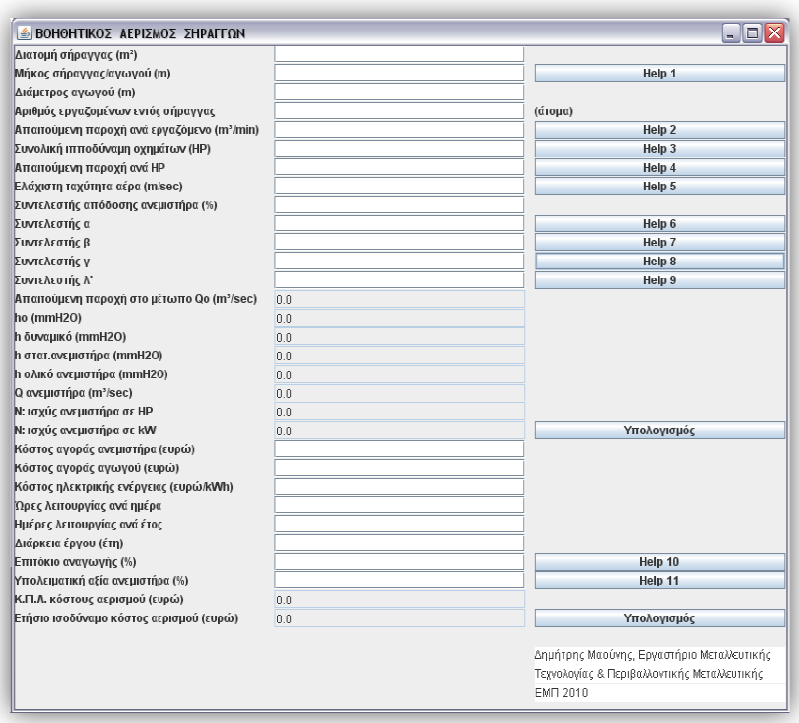
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΜΕΤΑΛΛΕΙΩΝ – ΜΕΤΑΛΛΟΥΡΓΩΝ

ΤΟΜΕΑΣ ΜΕΤΑΛΛΕΥΤΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΥΠΟΛΟΓΙΣΤΙΚΟΥ ΕΡΓΑΛΕΙΟΥ ΓΙΑ ΣΥΣΤΗΜΑΤΑ ΒΟΗΘΗΤΙΚΟΥ ΑΕΡΙΣΜΟΥ ΣΕ ΠΕΡΙΒΑΛΛΟΝ JAVA



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μαούνης Δημήτριος

Επιβλέπων: Δαμίγος Δημήτριος

Επίκουρος Καθηγητής

ΑΘΗΝΑ
ΙΟΥΛΙΟΣ 2010

ΑΘΗΝΑ
ΙΟΥΛΙΟΣ 2010

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ ΜΕΤΑΛΛΕΙΩΝ - ΜΕΤΑΛΛΟΥΡΓΩΝ

ΤΟΜΕΑΣ ΜΕΤΑΛΛΕΥΤΙΚΗΣ

ΑΝΑΠΤΥΞΗ ΥΠΟΛΟΓΙΣΤΙΚΟΥ ΕΡΓΑΛΕΙΟΥ ΓΙΑ ΣΥΣΤΗΜΑΤΑ ΒΟΗΘΗΤΙΚΟΥ ΑΕΡΙΣΜΟΥ ΣΕ ΠΕΡΙΒΑΛΛΟΝ JAVA

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Μαούνης Δημήτριος

Επιβλέπων: Δαμίγος Δημήτριος

Επίκουρος Καθηγητής

Εγκρίθηκε από την τριμελή επιτροπή στις /07/2010

Δαμίγος Δημήτριος, Επίκ. Καθηγητής

Καλιαμπάκος Δημήτριος, Καθηγητής

Μιχαλακόπουλος Θεόδωρος, Επίκ. Καθηγητής

ΑΘΗΝΑ
ΙΟΥΛΙΟΣ 2010

ΠΡΟΛΟΓΟΣ

Ο σκοπός της διπλωματικής εργασίας ‘‘*ΑΝΑΠΤΥΞΗ ΥΠΟΛΟΓΙΣΤΙΚΟΥ ΕΡΓΑΛΕΙΟΥ ΓΙΑ ΣΥΣΤΗΜΑΤΑ ΒΟΗΘΗΤΙΚΟΥ ΑΕΡΙΣΜΟΥ ΣΕ ΠΕΡΙΒΑΛΛΟΝ JAVA*’’ είναι η δημιουργία ενός εύχρηστου εργαλείου για την εκτίμηση τεχνικών παραμέτρων και οικονομικών μεγεθών που αφορούν στο σχεδιασμό κυκλωμάτων βοηθητικού αερισμού.

Η Διπλωματική Εργασία δομείται ως ακολούθως:

- Στο 1^{ο} κεφάλαιο της εργασίας αναφέρονται βασικά στοιχεία για την ποιότητα της ατμόσφαιρας και την ισχύουσα νομοθεσία για τον αερισμό των υπογείων έργων.
- Στο 2^{ο} κεφαλαίο αναφέρονται τα βασικά μεγέθη υπολογισμού κυκλωμάτων αερισμού.
- Στο 3^{ο} κεφαλαίο παρουσιάζονται τα κυκλώματα του βοηθητικού αερισμού σε υπό διάνοιξη υπόγεια έργα και οι μαθηματικοί τύποι για τον υπολογισμό τους.
- Στο 4^{ο} κεφάλαιο γίνεται οικονομική αξιολόγηση κυκλωμάτων βοηθητικού αερισμού.
- Στο 5^{ο} κεφάλαιο παρουσιάζεται η γλώσσα Java (ιστορικό, χαρακτηριστικά), η οποία χρησιμοποιήθηκε για τη δημιουργία του προγράμματος, η δομή του προγράμματος, οι δυνατότητες και η χρήση του.
- Στο 6^{ο} κεφάλαιο, η εργασία ολοκληρώνεται με τα συμπεράσματα.

Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον Επίκουρο Καθηγητή Δημήτριο Δαμίγο για την πολύτιμη συνεργασία του. Τη φοιτήτρια της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών του Ε.Μ.Π. Αλκμήνη Σγουρίτσα για τη σημαντική βοήθεια της καθώς επίσης την οικογένεια μου για την ηθική υποστήριξη.

Μαούνης Δημήτριος

Ιούλιος 2010

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια παρατηρείται μία άνθιση στην κατασκευή υπογείων έργων για την εξυπηρέτηση διαφόρων έργων υποδομής, καθώς η αύξηση του πληθυσμού και η έλλειψη χώρου έχουν οδηγήσει στην εκμετάλλευση της τρίτης διάστασης του αστικού και όχι μόνο χώρου. Τα υπόγεια έργα αποτελούν σήμερα αναπόσπαστο κομμάτι της καθημερινότητας στις ανεπτυγμένες χώρες, ενώ οι υπηρεσίες και το εύρος των εφαρμογών που προσφέρουν συνεχίζουν να αυξάνουν, προσπαθώντας να δώσουν λύσεις στις σύγχρονες ανάγκες.

Απόρροια των αναγκών αλλά και των τεχνολογικών εξελίξεων είναι η κατασκευή μεγαλύτερων και πιο σύνθετων υπόγειων έργων. Η χρήση εκρηκτικών υλών για την εξόρυξη του πετρώματος, ντιζελοκίνητου μηχανολογικού εξοπλισμού φόρτωσης και μεταφοράς, κ.λπ., δημιουργεί μεγαλύτερες απαιτήσεις ως προς τον αερισμό των μετώπων εργασίας με τη βοήθεια κυκλωμάτων βοηθητικού αερισμού (auxiliary ventilation). Κύριος σκοπός του αερισμού είναι η αραίωση και απομάκρυνση διαφόρων ουσιών που βρίσκονται στην ατμόσφαιρα των υπογείων έργων και η σωστή λειτουργία του αερισμού αποτελεί παράγοντα κρίσιμης σημασίας για την αποδοτική και ασφαλή λειτουργία του έργου, καθώς επίσης και για την υγιεινή των εργαζομένων.

Ο σκοπός της διπλωματικής εργασίας είναι η ανάπτυξη ενός εύχρηστου εργαλείου για την εκτίμηση τεχνικών παραμέτρων και οικονομικών μεγεθών που αφορούν στο σχεδιασμό κυκλωμάτων βοηθητικού αερισμού (π.χ. επιλογή κατάλληλου ανεμιστήρα, αγωγού, κ.ά.). Η υλοποίηση του εργαλείου αυτού γίνεται με τη χρήση της γλώσσας προγραμματισμού Java. Η Java είναι μία αξιόπιστη και ασφαλής γλώσσα που παρέχει πολλές προγραμματιστικές δυνατότητες. Στην εργασία αυτή χρησιμοποιήθηκε κυρίως η δυνατότητα δημιουργίας γραφικού περιβάλλοντος της Java, που καθιστά ευκολότερη και πιο άμεση τη χρήση του προγράμματος.

ABSTRACT

In recent years, the world has being experiencing a boom in underground infrastructure projects. Underground space can provide a number of solutions towards making a human-friendly urban environment. Nowadays, the construction of major utility projects such as metros and road tunnels, underground storage complexes, etc., has become an everyday practice. Nevertheless, this progress has extended the needs for more cost-effective construction methods. Towards this direction, modern construction techniques involve the use of explosive or mechanical excavation and heavy diesel loading and hauling equipment, which require huge amounts of fresh air. The latter is supplied by auxiliary ventilation systems. The basic objective of underground ventilation systems is to provide airflow in sufficient quantity and quality to dilute contaminants to safe concentrations in all parts of the facility where personnel are required to work. The overall requirement is to offer an environment that is safe and reasonably comfort.

The aim of this thesis is the development of a user-friendly tool by means of Java programming language for the estimation of technical and economic parameters regarding the design of auxiliary ventilation systems (e.g. fan parameters, ducts characteristics, etc.). Java was selected for its reliability, safety and flexibility, as well as its ability to create graphic environment, which facilitates the use of the program.

ΠΕΡΙΕΧΟΜΕΝΑ

1. Αερισμός των υπογείων έργων…….. 1
   1.1. Γενικά……..... 1
   1.2. Βασικά στοιχεία για την ποιότητα της ατμόσφαιρας των υπογείων έργων……... 2
      1.2.1. Οξυγόνο……..... 2
      1.2.2. Διοξείδιο του άνθρακα CO_2……..... 3
      1.2.3. Μεθάνιο……... 5
      1.2.4. Υδρόθειο……. 6
      1.2.5. Οξείδια του Αζώτου……..... 7
      1.2.6. Μονοξείδιο του Άνθρακα…… 8
      1.2.7. Άλλα αέρια…….. 10
      1.2.8. Κονιορτοί των υπόγειων τεχνικών έργων……... 10
   1.3. Νομοθεσία……. 12
      1.3.1. Υφιστάμενα όρια και προδιαγραφές…… 12
         1.3.1.1. Υφιστάμενα όρια και προδιαγραφές ως προς την παροχή και την ταχύτητα του αέρα……... 13
2. Βασικά μεγέθη υπολογισμού κυκλωμάτων αερισμού……... 15
   2.1. Βασικά στοιχεία ροής του αέρα - Μορφές πίεσης……..... 15
      2.1.1. Γενικά…….. 15
   2.2. Χαρακτηριστικά μεγέθη κυκλώματος αερισμού……..... 19
      2.2.1. Απώλειες στατικής πίεσης……... 19
      2.2.2. Αντίσταση αγωγού……... 22
   2.3. Χαρακτηριστική καμπύλη υπόγειου έργου……..... 25
   2.4. Αερισμός των υπογείων έργων με ανεμιστήρες…… 27
      2.4.1. Εισαγωγή του ανεμιστήρα στο σύστημα αερισμού-Σημείο λειτουργία….. 27
      2.4.2. Ανεμιστήρες σε σειρά…….. 29
      2.4.3. Ανεμιστήρες σε παράλληλη διάταξη……... 30
   2.5. Ενέργεια του αερισμού……... 31
3. Υπολογισμός του κυκλώματος του αερισμού σε υπό διάνοιξη υπόγεια έργα……... 32
   3.1. Σκοπός του βοηθητικού αερισμού – Γενικά……... 32
   3.2. Διαμορφώσεις βοηθητικού αερισμού – Γενικά…….. 32
      3.2.1. Φυσητικός αερισμός…… 32

3.2.2. Μυζητικός αερισμός…… 33
3.2.3. Μεικτός αερισμός…… 34
3.3. Εξοπλισμός βοηθητικού αερισμού……. 35
3.3.1. Ανεμιστήρες……. 35
3.3.2. Αγωγοί……. 37
3.3.3. Χαρακτηριστικά εύκαμπτων αγωγών…….. 37
3.3.4. Χαρακτηριστικά άκαμπτων αγωγών……... 37
3.3.5. Τύποι χρησιμοποιούμενων αγωγών……. 38
3.3.6. Συστήματα αναρτήσεως και σύνδεσης εύκαμπτων φυσητικών αεραγωγών……... 40
3.4. Εκτίμηση των χαρακτηριστικών μεγεθών ενός κυκλώματος βοηθητικού αερισμού……. 41
3.4.1. Εκτίμηση της απαιτούμενης παροχής…….. 42
3.4.2. Εκτίμηση αντιστάσεων και απωλειών πιέσεως στους αγωγούς…….. 43
3.5. Υπολογισμός παροχής και πιέσεως ανεμιστήρα…… 46
4. Οικονομική αξιολόγηση κυκλωμάτων βοηθητικού αερισμού…….. 49
4.1. Γενικά……. 49
4.2. Βασικοί ορισμοί…….. 50
4.2.1. Διαχρονική αξία του χρήματος…… 50
4.2.2. Υπολογισμός μελλοντικής και παρούσας αξίας ενός ποσού……... 51
4.2.3. Υπολογισμός μελλοντικής και παρούσας αξίας περιοδικών χρηματοροών 52
4.2.4. Συντελεστές ανάκτησης και συσσώρευσης κεφαλαίου……... 53
4.3. Τα κριτήρια αξιολόγησης……... 54
4.3.1. Η Καθαρή Παρούσα Αξία……... 54
4.3.2. Το Ετήσιο Ισοδύναμο Κόστος……. 55
4.4. Παράδειγμα υπολογισμού…….. 57
5. Προγραμματισμός σε περιβάλλον Java……. 58
5.1. Η γλώσσας Java…….. 58
5.1.1. Εισαγωγή……. 58
5.1.2. Σύντομο ιστορικό της γλώσσας Java……... 60
5.1.3. Χαρακτηριστικά της Java…… 61
5.1.3.1. Αντικειμενοστραφής προγραμματισμός…….. 61
5.1.3.2. Κληρονομικότητα (Inheritance)…….. 68

5.1.3.3. Χειριστές Εξαιρέσεων (Exceptions Handlers)…… 70
5.1.3.4. Γραφικές Διεπιφάνειες Χρήστη (GUI)…… 72
5.1.3.5. Χρήση Stream (είσοδος και έξοδος)…… 74
5.1.3.6. Δικτύωση……. 75
5.1.3.7. Δημιουργία Ανεξάρτητων Εφαρμογών και Applets…… 76
5.1.3.8. Διαπλατφορμική (Interpreted) Γλώσσα……... 77
5.1.3.9. Συλλογή Απορριμμάτων…….. 77
5.1.3.10. Κατανεμημένη (Distributed)……... 78
5.1.3.11. Ασφάλεια Κώδικα……... 78
5.1.3.12. Ασφάλεια (Secure)…….. 80
5.1.3.13. Multithreaded……... 80
5.1.3.14. Υποστήριξη Multimedia Εφαρμογών…….. 81
5.2. Το πρόγραμμα aux_v1.0…… 81
5.2.1. Περιγραφή του προγράμματος…… 81
5.2.2. Μεταγλώττιση των αρχείων του προγράμματος……. 84
5.2.3. Δημιουργία εκτελέσιμου αρχείου JAR…… 85
5.2.4. Ο κώδικας του προγράμματος……. 86
5.2.4.1. Αρχείο Main.java……. 86
5.2.4.2. Αρχείο Program.java……... 86
5.2.4.3. Αρχείο Help0.java……... 119
5.2.4.4. Αρχείο Isxus.java……. 121
5.2.4.5. Αρχείο Kostos.java…….. 122
5.3. Βασικές οδηγίες χρήσης του προγράμματος aux_v1.0…….. 123
6. Συμπεράσματα……... 126
Βιβλιογραφικές αναφορές……..... 128

ΠΕΡΙΕΧΟΜΕΝΑ ΠΙΝΑΚΩΝ

Πίνακας 1.1.: επιδράσεις στον άνθρωπο λόγω έλλειψης οξυγόνου…… 2
Πίνακας 1.2.: κατά βάρος στοιχειομετρικές αναλογίες αέριων και υγρών καυσίμων….. 3
Πίνακας 1.3.: επιπτώσεις, λόγω αύξησης του διοξειδίου του άνθρακα, στον άνθρωπο.... 4
Πίνακας 4.1.: Υπολογισμός ΚΠΑ κόστους αερισμού……..... 57

ΠΕΡΙΕΧΟΜΕΝΑ ΔΙΑΓΡΑΜΜΑΤΩΝ

Διάγραμμα 2.1.: Χαρακτηριστική καμπύλη υπογείου τεχνικού έργου…….. 26
Διάγραμμα 2.2.: Σημείο λειτουργίας ανεμιστήρα…….. 28
Διάγραμμα 2.3.: Συνδυασμένο αποτέλεσμα δύο διαφορετικών ανεμιστήρων σε σειρά…….. 29
Διάγραμμα 2.4.: Αποτέλεσμα ανεμιστήρων σε παράλληλη διάταξη……. 30
Διάγραμμα 3.5.: Νομόγραμμα……..... 45

ΠΕΡΙΕΧΟΜΕΝΑ ΕΙΚΟΝΩΝ

Εικόνα 3.1.: Φυσητικός αερισμός…… 33
Εικόνα 3.2.: Μυζητικός αερισμός……... 34
Εικόνα 3.3.: Μεικτός αερισμός……... 35
Εικόνα 3.4.: Διαφορετικές κατευθύνσεις αέρα σε αξονικούς ανεμιστήρες…… 36
Εικόνα 5.1.: Παραδείγματα GUI……..... 72
Εικόνα 5.2.: Παράθυρο προγράμματος……. 82
Εικόνα 5.3.: Παράθυρο που περιέχει ένα κείμενο βοήθειας…….. 83
Εικόνα 5.4.: Μήνυμα λάθους σε περίπτωση εισαγωγής γράμματος…….. 83
Εικόνα 5.5.: Μήνυμα λάθους σε περίπτωση εισαγωγής εισαγωγής αρνητικού αριθμού.. 84
Εικόνα 5.6.: Παράδειγμα μεταγλώττισης του αρχείου Main.java…….. 85
Εικόνα 5.7.: Εικονίδιο εκτελέσιμου αρχείου JAR…….. 85

1. ΑΕΡΙΣΜΟΣ ΤΩΝ ΥΠΟΓΕΙΩΝ ΕΡΓΩΝ

1.1. Γενικά

Γενικά, με το σύστημα αερισμού επιδιώκεται ο ποιοτικός έλεγχος, που αφορά στη διατήρηση της σύνθεσης της ατμόσφαιρας, της θερμοκρασίας και της υγρασίας των υπόγείων χώρων μέσα στα επιτρεπτά επίπεδα, ώστε να μην εκτίθενται σε κίνδυνο το προσωπικό και να λειτουργεί αποδοτικά ο μηχανικός εξοπλισμός. Στη βάση των ανωτέρω, γίνεται φανερό ότι τα υπόγεια έργα πρέπει να διαθέτουν ένα αποτελεσματικό σύστημα αερισμού το οποίο να εξασφαλίζει αφενός την αραίωση και απαγωγή των επικίνδυνων αερίων και κονιορτών και αφετέρου τη διατήρηση κατάλληλης θερμοκρασίας και υγρασίας. Για να ικανοποιηθούν οι προϋποθέσεις αυτές το σύστημα του αερισμού πρέπει να παρέχει κάθε στιγμή και σε οποιαδήποτε θέση του υπόγειου χώρου την απαραίτητη για κάθε περίπτωση και εργασία, παροχή αέρα.

Ο καθαρός ξηρός αέρας περιέχει, κατ' όγκο 20,93% Οξυγόνο, 78,10% Άζωτο, 0,03% Διοξείδιο του Άνθρακα και 0,94% Αργό. Θα πρέπει να σημειωθεί όμως ότι ξηρός αέρας δεν υφίσταται στην πράξη, δηλαδή στις συνήθεις ατμόσφαιρες. Η κατάσταση που γενικώς αντιμετωπίζεται αφορά σε υγρό ή «κανονικό» αέρα, ο οποίος είναι μίγμα ξηρού αέρα και υδρατμών σε ποσοστό 0,1 έως 4% κατ' όγκο.

Ο αέρας κατά την κυκλοφορία του σε υπόγειους αποθηκευτικούς χώρους, χάνει από το οξυγόνο του, συνεπεία της αναπνοής των εργαζομένων, της ενδεχόμενης λειτουργίας μηχανών εσωτερικής καύσης, της βραδείας οξείδωσης των ξύλων εντός του χώρου, κ.ά. Το οξυγόνο αυτό αντικαθίσταται κατά μεγάλο ποσοστό από διοξείδιο του άνθρακα. Για παράδειγμα, ο αέρας που εκπνέεται από τον άνθρωπο περιέχει 16,3% οξυγόνο, 79,32% άζωτο και 4,38% διοξείδιο του άνθρακα. Επίσης, σε περιπτώσεις πυρκαγιών, παράγονται σημαντικές ποσότητες διοξειδίου και μονοξειδίου του άνθρακα, οι οποίες πρέπει να απομακρυνθούν από το σύστημα αερισμού. Εάν το σύστημα παρουσιάζει αδυναμία να ανταποκριθεί στο σκοπό του, τότε ο χώρος καθίσταται ανασφαλής.

1.2. Βασικά στοιχεία για την ποιότητα της ατμόσφαιρας των υπογείων έργων

1.2.1. Οξυγόνο

Οι άνθρωποι αναπνέουν άνετα και εργάζονται καλύτερα όταν ο αέρας περιέχει το κανονικό ποσοστό οξυγόνου (21%). Οι παρατηρούμενες επιδράσεις στο ανθρώπινο οργανισμό από την έλλειψη οξυγόνου αναφέρονται στον Πίνακα 1.1. και εξαρτώνται από τον χρόνο έκθεσης του ατόμου (Hartman, 1997).

Πίνακας 1.1.: Επιδράσεις στον άνθρωπο λόγω έλλειψης οξυγόνου.

% περιεκτικότητα O_2 στον αέρα	**Επιδράσεις**
17	Ταχύτερη και βαθύτερη αναπνοή
15	Ταχυπαλμία, ζάλη, βουή στα αυτιά, συχνά και πονοκέφαλο
13	Έντονη ταχυπαλμία, εμετός και υπό ορισμένες συνθήκες απώλεια των αισθήσεων
9	Κυάνωση, κώμα και σπασμοί
7	Κίνδυνος απώλειας της ζωής
6	Θάνατος

Παρ’ όλον ότι απώλεια των αισθήσεων δεν επέρχεται παρά μόνο κάτω του 13% σε οξυγόνο, κανείς δεν πρέπει να εισέρχεται σε περιοχές που έχουν ποσοστά χαμηλότερα από 16,25% (Οικονομόπουλος, 1991). Η έλλειψη οξυγόνου στην ατμόσφαιρα μπορεί να είναι αποτέλεσμα ύπαρξης ενός ή και περισσότερων αερίων, τα οποία παρουσιάζουν υψηλή συγκέντρωση. Ακόμα και μη τοξικά αέρια μπορούν να οδηγήσουν τον ανθρώπινο οργανισμό σε ασφυξία, αν η συγκέντρωσή τους είναι αρκετή για να προκαλέσει έλλειψη οξυγόνου (McPherson, 1993).

Οι μηχανές εσωτερικής καύσης για να μπορέσουν να λειτουργήσουν με τη μέγιστη δυνατή απόδοση, πρέπει να υπάρχει η σωστή αναλογία οξυγόνου και καυσίμου στο θάλαμο καύσης. Ειδικότερα, στις βενζινοκίνητες μηχανές αυξημένη ποσότητα καυσίμου σε σχέση με το περιεχόμενο στο θάλαμο καύσης οξυγόνο, μπορεί να οδηγήσει σε αδυναμία καύσης του μίγματος και, κατ’ επέκταση, σε διακοπή της λειτουργίας του κινητήρα. Μολονότι οι μηχανές ντίζελ έχουν διαφορετική διαδικασία καύσης, και σε αυτή την περίπτωση το μίγμα οξυγόνου – καυσίμου πρέπει να έχει τη σωστή στοιχειομετρία.

Το βάρος και ο όγκος του μίγματος αέρα (επομένως και του περιεχόμενου σε αυτό οξυγόνου) και καυσίμου εξαρτάται από τη χημική σύσταση του καυσίμου. Στον πίνακα που ακολουθεί (Πίνακας 1.2.), δίνονται οι στοιχειομετρικές κατά βάρος αναλογίες για τη βενζίνη και το ντίζελ.

Πίνακας 1.2.: Κατά βάρος στοιχειομετρικές αναλογίες αέριων και υγρών καυσίμων.

Καύσιμο	Χημικός τύπος	Κατά βάρος αναλογία αέρα – καυσίμου
Βενζίνη (l)	C_8H_{18} (οκτάνια)	14.6 : 1
Ελαφρύ ντίζελ (l)	$C_{16}H_{34}$ (οκτάνια)	14.5 : 1

Προκειμένου να πραγματοποιηθεί πλήρης καύση του μίγματος απαιτείται, σύμφωνα με την ακόλουθη αντίδραση, η εξής αναλογία μίγματος ντίζελ - αέρα:

$$C_{16}H_{34} + 25(O_2 + 3{,}76N_2) \rightarrow 16CO_2 + 17H_2O + 94N_2 + ½\ O_2$$

Δεδομένου ότι 1 mol αέρα περιέχει 0,2095 moles O_2 (σύμφωνα με την κ.ό. περιεκτικότητα του αέρα σε O_2) είναι προφανές ότι σε περίπτωση μείωσης του O_2, απαιτείται ανάλογη αύξηση της ποσότητας του εισερχόμενου στο θάλαμο καύσης αέρα. Σε αντίθετη περίπτωση, η διαδικασία της καύσης δεν λαμβάνει χώρα.

1.2.2. Διοξείδιο του άνθρακα CO_2

Το αυξημένο ποσοστό του CO_2 στην ατμόσφαιρα εγκυμονεί κινδύνους για το προσωπικό, καθώς επιδρά στο αναπνευστικό και στο κεντρικό νευρικό σύστημα. Η διαλυτότητα του CO_2 είναι είκοσι φορές μεγαλύτερη από αυτή του οξυγόνου. Η διάχυση του στο αίμα είναι ταχύτατη και οι επιδράσεις στο αναπνευστικό σύστημα γρήγορα γίνονται εμφανείς. Ποσοστό 0,5% CO_2 στον αέρα προκαλεί την ανάγκη λίγο βαθύτερης και ταχύτερης αναπνοής, σε σχέση με την συνηθισμένη (McPherson, 1993). Αύξηση του ποσοστού σε 2%, 3% ή 5% προκαλεί αντίστοιχη αύξηση του απαιτούμενου αέρα κατά 50% 100% και 300%. Οι φυσιολογικές επιδράσεις του διοξειδίου του άνθρακα δίνονται στον Πίνακα1.3.

Πίνακας 1.3.: Επιπτώσεις, λόγω αύξησης του διοξειδίου του άνθρακα, στον άνθρωπο.

% περιεκτικότητα CO_2 στον αέρα	**Επιδράσεις**
0,03	Καμία, κανονική συγκέντρωση ατμόσφαιρας
0,5	Αύξηση της αναπνοής κατά 5%
2,0	Αύξηση της αναπνοής κατά 50%
3,0	Αύξηση της αναπνοής κατά 100%
5 – 10	Πονοκέφαλος, βίαιοι πόνοι
10 – 15	Ανυπόφοροι πόνοι, έντονοι πονοκέφαλοι, γρήγορες αναπνοές, θάνατος

Πηγή: Strang and MacKenzie-Wood, 1985

Οι παραπάνω επιδράσεις ισχύουν υπό την προϋπόθεση ότι το οξυγόνο παραμένει περίπου στα κανονικά όρια και ότι ο εργαζόμενος δεν πραγματοποιεί επίπονη εργασία. Σε αντίθετη περίπτωση τα συμπτώματα είναι πιο έντονα και φυσικά πιο επικίνδυνα, ακόμη και σε μικρότερες συγκεντρώσεις CO_2 (Οικονομόπουλος, 1991).

Η ύπαρξη CO_2 στην ατμόσφαιρα προκαλεί ενανθράκωση του σκυροδέματος. Η ενανθράκωση του σκυροδέματος δημιουργείται από την αντίδραση του CO_2 της ατμόσφαιρας με το υδροξείδιο του ασβεστίου του σκυροδέματος. Η αντίδραση αυτή πραγματοποιείται σε τρία στάδια.

Το πρώτο στάδιο αφορά στη διάχυση του CO_2 στους πόρους του σκυροδέματος.

Στο δεύτερο στάδιο πραγματοποιείται η αντίδραση του CO_2 με το νερό :

$$H_2O + CO_2 \rightarrow 2H^- + CO_3^{2-}$$

Στο τρίτο στάδιο, το υδροξείδιο του ασβεστίου αντιδρά με το διαλυμένο CO_2, ως ακολούθως:

$$Ca(OH)_2 + 2H^+ + CO_3^{2-} \rightarrow CaCO_3 + 2H_2O$$

Η ενανθράκωση του σκυροδέματος δεν μπορεί να πραγματοποιηθεί σε ξηρό ή σε κορεσμένο με νερό σκυρόδεμα. Οι πλέον ευνοϊκές συνθήκες για την ενανθράκωση είναι όταν η σχετική υγρασία είναι 60 – 80%. Η ταχύτητα της ενανθράκωσης εξαρτάται από το πορώδες, την κατανομή των πόρων, το ποσό του ελεύθερου υδροξειδίου του ασβεστίου, την ύπαρξη ρωγμών και την μερική πίεση του CO_2 στην επιφάνεια του σκυροδέματος. Η όλη διαδικασία επιταχύνεται με αύξηση της θερμοκρασίας.

Το φαινόμενο της ενανθράκωσης αρχίζει από την επιφάνεια του σκυροδέματος και προχωρεί προς το εσωτερικό του υπό την μορφή μετώπου. Η ενανθράκωση εμφανίζει θετικές και αρνητικές επιπτώσεις. Ως προς τις θετικές, αναφέρεται ότι θλιπτική και η καμπτική αντοχή του σκυροδέματος αυξάνει για σκυροδέματα από πολυθεϊκά τσιμέντα.

Επίσης, η ενανθράκωση μειώνει το πορώδες του σκυροδέματος, φαινόμενο το οποίο πιθανώς να οφείλεται στην εναπόθεση των προϊόντων της ενανθράκωσης στα τοιχώματα των τριχοειδών πόρων.

Από την άλλη πλευρά όμως, η ενανθράκωση συντελεί στην απόθεση αλάτων, η οποία γίνεται εμφανής με τη δημιουργία λευκών κηλίδων στην επιφάνεια του σκυροδέματος. Τέτοια φαινόμενα μπορεί να μειώσουν πολύ την αλκαλικότητα του σκυροδέματος λόγω απομάκρυνσης των ουσιών που συντελούν στη διατήρηση των υψηλών τιμών του pH, γεγονός που επιτρέπει την οξείδωση του οπλισμού. Αναγκαίες προϋποθέσεις για τη διάβρωση του οπλισμού είναι η ύπαρξη οξυγόνου και η ύπαρξη σχετικά υψηλής αγωγιμότητας. Όσον αφορά στη διάχυση του O_2, θα πρέπει να θεωρείται δεδομένη εφόσον υφίσταται διάχυση του CO_2. Εξαίρεση στον κανόνα παρατηρείται όταν στη μάζα του σκυροδέματος υπάρχουν χημικές ενώσεις που δεσμεύουν το οξυγόνο (ειδικά πρόσθετα όπως π.χ. $NaNO_2$). Άλλη φθορά που μπορεί να προκληθεί από την ενανθράκωση είναι η άτακτη ρηγμάτωση, η οποία οφείλεται σε μεταγενέστερη αλκαλική αντίδραση τσιμέντου-αδρανών και μπορεί να οδηγήσει σε εκτεταμένες απολεπίσεις του σκυροδέματος.

1.2.3. Μεθάνιο

Το μεθάνιο, αέριο άοσμο, δεν βρίσκεται μόνο στα περισσότερα ανθρακωρυχεία, αλλά και σε μερικά μεταλλικά ή άλλα υπόγεια έργα. Επίσης, ενδέχεται να ανιχνευθεί κατά την όρυξη υπόγειων τεχνικών έργων (π.χ. σηράγγων).

Στα ανθρακωρυχεία το αέριο προέρχεται από το κοίτασμα ή τα περιβάλλοντα πετρώματα, ενώ στα μεταλλεία και σε άλλα υπόγεια τεχνικά έργα ανιχνεύεται, κατά κανόνα, σε περιπτώσεις κατασκευής σε ανθρακούχους σχιστόλιθους ή όταν υπάρχουν περιβάλλοντα ανθρακούχα πετρώματα ή ακόμη και ανθρακοφόρα στρώματα.

Η έκλυση μεθανίου μπορεί να είναι μικρή και σταθερή, ή ακαριαία, οπότε απελευθερώνονται τεράστιες ποσότητες αερίου που κυμαίνονται από πλευράς μεγέθους μέσα σε ευρεία όρια και έχουν άμεση ή έμμεση εξάρτηση με τη διαπερατότητα των γεωλογικών σχηματισμών, την πίεση του αερίου και την παρουσία τεκτονικών ανωμαλιών, κλπ.

Η εισπνοή μεθανίου δεν προκαλεί επιβλαβείς αντιδράσεις στον ανθρώπινο οργανισμό, εάν όμως περιέχεται σε μεγάλα ποσοστά μπορεί να προκαλέσει ασφυξία, λόγω ελαττώσεως της περιεκτικότητας του οξυγόνου στην ατμόσφαιρα του υπόγειου έργου.

Το μεθάνιο καίγεται κατά την αντίδραση $CH_4 + 2O_2 = CO_2 + 2H_2O$, σε αναλογία δε με τον αέρα 5 – 15% (κατ' όγκο) και υπό την προϋπόθεση ότι το ποσοστό του οξυγόνου είναι μεγαλύτερο του 12,1%, εκρήγνυται, εφόσον υποστεί έναυση, με καταστρεπτικά συνήθως αποτελέσματα.

Είναι εξ άλλου γνωστό ότι η ξερή ανθρακόσκονη, που αιωρείται στον αέρα, είναι κατά κανόνα εξαιρετικά εκρηκτική. Στην περίπτωση κατά την οποία στην ατμόσφαιρα του υπόγειου έργου συνυπάρχει μεθάνιο, έστω και σε ποσοστό κατά πολύ χαμηλότερο του κατώτερου εκρηκτικού ορίου (5%), η έκρηξη της ανθρακόσκονης μεταδίδεται ταχύτερα.

Από τα προηγούμενα σαφώς προκύπτει ότι το μεθάνιο αποτελεί παράγοντα πρωταρχικής σημασίας για την ασφάλεια των υπογείων έργων και συνεπώς θα πρέπει να καταβάλλεται συνεχής προσπάθεια ώστε, με τη βοήθεια του κατάλληλου συστήματος αερισμού, το αέριο να βρίσκεται στις υπόγειες εκσκαφές σε όσο το δυνατόν μικρότερα ποσοστά και οπωσδήποτε κάτω του 0,5%, σε όλα τα κυκλώματα αερισμού.

Ο επικίνδυνος χαρακτήρας του μεθανίου απαιτεί την ανίχνευση και τον προσδιορισμό του στις υπόγειες εργασίες, τόσο υπό κανονικές συνθήκες, όσο και μετά από επισυμβαίνουσες πυρκαγιές ή εκρήξεις.

1.2.4. Υδρόθειο

Το δύσοσμο αυτό αέριο είναι πολύ δηλητηριώδες. Σε ποσοστό 0,005 – 0,010% προκαλεί ορισμένα ελαφρά συμπτώματα (ερεθισμό στα μάτια και στις αναπνευστικές οδούς) μετά από μία ώρα εκθέσεως.

Σε ποσοστό 0,02 έως 0,07% παρατηρείται έντονος ερεθισμός στα μάτια, πονοκέφαλος ναυτία, πόνοι στη μύτη, το λάρυγγα και το στήθος.

Εάν η συγκέντρωση αυξηθεί (0,07 - 0,1%) επέρχεται απώλεια των αισθήσεων, παύση της αναπνοής και θάνατος.

Σε ποσοστό 0,1 έως 0,2% ο θάνατος επισυμβαίνει σε λίγα λεπτά.

