων ελέγχων. Όπως φαίνεται στο σχήμα 4.1, δίπλα σε κάθε αρχείο ελέγχου είναι τα χρωματικά tags και χάρη στη CSS διακρίνεται το επιλεγμένο. Για την προσθήκη του blue tag, χρησιμοποιήθηκε η εικόνα από το hellenico, και αυτή προστέθηκε μετά το κίτρινο tag. Αντίστοιχα, προστέθηκε η περιγραφή του συγκεκριμένου tag και τροποποιήθηκε ο κώδικας που διαχειρίζεται το πάτημα του tag και τροποποιεί τη βάση.

Στον κώδικα που τρέχει κατά την υποβολή, επιλέγονται τα αρχεία ελέγχου που θα εκτελεστούν και γράφονται σε ένα αρχείο στον εξυπηρετητή ώστε να μπου στην ουρά του Kewii. Εκεί προστέθηκε ο νέος τύπος εκτέλεσης σε αυτούς που στέλνονται σε όλες τις υποβολές. Αντίστοιχα, στο callback script, το κομμάτι του Grader που τρέχει μόλις έχει ολοκληρωθεί η βασική αξιολόγηση μιας υποβολής από τον Kewii, υλοποιήθηκε η λογική των "μπλε" αρχείων ελέγχου, που ακόμα και να έχουν χαρακτηριστεί ως λανθασμένα, δεν επηρεάζουν την έκβαση.

4.2 Ομάδες αρχείων ελέγχου (testcase groups)

Έχοντας προσθέσει το blue tag, τροποποιώντας πολλαπλά σημεία του Grader τόσο για την υποβολή όσο και για την αξιολόγηση, μπορούσαμε να προχωρήσουμε στην υλοποίηση της αρκετά πιο σύνθετης δυνατότητας, των testcase groups.

4.2.1 Κίνητρο

Πριν τις αλλαγές μας ο διαχειριστής, κατά την δημιουργία του προβλήματος, έπρεπε να δημιουργήσει αρχεία ελέγχου και να επιλέξει στο καθένα βαθμολογία και τύπο εκτέλεσης. Τα αρχεία ελέγχου δεν είχαν δυνατότητα για οποιαδήποτε κατηγοριοποίηση ή διαχωρισμό. Ο διαχειριστής, σε κάθε σημείο, όφειλε να διατηρεί την ισορροπία όσον αφορά στις βαθμολογίες των αρχείων ελέγχου και να θυμάται την αναλογία εύκολων και δύσκολων αρχείων.

Άλλο ένα πρόβλημα που υπήρχε ήταν η δυνατότητα για δημιουργία ειδικών λύσεων σε συγκεκριμένα προβλήματα έτσι ώστε να λαμβάνουν αξιοπρεπή βαθμολογία έχοντας υλοποιήσει περιορισμένο μέρος του αλγόριθμου που να αρκεί για την ορθότητα της υποβολής. Κάθε αρχείο ελέγχου που αξιολογείται ως σωστό αθροίζεται στην βαθμολογία της τελικής αξιολόγησης και μια τέτοια υποβολή μπορεί να επιτύχει καλύτερη βαθμολογία από ότι θα έπρεπε.

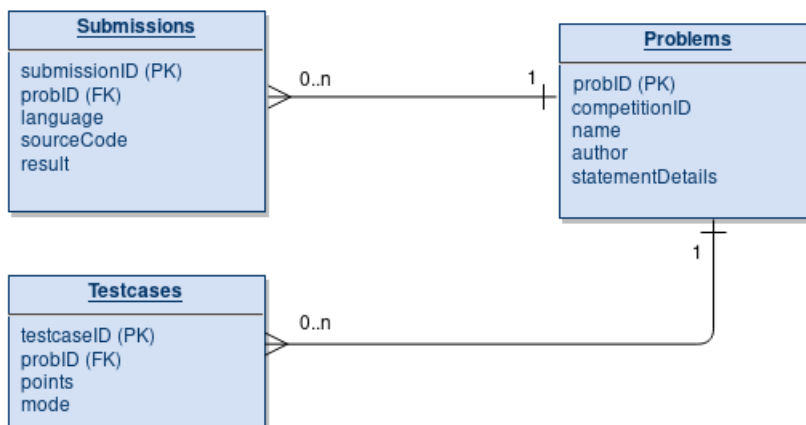
Η προσθήκη testcase groups θα επιτρέψει την ομαδοποίηση των αρχείων ελέγχου σε groups ανεξάρτητα στην αξιολόγηση, με δική τους βαθμολογία και με την ορθότητα τους να κρίνεται στην ορθότητα όλων των επιμέρους αρχείων ελέγχου. Έτσι, θα δοθεί μεγάλη ευελιξία στη δημιουργία διαφορετικών κατηγοριών αρχείων. Ένας χρήσιμος διαχωρισμός θα μπορεί να είναι ανάλογα με το μέγεθος της εισόδου, δηλαδή σε μικρά, μεσαία και μεγάλα. Με αυτό τον τρόπο ο διαχειριστής γλιτώνει πολύ κόπο πετυχαίνοντας την επιθυμητή αναλογία βαθμολογίας στις διάφορες λύσεις βάζοντας απλά τους βαθμούς που επιθυμεί στο group, ανεξάρτητα με το πόσα αρχεία ελέγχου ανήκουν σε κάθε group.

Επιπροσθέτως, λύνεται το πρόβλημα των προσαρμοσμένων λύσεων, αφού αν τα υπάρχοντα groups αποτελούνται από αρχεία πολλαπλών περιπτώσεων, οι λύσεις αυτού του τύπου δε θα καταφέρνουν να ικανοποιήσουν ταυτόχρονα τα αρχεία και δε θα βαθμολογούνται σε αυτά τα groups.

4.2.2 Προδιαγραφές

Για την καθοδήγηση της υλοποίησης τέθηκαν συγκεκριμένες προδιαγραφές όσον αφορά στις ομάδες αρχείων ελέγχου και τη λειτουργία τους.

- Οι ομάδες αρχείων ελέγχου, όπως περιγράφηκαν και παραπάνω, θα μπορούν να περιέχουν οποιοδήποτε υποσύνολο των διαθέσιμων αρχείων. Επιθυμούμε, για μεγαλύτερη ευελιξία, τα αρχεία ελέγχου να μπορούν να ανήκουν σε πολλαπλές ομάδες, με επιλογή για διαφορετικό τύπο εκτέλεσης σε κάθε ομάδα.
- Η αξιολόγηση θα γίνεται με βάση μόνο τις ομάδες, ανεξάρτητα με το πόσα αρχεία ελέγχου υπάρχουν. Μια υποβολή θα θεωρείται ορθή και θα μπορεί να θεθεί ως ενεργή αν ικανοποιεί τουλάχιστον μια ομάδα.
- Οι ομάδες θα μπορούν να αποκτήσουν τίτλο από το διαχειριστή και θα διαθέτουν πόντους. Τα παραπάνω καθιστούν πόντους και τύπους εκτέλεσης των αρχείων ελέγχου ανούσιους.
- Οι ομάδες που περιέχουν τουλάχιστον ένα αρχείο ελέγχου με τύπο εκτέλεσης μη τελικής αξιολόγησης, δηλαδή κίτρινο, πορτοκαλί ή μπλε, θα αξιολογούνται κατά τις κανονικές υποβολές και θα είναι εμφανείς στους διαγωνιζόμενους μαζί με τα αρχεία ελέγχου που αξιολογήθηκαν.



Σχήμα 4.2: Οι πίνακες και οι σχέσεις της βάσης πριν τις αλλαγές μας.

4.2.3 Υλοποίηση

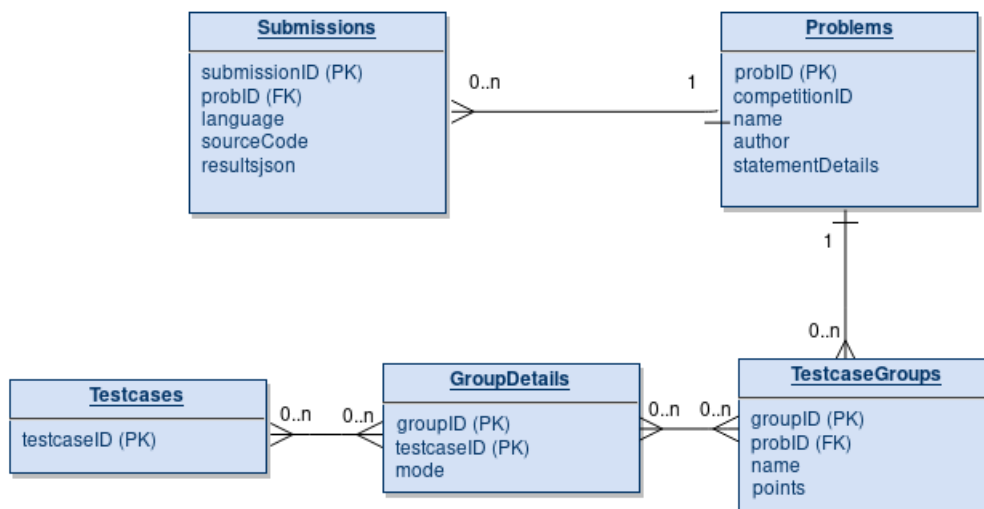
Με βάση τις προδιαγραφές που περιγράφηκαν παραπάνω διαμορφώνεται η εικόνα της υλοποίησης και των κρίσιμων μερών της.

Αρχικά, όσον αφορά στη βάση δεδομένων, θα χρειαστεί ένας νέος πίνακας για τα testcase groups και τα χαρακτηριστικά τους (όνομα, βαθμολογία). Δεδομένου ότι η σχέση αρχείων ελέγχου και groups είναι N-to-N, δηλαδή τόσο τα groups περιέχουν πολλαπλά αρχεία και τα αρχεία ανήκουν σε πολλαπλά groups, θα χρειαστεί και δεύτερος πίνακας, για την καταχώρηση όλων των αρχείων ελέγχου κάθε group.