Παρ' όλον ότι η έντονη οσμή του αερίου προδίδει την ύπαρξή του, πρέπει να σημειωθεί ότι παρατεινόμενη έκθεση των εργαζομένων και σε χαμηλές ακόμη συγκεντρώσεις υδρόθειου, παραλύει την όσφρηση με επικίνδυνες συνέπειες. Έκθεση επί 15 λεπτά σε ποσοστό 0,01% προκαλεί απώλεια της οσφρήσεως. Αυτό βεβαίως μπορεί να συμβεί και σε χαμηλότερες συγκεντρώσεις του αερίου εάν η έκθεση παραταθεί περισσότερο. Επίσης, σημειώνεται ότι αέρας που περιέχει 4,3 – 46 % H_2S θα εκραγεί αν υποστεί έναυση.

Το υδρόθειο το οποίο προέρχεται κυρίως από την αποσύνθεση των πυριτών, την αποσύνθεση των ξύλων που χρησιμοποιούνται για την υποστήριξη των υπογείων εκσκαφών, την αποσύνθεση εκρηκτικών υλών πλουσίων σε θείο και σπανιότερα εκλύεται μαζί με το μεθάνιο από τα ανθρακοφόρα κοιτάσματα. Πάντως, κατά κανόνα, απαντάται συχνότερα σε μεταλλικά μεταλλεία (θειούχων μεταλλευμάτων, γύψου κλπ.) παρά σε ανθρακωρυχεία.

1.2.5. Οξείδια του Αζώτου

Όπως είναι γνωστό το Άζωτο σχηματίζει διάφορα οξείδια (N_2O, NO, NO_2, N_2O_4, N_2O_3 και N_2O_5) τα οποία, με την εξαίρεση του N_2O είναι τοξικά.

Στην ατμόσφαιρα των υπογείων έργων απαντούν συχνότερα τα οξείδια NO και NO_2 που προέρχονται κυρίως από τα αέρια αποσυνθέσεως των εκρηκτικών υλών και τα καυσαέρια των μηχανών εσωτερικής καύσης.

Μικρές συγκεντρώσεις οξειδίων του Αζώτου προκαλούν ενοχλήσεις στο λάρυγγα και βήχα, ενώ σε μεγαλύτερα ποσοστά μπορεί να προκαλέσουν ερεθισμό των αναπνευστικών οδών, βρογχίτιδα ή και πνευμονία.

Ποσοστό 0,01% NO_2 προκαλεί επικίνδυνη κατάσταση μετά από εισπνοή λίγου χρόνου, ενώ ποσοστό 0,07% είναι θανατηφόρο μετά από περίπου μισή ώρα.

Γενικά, πρέπει να τονιστεί, ότι τα οξείδια του Αζώτου εισπνεόμενα διαλύονται στην υγρασία των πνευμόνων και σχηματίζουν νιτρώδη και νιτρικά οξέα τα οποία μπορεί να προκαλέσουν διαβρώσεις στις αναπνευστικές οδούς, ενώ τα σχετικά συμπτώματα εμφανίζονται με καθυστέρηση. Κατ’ αυτό τον τρόπο, άτομο εκτεθειμένο σε ατμόσφαιρα που περιέχει οξείδια του Αζώτου, είναι δυνατόν να πεθάνει αρκετές ώρες ή και ημέρες αργότερα.

Ειδικότερα, το NO επειδή είναι λιγότερο διαλυτό στο νερό, δεν συγκρατείται από την υγρασία των ανώτερων αναπνευστικών οδών και γίνεται εξαιρετικά επικίνδυνο εφόσον διεισδύσει βαθιά στους πνεύμονες. Επομένως, είναι περισσότερο τοξικό και βλαπτικό σε σύγκριση με το NO_2 σε ίσες συγκεντρώσεις.

Απαιτείται συνεπώς, κατάλληλος αερισμός για την αραίωση και ταχεία απαγωγή των καπνών που προέρχονται από την αποσύνθεση των εκρηκτικών υλών και τη λειτουργία των μηχανών εσωτερικής καύσεως.

1.2.6. Μονοξείδιο του Άνθρακα

Το μονοξείδιο του άνθρακα, αέριο άοσμο, είναι προϊόν ατελούς καύσεως ανθρακούχων υλών. Είναι αέριο εξαιρετικά επικίνδυνο, δεδομένου ότι εισπνεόμενο προκαλεί ελάττωση του οξυγόνου που μεταφέρει στους ιστούς το αίμα.

Όπως είναι γνωστό, η αιμοσφαιρίνη αποτελεί το κυριότερο συστατικό των ερυθρών αιμοσφαιρίων και επιτελεί σπουδαιότατες λειτουργίες στον οργανισμό. Μέσα στα τριχοειδή των πνευμονικών κυψελίδων συνδέει το Οξυγόνο του εισπνεόμενου αέρα – οπότε παράγεται οξυαιμοσφαιρίνη (HbO_2) - το οποίο μεταφέρει μέχρι τους ιστούς, όπου και το αποδίδει μετά από διάσπαση. Η αιμοσφαιρίνη που παράγεται από τη διάσπαση αυτή, συνδέει το ανθρακικό οξύ των ιστών και κατά την επιστροφή της στους πνεύμονες το αποβάλλει με τον εκπνεόμενο αέρα ως διοξείδιο του άνθρακα μετά από νέα διάσπαση.

Η αιμοσφαιρίνη ενώνεται με το μονοξείδιο του άνθρακα 200 έως 300 φορές ευκολότερα από ότι με το οξυγόνο και σχηματίζει με αυτό (το CO) την ανθρακοξυαιμοσφαιρίνη (HbCO), ένωση στην οποία δεν μπορεί πλέον να προστεθεί οξυγόνο. Κατ’ αυτό τον τρόπο η αιμοσφαιρίνη δεν είναι σε θέση να εκπληρώσει τον φυσιολογικό της προορισμό, με αποτέλεσμα την πρόκληση εσωτερικής ασφυξίας και φυσικά το θάνατο του ατόμου που εισπνέει το μονοξείδιο του άνθρακα.

Η μεγάλη, συνεπώς, συγγένεια της αιμοσφαιρίνης με το μονοξείδιο του άνθρακα, είναι και η αιτία του κινδύνου που προκαλείται από την εισπνοή του αερίου αυτού.

Το ποσοστό του CO που περιέχεται στο αίμα ή, με άλλη διατύπωση, το ποσοστό του αίματος που είναι ενωμένο με το CO (HbCO), αντί του O_2 (HbO_2) λέγεται συνήθως «ποσοστό κορεσμού» του αίματος.

Σύμφωνα με σχετικές έρευνες, συμπτώματα δηλητηριάσεως από εισπνοή CO μπορεί να εμφανιστούν ακόμη και σε ποσοστό κορεσμού 5%. Οπωσδήποτε όμως, τα πρώτα αισθητά συμπτώματα (κυρίως πονοκέφαλος) είναι περισσότερο έκδηλα σε ποσοστό κορεσμού 20 - 30%, ιδιαίτερα δε σε άτομα που βρίσκονται σε ηρεμία.

Σε ποσοστά 30 – 40%, παρατηρούνται ισχυρός πονοκέφαλος, ζάλη, ναυτία, εμετός.

Απώλεια των αισθήσεων συμβαίνει σε ποσοστό 50% περίπου, ενώ ο θάνατος μπορεί να επέλθει σε ποσοστά 65 - 80%.

Θα πρέπει ακόμα να σημειωθεί ότι, ο αριθμός των συμπτωμάτων ελαττώνεται με την αύξηση της ταχύτητας κορεσμού του αίματος. Με άλλα λόγια, αιφνίδια έκθεση του ατόμου σε υψηλές ποσότητες μονοξειδίου του άνθρακα μπορεί να προκαλέσει την κατάπτωση του, προτού εκδηλωθούν τα συνήθη αρχικά συμπτώματα. Για τον μηχανικό, έχει περισσότερο πρακτική αξία η γνώση των συμπτωμάτων που προέρχονται από την

εισπνοή μονοξειδίου του άνθρακα, σε συνάρτηση με το χρόνο εκθέσεως των εργαζομένων σε διάφορες συγκεντρώσεις του αερίου.

Τονίζεται, ότι μερικά άτομα μπορεί αποκτήσουν σχετική ανθεκτικότητα σε χαμηλές συγκεντρώσεις CO, με αποτέλεσμα να αντέχουν σε μεγαλύτερες ποσότητες αερίου, μετά την παρέλευση όμως ορισμένου χρόνου, τα άτομα αυτά γίνονται περισσότερο ευπαθή στο αέριο. Ενδεικτικά πάντως, αξίζει να σημειωθεί ότι, παραμονή σε ατμόσφαιρά που περιέχει 0,15% CO μπορεί να επιφέρει το θάνατο σε μία περίπου ώρα.

Οι κυριότερες πηγές από τις οποίες μπορεί να προέλθει το μονοξείδιο του Άνθρακα είναι:

- Οι εκρηκτικές ύλες, οι οποίες κατά την αποσύνθεση τους αποδίδουν διάφορα αέρια και κυρίως CO και NO_2. Η αναλογία των αερίων αυτών, εξαρτάται από τη χημική σύνθεση της εκρηκτικής ύλης, από το υλικό περιτυλίγματος των φυσιγγίων, τον τρόπο με τον οποίο διενεργείται η πυροδότηση κλπ.
- Οι μηχανές εσωτερικής καύσεως, στα παραγόμενα καυσαέρια των οποίων περιέχεται CO σε μικρές ή υπολογίσιμες ποσότητες, εξαρτώμενες από το είδος του κινητήρα, τα ο σύστημα εγχύσεως του καυσίμου, το είδος και την ποιότητα του καυσίμου, τις συνθήκες χρησιμοποιήσεως του μηχανήματος, της καταστάσεώς του από πλευράς συντηρήσεως κλπ.
- Οι εκρήξεις μειγμάτων αέρα-μεθανίου ή ανθρακόσκονης, οι οποίες μπορεί να προκαλέσουν την παραγωγή σημαντικών ποσοτήτων CO. Τα ποσοστά του αερίου ποικίλουν ανάλογα με τις συνθήκες αλλά οπωσδήποτε οι εκρήξεις ανθρακόσκονης έχουν ως αποτέλεσμα την παραγωγή μεγαλύτερων ποσοτήτων CO σε σύγκριση με τις εκρήξεις μεθανίου.
- Οι πυρκαγιές.
- Η οξείδωση του άνθρακα κατά την εκμετάλλευση των ανθρακωρυχείων, ο βαθμός της οποίας εξαρτάται από πολλούς παράγοντες, όπως η σύνθεση και η γεωλογική ηλικία του άνθρακα, τα γενικά χαρακτηριστικά του κοιτάσματος και των περιβαλλοντικών πετρωμάτων, το μέγεθος της επιφάνειας του άνθρακα που έρχεται σε επαφή με τον αέρα, το σύστημα αερισμού του μεταλλείου, οι συνθήκες που επικρατούν από πλευράς θερμοκρασίας και υγρασίας στις υπόγειες εκσκαφές, η μέθοδος εκμεταλλεύσεως που εφαρμόζεται κλπ.

Σημειώνεται ακόμη, ότι ο αέρας που περιέχει 12,5 έως 74% CO θα εκραγεί εάν υποστεί έναυση.

1.2.7. Άλλα αέρια

Άλλα αέρια που απαντούν σπανιότερα στην ατμόσφαιρα των υπόγειων τεχνικών έργων, είναι το διοξείδιο του θείου, το υδρογόνο, ορισμένοι υδρογονάνθρακες της σειράς του μεθανίου (όπως αιθάνιο, προπάνιο και βουτάνιο) και αλδεΰδες.

Ειδικότερα, το SO_2, βρίσκεται μέσα στη ζώνη κατακρημνίσεων κατά τις εκμεταλλεύσεις θειούχων κοιτασμάτων και είναι πολύ δηλητηριώδες σε μικρές ακόμη συγκεντρώσεις, πλην όμως ερεθίζει τους οφθαλμούς και τις αναπνευστικές οδούς σε τέτοιο βαθμό, ώστε να μη γίνεται ανεκτό ακόμα και σε ποσοστά που δεν είναι αμέσως επικίνδυνα.

Οι αλδεΰδες, προέρχονται κυρίως από τα καυσαέρια των μηχανών εσωτερικής καύσεως, μπορεί όμως να παραχθούν και κατά την απόσταξη ανθράκων που επιτελείται κατά τη διάρκεια πυρκαγιών σε ανθρακωρυχεία.

Ο ανθρώπινος οργανισμός είναι ιδιαιτέρα ευαίσθητος και σε μικρές ακόμα συγκεντρώσεις αλδεϋδών (οσμή ιδιάζουσα και ερεθισμός των αναπνευστικών και των οφθαλμών, ιδιαίτερα από τη φορμαλδεΰδη).

Δεδομένου ότι οι αλδεΰδες αποτελούν μερικώς οξειδωμένους υδρογονάνθρακες, η συγκέντρωση τους στα καυσαέρια συνδέεται αμέσως με το ποσοστό του καυσίμου που παραμένει άκαυστο. Το είδος και το ποσοστό των αλδεϋδών στα καυσαέρια δεν είναι δυνατόν να προκαθοριστεί, εξαρτάται δε σημαντικά και από τον τύπο της μηχανής εσωτερικής καύσης που χρησιμοποιείται.

Τα αποτελέσματα της παρατεταμένης εκθέσεως ατόμου σε ατμόσφαιρα περιχέουσα συγκεντρώσεις αλδεϋδών δεν είναι πάντοτε αρκετά σαφή.

1.2.8. Κονιορτοί των υπόγειων τεχνικών έργων

Από την εισπνοή ορισμένων κονιορτών είναι δυνατόν , υπό ορισμένες συνθήκες, να προκληθούν βαρύτατες ασθένειες στους εργαζομένους, ενώ από τους κονιορτούς των ανθρακωρυχείων, που είναι γενικά εύφλεκτοι, μπορεί να προκληθούν εκρήξεις με καταστρεπτικές συνήθως συνέπειες.

Από τους κονιορτούς που βρίσκονται στην ατμόσφαιρα των υπόγειων τεχνικών έργων, προέρχονται δυο είδη κινδύνων, διαφορετικής φύσεως, οι οποίοι μπορεί να εμφανίζονται χωριστά ή και συγχρόνως.

Η ταξινόμηση των κονιορτών είναι γενικά δύσκολη επειδή παρουσιάζουν μεγάλη ποικιλία και οι επιπτώσεις τους στον ανθρώπινο οργανισμό είναι πολλές και διάφορες.

Για τους λόγους αυτούς, στη σχετική βιβλιογραφία αναφέρονται, τόσο τα επί του ανθρωπίνου οργανισμού αποτελέσματα, όσο και οι εκρηκτικές ιδιότητες των κονιορτών.

Διακρίνονται:

- Κονιορτοί επιβλαβείς για το αναπνευστικό σύστημα. Πρόκειται κυρίως για τις σκόνες που προέρχονται από το ελεύθερο διοξείδιο του Πυριτίου, τα πυριτικά ορυκτά, τα μεταλλεύματα του Βηρυλλίου, του Κασσίτερου, του Σιδήρου (τουλάχιστον ορισμένα από αυτά), του Άνθρακα (κυρίως του ανθρακίτη και των πισσούχων ανθράκων) κλπ.
- Κονιορτοί που μπορεί να προκαλέσουν καρκίνο. Αφορούν κυρίως τα θυγατρικά προϊόντα από τη διάσπαση του ραδονίου, τον αμίαντο και τα μεταλλεύματα του Αρσενικού.
- Κονιορτοί τοξικοί (δηλητηριώδεις για ορισμένα όργανα και ιστούς του σώματος). Πρόκειται για τις σκόνες που προέρχονται από μεταλλεύματα του Βηρυλλίου, Αρσενικού, Μόλυβδου, Ουρανίου, Χρωμίου, Βαναδίου, Καδμίου, Αντιμονίου, Μαγγανίου, Νικελίου, Αργύρου, κλπ.
- Κονιορτοί ραδιενεργοί (επιβλαβείς λόγω των εκπεμπόμενων ακτινοβολιών α και β). Αφορούν τις σκόνες που προέρχονται από τα μεταλλεύματα Ουρανίου, Ραδίου και Θορίου.
- Κονιορτοί εκρηκτικοί. Πρόκειται για τις σκόνες του άνθρακα, όπως επίσης ορισμένων μετάλλων (Mg, Al, Zn, Sn) και θειούχων μεταλλευμάτων.
- Κονιορτοί ενοχλητικοί (με μικρή επιβλαβή επίδραση επί του οργανισμού). Αφορούν κυρίως τις σκόνες που προέρχονται από τον ασβεστόλιθο, το γύψο, την καολίνη κλπ.
- Κονιορτοί αδρανείς (χωρίς επιβλαβή επίδραση επί του οργανισμού). Καμία σκόνη δεν μπορεί με βεβαιότητα να υπαχθεί στην κατηγορία αυτή.

1.3. Νομοθεσία

1.3.1. Υφιστάμενα όρια και προδιαγραφές

Σύμφωνα με το Π.Δ. 225/1989 και τον Κ.Μ.Λ.Ε., η περιεκτικότητα του αέρα σε O_2, σε όλους τους υπόγειους χώρους εργασίας, δεν πρέπει να είναι μικρότερη από 19,5% κ.ό.

Με βάση το Π.Δ. 90/1999, αναφορικά με τις οριακές τιμές έκθεσης και τις ανώτατες οριακές τιμές έκθεσης των εργαζομένων σε χημικούς παράγοντες κατά τη διάρκεια της εργασίας τους, ισχύουν τα ακόλουθα:

- Όσον αφορά στο διοξείδιο του άνθρακα τόσο η οριακή όσο και η ανώτατη οριακή τιμή έκθεσης καθορίζεται στα 5000 ppm (0,5%).
- Για το μεθάνιο το κατώτερο εκρηκτικό όριο είναι 5% και δεν είναι επικίνδυνο σε εισπνοή.
- Η οριακή τιμή έκθεσης του υδρόθειου είναι 10 ppm και η ανώτατη 15 ppm.
- Για το μονοξειδίου του αζώτου η οριακή τιμή έκθεσης είναι 25 ppm.
- Του μονοξειδίου του άνθρακα η οριακή τιμή έκθεσης καθορίζεται στα 50 ppm και η ανώτατη τιμή στα 300 ppm.
- Για το διοξείδιο του θείου η οριακή τιμή έκθεσης καθορίζεται στα 2 ppm και η ανώτατη οριακή τιμή έκθεσης στα 5 ppm.
- Όσον αφορά στο προπάνιο και το βουτάνιο η οριακή τιμή έκθεσης είναι 1000 ppm

Ακόμη σύμφωνα με το Προεδρικό Διάταγμα 225/1989, Κεφ. Δ΄, Άρθρο 16, όσον αφορά στην ατμόσφαιρα του υπόγειου χώρου μετά κάθε ανατίναξη, συνιστάται να γίνεται αναρρόφηση από θέση η οποία πρέπει να είναι κοντά στο μέτωπο όπου έγινε η ανατίναξη ώστε να πραγματοποιείται μείωση του κονιορτού και των αερίων της έκρηξης που αιωρούνται. Η αναρρόφηση συνιστάται να επικουρείται με τοπική προσαγωγή αέρα με φορητό σύστημα για τη μη δημιουργία στο μέτωπο εργασίας ζώνης ακινήτου αέρα.

1.3.1.1. Υφιστάμενα όρια και προδιαγραφές ως προς την παροχή και την ταχύτητα του αέρα

Οι προδιαγραφές ως προς την απαιτούμενη παροχή και ταχύτητα (ελάχιστη και μέγιστη) του αέρα, οι οποίες πρέπει να ικανοποιούνται κατά τη διάνοιξη υπογείων έργων, καθορίζονται από το Προεδρικό Διάταγμα 225/1989 περί «Υγιεινής και ασφάλειας στα υπόγεια τεχνικά έργα» (Φ.Ε.Κ. 106/Α/2-5-1989) και τον Κανονισμό Μεταλλευτικών και Λατομικών Εργασιών (ΚΜΛΕ, 1984).

Σύμφωνα με το Π.Δ. 225/1989, πρέπει να ισχύουν τα ακόλουθα:

- Στις θέσεις εργασίας πρέπει να εξασφαλίζεται ποσότητα αέρα τουλάχιστον 200 ft^3/min και εργαζόμενο (5,66 m^3/min και εργαζόμενο ή 94,40 lt/s και εργαζόμενο). Οι παραπάνω ποσότητες αέρα πρέπει να αυξάνονται, κατά ποσότητα τουλάχιστον 2 m^3/min και HP μηχανών εσωτερικής καύσης.
- Η μέση ταχύτητα του ρεύματος αέρα σε κάθε θέση εργασίας και διάδρομο κυκλοφορίας δεν πρέπει να είναι μικρότερη από 0,1 m/sec ούτε να υπερβαίνει τα 6,0 m/s.
- Σε στοές οι οποίες βρίσκονται υπό διάτρηση και στις οποίες γίνεται χρήση εκρηκτικών, πρέπει να εισάγεται στο μέτωπο εργασίας ποσότητα αέρα τουλάχιστον 200 lt/s και m^2 της μεγαλύτερης διατομής της σήραγγας.
- Για τον υπολογισμό των ποσοτήτων αέρα που απαιτούνται κατά περίπτωση έργου πρέπει να λαμβάνονται υπόψη μεταξύ άλλων και τα ακόλουθα:
  - Το τελικό μήκος σήραγγας ή ο τελικός όγκος υπογείων χώρων τα οποία πρέπει να καλυφθούν.
  - Οι θέσεις στις οποίες ενδέχεται ο αέρας να μείνει στάσιμος (αλλαγές διεύθυνσης στοών κατά την οριζόντια ή κατακόρυφη έννοια, τυφλές εκσκαφές) ή να κινηθεί φυσιολογικά (είσοδοι και άλλα ανοίγματα, πηγές).
- Ο σχεδιασμός και η κατασκευή του κυκλώματος αερισμού πρέπει να εξασφαλίζει το διαχωρισμό του εισερχομένου ρεύματος καθαρού αέρα από το εξερχόμενο ρεύμα.

Οι απαιτήσεις σε αερισμό, όπως καθορίζονται από τον Κανονισμό Μεταλλευτικών και Λατομικών Εργασιών, έχουν πολλά κοινά σημεία με αυτές του Προεδρικού Διατάγματος. Οι διαφορές τους εντοπίζονται παρακάτω:

- Ο όγκος του καθαρού αέρα σ' όλες τις θέσεις εργασίας, πρέπει να είναι το λιγότερο, 2 m^3/min και εργαζόμενο και 2,3 m^3/min και HP μηχανών εσωτερικής καύσης, ανεξάρτητα απ' τον ετεροχρονισμό στη λειτουργία των μηχανημάτων.
- Η ταχύτητα του εισερχόμενου ρεύματος αέρα, πρέπει να ρυθμίζεται κατάλληλα, ώστε να συντελεί στην αποτελεσματική ανανέωση του αέρα σ' όλες τις θέσεις εργασίας, στη διατήρηση της θερμοκρασίας και υγρασίας σε επιτρεπόμενα όρια, αλλά και στην αποφυγή δημιουργίας κονιορτού. Η ταχύτητα δεν πρέπει να ξεπερνάει, σε κάθε θέση εργασίας, τα 8m/s.

2. ΒΑΣΙΚΑ ΜΕΓΕΘΗ ΥΠΟΛΟΓΙΣΜΟΥ ΚΥΚΛΩΜΑΤΩΝ ΑΕΡΙΣΜΟΥ

2.1. Βασικά στοιχεία ροής του αέρα - Μορφές πίεσης

2.1.1. Γενικά

Ο αερισμός των υπόγειων χώρων αποτελεί ουσιαστικά εφαρμογή των βασικών αρχών της ρευστομηχανικής στο δίκτυο των υπογείων τεχνικών έργων, που λειτουργούν ως αγωγοί ροής του αέρα. Στον αερισμό των υπόγειων χώρων, όπως άλλωστε και σε άλλα πεδία εφαρμογής των αρχών της μηχανικής των ρευστών, ενδιαφέρουν περισσότερο οι υφιστάμενες διαφορές πιέσεως παρά οι απόλυτες τιμές αυτών. Η ροή στους αγωγούς προκαλείται μόνο όταν κάποια διαφορά πίεσης δημιουργηθεί μεταξύ δύο σημείων του συστήματος. Συνεπώς, ενέργεια η οποία παρέχεται, είτε από φυσική (φυσικός ελκυσμός), είτε από μηχανική πηγή (ανεμιστήρας) και δημιουργεί αυτή τη διαφορά πίεσης, δαπανάται για να υπερνικηθούν οι απώλειες πίεσης κατά τη ροή. Η ενέργεια αυτή διατίθεται με τη μορφή της στατικής πίεσης και οι απώλειες προκαλούν αντίστοιχη μείωση της στατικής πίεσης του αέρα.

Κατά γενική θεώρηση ο αερισμός των υπογείων έργων αποτελεί ένα παράδειγμα μόνιμης ροής, παρόλο που ορισμένοι παράγοντες μπορεί να ανατρέψουν τις συνθήκες που επικρατούν κατά τη ροή αυτή. Τονίζεται ότι παρά του ότι ο αέρας αποτελεί ρευστό συμπιεστό, στα προβλήματα που αντιμετωπίζονται στα υπόγεια έργα μπορεί να θεωρείται ως ρευστό ασυμπίεστο, δεδομένου ότι η μεταβολή της πίεσης σε σχέση με την απόλυτη τιμή της είναι μικρή, οπότε αρκεί να λαμβάνεται σε κάθε περίπτωση ένα μέσο, σταθερό ειδικό βάρος του αέρα. Όπως είναι φανερό, η παραδοχή αυτή απλουστεύει σημαντικά τους σχετικούς υπολογισμούς.

Χαρακτηριστικό παράγοντα της ροής αποτελεί ο αδιάστατος αριθμός του Reynolds, που εκφράζεται ως συνάρτηση της ταχύτητας (ν), της πυκνότητας (ρ) και του ιξώδους (ξ) του ρευστού, ως επίσης και της διαμέτρου D του αγωγού δεδομένου ότι η κρίσιμη ταχύτητα μπορεί να υπολογιστεί σε συνάρτηση με τον αριθμό αυτό, πράγμα που σημαίνει ότι η

τυρβώδης ροή επέρχεται σε ορισμένη τιμή του αριθμού Reynolds. Στους αγωγούς των υπόγειων έργων είναι απαραίτητο να επικρατούν συνθήκες τυρβώδους ροής, για τον καλύτερο διασκορπισμό και απομάκρυνση των επικίνδυνων αερίων και κονιορτών.

Είναι γνωστό ότι η απόλυτη πίεση η οποία δείχνει βαρόμετρο εντός ρέοντος ρευστού καλείται απόλυτη πίεση αυτού. Η διαφορά μεταξύ της πίεσης αυτής και της πίεσης που δείχνει την ίδια στιγμή το βαρόμετρο που βρίσκεται εκτός του ρέοντος ρευστού καλείται υπερπίεση ή υποπίεση ανάλογα με το αλγεβρικό σημείο της διαφοράς.

Η πίεση αυτή, η οποία εισάγεται και στην εξίσωση καταστάσεως του ρευστού καλείται και πίεση στατική P_{Σ} (και είναι η πίεση που δέχονται τα τοιχώματα ενός αγωγού κατά την κίνηση του αέρα μέσα από αυτόν), αλλά είναι πρακτικά αδύνατον να ληφθεί εντελώς ελεύθερη από την επιρροή της ταχύτητα του ρεύματος του αέρα.

Εφ'όσον είναι γνωστή η μέση ταχύτητα, ν, του ρεύματος του αέρα η δυναμική πίεση, P_{Δ}, που οφείλεται στην κίνηση του αέρα και ασκείται κατά τη διεύθυνση της ροής μόνο, υπολογίζεται από τον τύπο:

$$P_{\Delta} = \frac{\delta * \nu^2}{2 * g}$$

όπου: δ = *το ειδικό βάρος του ρευστού*

ν = *η μέση ταχύτητα του ρεύματος του αέρα*

g = *η επιτάχυνση της βαρύτητας.*

Από το άθροισμα της στατικής και της δυναμικής πίεσης υπολογίζεται η ολική πίεση.

Κατά μήκος μιας ροϊκής γραμμής, υπό την προϋπόθεση ότι το ρευστό δεν έχει εσωτερικές τριβές και επικρατούν συνθήκες μόνιμης ροής, οι συνθήκες σε κάθε σημείο του πεδίου ροής παραμένουν αμετάβλητες ως προς το χρόνο, όπως ορίζεται και από τη διαφορική εξίσωση κινήσεως του Euler:

$$\left(\frac{1}{\rho}\right) dp + vdv + gdz = 0$$

όπου: ρ = *η πυκνότητα του ρευστού*

p = *η στατική πίεση*

g = *η επιτάχυνση της βαρύτητας*

z = *το δυναμικό ύψος και*

ν = *η ταχύτητα του ρευστού*

Με ολοκλήρωση της διαφορικής εξίσωσης και υπό την προϋπόθεση ότι το ρευστό είναι ασυμπίεστο, οπότε ρ = σταθερό, προκύπτει η εξίσωση Bernoulli:

$$\frac{p}{\rho} + \frac{v^2}{2} + g * z = c$$

όπου c η σταθερά της ολοκληρώσεως ή σταθερά του Bernoulli.

Σε εφαρμογή της παραπάνω εξίσωσης σε δύο σημεία της ίδιας ροϊκής γραμμής (1 & 2) προκύπτει η εξίσωση:

$$\frac{p}{\rho} + \frac{v_1^2}{2} + g * z_1 = \frac{p}{\rho} + \frac{v_2^2}{2} + g * z_2$$

από την οποία συνεπάγεται ότι η μέση τιμή του συνολικού ύψους ενέργειας διατηρείται σταθερή κατά μήκος αγωγού, εφόσον οι απώλειες μηχανικής ενέργειας είναι μηδενικές. Η παραπάνω εξίσωση μπορεί να ισχύσει με ορισμένες προϋποθέσεις ανεξάρτητα από ροϊκές γραμμές.

Σε εξέταση των ενεργειακών μεταβολών που συμβαίνουν σε θεωρούμενο φυσικό αέριο που ρέει σε δεδομένο αγωγό μεταξύ δύο διατομών (1 & 2) του συστήματος με παράλειψη μεταβολών θερμότητας και μηχανικής ενέργειας από εξωτερική πηγή διατυπώνεται ότι η ολική ενέργεια (ΟΕ) του αέρα στη διατομή 1 ισούται με την ολική ενέργεια του αέρα στη διατομή 2 πλέον τις απώλειες ενέργειας (ΑΕ) μεταξύ των διατομών 1 και 2, δηλαδή:

$$OE_1 = OE_2 + AE_{1_2} \text{ ή } OE_1 - OE_2 = AE_{1_2}$$

Με αμελητέες τις μεταβολές της εσωτερικής ενέργειας, ισχύει:

$$\frac{p}{\rho} + \frac{v_1^2}{2} + g * z_1 = \frac{p}{\rho} + \frac{v_2^2}{2} + g * z_2 + AE_{1_2}$$

$$\text{ή } p_1 + \frac{\delta * v_1^2}{2 * g} + \delta * z_1 = p_2 + \frac{\delta * v_2^2}{2 * g} + \delta * z_2 + AE_{1_2}$$

$$\text{ή } \frac{p_1}{\delta} + \frac{v_1^2}{2 * g} + \delta * z_1 = \frac{p_2}{\delta} + \frac{v_2^2}{2 * g} + z_2 + AE_{1_2}$$

όπου: *δ = το ειδικό βάρος του αέρα*

p = η στατική πίεση

g = η επιτάχυνση της βαρύτητας

z = το δυναμικό ύψος και

ν = η ταχύτητα του ρευστού

Είναι φανερό, ότι οι εξισώσεις αυτές αποτελούν τροποποιημένη μορφή της βασικής εξίσωσης Bernoulli, ώστε αυτή να προσαρμοστεί στις επισυμβαίνουσες απώλειες ενέργειας, οι δε όροι p/ρ, $ν^2/2g$ και g.z, που έχουν διαστάσεις L^2T^{-2} παριστάνουν ειδική στατική, ειδική κινητική και ειδική δυνητική ενέργεια (ενέργεια ανά μονάδα μάζας ρευστού) αντίστοιχα, ενώ ο όρος ΑΕ απώλειες ενέργειας, που μπορούν να εκφραστούν με αυτές τις διαστάσεις.

Συνεπώς, κατά τη ροή ασυμπίεστου ρευστού, με σύγχρονη εμφάνιση τριβών, από τη θέση 1 στη θέση 2, η ταχύτητα $ν^2$ εξαρτάται και από το ποσό ενέργειας που διατέθηκε για την υπερνίκηση των τριβών κατά τη ροή στο διάστημα 1 έως 2.

Δεδομένου ότι οι όροι p/δ, $ν^2/2g$ και z έχουν διαστάσεις μήκους, δηλαδή ύψους στήλης ρευστούς και λαμβάνοντας υπόψη την ισοδυναμία της ειδικής ενέργειας και του ύψους στήλης ρευστού, μπορεί να διατυπωθεί η εξίσωση:

$$H_{\Sigma 1} + H_{\Delta 1} + H_{Z1} = H_{\Sigma 2} + H_{\Delta 2} + H_{Z2} + H_A$$

όπου: H_Σ *= ύψος στήλης ρευστού που αναφέρεται στη στατική ενέργεια*

H_Δ *= ύψος στήλης ρευστού που αναφέρεται στην κινητική ενέργεια*

H_Z *= ύψος στήλης ρευστού που αναφέρεται στη δυνητική ενέργεια*

H_A *= ύψος στήλης ρευστού που αναφέρεται στις απώλειες ενέργειας.*

Στα προβλήματα αερισμού υπογείων χώρων τα ύψη στήλης εκφράζονται σε χιλιοστά της στήλης ύδατος (mmH_2O), ή σε Pascal (Pa) {1mm H_2O=10 Pa}

2.2. Χαρακτηριστικά μεγέθη κυκλώματος αερισμού

2.2.1. Απώλειες στατικής πίεσης

Οι απώλειες στατικής πίεσης (h_Σ) μπορεί να διαιρεθούν σε δύο διαφορετικές ομάδες, ανάλογα με το αίτιο από το οποίο προέρχονται:

- Απώλειες πίεσης εκ τριβών (hτ), προκαλούμενες από την επαφή των ρευμάτων αέρα με τα τοιχώματα των αγωγών.
- Απώλειες πίεσης εκ κρούσεων (hκ), προκαλούμενες από τη μεταβολή της διατομής των αγωγών στους οποίους κινούνται τα ρεύματα του αέρα. Οι απώλειες πίεσης, τόσο εκ τριβών όσο και εκ κρούσεων και επομένως και οι συνολικές στατικές απώλειες ($h_\Sigma = h_\tau + h_\kappa$) εκφράζονται σε Pa ή χιλιοστά στήλης ύδατος (mmH_2O).

α. Απώλειες πίεσης εκ τριβών

Οι απώλειες πίεσης εκ τριβών εξαρτώνται κατά κύριο λόγο από τις διαστάσεις των αγωγών που συνθέτουν το δίκτυο του υπόγειου χώρου, την κατάσταση των τοιχωμάτων τους και την ταχύτητα του ρεύματος του αέρα. Οι απώλειες αυτές είναι κατά κανόνα πολύ μεγαλύτερης σημασίας σε σύγκριση με τις απώλειες εκ κρούσεων και απαιτούν περισσότερη προσοχή και ακρίβεια στους υπολογισμούς. Κατά τη μελέτη των προβλημάτων αερισμού των υπόγειων μεταλλείων οι απώλειες πίεσης εκ τριβών υπολογίζονται από την ακόλουθη εξίσωση:

$$h_\tau = \frac{K * L * \Pi}{S} * \nu^2$$

όπου: *hτ = Απώλειες πίεσης εκ τριβών (Pa ή mmH_2O)*

L =Μήκος του αγωγού (m)

Π = Περίμετρος του αγωγού (m)

S = Διατομή του αγωγού (m^2)

ν = Ταχύτητα του ρεύματος του αέρα (m/s)

K = Συντελεστής τριβής ($k * g * m^{-3}$ ή $N * S^2 * m^{-4}$)

Δεδομένου ότι η παροχή Q(m^3/s) συνδέεται με τα μεγέθη S και ν με τον τύπο

Q = S * ν, εύκολα προκύπτει ότι :

$$h_\tau = \frac{K * L * \Pi}{S^3} * Q^2$$

Ο συντελεστής τριβής K θεωρείται ανεξάρτητος της παροχής και εξαρτάται από το σχήμα της διατομής των αγωγών, το είδος της υποστήριξης αυτών και την κατάσταση των τοιχωμάτων τους. Αντιστοιχεί ουσιαστικά στο συντελεστή τριβής που εφαρμόζεται στον υπολογισμό των αντιστάσεων κατά τη ροή ρευστών σε αγωγούς. Δίνεται από πίνακες (Οικονομόπουλος, 1991, SME, 1998, Hartman, et aI, 1997).

β. Απώλειες πίεσης εκ κρούσεων

Οι απώλειες πίεσης εκ κρούσεων προκαλούνται κατόπιν μεταβολής της επιφάνειας που καταλαμβάνεται από *τα* ρεύματα του αέρα και συνεπώς της σχετικής θέσης των τοιχωμάτων αναφορικά με τις γραμμές ροής. Επομένως, εάν στους αγωγούς του υπόγειου χώρου υφίστανται αλλαγές της διατομής ή της διεύθυνσης, συμβαίνουν απώλειες πίεσης εκ κρούσεων, οι οποίες επιτείνονται εάν παρεμβάλλονται και εμπόδια (εξοπλισμός, υλικά, κ.ά.).

Οι απώλειες αυτές ανέρχονται σε 10 έως 30% των συνολικών απωλειών από τριβές, αλλά ο ακριβής υπολογισμός τους δεν είναι εύκολος λόγω της ποικιλίας των συνθηκών υπό τις οποίες μπορεί να συμβούν. Οι απώλειες εκ κρούσεων είναι ανάλογες με τη δυναμική πίεση της ροής (h_Δ) που αντιστοιχεί στη μέση ταχύτητα του αέρα, καθώς επίσης και με ένα συντελεστή K' που ονομάζεται συντελεστής κρούσης και είναι αδιάστατος αριθμός.

$$h_k = K' h_\Delta = \frac{\delta * \nu^2}{2 * g}$$

όπου: h_κ = *Απώλειες πίεσης εκ κρούσεων (Pa ή mmH_2O)*

K΄ = Συντελεστής κρούσης

ν = Ταχύτητα του ρεύματος του αέρα (m/s)

δ = το ειδικό βάρος του αέρα

g = η επιτάχυνση της βαρύτητας

Προσεγγιστικά, ο υπολογισμός των απωλειών πίεσης εκ κρούσεων μπορεί να γίνει, με κατάλληλη αύξηση των απωλειών εκ τριβής. Για το σκοπό αυτό εισάγεται η έννοια του ισοδύναμου μήκους, δηλαδή του ευθύγραμμου μήκους του δεδομένου αγωγού στο οποίο σημειώνεται πτώση πίεσης ίση προς εκείνη που συμβαίνει λόγω της παρεμβολής εμποδίου στην ελεύθερη ροή του αέρα. Τιμές ισοδύναμων μηκών για χαρακτηριστικές περιπτώσεις δίνονται στη βιβλιογραφία (Οικονομόπουλος, 1991; SME, 1998; Hartman, et al., 1997). Πρέπει να σημειωθεί ότι για τις περιπτώσεις μεταβολής του μεγέθους της διατομής, όπως επίσης και για την είσοδο του αέρα, οι απώλειες εκ κρούσεων πρέπει να αναφέρονται στη διατομή του αγωγού που έπεται της αλλαγής (μεταβολής). Εξαίρεση γίνεται μόνο για την έξοδο του αέρα οπότε οι απώλειες αναφέρονται στη διατομή του αγωγού που προηγείται.
Είναι φανερό ότι με τη χρησιμοποίηση του ισοδύναμου μήκους (Le), οι ολικές απώλειες πίεσης (εκ τριβών και κρούσεων) υπολογίζονται με τη βοήθεια της ακόλουθης εξίσωσης:

$$h_{\Sigma} = h_{\tau} + h_{k} = \frac{K * \Pi * (L + L_e) * Q^2}{S^3}$$

όπου: h_{Σ} = *Απώλειες πίεσης εκ τριβών και κρούσεων (Pa ή mmH$_2$O)*

h_{τ} = *Απώλειες πίεσης εκ τριβών (Pa ή mmH$_2$O)*

h_{κ} = *Απώλειες πίεσης εκ κρούσεων (Pa ή mmH$_2$O)*

L = Μήκος του αγωγού (m)

L΄ = Ισοδύναμο μήκος αγωγού (m)

Π = Περίμετρος του αγωγού (m)

S = Διατομή του αγωγού (m^2)

v = Ταχύτητα του ρεύματος του αέρα (m/s)

K = Συντελεστής τριβής ($k * g * m^{-3}$ ή $N * S^2 * m^{-4}$)

Πρέπει σε αυτό το σημείο να τονιστεί ότι, κατά τον υπολογισμό των απωλειών πίεσης η προσέγγιση που επιτυγχάνεται σε σχέση με την πραγματικότητα δεν είναι πάντοτε ικανοποιητική. Αυτό οφείλεται, είτε σε άστοχη επιλογή της τιμής του συντελεστή, είτε σε αδυναμία ακριβούς επιλογής του, εάν η περίπτωση δεν καλύπτεται από το φάσμα των συνθηκών που αναφέρονται στη βιβλιογραφία.

2.2.2. Αντίσταση αγωγού

Στην τελευταία εξίσωση μπορεί να παρατηρηθεί ότι, για δεδομένο αγωγό, το μέγεθος $\frac{K*\Pi*(L+L_e)}{S^3}$ διατηρείται σταθερό. Αυτό καλείται αντίσταση του αγωγού, R, από την οποία προκύπτει η θεμελιώδης εξίσωση (Murgue, Atkinson) του αερισμού των μεταλλείων:

$$h = R * Q^2$$

Η αντίσταση εκφράζεται σε $k * g * f * s^2 * m^{-8}$ ή $N * s^2 * m^{-8}$. Στη Γαλλική κυρίως βιβλιογραφία η μονάδα αντίστασης καλείται Murgue. Κατά τις εφαρμογές χρησιμοποιείται κυρίως το kilomurgue (KMu) = 1000 Murgues.

Η μονάδα αυτή (kilomurgue) προκύπτει από τον τύπο $h = R * Q^2$ εφόσον τα h και Q εκφράζονται σε kgr/m^2 (ή mm H_2O) και m^3/s αντίστοιχα. Συνεπώς, η αντίσταση αγωγού ισούται προς το 1 kilomurgue εάν κατά τη ροή αέρα παροχής 1 m^3/s σημειώνονται απώλειες πιέσεως 1 mm H_2O.

Όπως θα αναφερθεί και στις επόμενες ενότητες, οι αντιστάσεις των αγωγών, που είναι απαραίτητες κατά τη μελέτη του δικτύου αερισμού των υπογείων έργων, μπορούν να υπολογισθούν με τη βοήθεια μετρήσεων της παροχής και των απωλειών πιέσεως μεταξύ των άκρων τους. Αυτό όμως δεν είναι πάντοτε δυνατόν, όπως συμβαίνει π.χ. στην περίπτωση υπογείων έργων υπό σχεδίαση ή ανάπτυξη, ή σε αγωγούς η όρυξη των οποίων δεν έχει αποπερατωθεί ή διαρρέονται από ρεύμα αέρα εξαιρετικά μικρής ταχύτητας, η οποία δεν είναι δυνατόν να μετρηθεί με την απαραίτητη ακρίβεια.

Υπό αυτές τις συνθήκες αρκεί να είναι γνωστά τα γεωμετρικά χαρακτηριστικά του αγωγού ως και του συντελεστή τριβής, προκειμένου να υπολογιστεί η αντίσταση από το σχετικό τύπο:

$$R = \frac{K * \Pi * (L + L_e)}{S^3}$$

Διευκρινίζεται ότι η παροχή Q είναι δυνατόν να αναφέρεται, είτε στον όγκο του ρέοντος στη μονάδα του χρόνου αέρα (Q_V) οπότε εκφράζεται σε m^3/s, είτε στο βάρος αυτού (Q_β) οπότε εκφράζεται σε kgf/s. προφανώς ισχύει η σχέση $Q_\beta = Qv * \delta$ όπου δ το ειδικό βάρος του αέρα που διαρρέει τον αγωγό. Είναι επίσης δυνατόν, να αναφέρεται σε όγκο αέρα υπό κανονικές συνθήκες (Q_n) δηλαδή 15 ºC και 760 mmHg. Πρόκειται για την παροχή (εκφραζόμενη σε m^3/s) που θα παρετηρείτο εάν ο αέρας βρισκόταν υπό κανονικές

συνθήκες, ενώ βεβαίως η κατά βάρος παροχή Q_β θα παραμένει η ίδια. Εν προκειμένω ισχύουν οι σχέσεις:

$$\frac{Q_\beta}{Q_\nu} = \frac{\delta}{\delta_0}$$

Επομένως, στη βασική σχέση $h = R * Q^2$ είναι δυνατόν να χρησιμοποιηθεί, είτε η πραγματική παροχή Qν , είτε η παροχή Q_β που αντιστοιχεί σε όγκο αέρα υπό κανονικές συνθήκες θερμοκρασίας και πιέσεως.

Κατ' αυτόν τον τρόπο προκύπτουν αντίστοιχα δύο αντιστάσεις, ήτοι:

$$R_\nu = \frac{h}{Q * \nu^2} \; \kappa\alpha\iota, R_n = \frac{h}{Q * n^2}$$

οπότε, με βάση τα προηγούμενα:

$$\frac{R_\nu}{R_n} = \left(\frac{\delta}{\delta_0}\right)^2$$

Η αντίσταση R_n αναφέρεται στη βιβλιογραφία και ως αντίσταση υπό κανονικές συνθήκες ή «κανονική αντίσταση». Πρόκειται για την πραγματική αντίσταση, η οποία έχει αναχθεί σε συνθήκες κανονικής παροχής Q_n. Όπως είναι πρόδηλο, οι αντιστάσεις R_ν και R_n δεν εξαρτώνται μόνο από τα χαρακτηριστικά του αγωγού αλλά και από το ειδικό βάρος του αέρα που τον διαρρέει και συνεπώς χαρακτηρίζουν το ζεύγος αγωγού-ρευστού.

Τέλος, ως ειδική αντίσταση R ή και απλώς αντίσταση χαρακτηρίζεται η αντίσταση την οποία παρουσιάζει αγωγός διαρρεόμενος από κανονικό αέρα, ειδικού βάρους δ_0 = 1,23 kgf/m^3, οπότε η R δεν εξαρτάται παρά μόνο από τα γεωμετρικά χαρακτηριστικά του αγωγού και την κατάσταση των τοιχωμάτων του. Πρόκειται για την αντίσταση που δίνεται στους σχετικούς πίνακες. Είναι φανερό ότι εάν δ= δ_0 είναι και Rν=Rn=R.

Τα προαναφερθέντα έχουν ασφαλώς περισσότερο θεωρητική αξία, δεδομένου ότι στη συνηθισμένη πράξη, εκτός ορισμένων περιπτώσεων, παραβλέπεται η τυχόν διαφοροποίηση του ειδικού βάρους του αέρα σε σύγκριση με το κανονικό.

Για τον υπολογισμό της αντιστάσεως μπορεί επίσης να χρησιμοποιηθεί ο τύπος :

$$R = \lambda * \frac{\delta_0 * \Pi}{8 * g * S^3} \text{ ή } R = 0{,}0156 * \lambda * \frac{L * \Pi}{S^3}$$

όπου : *R = η αντίσταση σε kilomurgues* ($k * g * r * s^2 * m^{-8}$)

λ = Συντελεστής τριβής, χωρίς διαστάσεις

δ_0 = Το κανονικό ειδικό βάρος του αέρα ($1.23\ k * g * f/m^3$)

g = Η επιτάχυνση της βαρύτητας (9,81 m/s^2)

L = Το μήκος του αγωγού (m)

Π: Η περίμετρος του αγωγού (m)

S : Η διατομή του αγωγού (m^2)

Ο συντελεστής τριβής λ, ανάλογος του συντελεστή απωλειών πιέσεως, ο οποίος υπεισέρχεται κατά τη μελέτη της ροής ρευστών δι' αγωγών, εξαρτάται από την κατάσταση των τοιχωμάτων (παρειών, οροφής και δαπέδου), τη μορφή και το μέγεθος τη διατομής, τον τρόπο υποστήριξης και την τυχόν ύπαρξη εμποδίων εντός του αγωγού.
Εφόσον ο αγωγός στηρίζεται από την ειδική του αντίσταση R, οι σχέσεις που συνδέουν την παροχή και τις απώλειες πιέσεως, είναι:

$$h = \left(\frac{\delta}{\delta_0}\right) * R * Q_{\nu^2} \ \kappa\alpha\iota \ \ h = \left(\frac{\delta_0}{\delta}\right) * R * Q_{n^2}$$

Τονίζεται ότι ο υπολογισμός της αντιστάσεως του δεδομένου αγωγού με τη χρησιμοποίηση οποιουδήποτε εκ των δύο τύπων:

$$R = K * \frac{L * \Pi}{S^3} \text{ ή } R = 0{,}0156 * \lambda * \frac{L * \Pi}{S^3}$$

πρέπει να οδηγήσει προφανώς στο ίδιο αποτέλεσμα, υπό την προϋπόθεση ότι υπάρχει ορθή αντιστοιχία στους συντελεστές K και λ που θα επιλεγούν.
Επομένως, και δεδομένου ότι πρόκειται ουσιαστικά για τον ίδιο τύπο θα μπορούσε να γίνει κατάλληλη μετατροπή των συντελεστών που δίνονται στη διεθνή βιβλιογραφία και αν παραταθεί ένας και μόνο τύπος. Εντούτοις, για διαφόρους λόγους, θεωρήθηκε σκόπιμη η αποφυγή μίας τέτοιας διαδικασίας και αντί αυτής προτιμήθηκε η παράθεση των τιμών των συντελεστών όπως αναφέρονται και σε σχετικά συγγράμματα.

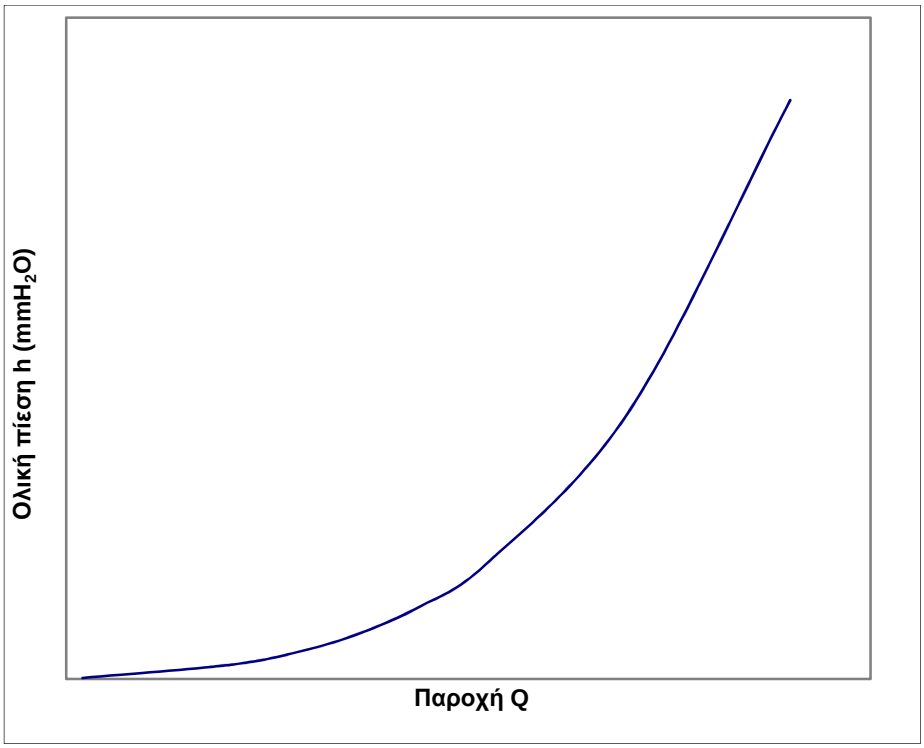
Κατ' αυτό τον τρόπο θα είναι δυνατόν να επιλέγεται, βάσει της περιγραφής των συνθηκών που επικρατούν (κατάσταση τοιχωμάτων, σχήμα και μέγεθος διατομής, ύπαρξη εμποδίων, κλπ), ο συντελεστής που προσιδιάζει καλύτερα στην περίπτωση που εξετάζεται και να ακολουθεί η εφαρμογή του αντίστοιχου τύπου για τον υπολογισμό της αντιστάσεως.

2.3. Χαρακτηριστική καμπύλη υπόγειου έργου

Όπως προκύπτει από την θεμελιώδη εξίσωση του αερισμού, $h = R * Q^2$, οι απώλειες πιέσεως μεταβάλλονται ανάλογα με το τετράγωνο παροχής. Συνεπώς, για δεδομένο σύστημα υφίσταται μια σταθερή αναλογία πιέσεων και παροχών και επομένως εφόσον το μέγεθος h καθοριστεί για συγκεκριμένη παροχή q, είναι εύκολο να προκύψουν οι τιμές οποιουδήποτε ζεύγους Q, h αφού ουσιαστικά η αντίσταση ($R = h * Q^2$) αποτελεί ένα συντελεστή αναλογίας.

Διευκρινίζεται ότι ο όρος «σύστημα» μπορεί να υποδηλώνει αγωγό, τμήμα ή και το σύνολο ενός υπόγειου έργου, υπό την προϋπόθεση ότι το τελευταίο εξυπηρετείται από ένα και μόνο κύριο ανεμιστήρα.

Εφόσον η τιμή της αντιστάσεως υπολογισθεί είτε θεωρητικά, είτε με κατάλληλες μετρήσεις, το θέμα μπορεί να αντιμετωπισθεί γραφικά με τη χάραξη παραβολικής καμπύλης (σε συντεταγμένες Q, h) η οποία, εάν αφορά στο σύνολο του δικτύου, καλείται χαρακτηριστική καμπύλη υπόγειου έργου (Διάγραμμα 2.1.)



Διάγραμμα 2.1.: Χαρακτηριστική καμπύλη υπογείου τεχνικού έργου.

Συνήθως η χάραξη της καμπύλης αυτή γίνεται με τη χρησιμοποίηση της στατικής πίεσης του υπόγειου έργου, είναι όμως συχνά χρήσιμο να αναφέρεται και στην ολική πίεση.

Παρατηρείται εύκολα ότι μπορεί να παραλληλιστεί η εξίσωση $h = R * Q^2$ με το νόμο του Ohm για τους ηλεκτρικούς αγωγούς ($V = R * I$), λόγω της αναλογίας της πτώσεως τάσεως (V) με τις απώλειες πιέσεως (h), της εντάσεως του ρεύματος (I) με την παροχή (Q) και της αντιστάσεως του ηλεκτρικού αγωγού (R) με την αντίσταση του αεραγωγού (R). Με βάση αυτή την αντιστοιχία, η τεχνική που χρησιμοποιείται για τη σύνθεση και την ανάλυση των ηλεκτρικών δικτύων, όπως παρουσιάζεται σε επόμενη ενότητα, μπορεί να εφαρμοσθεί με επιτυχία και στα δίκτυα αερισμού, χωρίς βεβαίως να παραβλέπεται ο γραμμικός χαρακτήρας της εξισώσεως του νόμου του Ohm, σε σύγκριση με την εξίσωση του αερισμού, που περιέχει όρο στη δεύτερη δύναμη.

2.4. Αερισμός των υπογείων έργων με ανεμιστήρες

2.4.1. Εισαγωγή του ανεμιστήρα στο σύστημα αερισμού-Σημείο λειτουργίας

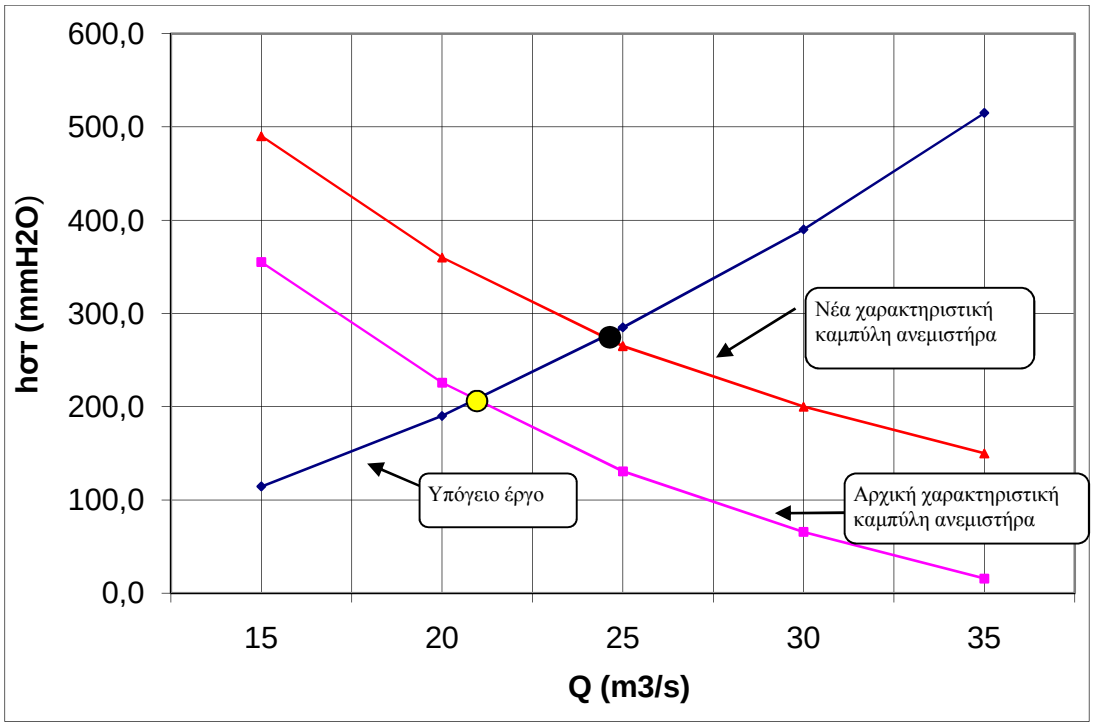
Ένα σύστημα αερισμού αποτελείται από ένα ή περισσότερους ανεμιστήρες, κατάλληλα συνδεδεμένους με ένα δίκτυο αγωγών δια μέσου των οποίων πρέπει να διέλθει το σύνολο της παροχής του αέρα. Όπως αναφέρθηκε, η αντίσταση του δικτύου εξαρτάται κυρίως από τη διάταξη και τα γεωμετρικά χαρακτηριστικά των αγωγών και τη μορφή των τοιχωμάτων τους και οπωσδήποτε μπορεί να επηρεασθεί από πλευράς μεγέθους από το μηχανικό, κατά τη φάση μελέτης και σχεδιασμού των έργων.

Αντίθετα, η χαρακτηριστική καμπύλη του ανεμιστήρα εξαρτάται από το μέγεθος και τον εν γένει σχεδιασμό του μηχανήματος από τον κατασκευαστή και συνεπώς ο μηχανικός οφείλει να προσαρμόσει τον κατά περίπτωση κατάλληλο ανεμιστήρα στο δίκτυο, ώστε να επιτύχει κατά τον καλύτερο τρόπο αερισμό του υπόγειου έργου.

Κάθε ανεμιστήρας που στρέφεται με δεδομένη, σταθερή ταχύτητα μπορεί να λειτουργήσει με άπειρους μεν αλλά καθορισμένους από πλευράς μεγεθών συνδυασμούς πιέσεων-παροχών, που προκύπτουν από την χαρακτηριστική του καμπύλη.

Εξάλλου και το δίκτυο του υπόγειου έργου καθορίζεται από τη χαρακτηριστική παραβολική του καμπύλη ($h = R * Q^2$) και δεδομένου ότι οι απώλειες πιέσεως που επισυμβαίνουν κατά τη ροή του αέρα πρέπει να ισοσταθμιστούν επακριβώς από την πίεση που θα προκαλέσει τη ροή (δηλ. την πίεση του ανεμιστήρα) είναι φανερό ότι ο ανεμιστήρας, εισαγόμενος στο σύστημα, θα επιλέξει ως σημείο λειτουργίας του το σημείο τομής των δύο χαρακτηριστικών καμπυλών που προαναφέρθηκαν (διάγραμμα 2.2.)

Το σημείο αυτό (σημείο κίτρινο) παραμένει σταθερό εφ'όσον ορισμένες παράμετροι του συστήματος (αντίσταση του δικτύου, ταχύτητα περιστροφής του ανεμιστήρα, πίεση φυσικού ελκυσμού, πυκνότητα αέρα) παραμένουν επίσης σταθερές. Εάν όμως η αντίσταση του υπόγειου έργου μεταβληθεί –πράγμα που για διαφόρους λόγους κατά κανόνα συμβαίνει- είναι ενδεχόμενο να απαιτηθεί κατάλληλη επέμβαση ώστε ο ανεμιστήρας να ανταποκριθεί στις νέες αυτές συνθήκες. Στην περίπτωση αυτή το σημείο τομής της νέας χαρακτηριστικής καμπύλης του ανεμιστήρα με την αντίστοιχη καμπύλη του υπόγειου έργου (σημείο μαύρο) καθορίζει το νέο σημείο λειτουργίας.



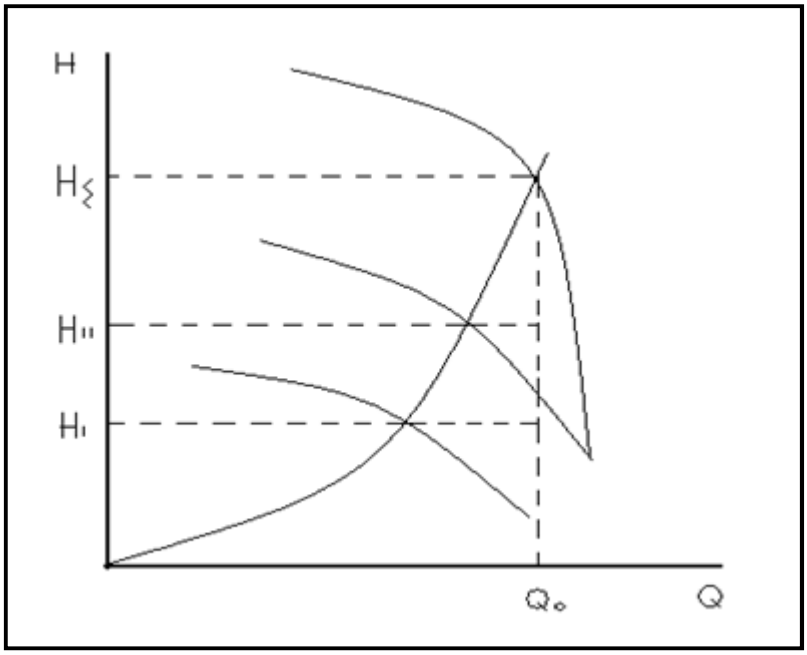
Διάγραμμα 2.2.: Σημείο λειτουργίας ανεμιστήρα.

Οι περισσότεροι ανεμιστήρες έχουν σχεδιαστεί κατά τρόπο ώστε να λειτουργούν ασφαλώς και οικονομικώς σε ορισμένο μόνο τμήμα των χαρακτηριστικών τους καμπυλών. Οι κατασκευαστές αναφέρονται συνήθως από πλευράς αποδόσεων σε αυτό το τμήμα της καμπύλης, στο οποίο θα πρέπει κατά το δυνατόν να περιορίζεται η λειτουργία του ανεμιστήρα.

Στα συνήθη υπόγεια έργα προκειμένου να καθοριστεί το σημείο λειτουργίας χρησιμοποιούνται οι χαρακτηριστικές καμπύλες πιέσεων-παροχών (ανεμιστήρα και μεταλλείου) που αναφέρονται στις στατικές πιέσεις. Οι δυναμικές πιέσεις παραλείπονται. Η διαδικασία αυτή δίνει ικανοποιητικά αποτελέσματα αν οι δυναμικές πιέσεις (ανεμιστήρα και υπόγειου έργου) είναι ίσες ή έχουν πολύ μικρές τιμές, ώστε να μπορούν να αγνοηθούν. Τονίζεται πάντως ότι πρέπει να καταβάλλεται προσπάθεια ώστε να ωφελούμαστε από τη δυναμική πίεση του ανεμιστήρα. Αν η πίεση αυτή υπερβαίνει την δυναμική πίεση του υπόγειου έργου ή μπορεί να μειωθεί στην έξοδο του αέρα από τον ανεμιστήρα τότε ένα μέρος της είναι δυνατόν να μετατραπεί σε στατική πίεση, προς όφελος του συστήματος.

2.4.2. Ανεμιστήρες σε σειρά

Κατά τη σε σειρά διάταξη οι ανεμιστήρες τοποθετούνται στο ίδιο κύκλωμα, καθένας δε από αυτούς αντιμετωπίζει το σύνολο της παροχής και εισφέρει μέρος της απαιτούμενης πίεσης προς επίτευξη της συγκεκριμένης ροής. Το σημείο λειτουργίας του συστήματος βρίσκεται στην τομή της χαρακτηριστικής καμπύλης του υπογείου έργου και της χαρακτηριστικής καμπύλης του συνδυασμένου αποτελέσματος από τη σύγχρονη λειτουργία των ανεμιστήρων. Θα πρέπει να τονιστεί, ότι για επιτυχημένα αποτελέσματα χρειάζεται κατάλληλη επιλογή ανεμιστήρων.



Διάγραμμα 2.3.: Συνδυασμένο αποτέλεσμα δύο διαφορετικών ανεμιστήρων σε σειρά.

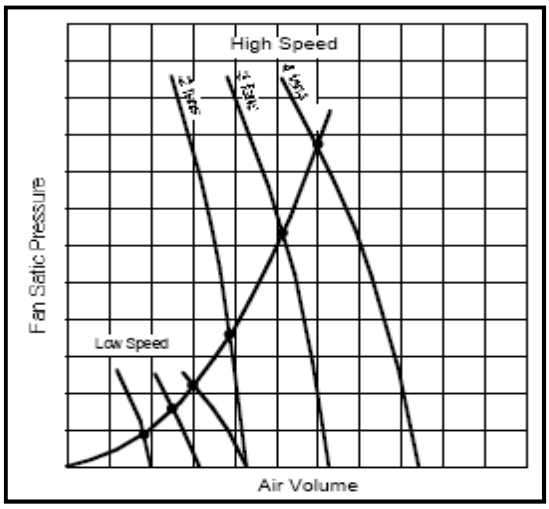
Οι επιλεγόμενοι ανεμιστήρες πρέπει να λειτουργούν στο τμήμα της χαρακτηριστικής τους καμπύλης στο οποίο επιτυγχάνεται η καλύτερη δυνατή απόδοση. Τέλος σε κάθε περίπτωση πρέπει να ελέγχεται το οικονομικό κόστος μιας ενέργειας εγκατάστασης ενός ανεμιστήρα σε σειρά ή η αντικατάστασή του από έναν άλλο μεγαλύτερο σε ένα δίκτυο με αυξημένες ανάγκες.

2.4.3. Ανεμιστήρες σε παράλληλη διάταξη

Δύο ή περισσότεροι ανεμιστήρες είναι δυνατόν να λειτουργήσουν τοποθετημένοι σε παράλληλη διάταξη, προκειμένου να αυξηθεί η ποσότητα του παρεχόμενου αέρα, εφόσον οι συνθήκες του υπογείου έργου το επιβάλλουν.

Πράγματι, σε περίπτωση συνεχιζόμενης ανάπτυξης του υπογείου έργου και δημιουργίας νέων τμημάτων εκμετάλλευσης η αντίσταση του δικτύου αυξάνεται, ενώ συγχρόνως απαιτούνται υψηλότερες παροχές αέρα προς εξυπηρέτηση των αναγκών του συστήματος. Σ' αυτήν ακριβώς την περίπτωση η τοποθέτηση ενός διαθέσιμου ανεμιστήρα ο οποίος θα λειτουργεί παράλληλα προς τον κύριο ανεμιστήρα του υπογείου έργου, δίνει ικανοποιητικά αποτελέσματα.

Γενικά, οι ανεμιστήρες μπορεί να τοποθετηθούν είτε στην αυτή στοά, είτε σε διαφορετικές στοές (αγωγούς) του δικτύου. Εφόσον τοποθετηθούν στην αυτή στοά, αντιμετωπίζουν το σύνολο του δικτύου, η πίεση δε έκαστου ανεμιστήρα ισούται με τις συνολικές απώλειες πιέσεως, ενώ η παροχή τους αποτελεί μέρος της συνολικής παροχής.



Διάγραμμα 2.4.: Αποτέλεσμα ανεμιστήρων σε παράλληλη διάταξη.

Συνηθέστερα, ανεμιστήρες που λειτουργούν κατά παράλληλη διάταξη τοποθετούνται σε διαφορετικές στοές.

Εάν οι αντιστάσεις των στοών αυτών είναι συγκριτικά υψηλές δεν αντιμετωπίζονται ιδιαίτερα προβλήματα λειτουργίας. Στην αντίθετη περίπτωση πρέπει να ληφθούν υπ' όψη, τόσο τα χαρακτηριστικά κάθε στοάς, όσο και τα χαρακτηριστικά του υπόλοιπου δικτύου – πέραν δηλαδή, του σημείου στο οποίο οι εν λόγω στοές συνδέονται – και να εφαρμοσθεί

γραφική λύση του προβλήματος προς καθορισμό των συνθηκών ροής και λειτουργίας των ανεμιστήρων.

Θα πρέπει πάντως να σημειωθεί ότι η χρησιμοποίηση περισσότερων του ενός ανεμιστήρων σε παράλληλη διάταξη απαιτεί προσοχή και προηγούμενη διερεύνηση, προς αποφυγή της εμφανίσεως του φαινόμενου της απώλειας στηρίξεως.

2.5. Ενέργεια του αερισμού

Η απαραίτητη ενέργεια για να υπερνικηθούν οι απώλειες πίεσης h που αντιστοιχούν στη στατική πίεση του υπόγειου χώρου (h_Σ) προκειμένου να επιτευχθεί μέσα από το δίκτυο η επιθυμητή παροχή αέρα, δίνεται από τον τύπο:

$$E = N * t$$

όπου: *E = απαιτούμενη ενέργεια (kWh)*

t = χρόνος (hrs)

N = ωφέλιμη ισχύς (kW) που δίνεται από την ακόλουθη εξίσωση:

$$N = \frac{Q * h}{1000}$$

Λαμβάνοντας υπόψη και την απόδοση της εγκατάστασης των ανεμιστήρων, που αποτελούν τις πηγές παροχής ενέργειας στο μηχανικό αερισμό, προκύπτει η ονομαστική ισχύς του συστήματος που δίνεται από την ακόλουθη εξίσωση:

$$N' = \frac{Q * h}{1000 * n}$$

όπου n ο βαθμός απόδοσης του συστήματος αερισμού.

3. ΥΠΟΛΟΓΙΣΜΟΣ ΤΟΥ ΚΥΚΛΩΜΑΤΟΣ ΑΕΡΙΣΜΟΥ ΣΕ ΥΠΟ ΔΙΑΝΟΙΞΗ ΥΠΟΓΕΙΑ ΕΡΓΑ

3.1. Σκοπός του βοηθητικού αερισμού - Γενικά

Στη διεθνή και ελληνική βιβλιογραφία έχει επικρατήσει ο όρος «κύριος αερισμός» ή απλά «αερισμός» για τα συστήματα αερισμού που αφορούν σε δίκτυα αγωγών (στοές, φρέατα, κλπ.) μεταλλευτικών και άλλων υπόγειων τεχνικών έργων και ο όρος «βοηθητικός αερισμός» (auxiliary ventilation) για τα συστήματα αερισμού που έχουν ως στόχο την εξασφάλιση του απαραίτητου αέρα σε «τυφλά» μέτωπα (π.χ. στοές ή σήραγγες υπό διάνοιξη). Το αντικείμενο της παρούσας διπλωματικής εργασίας αφορά στην ανάπτυξη ενός εύχρηστου εργαλείου για τον υπολογισμό οικονομοτεχνικών παραμέτρων σε κυκλώματα βοηθητικού αερισμού. Ωστόσο, για λόγους συντομίας αντί του όρου «βοηθητικός αερισμός» στις επόμενες ενότητες χρησιμοποιείται ο όρος «αερισμός».

Οι διατάξεις που αφορούν στον κατά περίπτωση υπολογισμό της απαιτούμενης παροχής αέρα είναι κοινές. Εντούτοις, ο τρόπος υπολογισμού διαφοροποιείται εξαιτίας των ιδιαιτεροτήτων που χαρακτηρίζουν τα συγκεκριμένα συστήματα αερισμού (π.χ. χρήση αγωγών για την προσαγωγή του αέρα στο μέτωπο). Στις επόμενες παραγράφους παρατίθενται τα στοιχεία εκείνα που απαιτούνται και χρησιμοποιούνται για τον υπολογισμό των βασικών μεγεθών του αερισμού σε υπό διάνοιξη υπόγεια έργα.

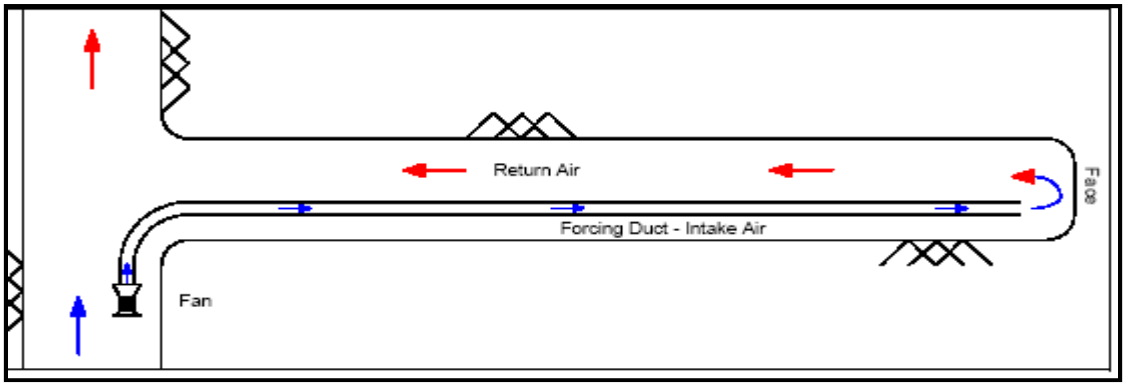
3.2. Διαμορφώσεις βοηθητικού αερισμού - Γενικά

Παρακάτω αναπτύσσονται τα χαρακτηριστικά των εναλλακτικών διαμορφώσεων του βοηθητικού αερισμού και τα λειτουργικά τους αποτελέσματα.