Ο δεύτερος αυτός πίνακας θα ονομαστεί GroupDetails και θα περιέχει, εκτός από τα πεδία testcaseID και groupID, πεδίο για τον τύπο εκτέλεσης του αρχείου ελέγχου στη συγκεκριμένο group, επιτρέποντας ένα αρχείο π.χ. να ανήκει ως φανερό (κίτρινο) σε ένα group και ως τελικό (πράσινο) σε ένα άλλο. Οι σχετικοί πίνακες παρουσιάζονται στα σχήματα 4.2 και 4.3.

Από τη στιγμή που υπάρχει η πιθανότητα δύο groups να έχουν κοινά αρχεία ελέγχου, δημιουργείται το θέμα της πιθανής αχρείαστης αξιολόγησης των ίδιων αρχείων ελέγχου πολλές φορές. Στην υλοποίηση μας αυτό θα αποφευχθεί διατηρώντας την αρχιτεκτονική της εφαρμογής όσον αφορά στις αρμοδιότητες Kewii και Grader απaráλλαχτη. Αυτό σημαίνει ότι ο Kewii θα συνεχίσει να παίρνει απλά μια λίστα με τα αρχεία ελέγχου που πρέπει να εκτελέσει και θα επιστρέφει το αποτέλεσμα τους. Πρακτικά, δε θα "αντιληφθεί" την αλλαγή, καθώς ο Grader θα παραμείνει υπεύθυνος, κατά την υποβολή μιας λύσης, να φιλτράρει τα μοναδικά αρχεία ελέγχου που του χρειάζονται και κατά το callback να συνθέτει τα αποτελέσματα που θα λάβει κατάλληλα ώστε να αποφανθεί για την ορθότητα κάθε group.

Για τη διαχείριση και τη δημιουργία των testcase groups κρίθηκε αναγκαίο να δημιουργηθεί μια νέα σελίδα, η οποία εισάχθηκε στην ήδη υπάρχουσα διαχείριση αρχείων ελέγχου. Η τελευταία χρειάστηκε σημαντικές τροποποιήσεις, καθώς πλέον τα αρχεία ελέγχου δεν μπορούν να έχουν βαθμολογίες και τύπους εκτέλεσης (tags) έξω των ομάδων που ανήκουν. Τα χρωματικά tags παρέμειναν, δίνοντας τη δυνατότητα στο διαχειριστή να αλλάξει μαζικά τον τύπο ενός αρχείου ελέγχου σε όλες τις ομάδες που ανήκει. Κρίθηκε σημαντικό η πληροφορία για τις ομάδες, τις βαθμολογίες τους και το ποια αρχεία εμπεριέχει η κάθε μια, να εμφανίζεται στη συγκεκριμένη σελίδα χωρίς να είναι απαραίτητη η μετάβαση στη σελίδα της διαχείρισης ομάδας αρχείων ελέγχου. Οι δύο προαναφερθείσες σελίδες εμφανίζονται στα σχήματα 4.4 και 4.5.



Σχήμα 4.3: Δημιουργήθηκε ο νέος πίνακας GroupDetails, που είναι απαραίτητος για την αντιστοίχιση testcase groups με αρχεία ελέγχου. Επίσης, προστέθηκε το πεδίο resultsjson για την αποθήκευση των υπολογισμένων αποτελεσμάτων της υποβολής.

Η εισαγωγή των testcase groups στο Grader επηρέασε, ακόμα, μεγάλο πλήθος σελίδων και λειτουργιών συμπεριλαμβανομένων των παρακάτω:

- Αλλαγές στη σελίδα των λεπτομερειών υποβολής. Στη συγκεκριμένη σελίδα πλέον εμφανίζονται ομάδες αντί για αρχεία, οι οποίες περιέχουν τα αρχεία που αντιστοιχούν σε αυτές μέσα σε πλαίσια. Οι χρωματικές ενδείξεις επιτυχίας (πράσινο φόντο) και αποτυχίας (κόκκινο) έχουν παραμείνει. Το φόντο του τίτλου του group χρωματίζεται και αυτό ανάλογα με την ορθότητα του.
- Αλλαγές στη σελίδα παρουσίασης όλων των υποβολών και επιλογής της ενεργής. Εδώ έπρεπε να προστεθεί μία ένδειξη ορθότητας κάθε υποβολής σε σχέση με το πόσα σωστά groups έχει. Πρόκειται απλά για ένα κλάσμα των σωστών groups προς όλα όσα ελέγχθηκε. Μια ορθή υποβολή θα είναι πράσινη αλλά μπορεί π.χ. να έχει περάσει μόνο το ένα από τα τρία testcase groups, κάτι που δεν είναι εμφανές μέχρι να μεταβεί ο διαγωνιζόμενος στις λεπτομέρειες υποβολής. Σε αυτή την περίπτωση θα αναγράφεται 1/3.
- Στη σελίδα διαχείρισης προβλημάτων και διαγωνισμών άλλαξαν τα στατιστικά δίπλα σε κάθε πρόβλημα που έως τώρα παρουσίαζαν την αναλογία αρχείων ελέγχου ανά τύπο εκτέλεσης. Πλέον, εμφανίζονται μόνο δύο αριθμοί, ο συνολικός αριθμός αρχείων ελέγχου και ομάδων.

Άλλη μια σημαντική πτυχή της υλοποίησης αποτελεί ο τρόπος που θα γίνει η μετάβαση από τον προηγούμενο τρόπο αξιολόγησης στο νέο. Εκτός από τις πολλαπλές τροποποιήσεις στον κώδικα και στη βάση, επιβάλλεται να τροποποιηθούν όλα τα προβλήματα, παλιά και τρέχοντα. Ο λόγος είναι ότι στο νέο σύστημα η αξιολόγηση όλων των υποβολών βασίζεται στα testcase groups αντί για τα μεμονωμένα αρχεία ελέγχου. Ως αποτέλεσμα, κάθε υπάρχον πρόβλημα θα πρέπει να αποκτήσει groups τα οποία, ιδανικά, θα είναι ισοδύναμα με τα υπάρχοντα αρχεία ελέγχου.

Ο τρόπος να επιτευχθεί η ισοδυναμία προηγούμενης και νέας κατάστασης είναι η δημιουργία ενός group που να περιέχει μόνο τα αρχεία ελέγχου κανονικών υποβολών, δηλαδή μπλε, κίτρινα και πορτοκαλί, με βαθμολογία 0. Αυτό το group, πετυχαίνει ουσιαστικά, την προσομοίωση της προηγούμενης κατάστασης όπου οι υποβολές ελέγχονταν στα συγκεκριμένα αρχεία ελέγχου και εφόσον τα περνούσαν όλα, η λύση ήταν σωστή και γινόταν ενεργή. Αυτό το group όμως δεν αρκεί, καθώς χρειάζονται

trena - Αρχεία ελέγχου

Αρχεία Ελέγχου

1ο αρχείο ελέγχου

2ο αρχείο ελέγχου

3ο αρχείο ελέγχου

= Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου προκειμένου να γίνουν δεκτές.

= Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου προκειμένου να γίνουν δεκτές και οι χρήστες θα μπορούν να δουν το συγκεκριμένο αρχείο ελέγχου.

= Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου την ώρα της υποβολής και ο χρήστης θα μπορεί να δει το αποτέλεσμα, αλλά όχι το αρχείο ελέγχου. Το αποτέλεσμα της εκτέλεσης δεν επηρεάζει το αν θα γίνει δεκτή η υποβολή (για προβλήματα με partial feedback).

= Το αρχείο ελέγχου θα χρησιμοποιηθεί μόνο στην τελική αξιολόγηση.

= Το αρχείο ελέγχου δεν θα χρησιμοποιηθεί για την αξιολόγηση.

[Δημιουργία groups / Επανυπολογισμός υποβολών](#)

Groups Αρχείων Ελέγχου

Initial (15 βαθμοί) ✘
 Αρχεία ελέγχου: **1, 2**

Final (10 βαθμοί) ✘
 Αρχεία ελέγχου: **1, 2, 3**

[+ Group](#)

Συνολική βαθμολογία προβλήματος: **25** βαθμοί

Καλωσήρθες admin
 Όνομα: admin
 Επίθετο: admin

Έξοδος

Σχήμα 4.4: Η σελίδα διαχείρισης των αρχείων ελέγχου μετά την προσθήκη των testcase groups. Διακρίνεται η επισκόπηση τους, μαζί με τα αρχεία ελέγχου που περιέχουν και τους βαθμούς τους.

Διαγωνισμοί
 Ημερίδα για Διαλμαίους

Δεν υπάρχουν αναπάντητες ερωτήσεις.

Αρχική | Διαγωνισμοί | Διαχείριση | Επικοινωνία

Ωρα Συστήματος

Τρίτη, 13 Μαρτίου 2018 - 21:38:51

Δημιουργία / Επεξεργασία Group Αρχείων Ελέγχου

Αρχείο Ελέγχου 1

Αρχείο Ελέγχου 2

Αρχείο Ελέγχου 3

= Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου προκειμένου να γίνουν δεκτές.

= Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου προκειμένου να γίνουν δεκτές και οι χρήστες θα μπορούν να δουν το συγκεκριμένο αρχείο ελέγχου.

= Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου την ώρα της υποβολής και ο χρήστης θα μπορεί να δει το αποτέλεσμα, αλλά όχι το αρχείο ελέγχου. Το αποτέλεσμα της εκτέλεσης δεν επηρεάζει το αν θα γίνει δεκτή η υποβολή (για προβλήματα με partial feedback).

= Το αρχείο ελέγχου θα χρησιμοποιηθεί μόνο στην τελική αξιολόγηση.

= Το αρχείο ελέγχου δεν θα χρησιμοποιηθεί για την αξιολόγηση.

Όνομα:

Πόντοι:

Υποβολή
Πίσω

Καλωσήρθες admin
 Όνομα: admin
 Επίθετο: admin

Έξοδος

Σχήμα 4.5: Εδώ φαίνεται η νέα σελίδα που δημιουργήθηκε για την δημιουργία και τροποποίηση μιας ομάδας αρχείων ελέγχου.