3.2.1. Φυσητικός αερισμός

Στην περίπτωση εφαρμογής φυσητικού αερισμού σε υπό διάνοιξη υπόγεια έργα ισχύουν τα εξής: Η ζώνη των ρύπων από τα καυσαέρια των μηχανημάτων ή τα αέρια μιας ανατίναξης, ωθείται από το ρεύμα του αέρα και αραιωμένη κινείται κατά μήκος της στοάς. Αν η

απόσταση μεταξύ του στομίου του αγωγού και του μετώπου είναι μικρή, η απομάκρυνση των καπνών είναι γρήγορη, δεδομένου ότι αναμειγνύονται συνεχώς με τον προσαγόμενο αέρα.



Εικόνα 3.1.: Φυσητικός αερισμός.

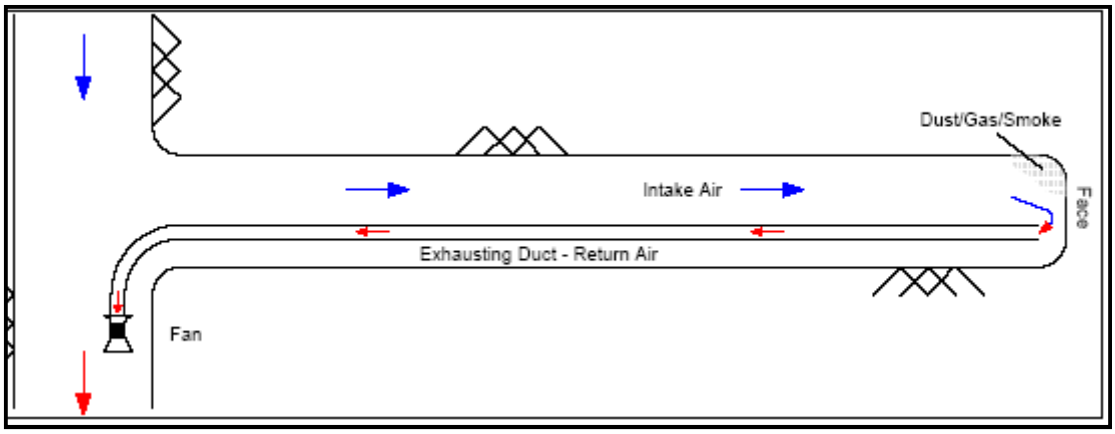
Αντίθετα αν η παραπάνω απόσταση είναι σχετικά μεγάλη, διατηρείται στο βάθος της στοάς ζώνη εντός της οποίας ο αέρας παραμένει ακίνητος (νεκρή ζώνη).Αρχικά είναι φανερό ότι πρέπει κατά το δυνατόν να μειώνεται η απόσταση μεταξύ του στομίου του αγωγού και του μετώπου προσέχοντας παράλληλα τυχών παραμορφώσεις του αγωγού. Επισημαίνεται ότι πρέπει να εξασφαλιστεί η κατάλληλη παροχή, ώστε να αποφευχθεί η συγκέντρωση των καπνών, που ήδη έχουν αρχίσει να αραιώνονται, σε σημεία προς στην είσοδο της στοάς και η περαιτέρω αύξηση της συγκέντρωσής τους.

Τέλος εκτός από την απόσταση του στομίου του αγωγού με του μετώπου και την παροχή σημαντικοί παράγοντες που επηρεάζουν τη διαδικασία αραίωσης και απαγωγής της μολυσμένης ατμόσφαιρας των υπογείων έργων είναι η διάμετρος των αγωγών και η θέση στην οποία έχουν τοποθετηθεί.

3.2.2. Μυζητικός αερισμός

Κατά την εφαρμογή του μυζητικού αερισμού το ρεύμα αέρα τείνει να εισέλθει βίαια μέσα στον αγωγό, με αποτέλεσμα να εξαφανίζεται οποιαδήποτε αισθητή κίνηση αέρα, ακόμη και σε πολύ μικρή απόσταση πέρα από το στόμιο σχεδόν του αγωγού. Με αυτόν τον τρόπο στο χώρο από το στόμιο περίπου του αγωγού μέχρι το μέτωπο ο αέρας είναι πρακτικά ακίνητος. Οι καπνοί που καταλαμβάνουν το χώρο της στοάς παρασύρονται μέσα στον

αγωγό, μαζί με ποσότητα καθαρού αέρα, ενώ οι καπνοί που βρίσκονται στη νεκρή ζώνη απάγονται με πολύ αργό ρυθμό.



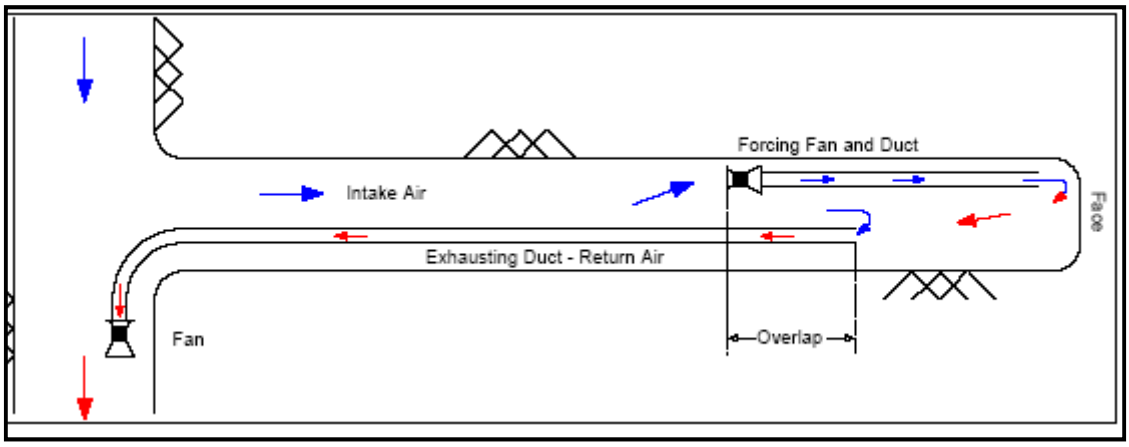
Εικόνα 3.2.: Μυζητικός αερισμός.

Το κύριο πλεονέκτημα του μυζητικού αερισμού είναι η αποφυγή δημιουργίας κινούμενης ζώνης καπναερίων μέσα στη στοά, αντίθετα όμως ο καθαρισμός του μετώπου είναι δύσκολος και εξαιρετικά αργός εκτός αν η απόσταση στομίου αγωγού-μετώπου είναι εξαιρετικά μικρή. Επομένως προκειμένου να επιτευχθούν καλύτερα αποτελέσματα ο μυζητικός αερισμός πρέπει να συνδυαστεί κατάλληλα με φυσητικό αερισμό, δηλαδή να εφαρμοσθεί μεικτός αερισμός.

3.2.3. Μεικτός αερισμός

Ο συνδυασμός φυσητικού και μυζητικού αερισμού (μεικτός αερισμός) επιτυγχάνεται, είτε διαδοχικώς είτε συγχρόνως.

Στην πρώτη περίπτωση χρησιμοποιείται ένας και ο αυτός αγωγός, εφαρμόζεται αρχικά μύζηση η οποία ακολουθείται από φύσηση. Κατά τη φάση της μύζησης απομακρύνεται το τμήμα της ζώνης καπνού που βρίσκεται μπροστά από το στόμιο του αγωγού. Στη συνέχεια ακολουθεί η φάση της φύσησης κατά την οποία καθαρίζεται από τους καπνούς η περιοχή μεταξύ του στομίου του αγωγού και του μετώπου και όπως είναι φυσικό μία ποσότητα καπνού θα μετατοπιστεί μέσα στη στοά με πολύ μικρότερο κίνδυνο λόγω της αραίωσης που έχει επέλθει. Πάντως ο συνολικός χρόνος για τον καθαρισμό του μετώπου είναι πολύ μεγαλύτερος από ότι στο αμιγές φυσητικό σύστημα.



Εικόνα 3.3.: Μεικτός αερισμός.

Στη δεύτερη περίπτωση είναι απαραίτητοι δύο ανεξάρτητοι αγωγοί, εκ των οποίων ο χρησιμοποιούμενος για τη φύσηση του αέρα μπορεί να είναι μικρού μήκους και διαμέτρου ώστε να μεταφέρεται εύκολα πολύ κοντά στο μέτωπο, μειώνοντας έτσι το χρόνο καθαρισμού του αέρα. Το στόμιο του αγωγού μύζησης μπορεί να τοποθετηθεί σε αρκετή απόσταση από το μέτωπο πράγμα που επιτρέπει τη χρησιμοποίηση αγωγού μεγάλης διαμέτρου χωρίς να υπάρχει ανάγκη προφύλαξής του.

Τέλος οι καπνοί που αναμοχλεύονται και απωθούνται από το φυσητικό αγωγό δεν θα κυκλοφορήσουν πίσω από το στόμιο του μυζητικού αγωγού αν η παροχή αυτού είναι μεγαλύτερη. Πρόκειται λοιπόν για την πιο αποτελεσματική διάταξη.

3.3. Εξοπλισμός βοηθητικού αερισμού

3.3.1. Ανεμιστήρες

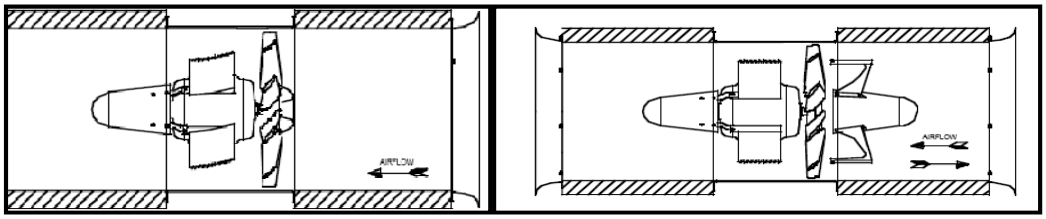
Οι χρησιμοποιούμενοι για τον αερισμό υπογείων τεχνικών έργων ανεμιστήρες είναι κατά κανόνα οι αξονικοί ελικοφόροι διότι είναι επιθυμητό να μην ελαττώνεται σημαντικά η παροχή σε περίπτωση αύξησης του μήκους του αγωγού, γεγονός που χαρακτηρίζει τους φυγοκεντρικούς ανεμιστήρες. Αμφότεροι λειτουργούν με ηλεκτρικό ρεύμα αφού η λειτουργία με πεπιεσμένο αέρα εγκαταλείπεται λόγο σημαντικών μειονεκτημάτων.

α. Φυγοκεντρικοί

Στους φυγοκεντρικούς ανεμιστήρες ο αέρας αναρροφείται με φυγοκέντριση εντός δίσκου (τροχού) με πτερύγια και στη συνέχεια εισέρχεται σε κέλυφος σχήματος σαλιγκαριού, η διατομή του οποίου αυξάνει προοδευτικά μέχρι την έξοδο του αέρα. Η παραγωγή πίεσης οφείλεται σε δύο ανεξάρτητες δράσεις: Στην κεντρόφυγα δύναμη, λόγω της περιστροφής του αέρα και στην κινητική ενέργεια που παρέχεται καθώς ο αέρας εγκαταλείπει το άκρο της κορυφής των λεπίδων του δίσκου.

β. Αξονικοί

Οι ελικοφόροι ανεμιστήρες περιλαμβάνουν, εντός κυλινδρικού ή ελαφρώς κωνικού κελύφους-αγωγού, έλικα που αποτελείται από ορισμένο αριθμό πτερυγίων τα οποία είναι κατάλληλα στερεωμένα σε πλήμνη ώστε να μπορούν να ρυθμίζονται. Η παραγωγή πίεσης έγκειται στο να δοθεί εφαπτομενική επιτάχυνση στον αέρα κατά τη δίοδό του μέσω της έλικας. Η ενέργεια που δημιουργείται κατά την περιστροφή πρέπει να μετατραπεί σε ενέργεια γραμμικής ροής και στατική πίεση καθώς ο αέρας εγκαταλείπει την έλικα, πράγμα δύσκολο πλην όμως ουσιαστικό για την καλή απόδοση του ανεμιστήρα. Για το σκοπό αυτό χρησιμοποιούνται διευθυντικά πτερύγια αμέσως μετά την έλικα. Ακόμη οι ελικοφόροι αξονικοί μπορούν να ρυθμιστούν σε αντίστροφη λειτουργία και να ικανοποιήσουν απαιτήσεις μεικτού αερισμού.



Εικόνα 3.4.: Διαφορετικές κατευθύνσεις αέρα σε αξονικούς ανεμιστήρες.

Τέλος οι σύγχρονοι ανεμιστήρες έχουν τη δυνατότητα της αυτόματης ρύθμισης της κλίσης των πτερυγίων χωρίς τη διακοπή της λειτουργίας του ανεμιστήρα για την μέγιστη απόδοση αυτού. Η δυνατότητα αυτή είναι σημαντική σε περιπτώσεις συχνών μεταβολών της αντίστασης δικτύου.

3.3.2. Αγωγοί

Οι αγωγοί που διοχετεύουν τον αέρα του ανεμιστήρα σε σύστημα αερισμού ενός υπόγειου τεχνικού έργου διακρίνονται σε άκαμπτους και σε εύκαμπτους και είναι φανερό ότι δεν είναι δυνατόν να συγκεντρώνουν όλες τις παρακάτω ιδιότητες.

Γενικά τα χαρακτηριστικά των αγωγών του αέρα πρέπει:

- Να παρέχουν ελάχιστη αντίσταση στη δίοδο του αέρα, με λίγες διαφυγές.
- Να προσαρμόζονται εύκολα στις συνθήκες των υπογείων χώρων
- Να παρουσιάζουν αντοχή στις κακώσεις και στη διάβρωση
- Να μεταφέρονται και να αποθηκεύονται εύκολα.
- Να κατασκευάζονται από άφλεκτες ύλες, για την αποφυγή μετάδοσης πυρκαγιών.
- Να παρουσιάζουν σχετική ηλεκτρική αγωγιμότητα, ώστε να αποφεύγεται η συγκέντρωση ηλεκτροστατικών φορτίων, τα οποία μπορεί να προκληθούν κυρίως από τριβή κονιορτών στην επιφάνεια των αγωγών.

3.3.3. Χαρακτηριστικά εύκαμπτων αγωγών

Οι εύκαμπτοι αγωγοί χρησιμοποιούνται ευρύτατα. Κατασκευάζονται, είτε από κατάλληλα αδιαβροχοποιημένα υφάσματα, είτε από αμιγείς πλαστικές ύλες, όπως χλωριούχο βινύλιο, νεοπρένιο, πολυαιθυλένιο, κλπ.

Το βάρος των εύκαμπτων αγωγών (0,7 έως 1 kg/m) είναι 10 έως 20 φορές μικρότερο, σε σύγκριση με το βάρος των μεταλλικών αγωγών, πράγμα που αποτελεί κύριο πλεονέκτημά τους. Παράλληλα, παρουσιάζουν αρκετά μειονεκτήματα, όπως αυξημένη αντίσταση στη δίοδο του αέρα, σημαντικές διαφυγές, μειωμένη σχετικά αντίσταση στη διάδοση της φλόγας και ελαττωμένη γενικά αντοχή στις κακώσεις που οδηγούν συνήθως σε σχισίματα.

3.3.4. Χαρακτηριστικά άκαμπτων αγωγών

Οι άκαμπτοι αγωγοί κατασκευάζονται από λαμαρίνα, από ξύλα κόντρα πλακέ και από άκαμπτη πλαστική ύλη. Οι μεταλλικοί αγωγοί χρησιμοποιούνται ευρέως, είναι ισχυρής κατασκευής, άφλεκτοι και παρέχουν μικρή αντίσταση στη δίοδο του αέρα, ενώ οι διαφυγές περιορίζονται στην περιοχή των συνδέσμων. Από την άλλη πλευρά, είναι δαπανηροί και

έχουν μεγάλο βάρος, γι' αυτό και πρέπει να χρησιμοποιούνται σε περιπτώσεις κατά τις οποίες η εγκατάσταση θα παραμένει σε λειτουργία αρκετό καιρό.

Οι αγωγοί από ξύλα έχουν το μισό βάρος από τους μεταλλικούς και διατίθενται σε τμήματα μήκους 3m. Παρουσιάζουν μικρή αντίσταση στη δίοδο του αέρα και η αντοχή τους, που είναι μειωμένη, βελτιώνεται σημαντικά με κατάλληλες επαλείψεις (βερνίκια).

Οι αγωγοί από άκαμπτη πλαστική ύλη κατασκευάζονται συνήθως από πολυεστέρες και έχουν γενικά καλές ιδιότητες (αντοχή στη διάβρωση, μικρή αντίσταση στη δίοδο του αέρα και μικρό σχετικά βάρος). Σε περίπτωση πυρκαγιάς δημιουργεί πολύ επικίνδυνες συνθήκες για το προσωπικό (πυκνός βαρύς καπνός από τοξικά αέρια).

3.3.5. Τύποι χρησιμοποιούμενων αγωγών

Παρακάτω παρουσιάζονται μερικά από τα ποιο ευρέως χρησιμοποιούμενα είδη αγωγών σε υπόγεια τεχνικά έργα.

α. Σπειροειδής αγωγός από χάλυβα

Χρησιμοποιείται εξίσου για μυζητικό και φυσητικό αερισμό. Ο αγωγός χάλυβα έχει το πλεονέκτημα ότι μπορεί να τυλιχτεί, και πολλά ορυχεία οργανώνουν ένα αφιερωμένο εργαστήριο για αυτήν την δραστηριότητα. Ο αγωγός μπορεί να κατασκευαστεί σε διάφορα μήκη. Οι τυπικές διάμετροι κυμαίνονται μεταξύ 150 mm και 2.450 mm.

Συμπερασματικά:

- Μικρό κόστος, μεικτός αγωγός
- Άφλεκτος
- Κρατά την υψηλή θετική και αρνητική πίεση
- Μέση διαρροή
- Χαμηλός παράγοντας τριβής

Καταλληλότερος για:

- Συστήματα εξαερισμού μήκους λιγότερο των 3000 μέτρων
- Μια χρήσεως εφαρμογές αφού σε περίπτωση επανασύνδεσής του εξαιτίας του υλικού του θα έχουμε φτωχή σύνδεση και απώλειες.

β. Αγωγός από Fibreglass

Ο αγωγός αυτός σχεδιάστηκε για να αντιστέκεται στις συγκρούσεις και στις υψηλές θετικές ή αρνητικές πιέσεις. Αυτή η άκαμπτη κατασκευή διοχέτευσης αέρα εξειδικεύεται σε κρίσιμες εφαρμογές με τα κύρια χαρακτηριστικά γνωρίσματά της που είναι: χαμηλό βάρος, άριστη αντίσταση διάβρωσης, υψηλή απόδοση και ικανότητα επαναχρησιμοποίησης. Παρέχεται και σε ωώδες σχήμα και τα τυπικά μεγέθη που χρησιμοποιείται είναι 300 έως 1.200 mm διάμετρο και 6 μέτρα μήκος.

Συμπερασματικά:

- Ανθεκτικότερος αγωγός μύζησης
- Ελαφρύς
- Κρατά την υψηλή θετική και αρνητική πίεση
- Χαμηλό παράγοντα τριβής
- Μέση διαρροή
- Σχετικά ακριβός

Καταλληλότερος για:

- Εξαερισμό μικρού μήκους (λιγότερο από 1.500 μ)
- Πολλαπλάσιες διαδικασίες επαναχρησιμοποίησης
- Ανάπτυξη ορυχείου

γ. Εύκαμπτος φυσητικός αγωγός

Ο αγωγός αυτός έχει εφαρμογή σε θετικές τιμές πίεσης και κυριαρχεί σε ορυχεία με σκληρά πετρώματα και σε σήραγγες λόγω χαμηλού κόστους και ευκολίας εγκατάστασής του. Οι ραφές και τα στρόφια πρέπει να ενωθούν σφιχτά και αεροστεγώς για να παρέχεται μικρή αντίσταση στη ροή του αέρα. Οι χαρακτηριστικές διάμετροι κυμαίνονται από 150 ως 2.450 mm και τα μήκη τους μπορεί να ξεπερνούν τα 300 μέτρα.

Συμπερασματικά:

- Υψηλή διαρροή στον εξαερισμό του μετώπου και σχετικά χαμηλή σε εφαρμογές με TBM λόγω του υλικού που χρησιμοποιείται.
- Μέσος παράγοντας τριβής
- Χαμηλότερο κόστος
- Εύκολη εγκατάσταση

- Ποικιλία υλικού κατασκευής του αγωγού για την επίτευξη των επιθυμητών αποτελεσμάτων.

Καταλληλότερος για:

- Όλα τα μήκη των εξαερισμών

δ. Εύκαμπτος μυζητικός αγωγός

Ο αγωγός αυτός χρησιμοποιείται όπου απαιτείται ελαφρύς μηχανισμός μυζητικού εξαερισμού, επειδή μεταφέρεται εύκολα. Κατασκευάζεται από PVC «ντυμένο» από νάιλον και πολυεστέρα και είναι ενισχυμένος με ελικοειδές σπείρωμα από χάλυβα. Χαρακτηριστικά μεγέθη: Διάμετρος 150 με 1500mm και μήκος 3, 4,5 και 7,5 μέτρα.

Συμπερασματικά:

- Το βάρος του εξαρτάται από τις εκτιμήσεις της αρνητικής πίεσης
- Μπορεί να αποθηκευτεί σε μικρή περιοχή
- Πρέπει να είναι σωστά τεντωμένος γιατί σε αντίθετη περίπτωση θα υπάρχουν μεγάλες απώλειες από την τριβή.

Καταλληλότερος για:

- Εκκινήσεις σηράγγων
- Μικρού μήκους συστήματα εξαερισμού
- Εύκαμπτες συνδέσεις σε άλλο τύπο διοχέτευσης αέρα.

3.3.6. Συστήματα αναρτήσεως και σύνδεσης εύκαμπτων φυσητικών αεραγωγών

Αυτού του είδους οι αεραγωγοί διαθέτουν γάντζους αναρτήσεως από χάλυβα θερμού γαλβανίσματος τοποθετημένους σε τακτά διαστήματα. Οι γάντζοι προσαρμόζονται με ειδικό ενισχυμένο ύφασμα το οποίο έχει βουλκανιστεί και συγκολληθεί στο ύφασμα του αεραγωγού. Το σύστημα αναρτήσεως είναι κατασκευασμένο με τέτοιο τρόπο ώστε σε ενδεχόμενη βίαιη και μεγάλη μετακίνηση του αεραγωγού να θραυστούν οι χαλύβδινοι γάντζοι χωρίς να τραυματιστεί ή να σκιστεί το ενισχυμένο ύφασμα συγκρατήσεως των γάντζων.

Οι γάντζοι αναρτώνται σε χαλύβδινο συρματόσχοινο που έχει αναρτηθεί στην οροφή της σήραγγας με αγκύρια μικρού μήκους.

Όσον αφορά τη σύνδεση των τεμαχίων των αεραγωγών, αυτή γίνεται ως εξής: το κάθε τεμάχιο αεραγωγού διαθέτει ένα χαλύβδινο δακτύλιο που έχει βουλκανιστεί στο ένα άκρο του αεραγωγού που δεν διαθέτει το χαλύβδινο δακτύλιο, ωθείται εξωτερικά ώστε να επικαλύψει το χαλύβδινο δακτύλιο στο άκρο του άλλου αεραγωγού. Ένας σφικτήρας σύνδεσης τοποθετείται εξωτερικά της επικαλυπτόμενης περιοχής έτσι ώστε η εγκοπή στον μέσο του να ευθυγραμμιστεί με το χαλύβδινο δακτύλιο. Με τη σύσφιξη του σφιχτήρα ασφαλίζεται η σύζευξη και επιτυγχάνεται η πλήρης αεροστεγής σύνδεση των αεραγωγών. Ο σφιχτήρας κατασκευάζεται από χαλύβδινο έλασμα θερμού γαλβανίσματος πάχους 1mm διαμορφωμένο σε ειδικό προφίλ πάχους 60mm. Η σύσφιξη πραγματοποιείται με τον αξονικό κοχλία από επιψευδαργυρομένο χάλυβα. Εκτός από τον παραπάνω τρόπο σύνδεσης υπάρχουν και πιο σύγχρονοι τύποι συνδέσεων με ισχυρής αντοχής φερμουάρ που εξασφαλίζουν αεροστεγή σύνδεση και απόλυτα επίπεδη και ευθυγραμμισμένη εσωτερική επιφάνεια αγωγού.

3.4. Εκτίμηση των χαρακτηριστικών μεγεθών ενός κυκλώματος βοηθητικού αερισμού

Τα χαρακτηριστικά μεγέθη ενός κυκλώματος βοηθητικού αερισμού είναι αντίστοιχα με αυτά του κύριου αερισμού, ήτοι η απαιτούμενη παροχή «φρέσκου» αέρα και η αντίσταση που προκαλείται εξαιτίας των τριβών και κρούσεων των μορίων του αέρα με τους αγωγούς. Οι διατάξεις που αφορούν στον κατά περίπτωση υπολογισμό της απαιτούμενης παροχής αέρα είναι κοινές. Εντούτοις, ο τρόπος υπολογισμού διαφοροποιείται εξαιτίας των ιδιαιτεροτήτων που χαρακτηρίζουν τα συγκεκριμένα συστήματα αερισμού (π.χ. χρήση αγωγών για την προσαγωγή του αέρα στο μέτωπο).

Στις επόμενες παραγράφους παρατίθενται τα στοιχεία εκείνα που απαιτούνται και χρησιμοποιούνται για τον υπολογισμό των βασικών μεγεθών του αερισμού σε υπό διάνοιξη υπόγεια έργα, καθώς και οι πλέον χρησιμοποιούμενες υπολογιστικές μέθοδοι για την εκτίμηση της παροχής και των απωλειών πίεσης που επισυμβαίνουν εξαιτίας της αντίστασης των αγωγών.

3.4.1. Εκτίμηση της απαιτούμενης παροχής

α. Με βάση τη νομοθεσία

Η εκτίμηση της απαιτούμενης παροχής στην προκειμένη περίπτωση καθορίζεται από την νομοθεσία (Προεδρικό Διάταγμα και Κ.Μ.Λ.Ε) με βάση τις παραμέτρους που πρέπει να ικανοποιηθούν και αφορούν τα m^3/min για τον αριθμό των εργαζομένων, τα m^3/min για το σύνολο της ιπποδύναμης των μηχανημάτων και το σύνολο του αέρα για την αραίωση των ρύπων.

β. Με εμπειρικές εξισώσεις

Ο καθορισμός της παροχής με βάση τον τύπο και την ποσότητα της καταναλισκόμενης εκρηκτικής ύλης είναι περισσότερο αξιόπιστος, καθ' όσον η όλη σχετική διαδικασία είναι περισσότερο κανονική, καθορισμένη και ελεγχόμενη.

Ενδεικτικά αναφέρεται ότι η απαιτούμενη παροχή αέρα μπορεί έστω και συγκριτικά να υπολογισθεί και από τον τύπο

$$Q = 5 * \frac{B * c}{t} \ (m^3/min)$$

όπου: *B = Η κατανάλωση εκρηκτικής ύλης (kg) ανά κύκλο πυροδοτήσεως*

c = Ο όγκος του παραγομένου CO κατά την αποσύνθεση 1 kg εκρηκτικής ύλης

t = Ο χρόνος απαγωγής των καπνών (min), όχι μεγαλύτερος των 30 min.

Σημειώνεται ότι σε όλους τους προταθέντες παρόμοιους τύπους υπεισέρχεται ο παράγοντας c, η τιμή του οποίου – κυμαινόμενη συνήθως μεταξύ 40 και 100 λίτρων CO ανά kg εκρηκτικής ύλης- είναι μάλλον αβέβαιη διότι εξαρτάται όχι μόνο από τη φύση της εκρηκτικής ύλης αλλά και από τους άλλους παράγοντες (πυκνότητα γομώσεως, τρόπος πυροδοτήσεως, κλπ.).

Η ανάγκη απλουστεύσεως της διαδικασίας καθορισμού της απαιτούμενης παροχής οδήγησε στην πρόταση εφαρμογής του τύπου

$$Q = q * S, \ (m^3/s \text{ ανά } m^2 \text{ επιφάνεια της διατομής}).$$

Παρ' όλον ότι η αναλογία παροχής και μεγέθους διατομής μπορεί να αμφισβητηθεί, μακροχρόνιες παρατηρήσεις και μετρήσεις αποδεικνύουν ότι δεν απέχει πολύ από την πραγματικότητα, τουλάχιστον για διατομές που δεν έχουν υπερβολικό μέγεθος. Η τιμή q κυμαίνεται από 200 έως 300 l/s με επικρατέστερη τιμή 250 l/s.

Για παράδειγμα σε σήραγγα διατομής S= 20 m^2 με κατανάλωση 20 kg εκρηκτικής ύλης (ανά κύκλο πυροδοτήσεως) κατά την αποσύνθεση της οποίας παράγονται 50 λίτρα CO ανά kg, με επιθυμητό χρόνο απαγωγής των καπνών 15 min και εφαρμόσουμε τους δυο προεκτεθέντες τύπους, με ειδική παροχή 250 l/s, βρίσκονται παροχές 5,56 m^3/s και 5 m^3/s αντίστοιχα.

Εν τούτοις, εάν ορισμένοι παράγοντες μεταβληθούν, είναι δυνατόν να προκύψει περισσότερο σημαντική διαφορά, πράγμα που επιβεβαιώνει τη σημασία της προσαρμογής στις εκάστοτε υφιστάμενες συνθήκες.

3.4.2. Εκτίμηση αντιστάσεων και απωλειών πιέσεως στους αγωγούς

Αρχικά εξετάζονται οι αγωγοί υπό την προϋπόθεση ότι είναι απόλυτα στεγανοί, παραδοχή η οποία δεν ισχύει στην πράξη. Η ανά μονάδα μήκους αντίσταση αγωγού κυκλικής διατομής είναι:

$$r = 0{,}0156 * \lambda' * \frac{\Pi}{S^3} = 0.1 * \frac{\lambda'}{D^5}$$

όπου: *λ΄ = συντελεστής τριβής χωρίς διαστάσεις*

Π = περίμετρος αγωγού (m)

S = Η διατομή του αγωγού (m^2)

D = Η διάμετρος του αγωγού (m)

Συνεπώς, οι απώλειες πιέσεως h σε άκαμπτο αγωγό μήκους l υπό παροχή Q είναι:

$$h = r * l * Q^2 \text{ ή } h = 0.1 * \frac{\lambda'}{D^5} * l * Q^2$$

όπου: *h = απώλειες πίεσης σε άκαμπτο αγωγό (mmH_2O)*

r = αντίσταση αγωγού κυκλικής διατομής σε kilomurgues ανά μονάδα μήκους (kilomurgues)

l = μήκος αγωγού (m)

λ΄= συντελεστής τριβής χωρίς διαστάσεις

D = ονομαστική διάμετρος του αγωγού

Q = παροχή (m^3/sec)

Η παραπάνω εξίσωση δεν έχει εφαρμογή σε εύκαμπτους αγωγούς για πολλούς λόγους, κυρίως διότι η διάμετρος αυτών αυξάνεται με την πίεση, με εξαίρεση τις διατομές συνδέσεως των τμημάτων, όπως επίσης και διότι οι αγωγοί σχηματίζουν βέλη κάμψεως μεταξύ δύο διαδοχικών σημείων στηρίξεως. Στην πράξη για εύκαμπτους αγωγούς χρησιμοποιείται η σχέση:

$$h = 0.1 * \frac{\lambda'}{D^5} * l * Q^a$$

όπου: *α = λαμβάνεται ίσος με 1,7*

D = είναι η ονομαστική διάμετρος του αγωγού (m)

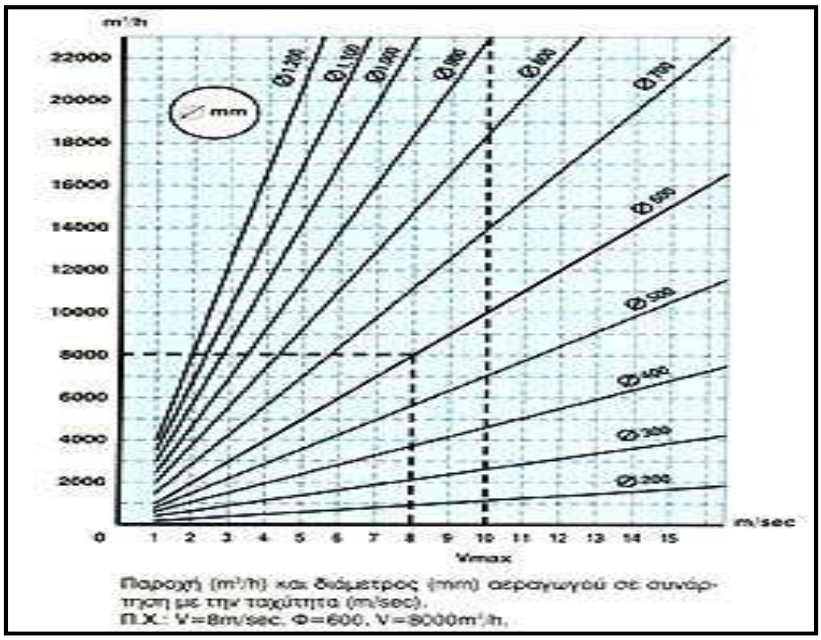
λ΄ = μεταξύ 0,021(αγωγός με καλή στήριξη) και 0,026 (αγωγός που παρουσιάζει βέλη κάμψεως)

l = μήκος αγωγού (m)

Q = Παροχή (m^3/sec)

h = απώλειες πίεσης σε εύκαμπτους αγωγούς (mmH_2O)

Στην πράξη, συχνά, οι απώλειες πίεσης εξαιτίας των τριβών και κρούσεων των μορίων του αέρα κατά τη ροή του μέσα από τους αγωγούς υπολογίζονται με τη βοήθεια κατάλληλων νομογραμμάτων που παρέχει ο κατασκευαστής του αγωγού, όπως αυτό που φαίνεται στο διάγραμμα.



Διάγραμμα 3.5.: Νομόγραμμα.

Προκειμένου να ληφθούν υπόψη οι διαφυγές του αέρα από τα σημεία σύνδεσης των τμημάτων του αγωγού, τις τυχόν οπές και σχισίματα αλλά ακόμη και από τους πόρους του υλικού στους εύκαμπτους αγωγούς, χρησιμοποιούνται κατά τους υπολογισμούς του απαιτούμενου ανεμιστήρα οι συντελεστές διαφυγής β και γ, οι οποίοι παρέχονται είτε από κατάλληλους πίνακες είτε από νομογράμματα.

Σε τμήμα αγωγού μικρού μήκους l, στο οποίο επισυμβαίνουν πολυάριθμες μικρές διαφυγές αέρα της αυτής φύσεως και ομοιόμορφα κατανεμημένες, η παροχή των διαφυγών q(m^3/sec) αυξάνει όσο αυξάνεται η διαφορά p(mmH_2O) της στατικής πίεσης στο εσωτερικό και στο εξωτερικό του αγωγού και εκφράζεται με τη σχέση:

$$q = \gamma * p^{\beta} * l$$

όπου γ και β χαρακτηριστικοί συντελεστές του αγωγού καλούμενοι και συντελεστές διαφυγών.

α. Τυπικές τιμές χαρακτηριστικών συντελεστών για εύκαμπτους αγωγούς

Αγωγοί εύκαμπτοι, από ύφασμα ή πλαστική ύλη σε τμήματα των 30m, $\gamma = 1{,}5*10^{-5}$

Αν ο εύκαμπτος αγωγός είναι πρόχειρα τοποθετημένος, παρουσιάζει οπές και σχίσματα και οι σύνδεσμοι των τμημάτων του είναι παραμορφωμένοι και σε κακή κατάσταση, τότε ο συντελεστής γ μπορεί να ανέλθει σε γ = 0,1.

Ο συντελεστής β για εύκαμπτους αγωγούς παίρνει τιμές β = 1,2 έως 1,5

β. Τυπικές τιμές χαρακτηριστικών συντελεστών για άκαμπτους αγωγούς

Αγωγοί άκαμπτοι σε τμήματα των 2.5m, $\gamma = 4{,}5*10^{-4}$ ως $3*10^{-6}$ ανάλογα με τον τύπο των συνδέσμων.

Ο συντελεστής β για άκαμπτο αγωγό παίρνει τιμές:

β = 0,9 ως 1,0 με καλή στεγανότητα και β = 0,5 ως 0,8 με σημαντικές διαφυγές.

Επειδή οι παραπάνω τιμές αφορούν αγωγούς διαμέτρου 0,6 m, αναφέρονται ως τάξη μεγέθους και θεωρούνται τιμές εφαρμογής, υπό την προϋπόθεση βεβαίως ότι έχει ληφθεί πρόνοια για την κατά το δυνατόν καλύτερη εγκατάσταση αγωγών. Εάν ο εξεταζόμενος αγωγός είναι διαμέτρου D(m) οι τιμές ανάγονται με πολλαπλασιασμό τους επί D/0.6.

3.5. Υπολογισμός παροχής και πιέσεως ανεμιστήρα

Η εκτίμηση της απαιτούμενης παροχής και των προκαλούμενων απωλειών πίεσης αποσκοπεί στον προσδιορισμό της παροχής Q και πίεσης h του ανεμιστήρα.

Κατά τον υπολογισμό της αντίστασης και κατά συνέπεια των απωλειών πίεσης που συμβαίνουν σε έναν αγωγό, το πραγματικό μήκος του είναι απαραίτητο να προσαυξηθεί (σύμφωνα με την έννοια του ισοδύναμου μήκους) προκειμένου να αντιμετωπισθούν οι αυξημένες αντιστάσεις στην περιοχή των συνδέσμων και οι απώλειες πιέσεως εκ κρούσεων των συνδέσμων, στις αλλαγές διεύθυνσης εξαιτίας των βελών κάμψης που παρουσιάζει ο αγωγός, κ.λπ. Τα ισοδύναμα αυτά μήκη παρέχονται συνήθως από τους κατασκευαστές. Για παράδειγμα, για αγωγό από πλαστική ύλη, με διάμετρο 0,6 m, θα πρέπει να ληφθεί υπόψη ισοδύναμο μήκος ανά σύνδεσμο από 1,5 έως 2 m και 10 m ανά στοιχείο αλλαγής διευθύνσεως κατά 90°. Προσεγγιστικά, είθισται να προσαυξάνεται το πραγματικό μήκος του αγωγού κατά 20% προκειμένου να συμπεριληφθούν αυτά τα μεγέθη .

Προκειμένου να εξασφαλιστεί παροχή Q_o στην έξοδο ενός αγωγού μήκους l (δηλ. στο μέτωπο εργασίας) πρέπει να ληφθούν υπόψη οι αναπόφευκτες διαφυγές του αέρα που επισυμβαίνουν κατά τη ροή του μέσα από τον αγωγό. Για να καλυφθούν οι διαφυγές, ο ανεμιστήρας πρέπει να παρέχει αυξημένη ποσότητα αέρα Q, η οποία όπως αποδεικνύεται, μπορεί υπολογίζεται από τον τύπο:

$$Q = \frac{Q_0 + \gamma * h_0^{\beta} * l}{\beta + 1}$$

όπου: *Q = Παροχή ανεμιστήρα (m^3/sec)*

Q_0 = Παροχή στην έξοδο του αγωγού (m^3/sec)

γ = χαρακτηριστικός συντελεστής αγωγού γ χωρίς διαστάσεις

h_0 = στατικό ύψος (mmH_2O)

β = χαρακτηριστικός συντελεστής αγωγού β χωρίς διαστάσεις

Όπως αναφέρθηκε στην προηγούμενη ενότητα, οι απώλειες ύψους σε έναν υποθετικά στεγανό εύκαμπτο αγωγό, κυκλικής διατομής, υπολογίζονται από τον ακόλουθο τύπο:

$$h_0 = 0.1 * \frac{\lambda'}{D^5} * l * Q_0^{1.7}$$

όπου: *h_0 = στατικό ύψος (mmH_2O)*

λ΄ = συντελεστής τριβής χωρίς διαστάσεις

Q_0 = Παροχή στην έξοδο του αγωγού (m^3/sec)

l = μήκος αγωγού (m)

D = είναι η ονομαστική διάμετρος του αγωγού (m)

Οι διαφυγές, όμως, του αέρα από τον αγωγό δεν επηρεάζουν μόνο την παροχή που φτάνει στο μέτωπο αλλά και την πίεση του αέρα. Έτσι, για να υπερνικηθούν οι εκτιμώμενες απώλειες πίεσης σε έναν πραγματικό αγωγό, η παρεχόμενη πίεση από τον ανεμιστήρα πρέπει να είναι προσαυξημένη, σύμφωνα με τον ακόλουθο τύπο:

$$h = h_0 * \left(1 + \frac{\alpha}{\beta + 2} * \frac{Q + Q_0}{Q_0}\right)$$

όπου: *h = παρεχόμενη πίεση ανεμιστήρα (mmH_2O)*

h_0 = στατικό ύψος (mmH_2O)

β = χαρακτηριστικός συντελεστής αγωγού β χωρίς διαστάσεις

Q_0 = Παροχή στην έξοδο του αγωγού (m^3/sec)

Q = Παροχή ανεμιστήρα (m^3/sec)

α = λαμβάνεται ίσος με 1,7

Τέλος, για τον υπολογισμό της πίεσης του ανεμιστήρα θα πρέπει να ληφθεί υπόψη και η απώλεια ενέργειας εξαιτίας της δυναμικής πιέσεως που επισυμβαίνει κατά την έξοδο του αέρα από το στόμιο του αγωγού. Η απώλεια αυτή πρέπει να προστεθεί στην υπολογιζόμενη στατική πίεση, ώστε να προκύψει η ολική απώλεια πιέσεως, η οποία βεβαίως αντιστοιχεί με την απαιτούμενη πίεση του ανεμιστήρα. Υπενθυμίζεται ότι ο υπολογισμός δυναμικού ύψους h_δ, πραγματοποιείται βάσει της ακόλουθης εξίσωσης:

$$h_\delta = 0.06 * v^2$$

όπου: *ν = ταχύτητα αέρα (m/s)*

h_δ = δυναμικού ύψους (mmH_2O)

Σύμφωνα με τα παραπάνω, η απαιτούμενη παροχή ισχύος του ανεμιστήρα υπολογίζεται από τον τύπο:

$$N = \frac{Q * h_{o\lambda}}{75 * \eta}$$

όπου: *hολ = πίεση ανεμιστήρα h + δυναμικό ύψος hδ σε (mmH_2O)*

η = Συντελεστής απόδοσης

Q = Παροχή ανεμιστήρα (m^3/sec)

N = Απαιτούμενη ισχύς ανεμιστήρα σε HP

4. ΟΙΚΟΝΟΜΙΚΗ ΑΞΙΟΛΟΓΗΣΗ ΚΥΚΛΩΜΑΤΩΝ ΒΟΗΘΗΤΙΚΟΥ ΑΕΡΙΣΜΟΥ

4.1. Γενικά

Η επιλογή του βέλτιστου συστήματος αερισμού μιας υπό διάνοιξη σήραγγας από πλευράς κόστους (με δεδομένη την ικανοποίηση των απαιτήσεων ασφάλειας από πλευράς επιτυγχανόμενης ποιότητας αέρα), αποτελεί ένα πρόβλημα, το οποίο δεν αντιμετωπίζεται μέχρι σήμερα, τουλάχιστον στην πλειοψηφία των περιπτώσεων, με την απαραίτητη προσοχή.

Η οικονομικότητα του συστήματος αερισμού εξαρτάται από:

- το κόστος αγοράς του αεραγωγού, το οποίο επίσης εξαρτάται από τα χαρακτηριστικά του αγωγού και του έργου (διάμετρος, ποιότητα, μήκος, κ.λπ.)
- το κόστος αγοράς του ανεμιστήρα, το οποίο αποτελεί συνάρτηση των χαρακτηριστικών του ανεμιστήρα και του έργου (ισχύς, διάμετρος, απαιτούμενη παροχή και πίεση, αυτοματισμοί, κ.λπ.) και
- το κόστος λειτουργίας του ανεμιστήρα (κόστος ενέργειας, συντήρηση, κ.λπ.).

Συχνά, το δίλημμα βρίσκεται ανάμεσα σε έναν ανεμιστήρα με χαμηλό κόστος αγοράς (ακόμη και μηδενικό, δεδομένου ότι σε πολλές περιπτώσεις χρησιμοποιούνται υπάρχοντες ανεμιστήρες) αλλά με υψηλό κόστος λειτουργίας ή σε έναν ανεμιστήρα με υψηλότερο κόστος αγοράς αλλά με χαμηλότερο κόστος λειτουργίας, ο οποίος μπορεί να αποδεικνύεται οικονομικότερος σε βάθος χρόνου.

Σύμφωνα με τα παραπάνω, προκειμένου να εξευρεθεί η βέλτιστη από οικονομικής πλευράς λύση, τα παραπάνω μεγέθη πρέπει να εξεταστούν συναρτήσει των ετήσιων ωρών εργασίας και της συνολικής διάρκειας του έργου. Στις ενότητες που ακολουθούν παρατίθενται συνοπτικά βασικές οικονομικές έννοιες για την καλύτερη κατανόηση των αρχών αξιολόγησης των εναλλακτικών κυκλωμάτων διαμόρφωσης, στη βάση των κριτηρίων της Καθαράς Παρούσας Αξίας (Net Present Value) και του Ετήσιου Ισοδύναμου Κόστους.

4.2. Βασικοί ορισμοί

4.2.1. Διαχρονική αξία του χρήματος

Το χρήμα έχει δύο χαρακτηριστικές ιδιότητες:

(α) την ιδιότητα να παράγει χρήμα (υπό μορφή νομισματικών μονάδων ή άλλων υλικών αγαθών) και

(β) την ιδιότητα να χάνει την αξία του.

Οι δύο αυτές ιδιότητες είναι σε άμεση συνάρτηση με *το* χρόνο.

Γενικά, ένα χρηματικό ποσό σήμερα έχει μεγαλύτερη χρησιμότητα και κατ' επέκταση μεγαλύτερη αξία από ένα ίδιο χρηματικό ποσό στο μέλλον. Αυτή η διαπίστωση σχετίζεται τόσο με τις βασικές αρχές του χρηματο-οικονομικού συστήματος όσο και με τις ανθρώπινες προτιμήσεις και συνοψίζεται στην επίδραση τριών παραμέτρων:

- του πληθωρισμού
- του κόστους ευκαιρίας και
- του ρίσκου.

Ο πληθωρισμός εκφράζει τη μείωση της αγοραστικής δύναμης του χρήματος, δηλ. το γεγονός ότι με την πάροδο του χρόνου με το ίδιο ποσό μπορούν να αγοραστούν ολοένα και λιγότερα αγαθά. Η πτώση της αξίας του χρήματος προκαλείται από την αύξηση των τιμών των διαφόρων αγαθών και για το λόγο αυτό ο δείκτης του πληθωρισμού μπορεί να εκτιμηθεί στην πράξη, καταγράφοντας τιε τιμές ενός συγκεκριμένου συνόλου καταναλωτικών αγαθών και υπηρεσιών (δείκτης τιμών καταναλωτή, δ), με τη βοήθεια του ακόλουθου τύπου:

$$f = \frac{\delta_1}{\delta_0} - 1$$

όπου δ_0 και δ_1 η αξία των αγαθών πριν από 1 έτος και σήμερα, αντίστοιχα.

Σημειώνεται ότι ο ρυθμός με τον οποίο το χρήμα χάνει την αξία του εξαιτίας του πληθωρισμού δεν είναι σταθερός για όλα τα αγαθά (ή τις υπηρεσίες).

Το κόστος ευκαιρίας γενικότερα αναφέρεται στη δέσμευση ενός πόρου σε μια συγκεκριμένη χρήση, η οποία έχει ως αποτέλεσμα την «εγκατάλειψη» άλλων εναλλακτικών επιλογών. Συχνά, το κόστος ευκαιρίας αναφέρεται στην αξία που παράγεται από έναν πόρο στην καλύτερη δυνατή εναλλακτική επιλογή.

Στην περίπτωση του χρήματος, το κόστος ευκαιρίας αναφέρεται συνήθως στην απώλεια μιας επενδυτικής ευκαιρίας, και κατά συνέπεια και του αντίστοιχου οφέλους, εξαιτίας της δέσμευσης των χρημάτων σε μια συγκεκριμένη επένδυση για ένα χρονικό διάστημα.

Το ρίσκο, τέλος, σχετίζεται με την αβεβαιότητα που υπάρχει στην πρόβλεψη των μελλοντικών συνθηκών, για παράδειγμα:

- στην αγορά των συντελεστών παραγωγής από πλευράς διαθεσιμότητας, τιμών, κ.λπ.
- στην αγορά του παραγόμενου προϊόντος, κυρίως από πλευράς ζήτησης, ανταγωνισμού που μπορεί να έχει επιπτώσεις στις τιμές, ανάπτυξης υποκατάστατων προϊόντων, κ.λπ.
- στο ευρύτερο οικονομικό περιβάλλον (π.χ. πληθωρισμός, θεσμικό πλαίσιο, πολιτική σταθερότητα, κ.ά.).

Η αντίληψη του ρίσκου είναι, τουλάχιστον σε κάποιο βαθμό, υποκειμενική και σχετίζεται με την ιδιοσυγκρασία του ατόμου. Έτσι, δεν είναι παράξενο δύο διαφορετικοί άνθρωποι να λάβουν, για την ίδια επένδυση, διαφορετική απόδοση ακόμη και αν στηρίζονται στα ίδια δεδομένα.

Οι παράμετροι αυτές λαμβάνονται υπόψη και συνυπολογίζονται στο επιτόκιο αναγωγής, το οποίο χρησιμοποιείται για τον υπολογισμό της μελλοντικής αξίας ενός σημερινού ποσού ή της σημερινής (ή παρούσας) αξίας ενός μελλοντικού ποσού. Στην περίπτωση αναγωγής ενός ποσού σε μελλοντική αξία, το επιτόκιο αναγωγής καλείται συχνά και επιτόκιο ανατοκισμού, ενώ στην περίπτωση υπολογισμού της παρούσας αξίας ενός ποσού, το επιτόκιο αναγωγής αναφέρεται ως επιτόκιο προεξόφλησης.

4.2.2. Υπολογισμός μελλοντικής και παρούσας αξίας ενός ποσού

Έστω ένα ποσό Κ, το οποίο επενδύεται σήμερα (χρόνος 0) με επιτόκιο αναγωγής ε. Η αξία που θα παράγει μετά από ένα έτος (δηλ. ο τόκος για το αντίστοιχο χρονικό διάστημα) θα είναι $K*\varepsilon$, το δε ποσό θα έχει ανέλθει σε $K + K * \varepsilon$ ή $K * (1 + \varepsilon)$.

Εάν ο τόκος δεν εισπραχθεί αλλά ενσωματωθεί στο αρχικό Κεφάλαιο (κεφαλαιοποίηση του τόκου), τότε το δεύτερο έτος ο τόκος θα ανέλθει σε $(K + K * \varepsilon) * \varepsilon$ ή $K * \varepsilon * (1 + \varepsilon)$. Η πρόσθεση του τόκου μιας χρονικής περιόδου στο κεφάλαιο και ο τοκισμός του καινούριου κεφαλαίου που πρόκυψε από την πρόσθεση αυτή (σύνθετος τόκος) καλείται και ανατοκισμός.

Το ποσό, στην περίπτωση αυτή, θα διαμορφωθεί ως ακολούθως:

$$K + K * \varepsilon + K * \varepsilon * (1 + \varepsilon) = K * [1 + \varepsilon + \varepsilon * (1 + \varepsilon)] = K * (1 + \varepsilon + \varepsilon + \varepsilon^2) = K * (1 + \varepsilon)^2$$

Ακολουθώντας την πρακτική της κεφαλαιοποίησης του τόκου, η τελική (ή μελλοντική) αξία του αρχικού ποσού *K* μετά από *t* έτη με ετήσιο επιτόκιο *ε* είναι:

$$MA_K = K * (1+\varepsilon)^t$$

Το επιτόκιο ε, όπως αναφέρθηκε καλείται επιτόκιο ανατοκισμού και ο συντελεστής $(1+\varepsilon)^t$ καλείται συντελεστής ανατοκισμού.

Από την παραπάνω εξίσωση της Μελλοντικής Αξίας είναι προφανές ότι εάν πρόκειται να πληρωθεί ένα ποσό *X* μετά από *t* έτη, τότε η αξία του ποσού σήμερα (τη χρονική στιγμή *0*), η οποία καλείται Παρούσα Αξία, θα είναι:

$$ΠA_X + X * (1+\varepsilon)^{-t}$$

Ο συντελεστής $(1+\varepsilon)^{-\nu}$ καλείται συντελεστής προεξόφλησης και το επιτόκιο *ε* επιτόκιο προεξόφλησης.

4.2.3. Υπολογισμός μελλοντικής και παρούσας αξίας περιοδικών χρηματοροών

Στην οικονομική αξιολόγηση των εναλλακτικών λύσεων για δοθεί μια περιγραφή των αναμενόμενων αποτελεσμάτων (εσόδων και εξόδων) χρησιμοποιούνται (συνήθως σε ετήσια βάση) σταθερές περιοδικές χρηματοροές ή «ράντες». Αντίστοιχα με τη μελλοντική ή την παρούσα αξία ενός ποσού, η μελλοντική και η παρούσα αξία μια σταθερής ράντας *P*, ληξιπρόθεσμης, για *ν* έτη και επιτόκιο *ε*, δίνεται από τους ακόλουθους τύπους:

α. Μελλοντική αξία ληξιπρόθεσμης ράντας

$$MA_P = P * \frac{(1+\varepsilon)^t - 1}{\varepsilon}$$

όπου: *P: η σταθερή περιοδική χρηματοροή*

ε: το επιτόκιο ανατοκισμού

t: η χρονική διάρκεια σε έτη

β. Παρούσα αξία ληξιπρόθεσμης ράντας

$$ΠΑ_P = P * \frac{(1+\varepsilon)^t - 1}{\varepsilon * (1+\varepsilon)^\nu} = P * \frac{1-(1+\varepsilon)^{-t}}{\varepsilon}$$

όπου: *P: η σταθερή περιοδική χρηματοροή*

ε: το επιτόκιο προεξόφλησης

t: η χρονική διάρκεια σε έτη

4.2.4. Συντελεστές ανάκτησης και συσσώρευσης κεφαλαίου

Ο συντελεστής ανάκτησης κεφαλαίου (ΣΑΚ) χρησιμοποιείται για να υπολογιστεί το ύψος μιας ετήσιας ληξιπρόθεσμης ράντας ώστε να κατανεμηθεί η παρούσα αξία ενός ποσού σε μια περίοδο t ετών. Ο ΣΑΚ πολλαπλασιαζόμενος με την παρούσα αξία του ποσού δίνει την ετήσια σταθερή δόση που θα πρέπει να καταβάλλεται. Η πιο συνηθισμένη χρήση του εν λόγω συντελεστή είναι για τον υπολογισμό της τοκοχρεολυτικής δόσης ενός δανείου.

Ο συντελεστής ανάκτησης κεφαλαίου υπολογίζεται από την εξίσωση:

$$ΣΑΚ = \frac{\varepsilon * (1+\varepsilon)^t}{(1+\varepsilon)^t - 1}$$

όπου: *ε: το επιτόκιο αναγωγής*

t: η χρονική διάρκεια σε έτη

Ο συντελεστής συσσώρευσης κεφαλαίου (ΣΣΚ) χρησιμοποιείται για να υπολογιστεί η ετήσια σταθερή ράντα που απαιτείται προκειμένου να συγκεντρωθεί ένα δεδομένο χρηματικό ποσό σε μια μελλοντική χρονική στιγμή (π.χ. πόσα χρήματα πρέπει να αποταμιεύει κάποιος κάθε χρόνο ώστε όταν συνταξιοδοτηθεί να έχει ένα Χ κεφάλαιο στη διάθεσή του). Ο ΣΣΚ πολλαπλασιαζόμενος με τη ζητούμενη τελική αξία του ποσού δίνει την ετήσια ισοδύναμη δόση.

Ο ΣΣΚ υπολογίζεται από την ακόλουθη εξίσωση:

$$\Sigma\Sigma K = \frac{\varepsilon}{(1+\varepsilon)^t - 1}$$

όπου: *ε: το επιτόκιο αναγωγής*

t: η χρονική διάρκεια σε έτη

4.3. Τα κριτήρια αξιολόγησης

4.3.1. Η Καθαρή Παρούσα Αξία

Η Καθαρά Παρούσα Αξία ορίζεται ως η παρούσα αξία των ετήσιων εισοδημάτων μείον τη παρούσα αξία των ετήσιων εξόδων, συμπεριλαμβανομένων των επενδύσεων. Στην πράξη, η ΚΠΑ υπολογίζεται ως η διαφορά των χρηματικών εισροών (καθαρών ταμειακών ροών μετά φόρων) μείον το κόστος των επενδύσεων, όπως, δίνεται από τον ακόλουθο τύπο:

$$K\Pi A = \left[\sum_{\tau=1}^{\nu} \frac{KTP_\tau}{(1+\varepsilon)^\tau}\right] - E_0$$

όπου: *ΚΠΑ = η Καθαρά Παρούσα Αξία του σχεδίου*

KTP_τ = η Καθαρή Ταμειακή Ροή το έτος τ

E_0 = η αρχική επένδυση το χρόνο τ=0

ν = η διάρκεια ζωής του επενδυτικού σχεδίου

ε = το επιτόκιο προεξόφλησης

Το επιτόκιο προεξόφλησης καθορίζεται από τον επενδυτικό φορέα με υποκειμενικά κατά βάση κριτήρια και εκφράζει είτε το κόστος κεφαλαίου της επιχείρησης είτε το ελάχιστο αποδεκτό επιτόκιο για να καλύψει τον κίνδυνο της επένδυσης έναντι μιας πιο ασφαλούς τοποθέτησης (π.χ. κρατικά ομόλογα).

Για την εφαρμογή του παραπάνω κριτηρίου είναι απαραίτητη η σύνταξη του πίνακα των ταμειακών ροών του σχεδίου. Η ταμειακή ροή του επενδυτικού σχεδίου ορίζεται ως το αλγεβρικό άθροισμα της ροής όλων των ετών της ζωής της επένδυσης. Για τους σκοπούς της συγκεκριμένης αξιολόγησης χρησιμοποιείται στον κώδικα που περιγράφεται στο επόμενο κεφάλαιο μια απλοποιημένη ταμειακή ροή, η οποία περιλαμβάνει το κόστος επένδυσης, το κόστος λειτουργίας και την υπολειμματική αξία του εξοπλισμού (η τελευταία ισούται με το αρχικό κόστος αγοράς μείον τις αποσβέσεις καθόλη τη διάρκεια του έργου). Δηλαδή η Καθαρή Ταμειακή Ροή (ΚΤΡ) του έτους t, ισούται με:

$$KTP_t = KE_t - K\Lambda_t + Y_t$$

όπου: *ΚΡΤ = Καθαρή Ταμειακή Ροή (€)*

ΚΕ = Κόστος επένδυσης (€)

ΚΛ = Κόστος λειτουργίας (€)

Υ = Υπολειμματική αξία (€)

4.3.2. Το Ετήσιο Ισοδύναμο Κόστος

Σε ορισμένες περιπτώσεις είναι χρήσιμη η εφαρμογή του κριτηρίου του ομοιόμορφου ετήσιου ισοδύναμου κόστους (uniform annual equivalent cost) ή απλά ετήσιου ισοδύναμου κόστους. Το κριτήριο αυτό χρησιμοποιείται επικουρικά με τη λογική ότι το χαμηλό κόστος λειτουργίας είναι ένα από τα κλειδιά της επιτυχίας. Για την εφαρμογή του κριτηρίου απαιτείται η αναγωγή όλων των σταθερών και μεταβλητών δαπανών, συμπεριλαμβανομένων των επενδυτικών, σε ετήσια βάση.

Για τον υπολογισμό του ετήσιου ισοδύναμου κόστους, υπάρχουν δύο βασικές προσεγγίσεις, οι οποίες δίδονται παρακάτω:

Προσέγγιση 1η:

Συνολικό ετήσιο ισοδύναμο κόστος = η παρούσα αξία της συνολικής ροής κόστους (έξοδα επένδυσης συν καθαρά κόστη λειτουργίας και συντήρησης) * συντελεστής ανάκτησης κεφαλαίου:

$$\textit{Συνολικό ετήσιο κόστος} = \left(\sum_{t=0}^{n} \frac{(C_t + OC_t)}{(1+\varepsilon)^t}\right) * \left(\frac{\varepsilon(1+\varepsilon)^n}{(1+\varepsilon)^n - 1}\right)$$

όπου: *t =0 είναι το έτος αναφοράς*

C_t = συνολικά έξοδα επένδυσης στην περίοδο t

OC_t = συνολικά καθαρά κόστη λειτουργίας και συντήρησης στην περίοδο t

ε = το επιτόκιο προεξόφλησης ανά περίοδο

n = η εκτιμώμενη διάρκεια ζωής του έργου σε έτη

Προσέγγιση 2η:

Συνολικό ετήσιο ισοδύναμο κόστος = (κόστος κεφαλαίου * συντελεστής ανάκτησης κεφαλαίου) + καθαρά ετήσια κόστη λειτουργίας και συντήρησης:

$$\textit{Συνολικό ετήσιο κόστος} = C_0 * \left(\frac{\varepsilon(1+\varepsilon)^n}{(1+\varepsilon)^n - 1}\right) + OC$$

όπου: *C_0 = η επένδυση το έτος 0 (έτος αναφοράς)*

ε = το επιτόκιο προεξόφλησης

n = η εκτιμώμενη διάρκεια ζωής του έργου σε έτη

OC = συνολικά καθαρά κόστη λειτουργίας και συντήρησης (σταθερά για κάθε έτος)

Στο πρόγραμμα που αναπτύχθηκε υιοθετείται για τους υπολογισμούς του ετήσιου ισοδύναμου κόστους η δεύτερη προσέγγιση.

4.4. Παράδειγμα υπολογισμού

Έστω ότι για κάποιο έργο απαιτείται ανεμιστήρας 55 kW, η τιμή αγοράς του οποίου υπολογίσθηκε σε 20.000 € και αγωγός διαμέτρου 1000 mm που η τιμή αγοράς του για μήκος 1500 μέτρα ανήλθε σε 15.000 €. Στο τέλος του τρίτου χρόνου υπολογίζεται η υπολειμματική αξία του ανεμιστήρα σε 8.000 € (γραμμική απόσβεση 20% ετησίως, επομένως η αξία του μειώνεται κατά 4.000 € κάθε έτος).

Όσον αφορά στο λειτουργικό κόστος, ο ανεμιστήρας θα λειτουργεί 8 h ανά ημέρα για 300 ημέρες ανά έτος. Το κόστος της ηλεκτρικής ενέργειας ανέρχεται σε 0,04 €/kWh. Η ετήσια κατανάλωση ρεύματος είναι: 55 kW * 8 h/d * 300 d/y = 132.000 kWh, συνεπώς το κόστος λειτουργίας ανέρχεται σε 132.000 kWh * 0,04 €/kWh = 5.280 €.

Στη συνέχεια υπολογίζεται η Παρούσα Αξία ταμειακών ροών και τέλος η Καθαρή Παρούσα Αξία με τις παραδοχές του πίνακα και με τον τρόπο που αναφέρθηκε στην ενότητα 4.2.4 η οποία εκτιμάται σε - 81.418 €

Η διάρκεια του έργου είναι τριετής και το επιτόκιο προεξόφλησης ανέρχεται σε 8%.

Τα αποτελέσματα αναφορικά με τον υπολογισμό της ΚΠΑ δίνονται στον ακόλουθο πίνακα.

Πίνακας 4.1.: Υπολογισμός ΚΠΑ κόστους αερισμού.

		Διάρκεια Έργου (έτη)			
		0	**1**	**2**	**3**
Κόστος αγοράς ανεμιστήρα (KW)		-20.000			
Κόστος αγοράς αεραγωγού		-15.000			
Λειτουργικό κόστος			-5.280	-5.280	-5.280
Υπολειμματική αξία ανεμιστήρα					8.000
Ταμειακές ροές		-35.000	-5.280	-5.280	2.720
Π.Α. ταμειακών ροών		-35.000	-4.889	-4.527	2.159

Η ΚΠΑ του κόστους αερισμού ανέρχεται σε 42.260 € περίπου.

Όσον αφορά στο ετήσιο ισοδύναμο κόστος, ο ΣΑΚ για ε = 8% και ν = 3 έτη υπολογίζεται σε 0,388. Συνεπώς, το ανηγμένο ετήσιο κόστος της επένδυσης των 35.000 € ανέρχεται σε 11.117 €. Σε αυτό το κόστος προστίθεται το ετήσιο κόστος λειτουργίας (5.280 €) και υπολογίζεται το συνολικό ετήσιο ισοδύναμο κόστος του αερισμού σε 16.400 € περίπου.

5. ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΕ ΠΕΡΙΒΑΛΛΟΝ JAVA

5.1. Η γλώσσας Java

5.1.1 Εισαγωγή

Η Java αποτέλεσε πηγή μεγάλων προσδοκιών όταν παρουσιάστηκε, εδώ και μια δεκαετία. Όταν η εταιρία Sun Microsystems παρουσίασε την Java ενσωματώνοντάς την σε εφαρμογές web browser, ένα πρωτοφανές ρεύμα δημοσιότητας καλωσόρισε την άφιξη της νέας γλώσσας.

Ο Bill Joy, συνιδρυτής της Sun, δεν υπερέβαλλε καθόλου όταν περιέγραψε τη νέα γλώσσα λέγοντας ότι «αντιπροσωπεύει το τελικό αποτέλεσμα προσπαθειών που διήρκεσαν σχεδόν 15 χρόνια, με στόχο τη δημιουργία μιας καλύτερης γλώσσας προγραμματισμού κι ενός καλύτερου περιβάλλοντος για την ανάπτυξη απλούστερου και πιο αξιόπιστου λογισμικού».

Στα χρόνια που πέρασαν από τότε, η Java δικαίωσε σε σημαντικό βαθμό τη φήμη της. Σήμερα, αποτελεί ένα ισχυρό και ευρύτατα χρησιμοποιούμενο εργαλείο ανάπτυξης λογισμικού.

Αρχικά η Java προτάθηκε σαν μία τεχνολογία για τον εμπλουτισμό ιστοθέσεων με προγράμματα τα οποία εκτελούνταν σε εφαρμογές περιήγησης (web browsers). Σήμερα είναι πιο πιθανό να τη βρει κανείς σε servers,να αποτελεί τη βάση δυναμικών web εφαρμογών που υποστηρίζονται από σχεσιακές βάσεις δεδομένων σε ορισμένες από τις μεγαλύτερες ιστοθέσεις στο Web.

Κάθε νέα έκδοση της Java βελτιώνει τις δυνατότητές της σαν μία γλώσσα προγραμματισμού γενικού σκοπού, κατάλληλη για πολλά άλλα περιβάλλοντα εκτός των εφαρμογών web browsers. Σήμερα, η Java χρησιμοποιείται σε εφαρμογές για επιτραπέζιους υπολογιστές, Internet servers, συσκευές PDA (προσωπικούς ψηφιακούς βοηθούς), ενσωματωμένες συσκευές και πολλά άλλα περιβάλλοντα.

Περισσότεροι από 3,5 εκατομμύρια προγραμματιστές έχουν μάθει τη γλώσσα και τη χρησιμοποιούν σε διάφορους φορείς και οργανισμούς όπως οι NASA, IBM, Kaiser Permanente και το Έργο Apache. Αποτελεί αναπόσπαστο μέρος της διδακτέας ύλης στο τμήμα της Επιστήμης των Υπολογιστών πολλών πανεπιστημίων, σε όλο τον κόσμο. Αν και αρχικά χρησιμοποιούνταν για τη δημιουργία απλών προγραμμάτων και ιστοσελίδων, σήμερα μπορεί κανείς να βρει την Java σε όλους τους ακόλουθους τομείς, αλλά και σε πολλούς άλλους:

- Web servers
- Σχεσιακές βάσεις δεδομένων
- Τηλεσκόπια που βρίσκονται σε τροχιά
- Προσωπικούς ψηφιακούς βοηθούς (PDA)
- Κινητά τηλέφωνα

Αν και η Java παραμένει χρήσιμη στους δημιουργούς εφαρμογών για το Web, εκτείνεται πλέον πολύ πέρα και έξω από το Web. Σήμερα, η Java είναι μία δημοφιλής γλώσσα προγραμματισμού γενικού σκοπού.

Η Java σχεδιάστηκε εξαρχής έτσι ώστε η εκμάθησή της να είναι ευκολότερη από την C++ κυρίως στα ακόλουθα σημεία:

- Η Java φροντίζει αυτόματα για τη δέσμευση και αποδέσμευση της μνήμης, απαλλάσσοντας τους προγραμματιστές από αυτήν την κοπιαστική και πολύπλοκη εργασία.
- Η Java δεν περιλαμβάνει δείκτες (pointers), ένα ισχυρό χαρακτηριστικό χρησιμοποιούμενο κυρίως από έμπειρους προγραμματιστές, το οποίο μπορεί εύκολα να τύχη εσφαλμένης χρήσης.
- Η Java υποστηρίζει μόνο μονή κληρονομικότητα (single inheritance) στον αντικειμενοστραφή προγραμματισμό.

Η έλλειψη δεικτών και η ύπαρξη της αυτόματης διαχείρισης της μνήμης είναι δύο βασικά στοιχεία της ασφάλειας στη Java.

5.1.2. Σύντομο ιστορικό της γλώσσας Java

Η ιστορία της γλώσσας Java είναι αρκετά γνωστή σήμερα. Στα μέσα της δεκαετίας του ’90, ο James Gosling και άλλοι προγραμματιστές στην Sun ασχολούνταν με τη δημιουργία μιας διαδραστικής τηλεόρασης, όταν ο Gosling απογοητεύτηκε από τη γλώσσα προγραμματισμού που χρησιμοποιούσαν (συγκεκριμένα τη C++, μια γλώσσα αντικειμενοστραφούς προγραμματισμού την οποία είχε αναπτύξει ο Bjarne Stroustrup στα Εργαστήρια της AT&T Bell δέκα χρόνια νωρίτερα, σαν μια επέκταση της γλώσσας C).

Ο Gosling κλείστηκε στο γραφείο του και δημιούργησε μία νέα γλώσσα η οποία ήταν κατάλληλη για το έργο που είχε αναλάβει και ταυτόχρονα έλυνε ορισμένα από τα προβλήματα που αντιμετώπιζε με την C++.

Η προσπάθεια της Sun να δημιουργήσει μία διαδραστική τηλεόραση απέτυχε, αλλά το έργο του Gosling πάνω στη νέα γλώσσα βρήκε αναπάντεχη εφαρμογή σ’ ένα νέο μέσο το οποίο μόλις άρχιζε να γίνεται δημοφιλές εκείνη την εποχή: το Web.

Η Java πρωτοπαρουσιάστηκε από τη Sun το φθινόπωρο του 1995. Αν και τα περισσότερα από τα χαρακτηριστικά της γλώσσας ήταν πρωτόγονα συγκριτικά με την C++ (και τη σημερινή Java), τα προγράμματα που γράφονταν με την Java, *οι αποκαλούμενες μικροεφαρμογές (applets)*, μπορούσαν να εκτελούνται σαν μέρος των ιστοσελίδων στην εφαρμογή περιήγησης Netscape Navigator.

Αυτή η λειτουργικότητα (η πρώτη μορφή διαδραστικού προγραμματισμού που υπήρξε ποτέ στο Web) βοήθησε στο να γίνει γνωστή η γλώσσα και να προσελκύσει αρκετές χιλιάδες προγραμματιστές στο πρώτο εξάμηνο της ζωής της.

Αφού πέρασε ο πρώτος καιρός και η Java έχασε το χαρακτηριστικό του νεωτερισμού, τα γενικά οφέλη της γλώσσας έγιναν σαφή και οι προγραμματιστές άρχισαν να την προτιμούν. Σήμερα υπάρχουν περισσότεροι επαγγελματίες προγραμματιστές Java συγκριτικά με τους προγραμματιστές σε C++.

5.1.3. Χαρακτηριστικά της Java

5.1.3.1. Αντικειμενοστραφής προγραμματισμός

Ο αντικειμενοστραφής προγραμματισμός (Object-Oriented Programming, OOP) είναι μία μεθοδολογία ανάπτυξης λογισμικού βάση της οποίας ένα πρόγραμμα θεωρείται σαν μία ομάδα αντικειμένων τα οποία συνεργάζονται μεταξύ τους προκειμένου να επιτευχθεί ο στόχος του προγράμματος. Κάθε αντικείμενο αποτελεί ένα ξεχωριστό μέρος του προγράμματος το οποίο αλληλεπιδρά με τα άλλα μέρη με συγκεκριμένους αυστηρά ελεγχόμενους τρόπους.

Τα αντικείμενα (objects) δημιουργούνται χρησιμοποιώντας πρότυπα τα οποία αποκαλούνται κλάσεις (classes). Η κλάση χρησιμοποιείται σαν *μήτρα* για την κατασκευή πανομοιότυπων αντιγράφων - αντικειμένων. Τα αντικείμενα - αντίγραφα, λέγονται στιγμιότυπα (instances) της κλάσης αυτής και η κατασκευή και αρχικοποίηση ενός από αυτά λέγεται instantiation. Οι κλάσεις ενσωματώνουν όλα τα χαρακτηριστικά ενός συγκεκριμένου συνόλου αντικειμένων. Σ’ ένα πρόγραμμα σε μία αντικειμενοστρεφή γλώσσα, δεν ορίζονται μεμονωμένα αντικείμενα, αντ’ αυτών ορίζονται κλάσεις οι οποίες χρησιμοποιούνται για τη δημιουργία των αντικειμένων.