Αρχική Διαγωνισμοί Επικοινωνία

Ωρα Συστήματος
 Τετάρτη, 14 Μαρτίου 2018 - 21:00:18

ICPC Training - Υποβολές
 << Επιστροφή στο 'ICPC Training'

Πρόβλημα 1 (trena)

✓ Υποβολή 271267	01:37:27 06/11/2017	Επιτυχής (6/6)	Active
✓ Υποβολή 271266	00:55:47 06/11/2017	Επιτυχής (4/6)	Activate
✓ Υποβολή 271265	21:32:16 28/10/2017	Επιτυχής (4/6)	Activate
✗ Υποβολή 271264	20:42:19 28/10/2017	Απέτυχε	
✓ Υποβολή 271263	22:52:39 27/10/2017	Επιτυχής (4/6)	Activate
✓ Υποβολή 271218	21:56:46 19/09/2017	Επιτυχής (1/6)	Activate



Καλωσήρθες gbrag
 Όνομα: Giannis
 Επίθετο: Paragiannis
 Έξοδος

Σχήμα 4.6: Η τροποποιημένη παρουσίαση όλων των υποβολών ενός διαγωνιζόμενου, όπου διακρίνεται ο βαθμός επιτυχίας της υποβολής ως προς τα testcase groups που είχε σωστά.

HelleniCO Διαγωνισμοί
 ΗΕΙΣΕΙΣ ΤΗΝ ΠΛΗΡΟΦΟΡΙΑ

Δεν υπάρχουν νέα μηνύματα.

Αρχική Διαγωνισμοί Επικοινωνία


Ωρα Συστήματος
 Τετάρτη, 14 Μαρτίου 2018 - 21:19:17

Αξιολόγηση για το πρόβλημα 'share'
 << Επιστροφή στο 'Προπόνηση για GCJ'

Γλώσσα Προγραμματισμού: C
 Κατάσταση: Η μεταγλώττιση ήταν επιτυχής.



Καλωσήρθες gbrag
 Όνομα: Giannis
 Επίθετο: Paragiannis
 Έξοδος



Ενεργοί Διαγωνισμοί
 • Προπόνηση για GCJ
 Απομένουν: 697:40:44

Easy			
✓ 1	Σωστή απάντηση (100%)	0 sec.	Προβολή
✓ 2	Σωστή απάντηση (100%)	0 sec.	

Medium			
✗ 1	Λάθος απάντηση	0 sec.	
✓ 2	Σωστή απάντηση (100%)	0 sec.	

Το πρόγραμμα αποθηκεύτηκε και θα αξιολογηθεί μετά το πέρας του διαγωνισμού.

Προβολή πηγαίου κώδικα

Όροι χρήσης · Οδηγίες για τους διαγωνισμούς · Συχνές απορίες · Πανελλήνιος Διαγωνισμός Πληροφορικής · Επικοινωνία
 © Copyright 2018 Χατζημήχος Γιάννης, Τάμος Χρήστος for Hellenico

Σχήμα 4.7: Η σελίδα τροποποιήθηκε σύμφωνα με τον νέο τρόπο αξιολόγησης, παρουσιάζοντας για κάθε group την αξιολόγηση της υποβολής για κάθε αρχείο ελέγχου που περιέχει.

και άλλα για την τελική αξιολόγηση. Αντίστοιχα, θα πρέπει να δημιουργηθούν ακόμα ίσος αριθμός groups με τα ενεργά αρχεία ελέγχου (όλα εκτός των μωβ), και πιο συγκεκριμένα, κάθε group θα περιέχει ένα αρχείο ελέγχου με χρώμα πράσινο και η βαθμολογία της θα είναι ίση με τη βαθμολογία του αρχείου ελέγχου. Έτσι, πετυχαίνουμε την προσομοίωση και της τελικής αξιολόγησης, καθώς όλα αυτά τα groups δε θα ελέγχονται στην κανονική υποβολή διότι περιέχουν μόνο πράσινα αρχεία ελέγχου (από ένα κάθε group). Για την αυτοματοποίηση της διαδικασίας μετατροπής που περιγράφηκε υλοποιήθηκε script, το οποίο εισάχθηκε στη σελίδα διαχείρισης αρχείων ελέγχου του προβλήματος.

Η υλοποίηση όσον αφορά στο frontend και τη λογική του Grader, συνδυάστηκε με μερικό refactoring των κλάσεων και των συναρτήσεων, χρησιμοποιώντας πολλές από τις αρχές που περιγράφονται στο Clean Code του Robert Martin [Mart09]. Μέρος των αλλαγών ήταν και η προσθήκη ενός πεδίου στον πίνακα των υποβολών, με τίτλο resultsjson (σχήμα 4.3), το οποίο περιέχει τα αναλυτικά αποτελέσματα μιας υποβολής, κωδικοποιημένα σε μορφή JSON, έτσι ώστε να μην υπολογίζονται κάθε φορά που απαιτούνται. Επιπλέον, αλλαγές έγιναν ώστε να αφαιρεθούν κομμάτια επαναλαμβανόμενου κώδικα με την αντίστοιχη δημιουργία νέων δομών και κλάσεων, αποσύνδεση της λογικής διαφορετικών αντικειμένων και μείωση της πολυπλοκότητας μεγάλων κομματιών κώδικα με δημιουργία μικρότερων συναρτήσεων με περιγραφικά ονόματα.

Κεφάλαιο 5

Σχεδίαση για διαχωρισμό Προβλημάτων - Διαγωνισμών

Στο παρόν κεφάλαιο θα περιγραφούν οι αλλαγές που έγιναν στο Grader για την τροποποίηση της λογικής του με σκοπό τη βελτίωση της ευκολίας χρήσης του στο πλαίσιο μαθημάτων προγραμματισμού και σειρών ασκήσεων. Αυτό απαιτεί την μερική αποσύνδεση των προβλημάτων από τους διαγωνισμούς ώστε αυτά να μπορούν να επαναχρησιμοποιηθούν. Θα αναλύσουμε πρώτα το κίνητρο για την αλλαγή αυτή και έπειτα τις λεπτομέρειες της υλοποίησης.

5.1 Κίνητρο

Ο πρωταρχικός στόχος που δημιουργήθηκε ο Grader ήταν η χρήση του για διαγωνισμούς πληροφορικής. Κάθε διαγωνισμός θα αποτελούσε ξεχωριστό, μεμονωμένο γεγονός με προβλήματα που θα είχαν δημιουργηθεί για το συγκεκριμένο διαγωνισμό και υποβολές που αντιστοιχούν αποκλειστικά στο πρόβλημα.

Τα προβλήματα, έπειτα από τη δημιουργία τους, παραμένουν ανένταχτα, στην κατηγορία "Προβλήματα εκτός διαγωνισμών" της σελίδας διαχείρισης όπως φαίνεται και στο σχήμα 5.1. Το επόμενο βήμα είναι η μετακίνηση τους σε κάποιον διαγωνισμό με χρήση του μενού στα δεξιά του προβλήματος. Η μετακίνηση αυτού του τύπου είναι ο μόνος τρόπος να επαναχρησιμοποιηθεί το πρόβλημα και σε άλλους διαγωνισμούς, αφού τελειώσει αυτός για τον οποίο δημιουργήθηκε (ουσιαστικά, ο πρώτος στον οποίο μετακινήθηκε). Το θέμα που δημιουργείται σε αυτό το σημείο είναι ότι κατά τη μετακίνηση του, το πρόβλημα διατηρεί όλες τις προηγούμενες υποβολές του, οι οποίες πρακτικά αγνοούνται στο νέο διαγωνισμό, ενώ ο προηγούμενος διαγωνισμός χάνει το πρόβλημα που είχε. Όλα τα παραπάνω οφείλονται στον τρόπο που είναι σχεδιασμένη η βάση, ο οποίος παρουσιάζεται στο σχήμα 5.2.

Στο σχήμα 5.2 αρχίζει να διακρίνεται το πρόβλημα που δημιουργείται. Η σύνδεση κάθε προβλήματος με το διαγωνισμό γίνεται μέσω του πεδίου `competitionID` στον πίνακα των προβλημάτων. Ως αποτέλεσμα, το μόνο που κάνει η λειτουργία της μετακίνησης προβλήματος σε άλλον διαγωνισμό είναι να αλλάξει αυτό το πεδίο. Επιπροσθέτως, όπως βλέπουμε, οι υποβολές συνδέονται άμεσα μόνο με τα προβλήματα και αυτός είναι ο λόγος που μεταφέρονται μαζί με αυτά κατά τη μετακίνηση τους.

Τέλος, είναι αξιοσημείωτος ο τρόπος που στον πίνακα `finalresults`, στον οποίο καταχωρούνται τα αποτελέσματα μετά την τελική αξιολόγηση, αποθηκεύονται τα επιμέρους σκορ των προβλημάτων του. Αυτό γίνεται στο πεδίο `scoreDetails`, όπου εισάγονται οι βαθμολογίες των προβλημάτων του διαγωνισμού χωρισμένες απλά με κόμμα, σύμφωνα με μια αύξουσα ταξινόμηση των `id` των προβλημάτων που περιείχε ο διαγωνισμός κατά την τελική αξιολόγηση.

Παραδείγματος χάρι, αν ο διαγωνισμός 15 περιέχει τα προβλήματα 48 και 51 και ένας διαγωνιζόμενος έχει λάβει 7 βαθμούς στο πρώτο και 9 στο δεύτερο, το πεδίο `score` θα έχει την τιμή 16 και το πεδίο `scoreDetails` θα έχει την τιμή 7,9. Όπως γίνεται αντιληπτό, όταν αλλάξει η σύνθεση ενός διαγωνισμού, χάνεται η ιστορικότητα των αποτελεσμάτων αφού δεν είναι δυνατό να ανακτηθεί από τη βάση η σύνδεση των βαθμολογιών με τα προβλήματα του διαγωνισμού.

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
Δεν υπάρχουν προβλήματα				

palios2

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
Δεν υπάρχουν προβλήματα				

Palios Diagwnismos

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
162	7/5	2/5/15	palio1	palio pronlima 1

Γλώσσες Προγραμματισμού 2 - 2016

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
Δεν υπάρχουν προβλήματα				

ICPC Training

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
159	3/2	2/36/119	trena	Πολύχρωμα τρένα
160	1/3	2/8/19	sudokugame	Σουντόκου για δύο

Εξετάση Προγραμματισμός Η.Υ. 2015-2016

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
Δεν υπάρχουν προβλήματα				

Προγραμματιστικές Τεχνικές 2015-2016

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
Δεν υπάρχουν προβλήματα				

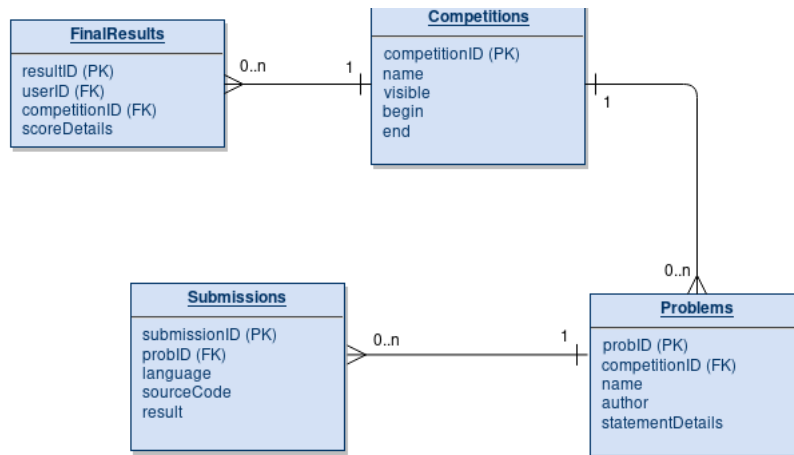
Προβλήματα εκτός διαγωνισμών

id	Αρχεία	Λύσεις	Όνομα	Τίτλος
177	0/0/0/0	0/0/0	test	Τεστ πρόβλημα

Άλλες ενέργειες

Εισαγωγή αρχείου διαγωνιζομένων

Σχήμα 5.1: Το κάτω μέρος της διαχείρισης διαγωνισμών περιέχει τα προβλήματα που δεν έχουν ενταχτεί σε κάποιο διαγωνισμό. Με το πράσινο κουμπί στα δεξιά μπορούν να μεταφερθούν σε διαγωνισμό. Με τον ίδιο τρόπο μπορούν όλα τα προβλήματα να μεταφερθούν σε καινούριους διαγωνισμούς ώστε να επαναχρησιμοποιηθούν.



Σχήμα 5.2: Οι πίνακες και οι σχέσεις τους πριν την αλλαγή μας. Παρατηρούμε, ότι οι υποβολές δεν έχουν σύνδεση με το διαγωνισμό, ενώ τα προβλήματα μπορούν να ανήκουν μόνο σε ένα διαγωνισμό.

5.2 Υλοποίηση

Η κύρια λειτουργική αλλαγή/δυνατότητα που θα προστεθεί είναι αυτή της προσθήκης των προβλημάτων σε νέους διαγωνισμούς χωρίς να επηρεάζονται οι προηγούμενοι, με αντιγραφή δηλαδή των προβλημάτων. Το κυριότερο μέρος της υλοποίησης έχει να κάνει με την αλλαγή των πινάκων της βάσης και των σχέσεων μεταξύ τους.

Αρχικά, θα πρέπει να δημιουργηθεί ένας πίνακας που να συνδέει κάθε διαγωνισμό με τα προβλήματα που διαθέτει. Το πεδίο στον πίνακα των προβλημάτων που έως τώρα χρησίμευε για αυτή τη σύνδεση δεν αρκεί αφού θέλουμε πλέον να υπάρχει σχέση "πολλά προς ένα" για προβλήματα και διαγωνισμούς. Ο νέος πίνακας χρειάζεται απλά να περιέχει τα πεδία `competitionID` και `probID`.

Όπως αναφέρθηκε και παραπάνω, οι υποβολές θα πρέπει να συνδέονται με το διαγωνισμό και όχι με το πρόβλημα. Αυτό θα επιτευχθεί με την προσθήκη του πεδίου `competitionID` στον πίνακα των υποβολών. Με αυτό τον τρόπο είναι εύκολο να γίνει ο διαχωρισμός των υποβολών ανά διαγωνισμό και πρόβλημα ώστε κάθε πρόβλημα να μπορεί να έχει ξεχωριστά δεδομένα υποβολών και αποτελεσμάτων σε κάθε διαγωνισμό που ανήκει.

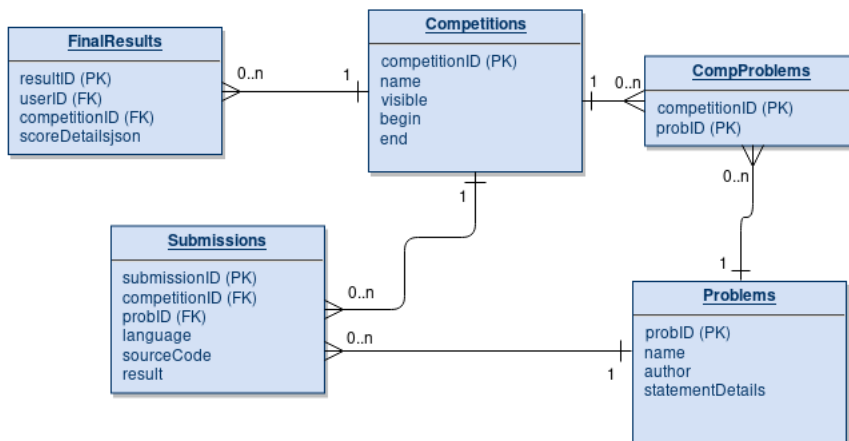
Άλλη μια αλλαγή που κρίθηκε σημαντική είναι η προσθήκη ενός πεδίου JSON στον πίνακα των αποτελεσμάτων σε αντικατάσταση του `scoreDetails`. Το `scoreDetailsjson` θα έχει την ίδια λογική, δηλαδή θα αναγράφει τις επιμέρους βαθμολογίες σε κάθε πρόβλημα του διαγωνισμού. Η διαφορά θα είναι ότι δε θα σημειώνεται στη βάση μόνο η βαθμολογία αλλά ζευγάρια `probID`: βαθμολογία. Στο παράδειγμα που χρησιμοποιήθηκε προηγουμένως, το πεδίο θα έχει τη μορφή `{48 : 7, 51 : 9}`.

Χάρη στην προσθήκη του πεδίου `scoreDetailsjson`, ένας διαχειριστής θα μπορεί να δει με λεπτομέρεια τα αποτελέσματα ενός διαγωνισμού για κάθε διαγωνιζόμενο ακόμα και αν έχει αλλάξει η δομή του, κάτι που πριν ήταν αδύνατο. Βεβαίως, από τη στιγμή που θα εισαχθεί η δυνατότητα αντιγραφής αντί μετακίνησης των προβλημάτων δε θα τίθεται συχνά θέμα αλλαγής της δομής ενός διαγωνισμού μετά τον υπολογισμό αποτελεσμάτων και την ολοκλήρωσή του.

Έπειτα από τις αλλαγές που περιγράφηκαν, η νέα μορφή της βάσης φαίνεται στο σχήμα 5.3. Ο πίνακας `CompProblems` είναι η νέα προσθήκη που είναι αναγκαία για τη σύνδεση προβλημάτων με διαγωνισμούς διατηρώντας μια κανονικοποιημένη βάση για αποφυγή του πλεονασμού δεδομένων. Νέα πεδία προστέθηκαν επίσης και στους πίνακες `Submissions` και `FinalResults`.

Έχοντας αλλάξει τη βάση, το επόμενο βήμα ήταν η υλοποίηση των αλλαγών στον Grader ώστε να δίνεται η δυνατότητα της αντιγραφής των προβλημάτων και να αξιοποιούνται τα νέα πεδία και πίνακες. Η πλειοψηφία των αλλαγών αφορούν τη σελίδα διαχείρισης, καθώς εκεί πρέπει να αλλάξει ο τρόπος που αντιστοιχίζονται τα προβλήματα με τους διαγωνισμούς, όπως και οι υποβολές αυτών. Ακόμα, η λειτουργία του πλήκτρου μετακίνησης αλλάζει σε αντιγραφή, πάλι επιλέγοντας διαγωνισμό. Τέλος, στο κάτω μέρος της σελίδας, όπου αναγράφονταν τα ανένταχτα προβλήματα, κρίθηκε προτιμότερο να αναγράφονται όλα τα προβλήματα ώστε να είναι ευκολότερο να αναζητηθεί και να αντιγραφεί κάποιο σε ένα νέο διαγωνισμό. Η νέα διαχείριση παρουσιάζεται στα σχήματα 5.4 και 5.5.

Μια ενδιαφέρουσα παρατήρηση είναι ότι η αντιγραφή ενός προβλήματος σε νέο διαγωνισμό, απλά αλλάζει τους συσχετισμούς στη βάση, προσθέτοντας μια νέα εγγραφή στον `CompProblems`. Τόσο τα αρχεία ελέγχου, που είναι αποθηκευμένα στο δίσκο του εξυπηρετητή για να διαβάζονται από τον Kewii, όσο και οι ομάδες αρχείων ελέγχου που αντιστοιχίζονται στη βάση με τα προβλήματα, δεν αντιγράφονται και ως αποτέλεσμα αντιστοιχούν στο πρόβλημα σε όλους τους διαγωνισμούς που αυτό ανήκει. Η συγκεκριμένη σχεδιαστική επιλογή στηρίχθηκε στο σκεπτικό ότι το πρόβλημα δεν αναμένεται να αλλάξει σημαντικά κατά την επαναχρησιμοποίησή του. Το ενδεχόμενο να χρειαστεί κάποια αλλαγή σε πρόβλημα που ανήκει ήδη σε παλαιότερους ή παράλληλους διαγωνισμούς δε μπορεί να αποκλειστεί, και για το λόγο αυτό προστέθηκε ένα προειδοποιητικό μήνυμα στη σελίδα διαχείρισης αρχείων ελέγχου, προς όποιο διαχειριστή επιχειρήσει τέτοιες αλλαγές.



Σχήμα 5.3: Δημιουργήθηκε ο πίνακας CompProblems που μας επιτρέπει τη σύνδεση προβλημάτων με πολλούς διαγωνισμούς ταυτόχρονα. Παράλληλα, οι υποβολές απέκτησαν σύνδεση με το διαγωνισμό ώστε να ανεξαρτητοποιηθούν από το πρόβλημα και να ανήκουν στο διαγωνισμό για τον οποίο δημιουργήθηκαν. Επίσης, προστέθηκε και το πεδίο scoreDetailsjson στο FinalResults.