Το αντικείμενο αυτό καθ' εαυτό είναι μια οντότητα στη μνήμη η οποία περιέχει δεδομένα καθώς και μεθόδους μέσω των οποίων μπορούμε να αλλάξουμε τα δεδομένα ή να επικοινωνήσουμε με το αντικείμενο. Αντίθετα η κλάση είναι απλώς ένα πρότυπο για την δημιουργία αντιγράφων του ίδιου αντικειμένου.

Μία κλάση μπορεί να περιέχει τα παρακάτω:

- Μεταβλητές
- Πίνακες
- Εκφράσεις και τελεστές
- Εντολές και μπλοκ
- Εντολές ελέγχου ροής
- Μεθόδους

Στον αντικειμενοστραφή προγραμματισμό θεωρείται ο κόσμος ως αντικείμενα που έχουν:

- Χαρακτηριστικά (τις μεταβλητές)
- Συμπεριφορές (τις μεθόδους)

α. Μεταβλητές

Κάθε μεταβλητή είναι μια θέση στη μνήμη και έχει όνομα, τύπο, μέγεθος και μια τιμή. Η τιμή τους μπορεί να αλλάξει σε οποιοδήποτε σημείο του προγράμματος γι' αυτό το λόγο ονομάζονται μεταβλητές. Για να δημιουργηθεί μία μεταβλητή θα πρέπει να της δοθεί ένα όνομα και να καθοριστεί ο τύπος των πληροφοριών που θα αποθηκεύει. Υπάρχουν οχτώ βασικοί τύποι δεδομένων: για την αποθήκευση ακέραιων τιμών (byte, short, int, long), αριθμών κινητής υποδιαστολής (float, double), χαρακτήρων (char) και λογικών τιμών (Boolean).

Υπάρχουν τρία είδη μεταβλητών στη Java: μεταβλητές υπόστασης, μεταβλητές κλάσης και τοπικές μεταβλητές.

- Οι μεταβλητές υπόστασης (instance variables) χρησιμοποιούνται για τον ορισμό των ιδιοτήτων ενός αντικειμένου.
- Οι μεταβλητές κλάσης (class variables) ορίζουν τις ιδιότητες μιας ολόκληρης κλάσης αντικειμένων και ισχύουν για όλες τις υποστάσεις της.
- Οι τοπικές μεταβλητές (local variables) χρησιμοποιούνται μέσα σε ορισμούς μεθόδων ή σε ακόμη μικρότερα μπλοκ προτάσεων μέσα σε μια μέθοδο. Μπορούν να χρησιμοποιηθούν μόνο κατά τη διάρκεια που εκτελείται η συγκεκριμένη μέθοδος ή μπλοκ προτάσεων από το διερμηνευτή Java. Από αυτό το σημείο και μετά παύουν να υπάρχουν.

β. Πίνακες

Οι πίνακες (arrays) είναι ένας τρόπος αποθήκευσης μιας λίστας στοιχείων τα οποία είναι του ίδιου τύπου δεδομένων, της ίδιας κλάσης, ή μιας κοινής πατρικής κλάσης. Κάθε στοιχείο της λίστας τοποθετείται στη δική του, αριθμημένη θέση, πράγμα το οποίο επιτρέπει την εύκολη προσπέλαση των πληροφοριών. Ο πίνακας δε δημιουργείται κατά τη δήλωσή του γιατί δεν έχει προσδιοριστεί το μέγεθός του. Κατά τη δήλωσή του υποδεικνύεται το αντικείμενο ή ο τύπος δεδομένων που θα περιέχει ο πίνακας και το

όνομά του. Για τη δημιουργία του πίνακα είναι απαραίτητη η δημιουργία ενός αντικειμένου πίνακα.

Μετά τη δημιουργία ενός πίνακα, είναι δυνατή η εισαγωγή, η ανάκτηση, η αλλαγή και ο έλεγχος των τιμών που υπάρχουν σε όλες τις θέσεις του πίνακα. Η προσπέλαση της τιμής μιας θέσης είναι εύκολη με τη χρήση ενός δείκτη που υποδεικνύει τη θέση αυτή.

Η Java δεν υποστηρίζει πολυδιάστατους πίνακες, αλλά μπορεί να επιτευχθεί η ίδια λειτουργικότητα δηλώνοντας έναν πίνακα πινάκων. Αυτοί οι πίνακες μπορούν επίσης να περιέχουν άλλους πίνακες, κ.ο.κ., για όσες διαστάσεις χρειάζονται.

γ. Εκφράσεις και τελεστές

Μία έκφραση είναι μία πρόταση η οποία μπορεί να παράγει μία τιμή και μπορεί να είναι οποιοσδήποτε συνδυασμός μεταβλητών, κυριολεκτικών και τελεστών. Οι εκφράσεις μπορούν επίσης να είναι κλήσεις μεθόδων, επειδή μία μέθοδος μπορεί να επιστρέφει μία τιμή στο αντικείμενο ή στην κλάση που την καλεί. Η τιμή που παράγει μία έκφραση αποκαλείται επιστρεφόμενη τιμή.

Οι τελεστές είναι ειδικά σύμβολα τα οποία χρησιμοποιούνται για μαθηματικές πράξεις, ορισμένους τύπους προτάσεων εκχώρησης και λογικές συγκρίσεις.

- Στη Java χρησιμοποιούνται πέντε τελεστές για την εκτέλεση των βασικών αριθμητικών πράξεων (+, -, *, /, % (ο τελεστής % παράγει το υπόλοιπο μιας διαίρεσης)). Κάθε τελεστής δέχεται δύο τελεστέους, έναν σε κάθε πλευρά του τελεστή. Ο τελεστής αφαίρεσης μπορεί επίσης να χρησιμοποιείται για να δημιουργήσει τον αρνητικό ενός τελεστέου, πράγμα το οποίο ισοδυναμεί με τον πολλαπλασιασμό αυτού του τελεστέου επί -1. Όταν εκτελούνται διαιρέσεις θα πρέπει να δίνεται προσοχή στο είδος των αριθμών που χρησιμοποιούνται. Εάν το αποτέλεσμα της διαίρεσης αποθηκεύεται σε μια ακέραια μεταβλητή, αυτό θα περικοπεί (θα στρογγυλοποιηθεί) στο επόμενο μικρότερο ακέραιο αριθμό, επειδή ο τύπος δεδομένων int δεν μπορεί να χειρίζεται αριθμούς κινητής υποδιαστολής.
- Η εκχώρηση μιας τιμής σε μία μεταβλητή είναι μία έκφραση, επειδή παράγει μία τιμή. Η χρήση μιας έκφρασης για την αλλαγή της τιμής μιας μεταβλητής

μπορεί να αντικατασταθεί με τη χρήση ενός τελεστή εκχώρησης τιμών (+=, -=, *=, /=). Έτσι η έκφραση x=x+y γίνεται x += y. Επίσης υπάρχουν ειδική τελεστές (++, --) για την πρόσθεση ή αφαίρεση του 1 από μία ακέραια μεταβλητή. Έτσι η έκφραση x=x+1 μπορεί να γίνει x = x++.

- Η Java έχει αρκετούς τελεστές για την εκτέλεση συγκρίσεων μεταξύ μεταβλητών, μεταβλητών και κυριολεκτικών, ή άλλων τύπων πληροφοριών σ' ένα πρόγραμμα. Αυτοί οι τελεστές (==, != (διάφορο), <, >, <=, >=) χρησιμοποιούνται σε εκφράσεις οι οποίες επιστρέφουν λογικές (Boolean) τιμές true ή false, ανάλογα με το εάν η πραγματοποιούμενη σύγκριση δίνει αποτέλεσμα αληθές ή ψευδές. Οι εκφράσεις που δίνουν σαν αποτέλεσμα λογικές (Boolean) τιμές μπορούν να συνδυάζονται μεταξύ τους για τη δημιουργία πιο πολύπλοκων εκφράσεων. Αυτό γίνεται με τη χρήση λογικών τελεστών (&&, ||), οι οποίοι χρησιμοποιούνται για τους λογικούς συνδέσμους AND (σύζευξη), OR (διάζευξη), XOR (αποκλειστική διάζευξη) και NOT (λογική άρνηση).

Όταν χρησιμοποιούνται περισσότεροι από ένας τελεστές σε μία έκφραση, η Java βασίζεται σε μία προκαθορισμένη προτεραιότητα για να καθορίσει τη σειρά με την οποία αποτιμώνται οι τελεστές. Σε πολλές περιπτώσεις, αυτή η προτεραιότητα καθορίζει την τελική τιμή της έκφρασης.

δ. Εντολές και μπλοκ

Μια πλήρης εκτέλεση μιας ή περισσοτέρων εκφράσεων καλείται εντολή (statement). Καμία ή περισσότερες εντολές κλεισμένες σε αγκύλες {} αποτελούν ένα μπλοκ. Επιπλέον ένα μπλοκ μπορεί να τοποθετηθεί μέσα σε άλλο μπλοκ. Είναι σημαντικό να αναφερθεί ότι με τη χρήση ενός μπλοκ δημιουργείται μία εμβέλεια – ένα πεδίο δράσης – για τις τοπικές μεταβλητές που δημιουργούνται μέσα στο μπλοκ. Εμβέλεια, ή πεδίο δράσης (scope), είναι το τμήμα ενός προγράμματος μέσα στο οποίο υπάρχει και μπορεί να χρησιμοποιηθεί μία μεταβλητή. Η χρήση μιας μεταβλητής έξω από την εμβέλειά της δημιουργεί σφάλμα. Στη Java η εμβέλεια μιας τοπικής μεταβλητής είναι το μπλοκ μέσα στο οποίο δημιουργήθηκε. Όταν μέσα σ' ένα μπλοκ δηλωθούν και χρησιμοποιηθούν τοπικές μεταβλητές, αυτές παύουν να υπάρχουν αφού ολοκληρωθεί η εκτέλεση του μπλοκ.

ε. Εντολές ελέγχου ροής

Ένα από τα βασικά στοιχεία οποιασδήποτε γλώσσας προγραμματισμού είναι ο τρόπος με τον οποίο δίνει σ' ένα πρόγραμμα τη δυνατότητα να λαμβάνει αποφάσεις. Αυτό γίνεται με τη χρήση ενός ειδικού τύπου εντολών οι οποίες αποκαλούνται εντολές ελέγχου ροής. Οι εντολές σε ένα πρόγραμμα εκτελούνται από πάνω προς τα κάτω με τη σειρά που έχουν διατυπωθεί. Οι εντολές ελέγχου ροής σπάνε τη ροή της εκτέλεσης εφαρμόζοντας έλεγχο συνθηκών, επαναλήψεις και διακλαδώσεις.

- Εντολές ελέγχου συνθηκών

  If, if-else, switch-case

  Η if χρησιμοποιεί μια λογική έκφραση για τον έλεγχο της συνθήκης – δηλαδή, για να αποφασίσει εάν μια πρόταση θα πρέπει να εκτελεστεί ή όχι. Εάν η έκφραση παράγει τιμή true, μία πρόταση, ή περισσότερες προτάσεις κλεισμένες μέσα σ' ένα μπλοκ, εκτελούνται.

  Η if-else χρησιμοποιεί μια λογική έκφραση για τον έλεγχο της συνθήκης. Εάν η έκφραση παράγει τιμή true, εκτελείται η πρόταση μετά το if, ενώ εάν η έκφραση παράγει τιμή false, εκτελείται η πρόταση μετά το else. Η switch χρησιμοποιείται για τη σύγκριση μιας μεταβλητής (που βρίσκεται στο switch) με τιμές (κάθε μία από τις οποίες βρίσκεται στα case). Η μεταβλητή ελέγχου (στο switch), η οποία μπορεί να είναι ενός από τους πρωταρχικούς τύπους byte, char, short ή int, συγκρίνεται με κάθε μία από τις τιμές των case, με τη σειρά. Εάν βρεθεί μία ταύτιση εκτελείται η πρόταση ή οι προτάσεις που ακολουθούν μετά από τον συγκεκριμένο έλεγχο. Εάν δε βρεθεί ταύτιση, εκτελείται η προεπιλεγμένη (default) πρόταση ή προτάσεις. Η παροχή μιας προεπιλεγμένης πρότασης είναι προαιρετική – εάν παραληφθεί και δεν υπάρχει ταύτιση για καμία από τις τιμές των case, η πρόταση switch ολοκληρώνεται χωρίς να εκτελέσει τίποτα.

  Η υλοποίηση της switch στην Java είναι περιορισμένη – οι έλεγχοι και οι τιμές μπορούν να είναι απλοί πρωταρχικοί τύποι δεδομένων που μπορούν να μετατρέπονται σε int. Δεν μπορούν να χρησιμοποιηθούν μεγαλύτεροι πρωταρχικοί τύποι όπως οι long ή float, string, ή άλλα αντικείμενα μέσα σε μία switch, ούτε και μπορεί να ελεγχθεί για οποιαδήποτε άλλη σχέση εκτός από την ισότητα. Αυτοί οι περιορισμοί δεν επιτρέπουν τη χρήση της switch παρά μόνο σε πολύ απλές περιπτώσεις. Εν αντιθέσει, οι ένθετες προτάσεις if δουλεύουν για

οποιοδήποτε είδος ελέγχου και οποιοδήποτε πιθανό τύπο δεδομένων. Βέβαια, η switch ενδείκνυται σε περιπτώσεις σύγκρισης μιας μεταβλητής με πολλές τιμές, κάτι που γίνεται γρήγορα κουραστικό με τη χρήση μόνο της if, ανάλογα με το πλήθος των τιμών που πρέπει να ελεγχθούν.

- Εντολές επαναλήψεων

  For, while, do-while

  Ένας βρόχος for χρησιμοποιείται για την επανάληψη μιας πρότασης μέχρι να ικανοποιηθεί μία συνθήκη. Αν και οι βρόχοι for χρησιμοποιούνται συχνά για απλές επαναλήψεις στις οποίες μια πρόταση επαναλαμβάνεται για ένα συγκεκριμένο αριθμό φορών, μπορούν να χρησιμοποιούνται για σχεδόν οποιοδήποτε είδος βρόχου. Ο βρόχος for περιλαμβάνει την αρχή, που έχει τρία μέρη, και την πρόταση. Η αρχή περιλαμβάνει την αρχικοποίηση, τον έλεγχο και την αύξηση. *Αρχικοποίηση*: είναι μια έκφραση η οποία αρχικοποιεί την έναρξη του βρόχου. Εάν υπάρχει μία μεταβλητή-δείκτης του βρόχου, θα μπορούσε να δηλώνεται και να αρχικοποιείται σ' αυτή την έκφραση. Οι μεταβλητές που δηλώνονται σ' αυτό το μέρος του βρόχου for είναι τοπικές για το βρόχο, παύουν να υπάρχουν αφού ολοκληρωθεί η εκτέλεση του βρόχου. *Έλεγχος:* είναι ο έλεγχος που λαμβάνει χώρα πριν από κάθε πέρασμα από το βρόχο. Ο έλεγχος πρέπει να είναι μια λογική έκφραση ή μια συνάρτηση η οποία επιστρέφει μία τιμή τύπου Boolean. Εάν ο έλεγχος είναι true, ο βρόχος εκτελείται. Όταν ο έλεγχος γίνει false, ο βρόχος σταματά να επαναλαμβάνεται. *Αύξηση:* είναι οποιαδήποτε έκφραση ή κλήση συνάρτησης. Συνήθως η αύξηση χρησιμοποιείται για την αλλαγή της τιμής του δείκτη του βρόχου, για να φέρει το βρόχο πλησιέστερα στην κατάσταση στην οποία επιστρέφει false, με αποτέλεσμα τον τερματισμό του. Η αύξηση συμβαίνει μετά από κάθε πέρασμα από το βρόχο. Παρόμοια με την ενότητα αρχικοποίησης μπορούν να τοποθετηθούν περισσότερες από μία εκφράσεις σ' αυτήν την ενότητα. Το τμήμα πρόταση του βρόχου for είναι η πρόταση που εκτελείται κάθε φορά που επαναλαμβάνεται ο βρόχος. Σ' αυτό το τμήμα μπορεί να χρησιμοποιηθεί είτε μία μεμονωμένη πρόταση, είτε μία πρόταση σε μορφή μπλοκ.

  Ο βρόχος while επαναλαμβάνει μία πρόταση για όσο χρόνο μία συγκεκριμένη συνθήκη παραμένει αληθής. Η συνθήκη που συνοδεύει τη δεσμευμένη λέξη while είναι μία λογική έκφραση. Όταν η έκφραση επιστρέφει true, ο βρόχος

while εκτελεί τις προτάσεις που περιλαμβάνει στον κορμό του και κατόπιν ελέγχει ξανά τη συνθήκη. Η διαδικασία αυτή επαναλαμβάνεται μέχρι η συνθήκη να γίνει false.

Ο βρόχος do-while μοιάζει με το βρόχο while με μια σημαντική διαφορά, τη θέση στην οποία ελέγχεται η συνθήκη μέσα στο βρόχο. Ένας βρόχος while ελέγχει τη συνθήκη πριν γίνει το πέρασμα από το βρόχο, οπότε εάν η συνθήκη είναι false την πρώτη φορά που ελέγχεται, ο κορμός του βρόχου δεν εκτελείται ποτέ. Ένας βρόχος do-while εκτελεί τις προτάσεις που περιέχονται στον κορμό του τουλάχιστον μία φορά πριν ελέγξει τη συνθήκη, οπότε εάν η συνθήκη είναι false την πρώτη φορά που ελέγχεται, ο κορμός του βρόχου θα έχει εκτελεστεί μία φορά.

- Εντολές διακλάδωσης

  Break, continue, return

  Η δεσμευμένη λέξη break, όταν χρησιμοποιείται σ' ένα βρόχο, τερματίζει άμεσα την εκτέλεση του τρέχοντος βρόχου. Εάν ο βρόχος όπου γίνεται το break είναι ένθετος μέσα σε άλλους βρόχους, η εκτέλεση συνεχίζεται με τον αμέσως επόμενο (πιο εξωτερικό) βρόχο. Αλλιώς, το πρόγραμμα συνεχίζει απλώς με την εκτέλεση της επόμενης εντολής μετά το βρόχο.

  Η δεσμευμένη λέξη continue ξεκινά την επόμενη επανάληψη του βρόχου. Για βρόχους while και do-while, αυτό σημαίνει ότι η εκτέλεση της πρότασης μπλοκ ξεκινάει από την αρχή, για βρόχους for, αποτιμάται η έκφραση αύξησης και κατόπιν εκτελείται η έκφραση μπλοκ.

  Οι break και continue μπορούν να έχουν μία προαιρετική ετικέτα η οποία λέει στη Java από ποιο σημείο να συνεχίσει την εκτέλεση του προγράμματος. Χωρίς μια ετικέτα η break από τον πλησιέστερο βρόχο μεταβαίνει στον βρόχο του αμέσως προηγούμενου (πιο εξωτερικού) επιπέδου, ή στην επόμενη πρόταση έξω από το βρόχο. Η continue επανεκκινεί το βρόχο μέσα στον οποίο περιέχεται. Η χρήση των break και continue με μια ετικέτα επιτρέπει τη χρήση της break για μετάβαση σ' ένα σημείο έξω από έναν ένθετο βρόχο και τη χρήση της continue για μετάβαση έξω από τον τρέχοντα βρόχο.

  Η return μεταφέρει την εκτέλεση του κώδικα έξω από την τρέχουσα μέθοδο αμέσως μετά την εντολή που την κάλεσε.

στ. Μέθοδοι

Μια μέθοδος είναι ένα κομμάτι κώδικα μέσα σε ένα μεγαλύτερο πρόγραμμα, που πραγματοποιεί μια συγκεκριμένη εργασία και είναι σχετικά ανεξάρτητη από τον υπόλοιπο κώδικα Οι μέθοδοι είναι το αντίστοιχο των συναρτήσεων, διαδικασιών, ρουτινών, υπορουτινών, υποπρογραμμάτων των άλλων γλωσσών προγραμματισμού Η δήλωση των μεθόδων περιέχει έξι στοιχεία κατά σειρά:

- Τροποποιητής πρόσβασης: π.χ. public, private
- Τύπος δεδομένων επιστροφής: π.χ. int, string, ή void για τίποτα
- Όνομα της μεθόδου: π.χ. Add, Connect, …
- Λίστα παραμέτρων σε παρενθέσεις: μια λίστα παραμέτρων εισόδου με τους τύπους δεδομένων τους, χωρισμένων με κόμματα, κλεισμένη σε παρενθέσεις
- Λίστα εξαιρέσεων: Μια λίστα με πιθανά προβλήματα που μπορεί να προκύψουν από την εκτέλεση της μεθόδου
- Σώμα της μεθόδου κλεισμένο σε άγκιστρα: ο κώδικας της μεθόδου, συμπεριλαμβανομένων των δηλώσεων.

5.1.3.2. Κληρονομικότητα (Inheritance)

Η κληρονομικότητα είναι ένας μηχανισμός ο οποίος επιτρέπει σε μία κλάση να κληρονομεί όλη τη συμπεριφορά και τις ιδιότητες μιας άλλης κλάσης. Μέσω της κληρονομικότητας, μία κλάση λαμβάνει άμεσα όλη τη λειτουργικότητα μιας υπάρχουσας κλάσης. Λόγω αυτού χρειάζεται να οριστεί μόνο πώς διαφέρει η νέα κλάση από την υπάρχουσα. Με την κληρονομικότητα, όλες οι κλάσεις –αυτές που δημιουργεί ο προγραμματιστής και αυτές που ανήκουν στη βιβλιοθήκη κλάσεων της Java και σε άλλες βιβλιοθήκες– οργανώνονται σε μία αυστηρή ιεραρχία. Μια κλάση η οποία κληρονομεί από μια άλλη κλάση αποκαλείται *υποκλάση* ή *δευτερεύουσα κλάση* (subclass). Η κλάση που παρέχει την κληρονομικότητα αποκαλείται υπερκλάση (superclass). Μια κλάση μπορεί να έχει μόνο μία υπερκλάση, αλλά κάθε κλάση μπορεί να έχει απεριόριστο αριθμό υποκλάσεων. Οι υποκλάσεις κληρονομούν όλες τις ιδιότητες και τη συμπεριφορά των υπερκλάσεών τους. Σε πρακτικό επίπεδο, αυτό σημαίνει ότι εάν η υπερκλάση έχει συμπεριφορά και ιδιότητες που χρειάζεται μία κλάση, δε χρειάζεται να την επαναπροσδιοριστεί ή να αντιγραφεί ο συγκεκριμένος κώδικας για να υλοποιηθούν οι ίδιες ιδιότητες και η ίδια συμπεριφορά στην κλάση αυτή. Η κλάση λαμβάνει αυτόματα αυτά τα

πράγματα από την υπερκλάση της, η οποία με τη σειρά της τα λαμβάνει από τη δική της υπερκλάση, κ.ο.κ., μέχρι την κορυφή της ιεραρχίας. Μια κλάση αποτελεί τον συνδυασμό των χαρακτηριστικών που ορίζονται σ' αυτήν και όλων των χαρακτηριστικών των κλάσεων που βρίσκονται πιο πάνω από αυτήν στην ιεραρχία. Στην κορυφή της ιεραρχίας κλάσεων της Java βρίσκεται η κλάση Object κι έτσι όλες οι κλάσεις κληρονομούν από αυτή την υπερκλάση. Η Object είναι η πιο γενική κλάση στην ιεραρχία και ορίζει συμπεριφορά η οποία κληρονομείται απ' όλες τις κλάσεις που περιέχονται στη βιβλιοθήκη κλάσεων της Java.

Κάθε κλάση που βρίσκεται σ' ένα χαμηλότερο επίπεδο της ιεραρχίας γίνεται πιο εξειδικευμένη για έναν συγκεκριμένο σκοπό. Μια ιεραρχία κλάσεων ορίζει αφηρημένες έννοιες στην κορυφή της. Αυτές οι έννοιες γίνονται πιο απτές κατεβαίνοντας στην ιεραρχία, στην αλυσίδα των υποκλάσεων.

Ο όρος δημιουργία υποκλάσεων (subclassing) αναφέρεται στη δημιουργία μιας νέας κλάσης η οποία κληρονομεί από μία υπάρχουσα κλάση. Για τη δήλωση της υπάρχουσας κλάσης που λειτουργεί ως υπερκλάση χρησιμοποιείται η δεσμευμένη λέξη extends. Η μόνη εργασία που πρέπει να γίνει στην υποκλάση είναι να καθοριστούν οι διαφορές στη συμπεριφορά και τις ιδιότητες μεταξύ αυτής και της υπερκλάσης της. Εάν η κλάση που δημιουργείται ορίζει εντελώς νέα συμπεριφορά και δεν είναι υποκλάση μιας άλλης κλάσης, μπορεί να οριστεί σαν απευθείας «απόγονο» της κλάσης Object. Αυτό επιτρέπει στην κλάση να ενταχθεί στην ιεραρχία κλάσεων της Java. Εάν δημιουργηθεί ένας ορισμός κλάσης στον οποίο δεν υποδεικνύεται καμιά υπερκλάση, η Java υποθέτει ότι η νέα κλάση κληρονομεί απευθείας από την κλάση Object.

Εάν δημιουργηθεί ένα μεγάλο σύνολο κλάσεων, είναι λογικό οι κλάσεις αυτές να κληρονομούν από την υπάρχουσα ιεραρχία κλάσεων και να σχηματίζουν από μόνες τους μια ιεραρχία επειδή αυτό παρέχει τα ακόλουθα πλεονεκτήματα:

- Η λειτουργικότητα που είναι κοινή σε πολλαπλές κλάσεις μπορεί να οριστεί στην υπερκλάση, πράγμα το οποίο επιτρέπει την επαναχρησιμοποίησή της σε όλες τις κλάσεις που βρίσκονται κάτω από αυτή στην ιεραρχία.
- Οι αλλαγές που γίνονται σε μία υπερκλάση αντικατοπτρίζονται αυτόματα σε όλες τις υποκλάσεις της, στις υποκλάσεις αυτών, κ.ο.κ. Δεν χρειάζεται να μεταβληθούν ή να μεταγλωττιστούν εκ νέου οι κλάσεις των κατωτέρων

επιπέδων της ιεραρχίας, αυτές λαμβάνουν τις νέες πληροφορίες μέσω της κληρονομικότητας.

Όταν δημιουργείται ένα νέο αντικείμενο, η Java παρακολουθεί κάθε μεταβλητή που έχει οριστεί γι' αυτό το αντικείμενο και κάθε μεταβλητή που έχει οριστεί για κάθε υπερκλάση αυτού του αντικειμένου. Με τον τρόπο αυτό, όλες οι κλάσεις συνδυάζονται για να σχηματίσουν ένα πρότυπο για το τρέχον αντικείμενο και κάθε αντικείμενο συμπληρώνει τις πληροφορίες που είναι κατάλληλες για την περίπτωσή του. Οι μέθοδοι λειτουργούν με παρόμοιο τρόπο. Ένα νέο αντικείμενο έχει πρόσβαση σε όλα τα ονόματα μεθόδων της κλάσης του και της υπερκλάσης της. Αυτό καθορίζεται δυναμικά, όταν χρησιμοποιηθεί μια μέθοδος κατά την εκτέλεση ενός προγράμματος. Εάν κληθεί μία μέθοδος ενός συγκεκριμένου αντικειμένου, ο διερμηνευτής Java την αναζητά κατ' αρχήν στην κλάση αυτού του αντικειμένου. Εάν η μέθοδος δε βρεθεί σ' αυτή την κλάση, ο διερμηνευτής την αναζητά στην υπερκλάση της κ.ο.κ., μέχρι να βρεθεί ο ορισμός της συγκεκριμένης μεθόδου.

Τα πράγματα αρχίζουν να περιπλέκονται όταν μία υποκλάση ορίζει μία μέθοδο η οποία ταιριάζει –στο όνομα και σε άλλες απόψεις– με μία μέθοδο ορισμένη σε μία υπερκλάση. Σ' αυτήν την περίπτωση, ο ορισμός της μεθόδου που εντοπίζεται πρώτος (ξεκινώντας από το κάτω άκρο της ιεραρχίας και προχωρώντας προς τα πάνω) είναι αυτός που θα χρησιμοποιηθεί. Λόγω αυτού μπορεί να δημιουργηθεί μία μέθοδος σε μία υποκλάση η οποία θα αποκλείει τη χρήση μιας μεθόδου σε μια υπερκλάση. Για το σκοπό αυτό θα πρέπει να δοθεί σ' αυτή τη μέθοδο το ίδιο όνομα, ο ίδιος τύπος επιστρεφόμενης τιμής και τα ίδια ορίσματα με τη μέθοδο της υπερκλάσης. Η διαδικασία αυτή αποκαλείται υπερκάλυψη (overriding).

Η μορφή κληρονομικότητας που υποστηρίζει η Java αποκαλείται *μονή κληρονομικότητα* (single inheritance) επειδή κάθε κλάση της Java μπορεί να έχει μόνο μία υπερκλάση (αν και οποιαδήποτε υπερκλάση μπορεί να έχει πολλαπλές υποκλάσεις).

5.1.3.3. Χειριστές Εξαιρέσεων *(Exceptions Handlers)*

Όταν σε ένα πρόγραμμα σε Java συμβεί κάποιο λάθος, για παράδειγμα περαστεί κάποια λάθος παράμετρος, τότε ο κώδικας που θα το ανιχνεύσει μπορεί να εγείρει (throw = πετάω) μία εξαίρεση! Η έγερση εξαίρεσης θα έχει ως αποτέλεσμα τον τερματισμό του

προγράμματος στο οποίο συνέβη το λάθος και θα τυπωθεί κάποιο μήνυμα. Ωστόσο τα προγράμματα μπορούν να ορίσουν *χειριστές εξαιρέσεων (exception handlers)* οι οποίοι θα *πιάνουν* την εξαίρεση και θα φροντίζουν ώστε το πρόγραμμα να ανανήψει από το λάθος.

Μερικές από τις εξαιρέσεις προκαλούνται από το runtime system, όπως στην περίπτωση διαίρεσης με το μηδέν. Ωστόσο, οποιαδήποτε κλάση μπορεί να ορίσει και να εγείρει δικές της εξαιρέσεις. Αυτό γίνεται ως εξής. Πρώτα δημιουργεί ένα αντικείμενο εξαίρεσης το οποίο θα πρέπει να είναι στιγμιότυπο της κλάσης Exception ή κάποιας υποκλάσης της. Έπειτα με την εντολή throw και το αντικείμενο εξαίρεσης προκαλείται exception. Η εκτέλεση του κώδικα θα διακοπεί στην εντολή throw και ο υπόλοιπος κώδικας που ακολουθεί δεν θα εκτελεστεί. Επίσης η μέθοδος μέσα στην οποία συνέβη η εξαίρεση δεν θα επιστρέψει κάποια τιμή. Το αντικείμενο, τώρα, της εξαίρεσης, (η εξαίρεση ουσιαστικά), θα δοθεί στον κατάλληλο exception handler, απ' όπου και συνεχίζεται η εκτέλεση του προγράμματος.

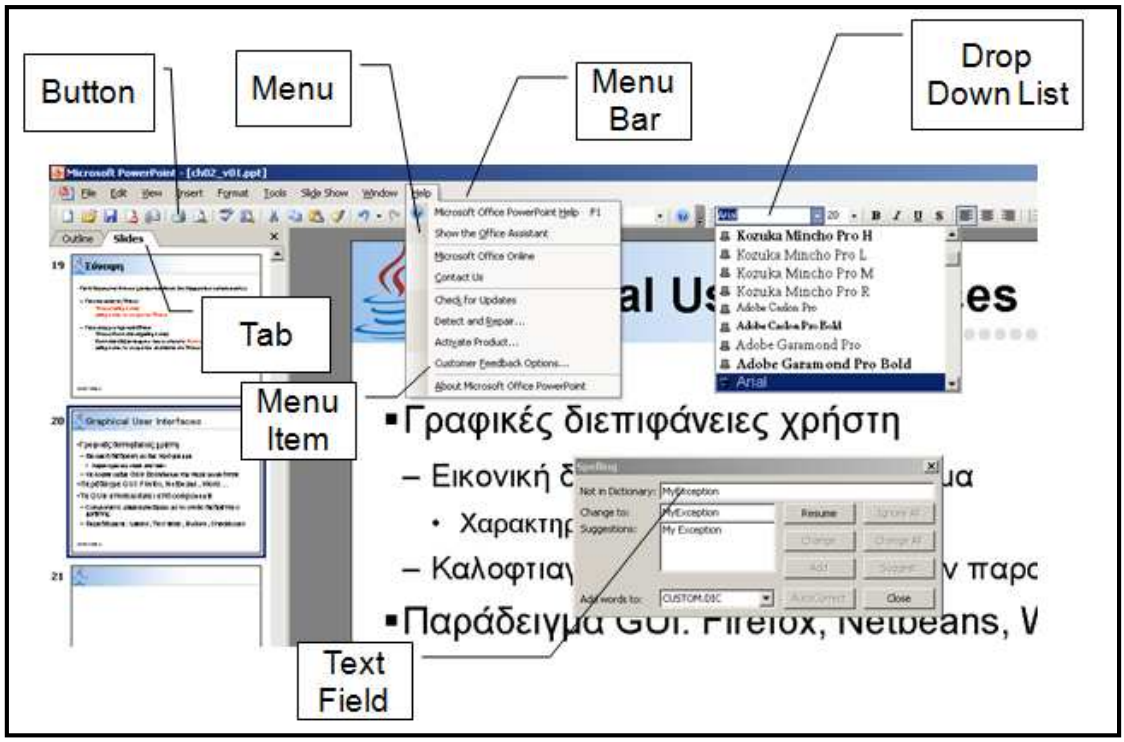
Για να ορίσουμε ένα χειριστή εξαιρέσεων (exception handler) θα πρέπει να κλείσουμε τον κώδικα που μπορεί να προκαλέσει την εξαίρεση μέσα σε μια εντολή try. Μετά την try θα πρέπει να βάλουμε μία ή περισσότερες εντολές catch. Κάθε catch θα μπορεί να πιάνει, μία μόνο κλάση εξαίρεσης (πχ MyException). Επίσης σε κάθε catch θα υπάρχει ο κατάλληλος κώδικας για τον χειρισμό της εξαίρεσης.

Οι χειριστές εξαιρέσεων μπορούν να φωλιαστούν επιτρέποντας έτσι ο χειρισμός να γίνει σε περισσότερα του ενός σημεία. Αυτό είναι χρήσιμο όταν ο πρώτος exception handler δεν μπορεί να διορθώσει πλήρως το λάθος. Προκειμένου, τώρα, να περαστεί ο χειρισμός της εξαίρεσης στον επόμενο, (υψηλότερου επιπέδου), exception handler, θα πρέπει να χρησιμοποιήσουμε την throw και για αντικείμενο εξαίρεσης θα δώσουμε την εξαίρεση που πιάσαμε. Σημειώστε ότι έπειτα από το throw η εκτέλεση του τρέχοντος κώδικα χειρισμού διακόπτεται οριστικά.

Υπάρχουν περιπτώσεις όπου θέλουμε ένα κομμάτι κώδικα να εκτελείται πάντα, ανεξάρτητα από το αν θα συμβεί κάποια εξαίρεση ή όχι. Αυτό το επιτυγχάνουμε με την εντολή *finally*.

5.1.3.4. Γραφικές Διεπιφάνειες Χρήστη (GUI)

Η γραφικές διεπιφάνιες χρήστη εξυπηρετούν την εικονική διάδραση με ένα πρόγραμμα, δηλαδή το χαρακτηριστικό “look and feel” για κάθε εφαρμογή. Παραδείγματα GUI είναι το Firefox, το Word κ.τ.λ. Τα GUIs αποτελούνται από components (συνιστώσες), τα οποία είναι αντικείμενα της Java και αποτελούν γενικά τα ορατά στοιχεία μιας γραφικής διεπιφάνειας χρήστη. Παραδείγματα components είναι τα Buttons (κουμπιά), τα Labels (ετικέτες), τα textfields (πεδία κειμένου) και τα textaerias (περιοχές κειμένου), frames (πλαίσια), drop-down lists (αναδιπλούμενες λίστες), checkboxes (πλαίσια ελέγχου) και radio buttons (στρογγυλά κουμπιά).



Εικόνα 5.1.: Παραδείγματα GUI.