The screenshot shows the HelleniCO Διαγωνισμοί website. At the top, there is a navigation bar with tabs for Αρχική, Διαγωνισμοί, Διαχείριση, and Επικοινωνία. Below this is a green banner for the system time: Τετάρτη, 14 Μαρτίου 2018 - 20:45:26. The main content area is titled "Διαχείριση Διαγωνισμών" and contains several sections for managing competitions and problems. On the right, there is a user profile for "Καλωσήρθες admin" with the name "admin" and role "admin", and a "Έξοδος" button.

Εξέταση Γλωσσών 2018

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
176	4/3	3/3/3	shops	Καταστήματα προγραμματιστών

Προπόνηση για GCJ

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
174	2/1	1/13/27	share	Δίκαιο μοίρασμα
173	1/0	1/25/66	icecream	Παγωτό για όλους

Προγραμματιστικές Τεχνικές 2017

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
160	1/3	0/0/0	sudokugame	Σουντόκου για δύο
167	5/3	3/9/12	onemore1	One more card

palios2

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
164	4/3	3/3/4	palio2	palio 2

Palios Diagwnismos

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
162	7/5	2/5/15	palio1	palio pronima 1

Γλώσσες Προγραμματισμού 2 - 2016

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος
160	1/3	2/8/19	sudokugame	Σουντόκου για δύο

Σχήμα 5.4: Η διαχείριση διαγωνισμών δεν έχει κάποια διαφορά εμφανισιακά, αλλά πλέον επιτρέπεται η χρήση ενός προβλήματος σε πολλαπλούς διαγωνισμούς, όπως φαίνεται στο πρόβλημα sudokugame. Το πρώην πλήκτρο μεταφοράς στα δεξιά κάθε προβλήματος πλέον αντιγράφει το πρόβλημα στον επιλεγμένο διαγωνισμό, ενώ το X το αφαιρεί χωρίς να το διαγράφει.

Προγραμματιστικές Τεχνικές 2015-2016

id	Αρχεία/Groups	Λύσεις	Όνομα	Τίτλος	
164	4/3	0/0/0	ralio2	ralio 2	
173	1/1	0/0/0	icecream	Παγωτό για όλους	

Όλα τα προβλήματα

id	Αρχεία	Λύσεις	Όνομα	Τίτλος	
159	3/2	2/36/119	trena	Πολύχρωμα τρένα	
160	1/3	2/8/19	sudokugame	Σουντόκου για δύο	
162	7/5	2/5/15	ralio1	ralio proñima 1	
164	4/3	3/3/4	ralio2	ralio 2	
167	5/3	6/13/16	onemore1	One more card	
173	1/1	1/25/66	icecream	Παγωτό για όλους	
174	2/1	1/13/27	share	Δίκαιο μοίρασμα	
176	4/3	7/11/11	shops	Καταστήματα προγραμματιστών	
177	0/0	0/0/0	test	Τεστ πρόβλημα	

Άλλες ενέργειες

Εισαγωγή αρχείου διαγωνιζομένων

Σχήμα 5.5: Το κάτω μέρος της διαχείρισης επανασχεδιάστηκε ώστε να περιέχει όλα τα προβλήματα για εύκολη επισκόπηση, τροποποίηση τους και αντιγραφή τους σε διαγωνισμούς.

Κεφάλαιο 6

Λοιπές Προσθήκες

Εκτός από τις νέες δυνατότητες που αναλύθηκαν στα προηγούμενα κεφάλαια, υλοποιήθηκαν και αλλαγές μικρότερης πολυπλοκότητας, που δε χρειάζονται ολόκληρο κεφάλαιο για να περιγραφούν ξεχωριστά. Στο συγκεκριμένο κεφάλαιο θα αναφερθούν οι σημαντικότερες από αυτές.

6.1 Προσθήκη γλώσσας προγραμματισμού Python

6.1.1 Προσθήκη γλωσσών στο Grader/Kewii

Ο Grader, όπως έχει αναφερθεί και σε προηγούμενα κεφάλαια, επιτρέπει στους διαγωνιζόμενους την υποβολή λύσεων σε οποιαδήποτε γλώσσα προγραμματισμού υποστηρίζεται. Οι υποστηριζόμενες γλώσσες προγραμματισμού είναι οι παρακάτω:

- C
- C++
- Pascal
- Pzascal
- F#
- OCaml
- Java
- Fortran
- Haskell

Η προσθήκη μιας νέας γλώσσας στο Grader δεν αποτελεί δύσκολη διαδικασία. Όσον αφορά στο frontend, αρκεί απλά η προσθήκη στο μενού επιλογής γλώσσας στην υποβολή και η αντίστοιχη κωδικοποίηση που θα χρησιμοποιηθεί για την καταχώρηση της υποβολής σε βάση και Kewii (γίνεται συνήθως με χρήση της επέκταση των πηγαίων αρχείων της γλώσσας, π.χ. cpp για C++). Όπως ο Kewii δεν εμπλέκεται στη διαδικασία της αξιολόγησης, ο Grader, αντίστοιχα, έχει καθήκον απλά να περάσει την επιλογή της γλώσσας και τον πηγαίο κώδικα ώστε να αναλάβει ο Kewii τον έλεγχο.

Ο Kewii αναλαμβάνει, για κάθε υποβολή, να μεταγλωττίσει τον πηγαίο κώδικα ώστε να παράξει το εκτελέσιμο αρχείο και έπειτα να τρέξει το εκτελέσιμο για κάθε αρχείο ελέγχου που του έχει ζητηθεί. Η μεταγλώττιση και η εκτέλεση των υποβολών γίνεται σε ένα περιορισμένο περιβάλλον, με τη χρήση μόνο των αναγκαίων μεταγλωττιστών και βιβλιοθηκών. Για την προσθήκη μιας γλώσσας θα πρέπει να εγκατασταθούν στο συγκεκριμένο περιβάλλον τα αντίστοιχα εκτελέσιμα και να τροποποιηθεί το αρχείο που περιέχει τις αντιστοιχίσεις γλωσσών και εντολών μεταγλώττισης/εκτέλεσης.

6.1.2 Επιλογή Python

Η Python είναι μια interpreted προγραμματιστική γλώσσα υψηλού επιπέδου γενικού σκοπού. Διαθέτει πολύ πλούσια βιβλιοθήκη για μια ευρεία γκάμα λειτουργιών και επιστημονικών πεδίων. Έχει σχεδιαστεί με έμφαση στην αναγνωσιμότητα και στη χρησιμοποίηση λιγότερου κώδικα, ενώ ευνοεί πολλαπλά προγραμματιστικά στυλ όπως είναι ο δομημένος, ο αντικειμενοστρεφής και ο συναρτησιακός προγραμματισμός.

Δεδομένης της δημοτικότητας της Python και της ευκολίας χρήσης της, είναι μια απαραίτητη προσθήκη στο Grader που πιθανότατα θα εκτιμηθεί από διαγωνιζόμενους μαθητές και φοιτητές, για τους οποίους είναι αρκετά πιθανό να αποτέλεσε πρώτη γλώσσα εισαγωγής στον προγραμματισμό. Η Python δε λείπει από κανένα από τα συστήματα αξιολόγησης που μελετήθηκαν, ενώ πλέον αποτελεί τη δημοφιλέστερη επιλογή στα κορυφαία αμερικάνικα πανεπιστήμια όσον αφορά στα εισαγωγικά μαθήματα των τμημάτων επιστήμης των υπολογιστών [Guo14]. Είναι μια από τις πιο αναπτυσσόμενες προγραμματιστικές γλώσσες σύμφωνα με στοιχεία του Stack Overflow [Robi17a] χάρη κυρίως στην καθιέρωση της σε πολλά προγράμματα προπτυχιακών σπουδών και στην ανάπτυξη των τομέων της ανάλυσης δεδομένων και μηχανικής μάθησης, στους οποίους κυριαρχεί ως εργαλείο [Robi17b].

Η διαφορά της Python με τις γλώσσες που υποστηρίζει ο Kewii είναι ότι αυτή αποτελεί μια interpreted γλώσσα και ως αποτέλεσμα δεν έχει ιδιαίτερο νόημα η μεταγλώττιση της σε ένα εκτελέσιμο. Σε κάθε υποβολή που χρησιμοποιείται, θα γίνεται απευθείας εκτέλεση με είσοδο τα αρχεία ελέγχου και θα εμφανίζονται μόνο τα σφάλματα κατά τον χρόνο εκτέλεσης στον διαγωνιζόμενο, αντίθετα με τις υπόλοιπες γλώσσες όπου υπάρχουν και σφάλματα μεταγλώττισης. Στο μέλλον θα μπορούσε να διερευνηθεί η δυνατότητα ενός εργαλείου ανάλυσης του πηγαίου κώδικα για τη διευκόλυνση των διαγωνιζόμενων με καλύτερα μηνύματα σφαλμάτων.

Η έκδοση της Python που επιλέχθηκε είναι η 2.7. Η υποστήριξη της συγκεκριμένης έκδοσης και γενικά της Python 2 θα διαρκέσει έως το 2020, οπότε καλό θα είναι να προστεθεί και η Python 3 στο μέλλον. Δεδομένης της δουλειάς που έγινε για την προσθήκη της έκδοσης 2, η προσθήκη της 3 θα είναι αρκετά εύκολη και άμεση.

6.2 Μαζικό ανέβασμα αρχείων ελέγχων/ομάδων

6.2.1 Κίνητρο

Κατά τη διαδικασία δημιουργίας ενός προβλήματος, είναι απαραίτητο να προστεθεί ένας συχνά μεγάλος αριθμός αρχείων ελέγχου. Ο μόνος τρόπος να γίνει αυτό είναι μέσω της σελίδας διαχείρισης των αρχείων ελέγχου, όπως φαίνεται στο σχήμα 4.1, και κάθε νέο αρχείο ανεβαίνει ξεχωριστά, δηλαδή δεν υπάρχει κάποια μαζική διαδικασία.

Για να αποφευχθεί η επαναληπτική και χρονοβόρα διαδικασία, οι έως τώρα διαχειριστές έγραφαν απλά PHP scripts για την ενημέρωση της βάσης με τα νέα αρχεία ελέγχου, και αντέγραφαν χειροκίνητα τα αρχεία στους αντίστοιχους φακέλους του Kewii. Μετά την προσθήκη των testcase groups, στη διαδικασία αυτή προστέθηκε και η δημιουργία και παραμετροποίηση των groups με τα επιθυμητά αρχεία και tags.

Για τους παραπάνω λόγους κρίθηκε αναγκαία η προσθήκη ενός αυτοματοποιημένου τρόπου μαζικής προσθήκης αρχείων για τα προβλήματα, στο οποίο να υπάρχει και η δυνατότητα ορισμού των κατάλληλων testcase groups. Το εργαλείο που θα υλοποιηθεί θα οφείλει τόσο να ενημερώνει τη βάση, όσο και να φορτώνει τα αρχεία στον εξυπηρετητή του Kewii. Για την επεξεργασία των ομάδων, θα πρέπει να χρησιμοποιηθεί ένα περιγραφικό αρχείο, ιδανικά σε ένα ανθρωπίνως αναγνώσιμο τύπο αρχείου με σκοπό την εύκολη δημιουργία του.

6.2.2 Υλοποίηση

Το εργαλείο που περιγράφηκε θα προστεθεί στη σελίδα διαχείρισης των αρχείων ελέγχου κάτω από το ήδη υπάρχον ανέβασμα μεμονωμένου αρχείου. Μετά την προσθήκη η σελίδα θα έχει τη μορφή του σχήματος 6.1. Ο διαχειριστής θα ανεβάζει ένα συμπιεσμένο αρχείο (zip) με τα αρχεία εισόδου και εξόδου που θέλει να προσθέσει στο πρόγραμμα. Στο αρχείο θα πρέπει, επιπλέον, να υπάρχει και ένα αρχείο με όνομα `descriptor.json` το οποίο θα αναλαμβάνει να περιγράψει στο εργαλείο τις προδιαγραφές αρχείων και ομάδων.

Η μορφή JSON επιλέχθηκε για τους λόγους που αναφέρθηκαν παραπάνω, δηλαδή για τα χαρακτηριστικά της ως αναγνώσιμη και με δυνατότητα εύκολης εισαγωγής αντικειμένων και λιστών. Στο συγκεκριμένο αρχείο θέλαμε να περιγράφονται τα παρακάτω:

- Ο τρόπος ονομασίας των αρχείων ελέγχου για τα αρχεία εισόδου και εξόδου
- Το πλήθος των ομάδων αρχείων ελέγχου
- Για κάθε ομάδα αρχείων ελέγχου το όνομα της, οι πόντοι και τα αρχεία ελέγχου που περιέχει
- Για κάθε αρχείο ελέγχου μέσα σε ομάδα ο τύπος εκτέλεσης του (tag)

Ο Grader περιμένει να βρει μέσα στο συμπιεσμένο αρχείο ζευγάρια εισόδου-εξόδου για κάθε αρχείο ελέγχου που πρέπει να προστεθεί. Επομένως, θα πρέπει να είναι γνωστή η μορφή των ονομάτων των αρχείων. Για μεγαλύτερη ευελιξία, δημιουργήθηκαν δύο προαιρετικά πεδία, για αρχεία εισόδου και εξόδου αντίστοιχα, τα οποία δέχονται τη μορφή ονομασίας που θα χρησιμοποιηθεί με χρήση ενός `format` παρόμοιου με το `printf` της PHP. Προφανώς, αυτό θα πρέπει να περιέχει ένα προσδιοριστή για ακέραιους αριθμούς ώστε να λειτουργεί σωστά η αντικατάσταση για την παραγωγή των ονομασιών των αρχείων. Αν τα πεδία αυτά δεν έχουν συμπληρωθεί χρησιμοποιείται η προεπιλεγμένη μορφή, δηλαδή `probname.in1` και `probname.out2`, για το πρώτο αρχείο εισόδου και εξόδου ενός προβλήματος με όνομα `probname`.

Όσον αφορά στις ομάδες αρχείων ελέγχου, αυτές πρέπει να περιέχονται μέσα στο `descriptor.json` σαν λίστες αντικειμένων. Κάθε αντικείμενο που αντιστοιχεί σε μια νέα ομάδα, έχει πεδία για το όνομα της, τους πόντους που αξίζει και τα αρχεία ελέγχου που θα περιέχει. Το τελευταίο πεδίο αποτελεί λίστα και περιέχει αντικείμενα με δύο πεδία: αριθμός αρχείου ελέγχου και τύπος εκτέλεσης. Ο τύπος εκτέλεσης θα δίνεται με το όνομα του χρώματος, δηλαδή θα μπορεί να έχει τις τιμές `orange`, `yellow`, `blue`, `green`, `purple`. Ακολουθεί ένα πρότυπο `descriptor.json` στο οποίο έχουν συμπληρωθεί όλα τα διαθέσιμα πεδία.

```
{
  "numberOfTests": 4,
  "numberOfGroups": 3,
  "testNameFormatInput": "onemore1.in%d",
  "testNameFormatOutput": "onemore1.out%d",
  "groups": [
    {
      "name": "Group 1",
      "points": 4,
      "tests": [
        {
          "number": 1,
          "mode": "blue"
        }
      ]
    }
  ]
}
```

```

},
{
  "name": "Group 2",
  "points": 7,
  "tests": [
    {
      "number": 1,
      "mode": "yellow"
    },
    {
      "number": 2,
      "mode": "green"
    },
    {
      "number": 3,
      "mode": "blue"
    }
  ]
},
{
  "name": "Final",
  "points": 20,
  "tests": [
    {
      "number": 1,
      "mode": "green"
    },
    {
      "number": 2,
      "mode": "green"
    },
    {
      "number": 3,
      "mode": "green"
    },
    {
      "number": 4,
      "mode": "green"
    }
  ]
}
]
}

```

Listing 6.1: Παράδειγμα descriptor.json. Το συγκεκριμένο αρχείο περιγράφει 4 αρχεία ελέγχου σε 3 ομάδες αρχείων. Ακόμα ορίζει τη μορφή ονομασίας των αρχείων εισόδου και εξόδου.

6.2.3 Εργαλείο δημιουργίας

Χάρη στην αναγνώσιμη μορφή του JSON είναι εύκολο ένας διαχειριστής να συντάξει ένα JSON αρχείο για το ανέβασμα αρχείων ελέγχου και καθορισμό groups. Παρόλα αυτά, θα είναι αρκετά χρήσιμο

Αρχική Διαγωνισμοί Διαχείριση Επικοινωνία

Ωρα Συστήματος
Τετάρτη, 14 Μαρτίου 2018 - 21:25:22

test - Αρχεία ελέγχου

Αρχεία Ελέγχου

- Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου προκειμένου να γίνουν δεκτές.
- Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου προκειμένου να γίνουν δεκτές και οι χρήστες θα μπορούν να δουν το συγκεκριμένο αρχείο ελέγχου.
- Οι υποβολές θα ελεγχθούν με το συγκεκριμένο αρχείο ελέγχου την ώρα της υποβολής και ο χρήστης θα μπορεί να δει το αποτέλεσμα, αλλά όχι το αρχείο ελέγχου. Το αποτέλεσμα της εκτέλεσης δεν επηρεάζει το αν θα γίνει δεκτή η υποβολή (για προβλήματα με partial feedback).
- Το αρχείο ελέγχου θα χρησιμοποιηθεί μόνο στην τελική αξιολόγηση.
- Το αρχείο ελέγχου δεν θα χρησιμοποιηθεί για την αξιολόγηση.

Δημιουργία groups / Επανυπολογισμός υποβολών

Groups Αρχείων Ελέγχου
+ Group

Νέο αρχείο ελέγχου

Αρχείο εισόδου: No file selected.

Αρχείο εξόδου: No file selected.

Μαζικό upload αρχείων/groups (διαγραφή υπαρχόντων)

Αρχείο zip: No file selected.

Καλωσήρθες admin
Όνομα: admin
Επίθετο: admin

Σχήμα 6.1: Η επιλογή μαζικού ανεβάσματος αρχείων ελέγχου τοποθετήθηκε κάτω από το ανέβασμα μεμονωμένων αρχείων. Στον τίτλο συμπεριλήφθηκε και η προειδοποίηση για τη διαγραφή υπαρχόντων αρχείων και ομάδων κατά το ανέβασμα.

το συγκεκριμένο αρχείο να μη συντάσσεται με το χέρι, ώστε να μην υπάρχει και η πιθανότητα σφάλματος. Για το λόγο αυτό δημιουργήθηκε ένα interactive script σε Python για την αυτόματη παραγωγή ενός descriptor.json αρχείου.

Το συγκεκριμένο script, μόλις εκτελεστεί, σε οποιοδήποτε περιβάλλον, διατυπώνει ερωτήσεις σχετικές με τον αριθμό και τα ονόματα των αρχείων ελέγχου και ομάδων και έπειτα για κάθε ομάδα σχετικά με το ποια αρχεία περιέχει και τους τύπους εκτέλεσης τους. Στο τέλος, παράγει το αρχείο που περιγράφηκε. Στο σχήμα 6.2 φαίνεται μια τυπική εκτέλεση του συγκεκριμένου εργαλείου.

```

# grader @ debian in ~/hellenico/kewii on git:mysql-to-PDO x [21:35:50]
$ python generatejson.py
How many tests would you like to upload? (Default: 0): 4
What is the name format of the testcases input? (Default: probname.in%d): onemore1.in%d
What is the name format of the testcases output? (Default: probname.out%d): onemore1.out%d
How many groups would you like to create? (Default: 0): 3
Would you like to describe the groups? [Y/n]:
New group (1/3)
How will be the new group named? [optional]: Group 1
How many points will it be worth? [optional]: 4

Input a string describing what tests should be included in this group.
An example would be: "1y 3b 4o 7g", to have 1 in yellow mode, 3 in
blue, 4 orange and 7 green

1y 2b
New group (2/3)
How will be the new group named? [optional]: Group 2
How many points will it be worth? [optional]: 10

Input a string describing what tests should be included in this group.
An example would be: "1y 3b 4o 7g", to have 1 in yellow mode, 3 in
blue, 4 orange and 7 green

2o 3o
New group (3/3)
How will be the new group named? [optional]: Group 3
How many points will it be worth? [optional]: 15

Input a string describing what tests should be included in this group.
An example would be: "1y 3b 4o 7g", to have 1 in yellow mode, 3 in
blue, 4 orange and 7 green

1g 2g 3g 4g

./descriptor.json has been created

You should now create a zip archive containing descriptor.json and test files

```

Σχήμα 6.2: Ένα παράδειγμα εκτέλεσης του generatejson.py για την αυτόματη παραγωγή του απαραίτητου descriptor.json. Οι εντολές του χρήστη είναι υπογραμμισμένες.