Τα GUIs υλοποιούνται με τη χρήση Swing, AWT (Abstract Windowing Toolkit) και Συμβάντων (Events)

I. Η βιβλιοθήκη κλάσεων της Java περιλαμβάνει ένα σύνολο πακέτων τα οποία συλλογικά αποκαλούνται Swing και επιτρέπουν στις εφαρμογές Java να έχουν γραφικό περιβάλλον λειτουργίας (διεπιφάνεια χρήστη, user interface) και να συλλέγουν την είσοδο που τους παρέχει ο χρήστης μέσω ποντικιού, πληκτρολογίου και άλλων συσκευών εισόδων. Βασικά στοιχεία του Swing είναι τα components που αναφέρθηκαν παραπάνω και τα containers (περιεκτές), τα

οποία είναι και αυτά components. Τα components τοποθετούνται σε μία «οθόνη» «προσθέτοντάς» τα σε ένα container. Γενικά ένα container περιέχει ένα ή περισσότερα components και μπορεί να περιέχει και άλλα containers. Το γεγονός ότι ένα container μπορεί να διατηρεί όχι απλά components, αλλά και containers είναι κρίσιμο και βασικό στη δημιουργία layouts που είναι ρεαλιστικής πολυπλοκότητας. Το Swing επιτρέπει να δημιουργηθεί μια εφαρμογή Java με διεπιφάνεια η οποία χρησιμοποιεί το στυλ του λειτουργικού συστήματος στο οποίο τρέχει. Κάθε ένα από αυτά τα στυλ που μπορεί να έχει μία διεπιφάνεια αποκαλείται εμφάνιση και αίσθηση (look and feel), επειδή περιγράφει τόσο την εμφάνιση της διεπιφάνειας, όσο και το πώς λειτουργούν τα συστατικά της όταν χρησιμοποιούνται. Τα συστατικά του Swing περιλαμβάνονται στο πακέτο javax.swing, το οποίο είναι στάνταρ μέρος της βιβλιοθήκης κλάσεων της Java. Για να χρησιμοποιηθεί μία κλάση του Swing θα πρέπει να γίνει διαθέσιμη στο πρόγραμμα με μία πρόταση import. Πριν μπορέσουν τα συστατικά να εμφανιστούν σε μία διεπιφάνεια χρήστη πρέπει να προστεθούν σ' έναν container.

II. Η βιβλιοθήκη κλάσεων της Java περιλαμβάνει επίσης το πακέτο java.awt, το οποίο περιέχει κλάσεις για την παραγωγή συνιστωσών GUI. Το AWT χρησιμοποιεί τα components και containers με τον ίδιο τρόπο που τα χρησιμοποιεί το Swing. Το AWT όμως παρέχει τη δυνατότητα ορισμού της θέσης ενός συστατικού μέσα σ' ένα παράθυρο επακριβώς από τις x, y συντεταγμένες του. Η θέση ενός component σε ένα container καθορίζεται από το layout manager. Ένα container διατηρεί μία αναφορά σε ένα συγκεκριμένο στιγμιότυπο ενός LayoutManager. Όταν ένα container χρειάζεται να τοποθετήσει ένα component καλεί το layout manager για να το κάνει. Η ίδια εξουσιοδότηση (delegation) συμβαίνει και όταν θέλει να αποφασίσει το μέγεθος του component.

III. Για να είναι πλήρως λειτουργική η διεπιφάνεια χρήστη ενός προγράμματος Java, θα πρέπει να μπορεί να δέχεται συμβάντα (events) προκαλούμενα από το χρήστη. Τα Swing και AWT χειρίζονται τα συμβάντα χρησιμοποιώντας ένα σύνολο διασυνδέσεων (interfaces) που αποκαλούνται ακροατές συμβάντων (event listeners). Δημιουργείται ένα αντικείμενο ακροατή που συσχετίζεται με το συστατικό της διεπιφάνειας χρήστη που επιθυμείται η παρακολούθησή του. Για παράδειγμα, ένα κλικ με το ποντίκι σε ένα button component θα παράγει (θα

είναι η πηγή για) ένα ActionEvent. Το ActionEvent είναι ένα αντικείμενο (κλάση) που περιέχει πληροφορία σχετικά με την κατάσταση του συμβάντος. Όταν λαμβάνει χώρα ένα συμβάν, το συμβάν λαμβάνεται από ένα component με το οποίο ο χρήστης είχε αλληλεπίδραση. Για παράδειγμα, το button, slider, text field κλπ μέσα σε ένα frame. Ένας χειριστής συμβάντων (event handler) είναι η μέθοδος που λαμβάνει το αντικείμενο Event έτσι ώστε το πρόγραμμα να μπορεί να επεξεργαστεί την αλληλεπίδραση με το χρήστη. Όταν δημιουργείται μία κλάση ακροατή συμβάντων πρέπει να οριστεί ένας συγκεκριμένος τύπος συμβάντος που θα "ακούει" αυτή την κλάση. Ωστόσο, το συμβάν δε θα ακουστεί πραγματικά παρά μόνο αφού γίνει ένα δεύτερο βήμα: θα πρέπει να προστεθεί ο κατάλληλος ακροατής στο συστατικό (διαφορετικός για κάθε είδος συμβάντων). Αυτός ο ακροατής παράγει τα συμβάντα όταν χρησιμοποιείται το συστατικό. Το πλαίσιο λειτουργίας της παρακολούθησης (listening) των AWT συμβάντων επιτρέπει να προσκολληθούν πολλοί listeners στο ίδιο component.

Επειδή οι κλάσεις του Swing και οι κλάσεις του AWT κληρονομούν πολλές από τις ίδιες υπερκλάσεις, είναι δυνατό να χρησιμοποιούνται συστατικά του Swing και του AWT μαζί, στην ίδια διεπιφάνεια. Ωστόσο σε ορισμένες περιπτώσεις οι δύο τύποι συστατικών δεν αποδίδονται σωστά όταν τοποθετούνται μέσα σ' ένα αντικείμενο container. Για να αποφευχθούν τέτοια προβλήματα, είναι προτιμότερο να χρησιμοποιούνται πάντα τα συστατικά του Swing – υπάρχει ένα συστατικό του Swing για κάθε συστατικό του AWT.

5.1.3.5. Χρήση Stream (είσοδος και έξοδος)

Στη Java, η αποθήκευση και ανάκτηση πληροφοριών μπορεί να γίνεται με τη χρήση ενός ειδικού συστήματος επικοινωνίας, των αποκαλούμενων streams, τα οποία υλοποιούνται στο πακέτο java.io. Στη Java, όλα τα δεδομένα γράφονται και διαβάζονται με τη χρήση streams. Ένα stream είναι μία διαδρομή την οποία διανύουν τα δεδομένα σ' ένα πρόγραμμα. Ένα stream εισόδου στέλνει δεδομένα από τη μία πηγή σ' ένα πρόγραμμα, ένα stream εξόδου στέλνει δεδομένα από ένα προορισμό. Υπάρχουν διάφορα είδη stream όπως byte streams και streams χαρακτήρων. Τα byte streams μεταφέρουν ακέραιους με τιμές που κυμαίνονται από 0 έως 255. Υπάρχουν πολλά διαφορετικά είδη δεδομένων που μπορούν να εκφραστούν σε μορφή byte, συμπεριλαμβανομένων των αριθμητικών δεδομένων, των εκτελέσιμων προγραμμάτων, των επικοινωνιών στο Internet και της

μορφής bytecode (των αρχείων κλάσης που εκτελούνται από μία εικονική μηχανή Java). Στην πραγματικότητα κάθε είδος δεδομένων μπορεί να εκφραστεί χρησιμοποιώντας είτε μεμονωμένα bytes, είτε μια σειρά από bytes που συνδυάζονται μεταξύ τους. Τα streams χαρακτήρων είναι ένας εξειδικευμένος τύπος byte stream που χειρίζεται μόνο δεδομένα κειμένου. Αυτά τα streams διαφοροποιούνται από τα byte streams επειδή το σετ χαρακτήρων της Java υποστηρίζει το Unicode, ένα πρότυπο που περιλαμβάνει πολύ περισσότερους χαρακτήρες απ' όσους θα μπορούσαν να εκφραστούν να εκφραστούν εύκολα με bytes. Οποιοδήποτε είδος δεδομένων περιλαμβάνει κείμενο θα πρέπει να χρησιμοποιεί streams χαρακτήρων, συμπεριλαμβανομένων των αρχείων κειμένου, των ιστοσελίδων και των άλλων κοινών τύπων κειμένου.

Στο πακέτο java.io ορισμένα streams είναι «κομβικά» streams, δηλαδή διαβάζουν από ή γράφουν σε ένα συγκεκριμένο σημείο όπως ένα αρχείο στο δίσκο ή σε μία περιοχή μνήμης. Άλλα streams καλούνται φίλτρα. Ένα stream εισόδου φίλτρο δημιουργείται με μία σύνδεση σε ένα υπάρχον stream εισόδου, έτσι ώστε όταν γίνεται ανάγνωση από ένα stream εισόδου φίλτρο αυτό παρέχει τους χαρακτήρες που έχουν αρχικά προέλθει από ένα άλλο stream εισόδου.

5.1.3.6. Δικτύωση

Για εφαρμογές δικτύωσης με δυνατότητες πέραν αυτών που παρέχουν οι κλάσεις URL και URL Connection (π.χ. για άλλα πρωτόκολλα ή πιο γενικές εφαρμογές δικτύωσης), η Java παρέχει τις κλάσεις Socket και Server Socket, σα μία αφηρημένη αναπαράσταση των στάνταρ τεχνικών προγραμματισμού socket του πρωτοκόλλου TCP (Transmission Control Protocol, πρωτόκολλο ελέγχου μετάδοσης). Τα "sockets" είναι το όνομα που δίνεται, σε ένα συγκεκριμένο προγραμματιστικό μοντέλο, στα τελικά άκρα των συνδέσεων επικοινωνίας ανάμεσα στις διεργασίες. Ένα socket είναι μία αξιόπιστη σύνδεση για τη μετάδοση δεδομένων μεταξύ δύο κόμβων. Η κλάση Socket παρέχει μία διασύνδεση socket (socket interface) στην πλευρά του client συστήματος. Δημιουργείται μια νέα υπόσταση της κλάσης Socket για το άνοιγμα μιας σύνδεσης, όπου το hostName είναι ο κεντρικός υπολογιστής (host) με τον οποίο θα γίνει η σύνδεση και portNumber ο αριθμός θύρας. Το hostName είναι η IP διεύθυνση, ή το όνομα του κεντρικού υπολογιστή και η θύρα (port) είναι μία αφαιρετική αναπαράσταση της μνήμης του υπολογιστή για την αντιστοίχηση εισερχόμενων δεδομένων σε διαφορετικές υπηρεσίες με έναν αριθμό από το

1 έως το 65535. Τα 1023 πρώτα ports είναι δεσμευμένα, τα πιο γνωστά ports είναι το 20 που αντιστοιχεί σε FTP δεδομένα, το 25 που αντιστοιχεί σε smtp και το 80 που αντιστοιχεί σε HTTP. Τα sockets στην πλευρά του server (διακομιστή) δουλεύουν παρόμοια με τα sockets στην πλευρά του client (πελάτη), με εξαίρεση της μεθόδου accept(). Ένα server socket ακροάζεται τη θύρα του TCP, αναμένοντας μια σύνδεση από έναν client. Όταν ένας client συνδεθεί σ' αυτή τη θύρα, η μέθοδος accept() δέχεται τη σύνδεση από αυτό το client σύστημα. Χρησιμοποιώντας σε συνδυασμό client και server sockets, μπορούν να δημιουργούνται εφαρμογές ικανές να επικοινωνούν με servers και clients μέσω του διαδικτύου.

Όταν οι διεργασίες επικοινωνούν πάνω από ένα δίκτυο, η Java χρησιμοποιεί και πάλι τα streams. Στην περίπτωση αυτή, ένα socket κατέχει δύο streams, ένα stream εισόδου και ένα stream εξόδου. Μία διεργασία στέλνει τα δεδομένα της σε μία άλλη διεργασία μέσω του δικτύου γράφοντας απλά στο stream εξόδου που σχετίζεται με το socket. Η διεργασία διαβάζει δεδομένα που γράφονται από τη διεργασία στο «άλλο άκρο» της σύνδεσης διαβάζοντας απλά από το stream εισόδου που σχετίζεται με το socket.

5.1.3.7. Δημιουργία Ανεξάρτητων Εφαρμογών και Applets

Τα Java Applets είναι προγράμματα Java τα οποία είναι σχεδιασμένα για να εκτελούνται από ένα πρόγραμμα πλοήγησης (browser). Ένα applet είναι ένα κομμάτι κώδικα Java που εκτελείται σε ένα περιβάλλον πλοηγού (browser). Διαφέρει από μία εφαρμογή στον τρόπο με τον οποίο εκτελείται. Μία εφαρμογή ξεκινά όταν καλείται η μέθοδος main(). Αντίθετα ο κύκλος ζωής ενός applet είναι μάλλον περισσότερο πολύπλοκος.

Επειδή ένα applet εκτελείται σε ένα Web browser δεν ξεκινά άμεσα πληκτρολογώντας μία εντολή. Αντ' αυτού πρέπει να δημιουργηθεί ένα αρχείο τύπου HTML που θα λέει στον browser τι να φορτώσει και πώς να το εκτελέσει. Στη συνέχεια εισάγεται στο browser το URL που καθορίζει το αρχείο HTML.

Επειδή τα applets είναι κομμάτια κώδικα που φορτώνονται πάνω από το δίκτυο, αναπαριστούν μία εγγενώς επικίνδυνη κατάσταση. Ένας σοβαρός κίνδυνος είναι κάποιος να γράψει ένα «κακό» πρόγραμμα που διαβάζει το αρχείο με τους κωδικούς ενός χρήστη και να το αποστέλλει μέσω Internet Ο τρόπος με τον οποίο η Java αποτρέπει κάτι τέτοιο είναι παρέχοντας μία κλάση SecurityManager που ελέγχει την πρόσβαση σε σχεδόν κάθε

κλήση επιπέδου συστήματος στη Java Virtual Machine. Το μοντέλο αυτό ονομάζεται μοντέλο ασφαλείας "sandbox" – η JVM παρέχει ένα sandbox που απαγορεύει στα applets να βγουν έξω από το sandbox τους. Το βάθος στο οποίο ελέγχεται η ασφάλεια υλοποιείται σε επίπεδο browser. Οι περισσότεροι browsers (συμπεριλαμβανομένου του Netscape) αποτρέπουν τα ακόλουθα:

- Εκτέλεση άλλου προγράμματος κατά το χρόνο εκτέλεσης
- Οποιαδήποτε είσοδο/έξοδο σε αρχείο
- Κλήσεις σε οποιαδήποτε ιθαγενή μέθοδο (native method)
- Προσπάθειες να ανοίξει ένα socket σε οποιαδήποτε άλλη διεργασία που δε βρίσκεται στον υπολογιστή που παρείχε το applet.

5.1.3.8. Διαπλατφορμική (Interpreted) Γλώσσα

Τα προγράμματα που γράφονται με Java έχουν τη δυνατότητα να εκτελούνται χωρίς να απαιτούνται τροποποιήσεις τους για διαφορετικά περιβάλλοντα υπολογιστών. Πιο αναλυτικά, τα προγράμματα αυτά μεταγλωττίζονται σε μια μορφή η οποία αποκαλείται bytecode και δεν είναι εκτελέσιμη απευθείας από μία μηχανή αλλά εκτελείται από την εικονική μηχανή Java (Java Virtual Machine – JVM). Αυτό δίνει στην Java το πλεονέκτημα της ανεξαρτησίας από την συγκεκριμένη μηχανή στην οποία εκτελείται. Δηλαδή η μορφή αυτή bytecode μπορεί να εκτελείται σε οποιοδήποτε λειτουργικό σύστημα, λογισμικό ή συσκευή που διαθέτει έναν διερμηνευτή (interpreter) Java. Είναι δυνατή λοιπόν η δημιουργία ενός προγράμματος Java σ' έναν υπολογιστή με τα Windows Vista το οποίο θα εκτελείται επίσης σ' έναν web server με λειτουργικό Linux, σ' έναν Mac της Apple που χρησιμοποιείται το λειτουργικό OS X, ή σε μία συσκευή PDA (προσωπικό ψηφιακό βοηθό) Palm. Εφόσον η πλατφόρμα διαθέτει έναν interpreter Java, μπορεί να εκτελέσει το πρόγραμμα μορφής bytecode. Αυτή η επινόηση έμεινε γνωστή σαν «Γράψε μια φορά για να τρέχει παντού» (Write Once – Run Everywhere).

5.1.3.9. Συλλογή Απορριμμάτων

Πολλές γλώσσες προγραμματισμού επιτρέπουν τη δυναμική δέσμευση μνήμης κατά την εκτέλεση. Η διαδικασία της δέσμευσης μνήμης ποικίλει ανάλογα με το συντακτικό της γλώσσας, αλλά πάντα έχει ως αποτέλεσμα ένα δείκτη σε μία αρχική διεύθυνση ενός μπλοκ

μνήμης. Από τη στιγμή που η δεσμευμένη μνήμη δε χρειάζεται πλέον (ο δείκτης που αναφέρεται στη μνήμη βρίσκεται *εκτός πεδίου εμβέλειας*) είναι σώφρον για το πρόγραμμα ή το περιβάλλον εκτέλεσης να αποδεσμεύει τη μνήμη ώστε να αποτραπεί η κατάσταση κατά την οποία το πρόγραμμα (ή το σύστημα) δεν έχει άλλη διαθέσιμη μνήμη.

Στη C και τη C++ (αλλά και σε άλλες γλώσσες) αυτός που αναπτύσσει το πρόγραμμα έχει και την ευθύνη για την αποδέσμευση της μνήμης. Αυτό μπορεί σε ορισμένες περιπτώσεις να φανεί αρκετά δύσκολο, καθώς δεν είναι πάντα γνωστό εκ των προτέρων πότε θα πρέπει να αποδεσμευτεί η μνήμη. Τα προγράμματα που δεν αποδεσμεύουν μνήμη μπορούν τελικά να καταρρεύσουν όταν δεν υπάρχει άλλη διαθέσιμη μνήμη προς δέσμευση στο σύστημα. Τα προγράμματα αυτά λέμε ότι έχουν «διαρροή μνήμης» (memory leak).

Η Java απαλλάσσει τον προγραμματιστή από την ευθύνη της αποδέσμευσης της μνήμης, παρέχοντας ένα νήμα (thread) επιπέδου συστήματος το οποίο εκτελείται στο παρασκήνιο που καταγράφει κάθε δέσμευση μνήμης και διατηρεί ένα μετρητή του πλήθους των αναφορών σε κάθε δείκτη στη μνήμη. Κατά τη διάρκεια των κύκλων χωρίς εκτέλεση άλλης εντολής στο σύστημα εκτέλεσης της JVM, το νήμα συλλογής απορριμμάτων (garbage collection thread) θα ελέγχει για να δει αν υπάρχουν δείκτες στη μνήμη των οποίων το πλήθος των αναφορών έχει μηδενιστεί. Αν υπάρχουν, το τμήμα αυτό της μνήμης που έχει σημειωθεί από το νήμα συλλογής απορριμμάτων «σκουπίζεται» (αποδεσμεύεται). Η συλλογή απορριμμάτων γίνεται αυτόματα κατά τη διάρκεια ζωής ενός προγράμματος Java και απαλείφει την ανάγκη για αποδέσμευση μνήμης και τις διαρροές μνήμης.

5.1.3.10. Κατανεμημένη (Distributed)

Δηλαδή ένα πρόγραμμα σε Java είναι δυνατό να έρθει από το δίκτυο και να εκτελεστεί από έναν προσωπικό υπολογιστή. Επίσης είναι δυνατό διαφορετικά κομμάτια του προγράμματος να έρθουν από διαφορετικά sites.

5.1.3.11. Ασφάλεια Κώδικα

Τα αρχεία Java «μεταγλωττίζονται» με την έννοια ότι μετατρέπονται από τη μορφή κειμένου στην οποία τα γράφουν οι προγραμματιστές σε ένα σύνολο από κώδικα byte, ανεξάρτητο από τη φυσική αρχιτεκτονική (το υλικό – hardware) του υπολογιστή. Κατά

την εκτέλεση ο κώδικας byte που αποτελεί το πρόγραμμα Java φορτώνεται, ελέγχεται και στη συνέχεια εκτελείται σε ένα διερμηνέα. Ο διερμηνέας έχει δύο εργασίες: εκτελεί τον κώδικα byte της Java και κάνει τις απαραίτητες κλήσεις μέσω του συστήματος εκτέλεσης στο υποκείμενο υλικό. Σε ορισμένα περιβάλλοντα εκτέλεσης της Java μέρος του επιβεβαιωμένου κώδικα byte μεταγλωττίζεται σε ιθαγενή (native) κώδικα μηχανής και εκτελείται άμεσα από την πλατφόρμα υλικού.

Ο διερμηνέας Java πρέπει να εκτελεί τον κώδικα που μεταγλωττίζεται για τη JVM. Ένας διερμηνέας Java έχει τρεις βασικές εργασίες:

- Φόρτωση του κώδικα – πραγματοποιείται από το φορτωτή κώδικα (class loader).
- Επιβεβαίωση κώδικα – πραγματοποιείται από τον επιβεβαιωτή κώδικα byte (byte code verifier).
- Εκτέλεση κώδικα – πραγματοποιείται από τον διερμηνέα χρόνου εκτέλεσης (runtime interpreter).

Ο *φορτωτής κώδικα* φορτώνει όλες τις κλάσεις που απαιτούνται για την εκτέλεση του προγράμματος. Ο φορτωτής κώδικα προσθέτει ασφάλεια διαχωρίζοντας τις περιοχές ονομάτων για τις κλάσεις που προέρχονται από το τοπικό σύστημα αρχείων από εκείνες που είναι για τις κλάσεις που προέρχονται από το δίκτυο. Αυτό περιορίζει πιθανές εφαρμογές «Δούρειους Ίππους» (Trojan horses), γιατί ελέγχονται πάντα πρώτα οι ενσωματωμένες (τοπικές) κλάσεις. Όταν έχουν φορτωθεί όλες οι κλάσεις καθορίζεται η μορφή που πρέπει να έχει η μνήμη για το εκτελέσιμο αρχείο. Στο σημείο αυτό, ανατίθενται σε συγκεκριμένες διευθύνσεις μνήμης συμβολικές αναφορές και δημιουργείται ο πίνακας ευρετηρίου (lookup table) των μεθόδων των κλάσεων. Επειδή η διαμόρφωση της μνήμης λαμβάνει χώρα κατά το χρόνο εκτέλεσης, ο διερμηνέας Java προσθέτει προστασία ενάντια σε μη νόμιμη διευθυνσιοδότηση του κώδικα που θα μπορούσε να κάνει το πρόγραμμα να απευθύνεται σε χώρους μνήμης που κατέχει το λειτουργικό σύστημα και να δημιουργηθούν προβλήματα.

Ο κώδικας Java περνάει από διάφορους ελέγχους προτού να εκτελεστεί πραγματικά στον υπολογιστή. Το πρόγραμμα εκτελεί τον κώδικα μέσω ενός επιβεβαιωτή κώδικα byte (όλα τα αρχεία κλάσεων που εισάγονται μέσω δικτύου περνάνε από τον τοπικό επιβεβαιωτή κώδικα byte) που ελέγχει τη διαμόρφωση των τμημάτων του κώδικα και εφαρμόζει ένα

πρόγραμμα απόδειξης θεωρημάτων για να ελέγξει με βάση κανόνες τα τμήματα του κώδικα για μη επιτρεπτό κώδικα – κώδικα που παραβιάζει τους δείκτες, παραβιάζει τα δικαιώματα προσπέλασης στα αντικείμενα ή προσπαθεί να αλλάξει τον τύπο των αντικειμένων ή των κλάσεων.

Ο *επιβεβαιωτής κώδικα* byte κάνει τέσσερα περάσματα στον κώδικα ενός προγράμματος. Επιβεβαιώνει ότι ο κώδικας υπακούει στις προδιαγραφές της JVM και δεν παραβιάζει την ακεραιότητα του συστήματος. Αν ο επιβεβαιωτής ολοκληρώσει και τα τέσσερα περάσματα χωρίς να επιστρέψει μήνυμα λάθους τότε είναι σίγουρα τα ακόλουθα:

- Οι κλάσεις υπακούουν στη διαμόρφωση των αρχείων κλάσεων που έχουν τεθεί από τις προδιαγραφές της JVM.
- Δεν υπάρχουν παραβιάσεις στους περιορισμούς πρόσβασης.
- Ο κώδικας δεν θα δημιουργήσει υπερχειλίσεις (overflow) ή υποχειλίσεις (underflow) της στοίβας.
- Οι τύποι των παραμέτρων όλου του κώδικα που κάνει πράξεις είναι γνωστό ότι είναι πάντα ορθοί.
- Δεν λαμβάνουν χώρα μη νόμιμες μετατροπές δεδομένων, όπως η μετατροπή ακεραίων σε δείκτες.
- Οι προσβάσεις στα πεδία των αντικειμένων είναι νόμιμες.

5.1.3.12. Ασφάλεια (Secure)

Στο δίκτυο ελλοχεύουν πολλοί κίνδυνοι για τον χρήστη - παραλήπτη μιας δικτυακής εφαρμογής, γι' αυτό η Java έχει σχεδιαστεί έτσι ώστε να ελαχιστοποιείται η πιθανότητα προσβολής του συστήματος του χρήστη από κάποιο applet γραμμένο για τέτοιο σκοπό.

5.1.3.13. Multithreaded

Ένα πρόγραμμα είναι κάτι στατικό. Όταν εκτελείται όμως δημιουργείται ένα νήμα εκτέλεσης για την εκτέλεση αυτού του προγράμματος. Το νήμα εκτέλεσης έχει το δικό του μετρητή προγράμματος (program counter) και το δικό του περιβάλλον (context) το οποίο είναι η κατάσταση του εκτελούμενου προγράμματος. Έτσι λοιπόν το νήμα εκτέλεσης του προγράμματος είναι σαν ένα κουβάρι που ξετυλίγεται και περνά από τις εντολές του προγράμματος που εκτελούνται κάθε φορά. Φυσικά το κουβάρι αυτό μπορεί να περάσει

από ένα σύνολο εντολών τη μια φορά και από ένα τελείως διαφορετικό σύνολο εντολών την επόμενη. Αυτό εξαρτάται από το τι θα γίνει κάθε φορά, για παράδειγμα εξαρτάται από το ποια πλήκτρα εντολών θα πατήσει ο χρήστης σε μια παραθυρική εφαρμογή ή ένα Applet. Η Java υποστηρίζει εγγενώς την χρήση πολλών threads (νημάτων). Προκειμένου να το επιτύχει αυτό σε συστήματα με έναν επεξεργαστή, το Java runtime system (interpreter) υλοποιεί ένα δικό του χρονοδρομολογητή (scheduler), ενώ σε συστήματα που υποστηρίζουν πολυεπεξεργασία η δημιουργία των threads ανατίθεται στο λειτουργικό σύστημα. Φυσικά όλα αυτά είναι αόρατα τόσο στον προγραμματιστή όσο και στον χρήστη.

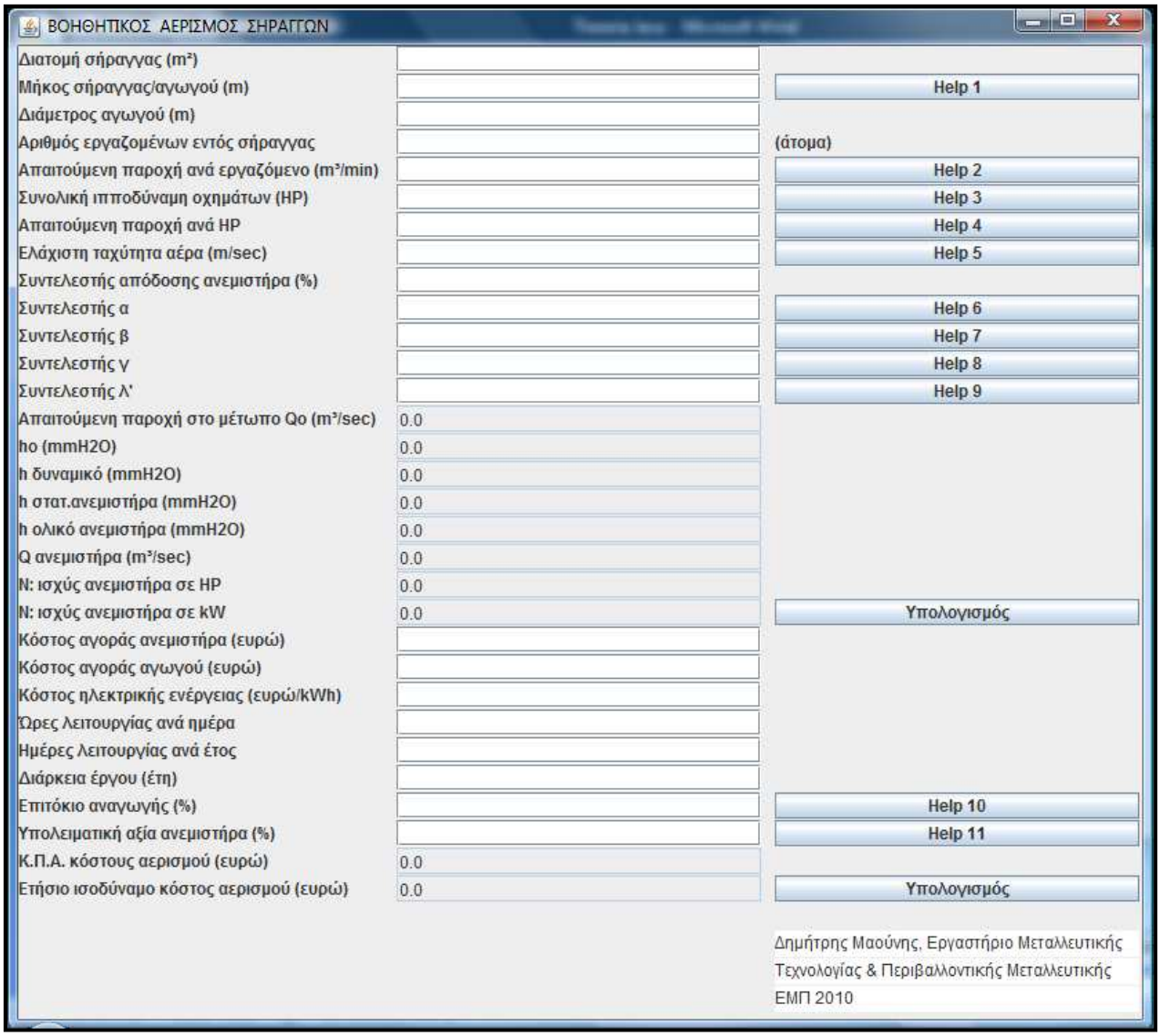
5.1.3.14. Υποστήριξη Multimedia Εφαρμογών

Η Java παρέχει ευκολίες στη δημιουργία multimedia εφαρμογών. Αυτό επιτυγχάνεται τόσο με την ευελιξία της σαν γλώσσα όσο και με τις πλούσιες και συνεχώς εμπλουτιζόμενες βιβλιοθήκες της.

5.2. Το πρόγραμμα aux_v1.0

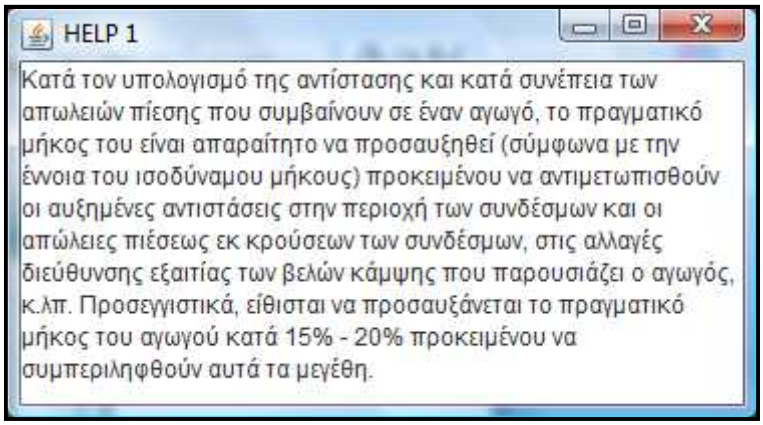
5.2.1. Περιγραφή του προγράμματος

Ο κώδικας για το πρόγραμμα της Java περιέχεται στα αρχεία Main.java, Program.java, Isxus.java, Kostos.java, Help1.java, Help2.java, Help3.java, Help4.java, Help5.java, Help6.java, Help7.java, Help8.java, Help9.java, Help10.java και Help11.java. Η ροή του προγράμματος ξεκινάει από το αρχείο Main.java το οποίο περιέχει τη μέθοδο main. Μέσα στη μέθοδο main δημιουργείται ένα αντικείμενο του οποίου ο κώδικας βρίσκεται στο αρχείο Program.java. Ο κώδικας αυτός δημιουργεί το παρακάτω παράθυρο, στο οποίο ο χρήστης μπορεί να εισάγει δεδομένα στα κενά πεδία κειμένου (Εικόνα 5.2.). Για τη δημιουργία του παραθύρου χρησιμοποιήθηκαν εργαλεία του Swing και του AWT, ενώ για το χειρισμό συμβάντων, π.χ. το πάτημα ενός κουμπιού, χρησιμοποιήθηκε το ActionListener. Όταν πατηθεί ένα κουμπί εκτελείται ο κώδικας της μεθόδου actionPerformed, όπου ελέγχεται ποιο κουμπί πατήθηκε και γίνονται οι αντίστοιχες ενέργειες.



Εικόνα 5.2.: Παράθυρο προγράμματος.

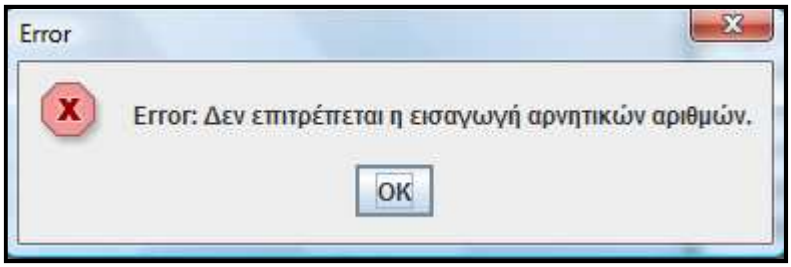
Ανεξάρτητα από το τι έχει εισάγει ο χρήστης στα κενά πεδία κειμένου, μπορεί να πατήσει οποιοδήποτε από τα κουμπιά Help. Μόλις πατηθεί κάποιο κουμπί Help δημιουργείται ένα αντικείμενο το οποίο υλοποιεί τον κώδικα που βρίσκεται στο αντίστοιχο αρχείο Help_.java. Ο κώδικας αυτός ανοίγει ένα μικρό παράθυρο που περιέχει ένα κείμενο βοήθειας για την αντίστοιχη παράμετρο (Εικόνα 5.3.). Το παράθυρο αυτό κλείνει πατώντας το “x” πάνω δεξιά.



Εικόνα 5.3.: Παράθυρο που περιέχει ένα κείμενο βοήθειας.

Ο χρήστης έχει τη δυνατότητα να γράφει μόνο στα κενά πεδία κειμένου και όχι σε αυτά που περιέχουν το “0.0”. Στα πεδία που περιέχουν το “0.0” εμφανίζονται τα αποτελέσματα από τους υπολογισμούς. Όταν πατηθεί το πρώτο κουμπί υπολογισμού δημιουργείται αντικείμενο που υλοποιεί τον κώδικα που βρίσκεται στο αρχείο Isxus.java. Ο κώδικας αυτός υπολογίζει την απαιτούμενη παροχή στο μέτωπο (Q_0), το στατικό ύψος (h_0), το δυναμικό ύψος (h_δ), η πίεση ανεμιστήρα (h), το ολικό ύψος του ανεμιστήρα ($h_{o\lambda}$), την παροχή του ανεμιστήρα (Q), την ισχύ του ανεμιστήρα σε HP (N) και την ισχύ του ανεμιστήρα σε kW (N). Τα αποτελέσματα από τους υπολογισμούς εμφανίζονται στα αντίστοιχα μη εγγράψιμα πεδία κειμένου. Βέβαια το πρώτο κουμπί υπολογισμού δίνει αποτελέσματα μόνο όταν και τα 13 πρώτα πεδία κειμένου περιέχουν δεδομένα και φυσικά τα δεδομένα αυτά είναι μη αρνητικοί αριθμοί. Σε περίπτωση εισαγωγής γράμματος π.χ. “g” εμφανίζεται μήνυμα λάθους (Εικόνα 5.4.) όπως και στην περίπτωση εισαγωγής αρνητικού αριθμού (Εικόνα 5.5.).

Εικόνα 5.4.: Μήνυμα λάθους σε περίπτωση εισαγωγής γράμματος.

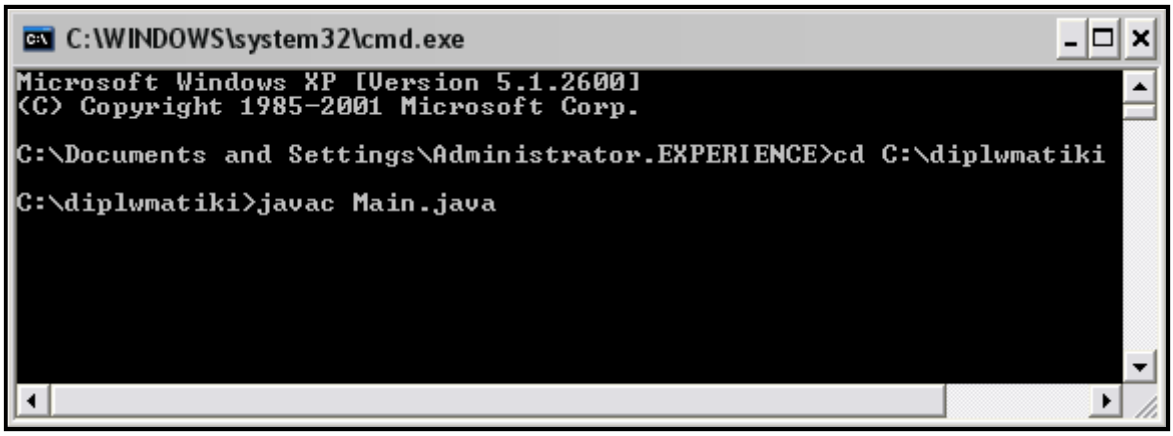


Εικόνα 5.5.: Μήνυμα λάθους σε περίπτωση εισαγωγής εισαγωγής αρνητικού αριθμού.