6.3 Αλλαγή MySQL connector

Μια τροποποίηση που θεωρήθηκε απαραίτητη ήταν η αλλαγή της επέκτασης επικοινωνίας της PHP με τη MySQL βάση δεδομένων μας. Η επέκταση, που χρησιμοποιείται τόσο από τον Grader όσο και από τον Kewii, είναι το πρωτότυπο (original) MySQL API. Το συγκεκριμένο API έχει αρκετά μειονεκτήματα, που έχουν οδηγήσει και στην παύση της υποστήριξης του όπως περιγράφεται στο [Harv12].

Δεδομένου ότι η συγκεκριμένη επέκταση δεν συντηρείται πλέον και ακόμα χειρότερα εμφανίζει σφάλματα ήδη από την έκδοση 5.5 της PHP και στην 7 δεν υπάρχει, είναι επιτακτικό να αφαιρεθεί από τον κώδικα της εφαρμογής μας και να αντικατασταθεί με μια πιο σύγχρονη. Οι επίσημα υποστηριζόμενες επεκτάσεις που μπορούμε να χρησιμοποιήσουμε είναι οι mysqli και PDO. Για την επιλογή χρησιμοποιήθηκε ο πίνακας σύγκρισης των προαναφερθέντων επεκτάσεων που υπάρχει στο manual της PHP [Conta] και φαίνεται στο σχήμα 6.3.

Όπως βλέπουμε, τόσο το mysqli όσο και το PDO υποστηρίζονται και προτείνονται για νέες υλοποιήσεις. Κάτι που δεν γίνεται εμφανές στο συγκεκριμένο πίνακα είναι το γεγονός ότι το PDO αποτελεί μια γενικευμένη διεπαφή ή οποία διαθέτει οδηγούς για πολλές διαφορετικές βάσεις δεδομένων εκτός από την SQL, προσφέροντας στο χρήστη ένα API υψηλού επιπέδου και επιτρέποντας του να αλλάξει πολύ εύκολα βάση δεδομένων, αν χρειαστεί. Και οι δύο επεκτάσεις έχουν αρκετές δυνατότητες που μας ενδιαφέρει να υλοποιηθούν με σημαντικότερη αυτή των prepared statements.

	ext/mysql	PDO_MySQL	ext/mysql
PHP version introduced	5.0	5.1	2.0
Included with PHP 5.x	Yes	Yes	Yes
Included with PHP 7.x	Yes	Yes	No
Development status	Active	Active	Maintenance only in 5.x; removed in 7.x
Lifecycle	Active	Active	Deprecated in 5.x; removed in 7.x
Recommended for new projects	Yes	Yes	No
OOP Interface	Yes	Yes	No
Procedural Interface	Yes	No	Yes
API supports non-blocking, asynchronous queries with mysqlnd	Yes	No	No
Persistent Connections	Yes	Yes	Yes
API supports Charsets	Yes	Yes	Yes
API supports server-side Prepared Statements	Yes	Yes	No
API supports client-side Prepared Statements	No	Yes	No
API supports Stored Procedures	Yes	Yes	No
API supports Multiple Statements	Yes	Most	No
API supports Transactions	Yes	Yes	No
Transactions can be controlled with SQL	Yes	Yes	Yes
Supports all MySQL 5.1+ functionality	Yes	Most	No

Σχήμα 6.3: Οι δυνατότητες των επίσημων επεκτάσεων της PHP για επικοινωνία με MySQL βάση. Παρατηρούμε ότι mysqli και PDO_MySQL είναι εξίσου καλές επιλογές.

Τα prepared statements επιτρέπουν την εκτέλεση συγκεκριμένων ερωτημάτων (queries) με αντικατάσταση των παραμέτρων που αλλάζουν κάθε φορά. Αυτό έχει δύο σημαντικά πλεονεκτήματα. Αρχικά, αυξάνει ιδιαίτερα την απόδοση στην εκτέλεση διαδοχικών queries βασισμένων στο ίδιο prepared statement. Επιπλέον, βελτιώνει σε μεγάλο βαθμό την ασφάλεια της εφαρμογής μας, καθώς αποκλείει ουσιαστικά την πιθανότητα επίθεσης έκχυσης (injection) κώδικα SQL.

Η επίθεση με SQL injection γίνεται δυνατή λόγω του τρόπου που δημιουργείται το κάθε query. Αν δεν ελεγχθεί σωστά η είσοδος του χρήστη, υπάρχει η περίπτωση ο τελευταίος να εκμεταλλευτεί ειδικούς χαρακτήρες της SQL, π.χ. τον χαρακτήρα που ξεκινάει ένα σχόλιο, και να αλλοιώσει το query, επηρεάζοντας την εκτέλεση και επιτυγχάνοντας, ιδανικά για τον επιτιθέμενο, την πρόσβαση σε ευαίσθητα δεδομένα ή σελίδες.

Ο τρόπος που επιτυγχάνεται η ανοχή στις προαναφερθείσες επιθέσεις βασίζεται στον τρόπο λειτουργίας των prepared statements. Σε πρώτη φάση, δημιουργείται το πρότυπο ερωτήματα δίχως τις παραμέτρους και μεταγλωττίζεται ξεχωριστά. Οι παράμετροι, στέλνονται έπειτα με διαφορετικό πρότυπο επικοινωνίας με αποτέλεσμα να μην είναι δυνατόν να εισαχθούν ειδικοί χαρακτήρες που θα επηρεάσουν την εκτέλεση του query.

Για την υλοποίηση επιλέχθηκε, τελικά, το PDO για τους λόγους επεκτασιμότητας που αναφέρθηκαν παραπάνω δεδομένου ότι στα υπόλοιπα κριτήρια δεν υστερεί σε σχέση με το mysqli, και η υπάρχουσα σχεδίαση μας είναι αντικειμενοστρεφής ούτως ή άλλως, που υποστηρίζεται από το PDO (που δεν υποστηρίζει στυλ δομημένου προγραμματισμού).

6.3.1 Συνοπτική περιγραφή επέκτασης PDO

Η επέκταση PDO (PHP Data Objects), αποτελεί μια μικρού μεγέθους, συνεπή (για τις διαφορετικές βάσεις δεδομένων) διεπαφή για την πρόσβαση και χρήση βάσεων δεδομένων σε PHP. Μέσω των διάφορων, χαμηλού επιπέδου, οδηγιών της επιτρέπει την ενοποίηση του πλήθους των μεθόδων κάθε βάσης σε μια κοινή, πλούσια διεπαφή που περιστρέφεται γύρω από κοινά αντικείμενα που αντιστοιχούν σε κάθε πίνακα ξεχωριστά [Contb].

Η σχεδίαση της συγκεκριμένης επέκτασης έχει γίνει με έμφαση στην ευκολία χρήσης και την επαναχρησιμοποίηση του ίδιου κώδικα για διαφορετικές βάσεις δεδομένων και συναφείς λειτουργίες. Η σύνδεση στην εκάστοτε βάση δεδομένων γίνεται με τη χρήση μιας σειριακής δομής δεδομένων, με όνομα Data Source Name [cont17], έπειτα από την οποία, δημιουργείται ένα αντικείμενο που αντιστοιχεί στη σύνδεση. Τα queries εκτελούνται με τη χρήση της μεθόδου query, εκτός αν χρησιμοποιηθούν prepared statements (PDOStatement), όπου χρησιμοποιούνται οι μέθοδοι prepare και execute. Η επέκταση εμπεριέχει, επίσης, και δική της κλάση εξαιρέσεων (PDOException). Ακολουθούν παραδείγματα απλών λειτουργιών χρησιμοποιώντας τη συγκεκριμένη επέκταση/API.

```
<?php
// Connecting to a MySQL database
$user = "root";
$password = "root";
$charset = "utf8";
$dns = "mysql:host=localhost;dbname=test;charset=$charset";

$db = new PDO($dns, $user, $password);

// Executing a simple query
$val = 42;

$query = "SELECT * FROM foo WHERE bar = " . $db->quote($val);
$result = $db->query($query);

while($row = $result->fetch(PDO::FETCH_ASSOC)) {
    print_r($row);
}

// Sequential executions of a prepared statement
$query = "INSERT INTO REGISTRY (name, value)
VALUES (:name, :value)";
$stmt = $db->prepare($query);
$stmt->bindParam(':name', $name);
$stmt->bindParam(':value', $value);

// First execution
$name = 'one';
$value = 1;
$stmt->execute();

// Second execution with different values
$name = 'two';
$value = 2;
$stmt->execute();
?>
```

Listing 6.2: Παραδείγματα χρήσης PDO για σύνδεσή με βάση εκτέλεση απλών queries και εκτέλεση prepared statement.

Κεφάλαιο 7

Συμπεράσματα

7.1 Καταληκτικές Παρατηρήσεις

Στη συγκεκριμένη εργασία έγινε μια προσπάθεια βελτίωσης του συστήματος αυτόματης αξιολόγησης Grader και προσθήκης δυνατοτήτων που θα το καταστήσουν πιο ευέλικτο και εύκολο στη χρήση του. Για να επιτευχθεί αυτό, διερευνήθηκε ο τρόπος λειτουργίας του όπως και ο τρόπος λειτουργίας άλλων παρόμοιων συστημάτων. Για τις τροποποιήσεις και τις προσθήκες που υλοποιήθηκαν, έγινε προσπάθεια να χρησιμοποιηθούν σωστές σχεδιαστικές επιλογές με έμφαση στη βελτίωση της ποιότητας του κώδικα και στην μετέπειτα ευκολία συντήρησης και επέκτασης του.

Στο Κεφάλαιο 2, παρουσιάστηκαν συστήματα αυτόματης αξιολόγησης ελεύθερου λογισμικού που προσφέρονται για τη διενέργεια διαγωνισμών και λειτουργούν με παρόμοιο τρόπο με το Grader. Αναλύθηκε η σχεδίαση, οι δυνατότητες και ο τρόπος χρήσης τους.

Στο κεφάλαιο 3, είδαμε τον τρόπο λειτουργίας του Grader και του Kewii, της εφαρμογής που τρέχει στον εξυπηρετητή και αναλαμβάνει την εκτέλεση των υποβληθέντων προγραμμάτων. Αναλύθηκε η αρχιτεκτονική και ο τρόπος επικοινωνίας μεταξύ τους. Αναφέρθηκαν, ακόμα, όλα τα βασικά στοιχεία του συστήματος, οι συσχετισμοί τους και η χρήση τους από Kewii και Grader. Το συγκεκριμένο κεφάλαιο έχει τον επιπρόσθετο σκοπό να λειτουργήσει στο μέλλον σαν έγγραφο αναφοράς για όποιον επιθυμεί να επεκτείνει το σύστημα, βοηθώντας τον να αντιληφθεί γρηγορότερα τον σχεδιασμό του.

Στα κεφάλαια 4,5 και 6, αναλύθηκαν οι σημαντικότερες αλλαγές που έγιναν κατά τη διάρκεια της παρούσας εργασίας. Αυτές συμπεριλάμβαναν τη δημιουργία μιας νέας έννοιας για το σύστημα, αυτής των ομάδων αρχείων ελέγχου (και ως αποτέλεσμα την επέκταση ολόκληρου του συστήματος για την υποστήριξη της), την αλλαγή της αρχιτεκτονικής προβλημάτων, διαγωνισμών και υποβολών προς διευκόλυνση της λειτουργίας του Grader και ένα σύνολο άλλων, μικρότερης σημασίας, προσθηκών.

Κατά τη διαδικασία προσθήκης των παραπάνω δυνατοτήτων έγινε προσπάθεια βελτίωσης της εκάστοτε υλοποίησης με αναδιαμόρφωση κλάσεων μεθόδων και δομών δεδομένων ώστε ο κώδικας να είναι περισσότερο κατανοητός και διαχειρίσιμος από τους μελλοντικούς συντηρητές του. Επίσης, καταγράφηκε η διαδικασία εγκατάστασης της συνολικής εφαρμογής του σε ένα καινούριο σύστημα ώστε να είναι εύκολο να μετεγκατασταθεί στο μέλλον ή να δημιουργηθεί περιβάλλον δοκιμών.

7.2 Μελλοντική Εργασία

Υπάρχουν πολλές προσθήκες που θα μπορούσαν να γίνουν σε Grader και Kewii για τη βελτιστοποίηση τους. Όσον αφορά στον πρώτο, θα μπορούσε να προστεθεί η δυνατότητα επισκόπησης και σχολιασμού υποβληθέντων λύσεων (code reviews) όπου οι διαχειριστές ή υπεύθυνοι για τους διαγωνισμούς θα μπορούσαν να σχολιάζουν τις λύσεις των διαγωνιζόμενων και εκεί να γίνεται συζήτηση μεταξύ τους για τυχόν βελτιστοποιήσεις και σφάλματα.

Επιπροσθέτως, θα ήταν αρκετά θετικό αν μπορούσε να διενεργηθεί ένας επανασχεδιασμός της διαδικτυακής εφαρμογής σύμφωνα με μια αρχιτεκτονική Model-View-Controller (MVC) ή παρόμοια με

χρήση κάποιου σύγχρονου πλαισίου (framework) ώστε να γίνει άμεσα πιο κατανοητός ο διαχωρισμός παρουσίας και υλοποίησης της λογικής της αξιολόγησης και των υπόλοιπων λειτουργιών. Κάτι τέτοιο θα είχε ως προϋπόθεση πλήρη κατανόηση του Grader και του συνόλου των μερών του, αλλά θα επιβράβευε άμεσα όσους θα ήταν υπεύθυνοι για την περαιτέρω ανάπτυξη και συντήρηση του.

Μελλοντικά, θα μπορούσε να υπάρξει βελτίωση και του Kewii σε αρκετά θέματα. Αρχικά, όπως έχει αναφερθεί και στο κεφάλαιο 6.1.2 για την Python, οι διαγωνιζόμενοι θα επωφελούνταν στην περίπτωση προσθήκης ενός μηχανισμού για ανάλυση του πηγαιού κώδικα των υποβολών τους, καθώς δεν εκτελείται μεταγλώττιση και όλα τα σφάλματα εμφανίζονται κατά την εκτέλεση. Ένα εργαλείο που θα μπορούσε να χρησιμοποιηθεί για το συγκεκριμένο σκοπό είναι το `pylint`¹. Το συγκεκριμένο πρόγραμμα, έχει τη δυνατότητα τόσο να εντοπίζει σφάλματα πριν την εκτέλεση, όσο και να ελέγχει την ποιότητα του κώδικα σύμφωνα με συγκεκριμένα στάνταρ όπως είναι π.χ. το PEP 8².

Άλλο ένα θέμα που επιδέχεται βελτίωση είναι η ασφάλεια εκτέλεσης του Kewii. Προφανώς, είναι αρκετά δύσκολο να επιτυγχάνεται η απόλυτη ασφάλεια όταν εξ ορισμού εκτελείται άγνωστος κώδικας σε έναν εξυπηρετητή. Παρόλα αυτά, μπορεί να υλοποιηθεί ένα πιο αποκλεισμένο (sandboxed) περιβάλλον, πιθανόν με τη χρήση ενός εικονικού μηχανήματος που να έχει ως στόχο τον αποκλεισμό των εκτελούμενων προγραμμάτων ή με χρήση ειδικευμένου λογισμικού ως container, π.χ. Docker³.

Τέλος, θα ήταν ωφέλιμο για την απόδοση του Kewii να υλοποιηθεί μια παραλληλοποίηση των εκτελέσεων πολλαπλών υποβολών ταυτόχρονα, που θα μείωνε σε μεγάλο βαθμό το χρόνο εκτέλεσης, εάν χρησιμοποιούνταν με χρήση συστάδας εικονικών ή φυσικών μηχανημάτων. Η συγκεκριμένη υλοποίηση έχει ήδη γίνει στο [Ts15] και είναι συμβατή με το σύστημα μας. Μένει να διερευνηθούν οι λεπτομέρειες της προσθήκης και να υλοποιηθεί, με την προϋπόθεση να υπάρχει δυνατότητα πλήρους αξιοποίησης του νέου συστήματος σε θέμα πόρων.

¹ <https://www.pylint.org/>

² <https://www.python.org/dev/peps/pep-0008/>

³ <https://www.docker.com/>

Βιβλιογραφία

- [Conta] PHP Contributors, “Choosing an API”, <https://secure.php.net/manual/en/mysqlinfo.api.choosing.php>. [Online; accessed 6-March-2018].
- [Contb] PHP Contributors, “PHP Data Objects”, [Online; accessed 7-March-2018]. <https://secure.php.net/manual/en/book.pdo.php>.
- [cont17] Wikipedia contributors, “Data source name — Wikipedia, The Free Encyclopedia”, https://en.wikipedia.org/w/index.php?title=Data_source_name&oldid=796073781, 2017. [Online; accessed 17-March-2018].
- [Guo14] Philip Guo, “Python Is Now the Most Popular Introductory Teaching Language at Top U.s. Universities”, <https://cacm.acm.org/blogs/blog-cacm/176450>, July 2014. [Online; accessed 5-March-2018].
- [Hans02] Marit Hansen, Kristian Köhntopp and Andreas Pfitzmann, “The Open Source approach—opportunities and limitations with respect to security and privacy”, *Computers & Security*, vol. 21, no. 5, pp. 461–471, 2002.
- [Harv12] Adam Harvey, “Request for Comments: ext/mysql deprecation”, https://wiki.php.net/rfc/mysql_deprecation, December 2012. [Online; accessed 11-March-2018].
- [Leal03] José Paulo Leal and Fernando Silva, “Mooshak: A Web-based multi-site programming contest system”, *Software: Practice and Experience*, vol. 33, no. 6, pp. 567–581, 2003.
- [Leal08] José Paulo Leal and Fernando Silva, “Using Mooshak as a competitive learning tool”, in *The 2008 Competitive Learning Symposium, ACM-ICPC World Finals*, 2008.
- [Magg12] Stefano Maggiolo and Giovanni Mascellani, “Introducing CMS: A Contest Management System.”, *Olympiads in Informatics*, vol. 6, 2012.
- [MAGG14] Stefano MAGGILOLO, Giovanni MASCELLANI and Luca WEHRSTEDT, “CMS: a Growing Grading System.”, *Olympiads in Informatics*, vol. 8, 2014.
- [Mart09] Robert C Martin, *Clean code: a handbook of agile software craftsmanship*, Pearson Education, 2009.
- [Ribe08] Pedro Ribeiro and Pedro Guerreiro, “Early introduction of competitive programming”, *Olympiads in Informatics*, vol. 2, pp. 149–162, 2008.
- [Robi17a] David Robinson, “The Incredible Growth of Python”, <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>, September 2017. [Online; accessed 6-March-2018].
- [Robi17b] David Robinson, “Why is Python Growing So Quickly?”, <https://stackoverflow.blog/2017/09/14/python-growing-quickly/>, September 2017. [Online; accessed 6-March-2018].

- [Τσ15] Νικόλαος Τσιαμήτρος, “Σχεδίαση και Υλοποίηση Καταναμημένου Συστήματος για την Αυτόματη Αξιολόγηση Προγραμματιστικών Ασκήσεων”, Master’s thesis, Εθνικό Μετσόβιο Πολυτεχνείο, Ελλάδα, 2015.
- [Μ04] Рожков М., “Система автоматического тестирования программ и организации соревнований по программированию”, Master’s thesis, FEFU, Russia, 2004.