Όταν πατηθεί το δεύτερο κουμπί υπολογισμού δημιουργείται αντικείμενο που υλοποιεί τον κώδικα που βρίσκεται στο αρχείο Kostos.java. Ο κώδικας αυτός υπολογίζει την καθαρή παρούσα αξία του κόστους αερισμού και το ετήσιο ισοδύναμο κόστος αερισμού. Τα αποτελέσματα από τους υπολογισμούς εμφανίζονται στα αντίστοιχα μη εγγράψιμα πεδία κειμένου. Το δεύτερο κουμπί υπολογισμού δίνει αποτελέσματα μόνο όταν όλα τα εγγράψιμα πεδία κειμένου περιέχουν δεδομένα και φυσικά τα δεδομένα αυτά είναι μη αρνητικοί αριθμοί. Σε αντίθετη περίπτωση εμφανίζονται τα παραπάνω μηνύματα λάθους. Το δεύτερο κουμπί υπολογισμού μπορεί να πατηθεί ανεξάρτητα με το εάν έχει πατηθεί προηγμένως το πρώτο κουμπί υπολογισμού και εμφανίζει τα αποτελέσματα όχι μόνο στα δύο τελευταία μη εγγράψιμα πεδία κειμένου, αλλά σε όλα τα μη εγγράψιμα πεδία κειμένου.

5.2.2. Μεταγλώττιση των αρχείων του προγράμματος

Για να μπορέσει να τρέξει ο κώδικας του προγράμματος θα πρέπει πρώτα να μεταγλωττιστεί. Για τη μεταγλώττιση ενός αρχείου Java χρησιμοποιήθηκε το java developer kit (JDK) που περιέχει τη Java Virtual Machine (JVM). Η μεταγλώττιση όλων των αρχείων έγινε με την εντολή “javac *όνομα_αρχείου*.java” από τη Γραμμή εντολών. Ο java compiler δεν παράγει εκτελέσιμο κώδικα αλλά μια μορφή ψευδοκώδικα (bytecode) ο οποίος αποθηκεύεται σε αρχεία κλάσεων που δημιουργούνται και είναι της μορφής *όνομα_αρχείου*.class. Μετά τη μεταγλώττιση όλων των αρχείων του προγράμματος, η εκτέλεση του προγράμματος γίνεται πάλι από τη Γραμμή εντολών με την εντολή “java Main” (όπου στο αρχείο Main.java βρίσκεται η μέθοδος main απ’ όπου ξεκινάει η ροή του προγράμματος).



Εικόνα 5.6.: Παράδειγμα μεταγλώττισης του αρχείου Main.java.

5.2.3. Δημιουργία εκτελέσιμου αρχείου JAR

Ο τρόπος εκτέλεσης του προγράμματος από τη Γραμμή εντολών και η ύπαρξη πολλών αρχείων κλάσεων δημιουργούν δυσκολίες στη χρήση και διανομή του προγράμματος. Για τη διευκόλυνση αυτών των εργασιών, το Kit περιλαμβάνει ένα εργαλείο με όνομα jar το οποίο μπορεί να «συσκευάσει» όλα τα αρχεία του προγράμματος σ' ένα αρχείο αρχειοθέτησης (Java archive), που αποκαλείται επίσης και αρχείο JAR. (Το αρχείο jar μπορεί επίσης να χρησιμοποιείται για την «αποσυσκευασία» των αρχείων που περιέχονται σ' ένα τέτοιο αρχείο αρχειοθέτησης). Η δημιουργία του αρχείου jar έγινε με την εντολή "jar cmf Manifest.txt Αερισμός_Σηράγγων.jar *.class", όπου το αρχείο Manifest.txt είναι ένα αρχείο τύπου txt που περιέχει την κλάση στην οποία βρίσκεται η μέθοδος main (το περιεχόμενο του Manifest.txt στο συγκεκριμένο πρόγραμμα είναι «Main-Class: Main», αφού μέσα στην κλάση Main βρίσκεται η μέθοδος main), το aux_v1.0.jar είναι το όνομα του εκτελέσιμου αρχείου jar και το *.class περιλαμβάνει όλα τα αρχεία τύπου class (Εικόνα 5.6.). Το εκτελέσιμο αρχείο jar ανοίγει με διπλό κλικ και μπορεί να εκτελείται χωρίς την ύπαρξη των υπόλοιπων αρχείων τύπου java ή class και χωρίς την εγκατάσταση του JDK.

Εικόνα 5.7.: Εικονίδιο εκτελέσιμου αρχείου JAR.

5.2.4. Ο κώδικας του προγράμματος

Για την καλύτερη ανάγνωση του κώδικα αναφέρεται ότι οι σειρές που αρχίζουν με «//» και τα κείμενα που βρίσκονται ανάμεσα στα σύμβολα «/*» και «*/» αποτελούν επεξηγηματικά σχόλια του κώδικα και δε μεταγλωττίζονται.

5.2.4.1. Αρχείο Main.java

```
public class Main {
	//Αρχή του προγράμματος από την κλάση main
	public static void main(String[] args) {
		/* Δημιουργία αντικειμένου της κλάσης Program με όνομα program. Ο κώδικας
		του βρίσκεται στο αρχείο με όνομα Program.java */
		Program program = new Program();
	}
}
```

5.2.4.2. Αρχείο Program.java

```
//Απαραίτητες βιβλιοθήκες που πρέπει να συμπεριληφθούν
import javax.swing.*;
import java.util.*;
import java.io.*;
import java.awt.Dimension;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

/* Η κλάση Program δημιουργείται σαν υποκλάση της JFrame (που υπάρχει στις βιβλιοθήκες) και
κληρονομεί τα χαρακτηριστικά της. Η JFrame είναι απαραίτητη για τη δημιουργία διεπιφάνειας
χρήστη (στο πρόγραμμά μας είναι ένα πλαίσιο με διάφορα πεδία). Με τη διασύνδεση
ActionListener, η διεπιφάνεια χρήστη μπορεί να δέχεται συμβάντα προκαλούμενα από το χρήστη
(στο πρόγραμμά μας το πάτημα ενός κουμπιού)*/
public class Program extends JFrame implements ActionListener {
	//Δήλωση μεταβλητης για το πάνελ
```

```
private JPanel jContentPane = null;
//Δήλωση μεταβλητών για ετικέτες που θα μπουν μέσα στο πάνελ
private JLabel jLabel011 = null;
private JLabel jLabel021 = null;
private JLabel jLabel031 = null;
private JLabel jLabel041 = null;
private JLabel jLabel051 = null;
private JLabel jLabel061 = null;
private JLabel jLabel071 = null;
private JLabel jLabel081 = null;
private JLabel jLabel091 = null;
private JLabel jLabel101 = null;
private JLabel jLabel111 = null;
private JLabel jLabel121 = null;
private JLabel jLabel131 = null;
private JLabel jLabel141 = null;
private JLabel jLabel151 = null;
private JLabel jLabel161 = null;
private JLabel jLabel171 = null;
private JLabel jLabel181 = null;
private JLabel jLabel191 = null;
private JLabel jLabel201 = null;
private JLabel jLabel211 = null;
private JLabel jLabel221 = null;
private JLabel jLabel231 = null;
private JLabel jLabel241 = null;
private JLabel jLabel251 = null;
private JLabel jLabel261 = null;
private JLabel jLabel271 = null;
private JLabel jLabel281 = null;
private JLabel jLabel291 = null;
```

```
private JLabel jLabel301 = null;
private JLabel jLabel311 = null;
private JLabel jLabel321 = null;
private JLabel jLabel331 = null;
private JLabel jLabel341 = null;
private JLabel jLabel351 = null;

private JLabel jLabel012 = null;
private JLabel jLabel032 = null;
private JLabel jLabel042 = null;
private JLabel jLabel092 = null;
private JLabel jLabel142 = null;
private JLabel jLabel152 = null;
private JLabel jLabel162 = null;
private JLabel jLabel172 = null;
private JLabel jLabel182 = null;
private JLabel jLabel192 = null;
private JLabel jLabel202 = null;
private JLabel jLabel222 = null;
private JLabel jLabel232 = null;
private JLabel jLabel242 = null;
private JLabel jLabel252 = null;
private JLabel jLabel262 = null;
private JLabel jLabel272 = null;
private JLabel jLabel302 = null;
private JLabel jLabel322 = null;
private JLabel jLabel323 = null;
private JLabel jLabel332 = null;
private JLabel jLabel342 = null;
private JLabel jLabel352 = null;
```

```
//Δήλωση μεταβλητών για κενά πεδία κειμένου που θα μπουν μέσα στο πάνελ
private JTextField jTextField01 = null;
private JTextField jTextField02 = null;
private JTextField jTextField03 = null;
private JTextField jTextField04 = null;
private JTextField jTextField05 = null;
private JTextField jTextField06 = null;
private JTextField jTextField07 = null;
private JTextField jTextField08 = null;
private JTextField jTextField09 = null;
private JTextField jTextField10 = null;
private JTextField jTextField11 = null;
private JTextField jTextField12 = null;
private JTextField jTextField13 = null;
private JTextField jTextField14 = null;
private JTextField jTextField15 = null;
private JTextField jTextField16 = null;
private JTextField jTextField17 = null;
private JTextField jTextField18 = null;
private JTextField jTextField19 = null;
private JTextField jTextField20 = null;
private JTextField jTextField21 = null;
private JTextField jTextField22 = null;
private JTextField jTextField23 = null;
private JTextField jTextField24 = null;
private JTextField jTextField25 = null;
private JTextField jTextField26 = null;
private JTextField jTextField27 = null;
private JTextField jTextField28 = null;
private JTextField jTextField29 = null;
private JTextField jTextField30 = null;
```

```
private JTextField jTextField31 = null;

private JTextArea jTextArea33 = null;

private JTextArea jTextArea34 = null;

private JTextArea jTextArea35 = null;


/* Δήλωση μεταβλητών για κουμπιά που θα μπουν μέσα στο πάνελ και θα μπορεί να
πατάει ο χρήστης */

private JButton jButton02 = null;

private JButton jButton05 = null;

private JButton jButton06 = null;

private JButton jButton07 = null;

private JButton jButton08 = null;

private JButton jButton10 = null;

private JButton jButton11 = null;

private JButton jButton12 = null;

private JButton jButton13 = null;

private JButton jButton21 = null;

private JButton jButton28 = null;

private JButton jButton29 = null;

private JButton jButton31 = null;


public Program(){

    /* Ορίζεται ο τίτλος του πλαισίου, το μέγεθός του και η ιδιότητα του να κλείνει το
    πλαίσιο από το χρήστη */

    super("ΒΟΗΘΗΤΙΚΟΣ  ΑΕΡΙΣΜΟΣ  ΣΗΡΑΓΓΩΝ");

    setBounds(0, 0, 830, 740);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


    //Αν το πάνελ δεν έχει δημιουργηθεί γίνονται τα παρακάτω

    if (jContentPane == null) {

        /* Δημιουργείται ένα καινούργιο πάνελ με 31 γραμμές, 3 στήλες  και
        ορίζονται τα κενά αναμεσα στις γραμμές και τις στήλες */
```

```
jContentPane = new JPanel();
jContentPane.setLayout(new GridLayout(35, 3, 10, 1));
//Δημιουργούνται οι ετικέτες και ορίζεται το περιεχόμενό τους
jLabel011 = new JLabel();
jLabel011.setText("Διατομή σήραγγας (m²)");
jLabel021 = new JLabel();
jLabel021.setText("Μήκος σήραγγας/αγωγού (m)");
jLabel031 = new JLabel();
jLabel031.setText("Διάμετρος αγωγού (m)");
jLabel041 = new JLabel();
jLabel041.setText("Αριθμός εργαζομένων εντός σήραγγας");
jLabel051 = new JLabel();
jLabel051.setText("Απαιτούμενη παροχή ανά εργαζόμενο (m³/min)");
jLabel061 = new JLabel();
jLabel061.setText("Συνολική ιπποδύναμη οχημάτων (HP)");
jLabel071 = new JLabel();
jLabel071.setText("Απαιτούμενη παροχή ανά HP");
jLabel081 = new JLabel();
jLabel081.setText("Ελάχιστη ταχύτητα αέρα (m/sec)");
jLabel091 = new JLabel();
jLabel091.setText("Συντελεστής απόδοσης ανεμιστήρα (%)");
jLabel101 = new JLabel();
jLabel101.setText("Συντελεστής α");
jLabel111 = new JLabel();
jLabel111.setText("Συντελεστής β");
jLabel121 = new JLabel();
jLabel121.setText("Συντελεστής γ");
jLabel131 = new JLabel();
jLabel131.setText("Συντελεστής λ'");
jLabel141 = new JLabel();
jLabel141.setText("Απαιτούμενη παροχή στο μέτωπο Qo (m³/sec)");
```

```
jLabel151 = new JLabel();
jLabel151.setText("ho (mmH2O)");
jLabel161 = new JLabel();
jLabel161.setText("h δυναμικό (mmH2O)");
jLabel171 = new JLabel();
jLabel171.setText("h στατ.ανεμιστήρα (mmH2O)");
jLabel181 = new JLabel();
jLabel181.setText("h ολικό ανεμιστήρα (mmH2O)");
jLabel191 = new JLabel();
jLabel191.setText("Q ανεμιστήρα (m³/sec)");
jLabel201 = new JLabel();
jLabel201.setText("N: ισχύς ανεμιστήρα σε HP");
jLabel211 = new JLabel();
jLabel211.setText("N: ισχύς ανεμιστήρα σε kW");
jLabel221 = new JLabel();
jLabel221.setText("Κόστος αγοράς ανεμιστήρα (ευρώ)");
jLabel231 = new JLabel();
jLabel231.setText("Κόστος αγοράς αγωγού (ευρώ)");
jLabel241 = new JLabel();
jLabel241.setText("Κόστος ηλεκτρικής ενέργειας (ευρώ/kWh)");
jLabel251 = new JLabel();
jLabel251.setText("Ώρες λειτουργίας ανά ημέρα");
jLabel261 = new JLabel();
jLabel261.setText("Ημέρες λειτουργίας ανά έτος");
jLabel271 = new JLabel();
jLabel271.setText("Διάρκεια έργου (έτη)");
jLabel281 = new JLabel();
jLabel281.setText("Επιτόκιο αναγωγής (%)");
jLabel291 = new JLabel();
jLabel291.setText("Υπολειματική αξία ανεμιστήρα (%)");
jLabel301 = new JLabel();
```

```
jLabel301.setText("Κ.Π.Α. κόστους αερισμού (ευρώ)");
jLabel311 = new JLabel();
jLabel311.setText("Ετήσιο ισοδύναμο κόστος αερισμού (ευρώ)");

jLabel012= new JLabel();
jLabel032 = new JLabel();
jLabel042 = new JLabel();
jLabel042.setText("(άτομα)");
jLabel092 = new JLabel();
jLabel142 = new JLabel();
jLabel152 = new JLabel();
jLabel162 = new JLabel();
jLabel172 = new JLabel();
jLabel182 = new JLabel();
jLabel192 = new JLabel();
jLabel202 = new JLabel();
jLabel222 = new JLabel();
jLabel232 = new JLabel();
jLabel242 = new JLabel();
jLabel252 = new JLabel();
jLabel262 = new JLabel();
jLabel272 = new JLabel();
jLabel302 = new JLabel();
jLabel321 = new JLabel();
jLabel322 = new JLabel();
jLabel323 = new JLabel();
jLabel331 = new JLabel();
jLabel332 = new JLabel();
jLabel341 = new JLabel();
jLabel342 = new JLabel();
jLabel351 = new JLabel();
```

```
jLabel352 = new JLabel();


/* Προστίθονται στο πάνελ οι ετικέτες, τα πεδία κειμένου και τα κουμπιά
με τη σειρά. Τα πεδία κειμένου και τα κουμπιά δημιουργούνται από
μεθόδους (με όνομα π.χ. getJTextField01() και getJButton02() ), ο κώδικας
των οποίων βρίσκεται πιο κάτω.*/

jContentPane.add(jLabel011, null);

jContentPane.add(getJTextField01(), null);

jContentPane.add(jLabel012, null);

jContentPane.add(jLabel021, null);

jContentPane.add(getJTextField02(), null);

jContentPane.add(getJButton02(), null);

jContentPane.add(jLabel031, null);

jContentPane.add(getJTextField03(), null);

jContentPane.add(jLabel032, null);

jContentPane.add(jLabel041, null);

jContentPane.add(getJTextField04(), null);

jContentPane.add(jLabel042, null);

jContentPane.add(jLabel051, null);

jContentPane.add(getJTextField05(), null);

jContentPane.add(getJButton05(), null);

jContentPane.add(jLabel061, null);

jContentPane.add(getJTextField06(), null);

jContentPane.add(getJButton06(), null);

jContentPane.add(jLabel071, null);

jContentPane.add(getJTextField07(), null);

jContentPane.add(getJButton07(), null);

jContentPane.add(jLabel081, null);

jContentPane.add(getJTextField08(), null);

jContentPane.add(getJButton08(), null);

jContentPane.add(jLabel091, null);

jContentPane.add(getJTextField09(), null);
```

```
jContentPane.add(jLabel092, null);
jContentPane.add(jLabel101, null);
jContentPane.add(getJTextField10(), null);
jContentPane.add(getJButton10(), null);
jContentPane.add(jLabel111, null);
jContentPane.add(getJTextField11(), null);
jContentPane.add(getJButton11(), null);
jContentPane.add(jLabel121, null);
jContentPane.add(getJTextField12(), null);
jContentPane.add(getJButton12(), null);
jContentPane.add(jLabel131, null);
jContentPane.add(getJTextField13(), null);
jContentPane.add(getJButton13(), null);
jContentPane.add(jLabel141, null);
jContentPane.add(getJTextField14(), null);
jContentPane.add(jLabel142, null);
jContentPane.add(jLabel151, null);
jContentPane.add(getJTextField15(), null);
jContentPane.add(jLabel152, null);
jContentPane.add(jLabel161, null);
jContentPane.add(getJTextField16(), null);
jContentPane.add(jLabel162, null);
jContentPane.add(jLabel171, null);
jContentPane.add(getJTextField17(), null);
jContentPane.add(jLabel172, null);
jContentPane.add(jLabel181, null);
jContentPane.add(getJTextField18(), null);
jContentPane.add(jLabel182, null);
jContentPane.add(jLabel191, null);
jContentPane.add(getJTextField19(), null);
jContentPane.add(jLabel192, null);
```

```
jContentPane.add(jLabel201, null);
jContentPane.add(getJTextField20(), null);
jContentPane.add(jLabel202, null);
jContentPane.add(jLabel211, null);
jContentPane.add(getJTextField21(), null);
jContentPane.add(getJButton21(), null);
jContentPane.add(jLabel221, null);
jContentPane.add(getJTextField22(), null);
jContentPane.add(jLabel222, null);
jContentPane.add(jLabel231, null);
jContentPane.add(getJTextField23(), null);
jContentPane.add(jLabel232, null);
jContentPane.add(jLabel241, null);
jContentPane.add(getJTextField24(), null);
jContentPane.add(jLabel242, null);
jContentPane.add(jLabel251, null);
jContentPane.add(getJTextField25(), null);
jContentPane.add(jLabel252, null);
jContentPane.add(jLabel261, null);
jContentPane.add(getJTextField26(), null);
jContentPane.add(jLabel262, null);
jContentPane.add(jLabel271, null);
jContentPane.add(getJTextField27(), null);
jContentPane.add(jLabel272, null);
jContentPane.add(jLabel281, null);
jContentPane.add(getJTextField28(), null);
jContentPane.add(getJButton28(), null);
jContentPane.add(jLabel291, null);
jContentPane.add(getJTextField29(), null);
jContentPane.add(getJButton29(), null);
jContentPane.add(jLabel301, null);
```

```
                jContentPane.add(getJTextField30(), null);
                jContentPane.add(jLabel302, null);
                jContentPane.add(jLabel311, null);
                jContentPane.add(getJTextField31(), null);
                jContentPane.add(getJButton31(), null);
                jContentPane.add(jLabel321, null);
                jContentPane.add(jLabel322, null);
                jContentPane.add(jLabel323, null);
                jContentPane.add(jLabel331, null);
                jContentPane.add(jLabel332, null);
                jContentPane.add(getJTextArea33(), null);
                jContentPane.add(jLabel341, null);
                jContentPane.add(jLabel342, null);
                jContentPane.add(getJTextArea34(), null);
                jContentPane.add(jLabel351, null);
                jContentPane.add(jLabel352, null);
                jContentPane.add(getJTextArea35(), null);
                add(jContentPane);
        }
        //Το πάνελ τίθεται ορατό.
        setVisible(true);
}
```

/* Δημιουργία κενών πεδίων κειμένων. Τα πεδία αυτά είναι δύο ειδών. Από το 14^{ο} έως το 21^{ο} και το 30^{ο} και 31^{ο} πεδίο κειμένου ο χρήστης δε μπορεί να γράψει στα πεδία αυτά και χρησιμοποιούνται μόνο για εμφάνιση των αποτελεσμάτων. Τα υπόλοιπα πεδία κειμένου χρησιμοποιούνται για εισαγωγή δεδομένων από το χρήστη. */

```
private JTextField getJTextField01() {
        if (jTextField01 == null) {
                jTextField01 = new JTextField();
                jTextField01.setName("jTextField01");
        }
```

```
        return jTextField01;
    }

    private JTextField getJTextField02() {
        if (jTextField02 == null) {
            jTextField02 = new JTextField();
            jTextField02.setName("jTextField02");
        }
        return jTextField02;
    }

    private JTextField getJTextField03() {
        if (jTextField03 == null) {
            jTextField03 = new JTextField();
            jTextField03.setName("jTextField03");
        }
        return jTextField03;
    }

    private JTextField getJTextField04() {
        if (jTextField04 == null) {
            jTextField04 = new JTextField();
            jTextField04.setName("jTextField04");
        }
        return jTextField04;
    }

    private JTextField getJTextField05() {
        if (jTextField05 == null) {
            jTextField05 = new JTextField();
            jTextField05.setName("jTextField05");
```

```
        }
        return jTextField05;
    }

    private JTextField getJTextField06() {
        if (jTextField06 == null) {
            jTextField06 = new JTextField();
            jTextField06.setName("jTextField06");
        }
        return jTextField06;
    }

    private JTextField getJTextField07() {
        if (jTextField07 == null) {
            jTextField07 = new JTextField();
            jTextField07.setName("jTextField07");
        }
        return jTextField07;
    }

    private JTextField getJTextField08() {
        if (jTextField08 == null) {
            jTextField08 = new JTextField();
            jTextField08.setName("jTextField08");
        }
        return jTextField08;
    }

    private JTextField getJTextField09() {
        if (jTextField09 == null) {
            jTextField09 = new JTextField();
```

```
                jTextField09.setName("jTextField09");
            }
            return jTextField09;
        }

        private JTextField getJTextField10() {
            if (jTextField10 == null) {
                jTextField10 = new JTextField();
                jTextField10.setName("jTextField10");
            }
            return jTextField10;
        }

        private JTextField getJTextField11() {
            if (jTextField11 == null) {
                jTextField11 = new JTextField();
                jTextField11.setName("jTextField11");
            }
            return jTextField11;
        }

        private JTextField getJTextField12() {
            if (jTextField12 == null) {
                jTextField12 = new JTextField();
                jTextField12.setName("jTextField12");
            }
            return jTextField12;
        }

        private JTextField getJTextField13() {
            if (jTextField13 == null) {
```

```
                jTextField13 = new JTextField();
                jTextField13.setName("jTextField13");
        }
        return jTextField13;
}

private JTextField getJTextField14() {
        if (jTextField14 == null) {
                jTextField14 = new JTextField();
                jTextField14.setText("0.0");
                jTextField14.setEditable(false);
                jTextField14.setName("jTextField14");
        }
        return jTextField14;
}

private JTextField getJTextField15() {
        if (jTextField15 == null) {
                jTextField15 = new JTextField();
                jTextField15.setText("0.0");
                jTextField15.setEditable(false);
                jTextField15.setName("jTextField15");
        }
        return jTextField15;
}

private JTextField getJTextField16() {
        if (jTextField16 == null) {
                jTextField16 = new JTextField();
                jTextField16.setText("0.0");
                jTextField16.setEditable(false);
```

```
                jTextField16.setName("jTextField16");
        }
        return jTextField16;
}

private JTextField getJTextField17() {
        if (jTextField17 == null) {
                jTextField17 = new JTextField();
                jTextField17.setText("0.0");
                jTextField17.setEditable(false);
                jTextField17.setName("jTextField17");
        }
        return jTextField17;
}

private JTextField getJTextField18() {
        if (jTextField18 == null) {
                jTextField18 = new JTextField();
                jTextField18.setText("0.0");
                jTextField18.setEditable(false);
                jTextField18.setName("jTextField18");
        }
        return jTextField18;
}

private JTextField getJTextField19() {
        if (jTextField19 == null) {
                jTextField19 = new JTextField();
                jTextField19.setText("0.0");
                jTextField19.setEditable(false);
                jTextField19.setName("jTextField19");
```

```
        }
        return jTextField19;
    }

    private JTextField getJTextField20() {
        if (jTextField20 == null) {
            jTextField20 = new JTextField();
            jTextField20.setText("0.0");
            jTextField20.setEditable(false);
            jTextField20.setName("jTextField20");
        }
        return jTextField20;
    }

    private JTextField getJTextField21() {
        if (jTextField21 == null) {
            jTextField21 = new JTextField();
            jTextField21.setText("0.0");
            jTextField21.setEditable(false);
            jTextField21.setName("jTextField21");
        }
        return jTextField21;
    }

    private JTextField getJTextField22() {
        if (jTextField22 == null) {
            jTextField22 = new JTextField();
            jTextField22.setName("jTextField22");
        }
        return jTextField22;
    }
```

```
private JTextField getJTextField23() {
        if (jTextField23 == null) {
                jTextField23 = new JTextField();
                jTextField23.setName("jTextField23");
        }
        return jTextField23;
}

private JTextField getJTextField24() {
        if (jTextField24 == null) {
                jTextField24 = new JTextField();
                jTextField24.setName("jTextField24");
        }
        return jTextField24;
}

private JTextField getJTextField25() {
        if (jTextField25 == null) {
                jTextField25 = new JTextField();
                jTextField25.setName("jTextField25");
        }
        return jTextField25;
}

private JTextField getJTextField26() {
        if (jTextField26 == null) {
                jTextField26 = new JTextField();
                jTextField26.setName("jTextField26");
        }
        return jTextField26;
```

```
}

private JTextField getJTextField27() {
        if (jTextField27 == null) {
                jTextField27 = new JTextField();
                jTextField27.setName("jTextField27");
        }
        return jTextField27;
}

private JTextField getJTextField28() {
        if (jTextField28 == null) {
                jTextField28 = new JTextField();
                jTextField28.setName("jTextField28");
        }
        return jTextField28;
}

private JTextField getJTextField29() {
        if (jTextField29 == null) {
                jTextField29 = new JTextField();
                jTextField29.setName("jTextField29");
        }
        return jTextField29;
}

private JTextField getJTextField30() {
        if (jTextField30 == null) {
                jTextField30 = new JTextField();
                jTextField30.setText("0.0");
                jTextField30.setEditable(false);
```

```
                jTextField30.setName("jTextField30");
        }
        return jTextField30;
}

private JTextField getJTextField31() {
        if (jTextField31 == null) {
                jTextField31 = new JTextField();
                jTextField31.setText("0.0");
                jTextField31.setEditable(false);
                jTextField31.setName("jTextField31");
        }
        return jTextField31;
}

private JTextArea getJTextArea33() {
        if (jTextArea33 == null) {
                jTextArea33 = new JTextArea("Δημήτρης Μαούνης, Εργαστήριο
                                                        Μεταλλευτικής");
                jTextArea33.setEditable(false);
                jTextArea33.setLineWrap(true);
                jTextArea33.setWrapStyleWord(true);
                jTextArea33.setName("jTextArea33");
        }
        return jTextArea33;
}

private JTextArea getJTextArea34() {
        if (jTextArea34 == null) {
                jTextArea34 = new JTextArea("Τεχνολογίας & Περιβαλλοντικής
                                                        Μεταλλευτικής");
                jTextArea34.setEditable(false);
```

```
                jTextArea34.setLineWrap(true);
                jTextArea34.setWrapStyleWord(true);
                jTextArea34.setName("jTextArea34");
        }
        return jTextArea34;
}

private JTextArea getJTextArea35() {
        if (jTextArea35 == null) {
                jTextArea35 = new JTextArea("ΕΜΠ 2010");
                jTextArea35.setEditable(false);
                jTextArea35.setLineWrap(true);
                jTextArea35.setWrapStyleWord(true);
                jTextArea35.setName("jTextArea35");
        }
        return jTextArea35;
}

/* Δημιουργία κουμπιών τα οποία όταν πατηθούν είτε ανοίγουν παράθυρο βοήθειας είτε
εμφανίζουν αποτελέσματα σε αντίστοιχα πεδία. Τα κουμπιά 21 και 31 είναι κουμπιά
υπολογισμού και τα υπόλοιπα είναι κουμπιά Help. Τα κουμπιά προστίθονται στον
ActionListener ώστε όταν πατηθούν να γίνουν οι αντίστοιχες ενέργειες. */

private JButton getJButton02() {
        if (jButton02 == null) {
                jButton02 = new JButton();
                jButton02.setText("Help 1");
                jButton02.setName("jButton02");
                jButton02.addActionListener(this);
        }
        return jButton02;
}
```

```
private JButton getJButton05() {
        if (jButton05 == null) {
                jButton05 = new JButton();
                jButton05.setText("Help 2");
                jButton05.setName("jButton05");
                jButton05.addActionListener(this);
        }
        return jButton05;
}

private JButton getJButton06() {
        if (jButton06 == null) {
                jButton06 = new JButton();
                jButton06.setText("Help 3");
                jButton06.setName("jButton06");
                jButton06.addActionListener(this);
        }
        return jButton06;
}

private JButton getJButton07() {
        if (jButton07 == null) {
                jButton07 = new JButton();
                jButton07.setText("Help 4");
                jButton07.setName("jButton07");
                jButton07.addActionListener(this);
        }
        return jButton07;
}
```

```
private JButton getJButton08() {
        if (jButton08 == null) {
                jButton08 = new JButton();
                jButton08.setText("Help 5");
                jButton08.setName("jButton08");
                jButton08.addActionListener(this);
        }
        return jButton08;
}

private JButton getJButton10() {
        if (jButton10 == null) {
                jButton10 = new JButton();
                jButton10.setText("Help 6");
                jButton10.setName("jButton10");
                jButton10.addActionListener(this);
        }
        return jButton10;
}

private JButton getJButton11() {
        if (jButton11 == null) {
                jButton11 = new JButton();
                jButton11.setText("Help 7");
                jButton11.setName("jButton11");
                jButton11.addActionListener(this);
        }
        return jButton11;
}

private JButton getJButton12() {
```

```
        if (jButton12 == null) {
            jButton12 = new JButton();
            jButton12.setText("Help 8");
            jButton12.setName("jButton12");
            jButton12.addActionListener(this);
        }
        return jButton12;
    }

    private JButton getJButton13() {
        if (jButton13 == null) {
            jButton13 = new JButton();
            jButton13.setText("Help 9");
            jButton13.setName("jButton13");
            jButton13.addActionListener(this);
        }
        return jButton13;
    }

    private JButton getJButton21() {
        if (jButton21 == null) {
            jButton21 = new JButton();
            jButton21.setText("Υπολογισμός");
            jButton21.setName("jButton21");
            jButton21.addActionListener(this);
        }
        return jButton21;
    }

    private JButton getJButton28() {
        if (jButton28 == null) {
```

```
                jButton28 = new JButton();
                jButton28.setText("Help 10");
                jButton28.setName("jButton28");
                jButton28.addActionListener(this);
        }
        return jButton28;
}

private JButton getJButton29() {
        if (jButton29 == null) {
                jButton29 = new JButton();
                jButton29.setText("Help 11");
                jButton29.setName("jButton29");
                jButton29.addActionListener(this);
        }
        return jButton29;
}

private JButton getJButton31() {
        if (jButton31 == null) {
                jButton31 = new JButton();
                jButton31.setText("Υπολογισμός");
                jButton31.setName("jButton31");
                jButton31.addActionListener(this);
        }
        return jButton31;
}

/* Ακολουθεί η μέθοδος actionPerform από την ActionListener. Ο κώδικας αυτός εκτελείται όταν πατηθεί κάποιο κουμπί. */
```

```
public void actionPerformed(ActionEvent e) {

        boolean k = true, l = true, thetikoi = true;


        /* Οι μεταβλητές k, l και thetikoi παίρνουν τιμή true ή false και χρησιμοποιούνται
        για την ομαλή λειτουργία ελέγχου των δεδομένων εισαγωγής από το χρήστη. Πιο
        συγκεκριμένα το k όταν γίνεται false υποδηλώνει ότι έχει πατηθεί κάποιο κουμπί
        Help. Το l όταν γίνεται false υποδηλώνει ότι έχει γίνει έλεγχος των 13 πρώτων
        πεδίων κειμένου ή ότι έχει πατηθεί το πρώτο κουμπί υπολογισμού. Το thetikoi
        όταν γίνεται false υποδηλώνει ότι ένα τουλάχιστον από τα πρώτα 13 πεδία
        κειμένου περιέχει αρνητικό αριθμό. */


        /* Γίνεται έλεγχος αν έχει πατηθεί κάποιο κουμπί Help. Αν έχει πατηθεί,
        δημιουργείται ένα καινούργιο αντικείμενο, ο κώδικας του οποίου βρίσκεται σε
        αρχείο με όνομα Help_.java (π.χ. Help0.java) ανάλογα με το κουμπί Help που έχει
        πατηθεί. Επίσης η μεταβλητή k γίνεται false. */


        if (e.getSource().equals(jButton02)) {

                k = false;

                new Help1();

        }


        if (e.getSource().equals(jButton05)) {

                k = false;

                new Help2();

        }


        if (e.getSource().equals(jButton06)) {

                k = false;

                new Help3();

        }


        if (e.getSource().equals(jButton07)) {

                k = false;

                new Help4();

        }
```

```
if (e.getSource().equals(jButton08)) {
                k = false;
                new Help5();
        }

if (e.getSource().equals(jButton10)) {
                k = false;
                new Help6();
        }

if (e.getSource().equals(jButton11)) {
                k = false;
                new Help7();
        }

if (e.getSource().equals(jButton12)) {
                k = false;
                new Help8();
        }

if (e.getSource().equals(jButton13)) {
                k = false;
                new Help9();
        }

if (e.getSource().equals(jButton28)) {
                k = false;
                new Help10();
        }
```

```
if (e.getSource().equals(jButton29)) {
                k = false;
                new Help11();
        }
```

/* Δήλωση μεταβλητών όπου θα αποθηκευτούν τα δεδομένα που εισήγαγε ο χρήστης. */

```
double var14 = 0, var15 = 0, var16 = 0, var17 = 0, var18 = 0, var19 = 0;
int var20 = 0, var21 = 0, var30 = 0, var31 = 0;
```

/* Τα δεδομένα που εισάγει ο χρήστης παίρνονται σε μορφή String (συμβολοσειρά) και πρέπει να μετατραπούν σε μορφή Integer (ακέραιος) για τα 13 πρώτα πεδία κειμένου εγγραφής. Οι παρακάτω εντολές εσωκλείονται σε ένα block try που ακολουθείται από catch. Έτσι αν σε κάποιο πεδίο δεν έχει εισαχθεί κάποιος αριθμός (π.χ. έχει εισαχθεί κάποιος άλλος χαρακτήρας ή κενό) δημιουργείται exception (εξέραιση) που γίνεται catch και εμφανίζεται αντίστοιχο μήνυμα λάθους. */

```
try {
        double var1 = Double.parseDouble(jTextField01.getText());
        double var2 = Double.parseDouble(jTextField02.getText());
        double var3 = Double.parseDouble(jTextField03.getText());
        int var4 = Integer.parseInt(jTextField04.getText());
        double var5 = Double.parseDouble(jTextField05.getText());
        double var6 = Double.parseDouble(jTextField06.getText());
        double var7 = Double.parseDouble(jTextField07.getText());
        double var8 = Double.parseDouble(jTextField08.getText());
        double var9 = Double.parseDouble(jTextField09.getText());
        double var10 = Double.parseDouble(jTextField10.getText());
        double var11 = Double.parseDouble(jTextField11.getText());
        double var12 = Double.parseDouble(jTextField12.getText());
        double var13 = Double.parseDouble(jTextField13.getText());
```

/* Γίνεται έλεγχος αν οι αριθμοί είναι θετικοι. Αν έστω και ένας δεν είναι θετικός εμφανίζεται μήνυμα λάθους και η μεταβλητή thetikoi γίνεται false. Επίσης και η μεταβλητή l γίνεται false. */

```
if ((var1 < 0) || (var2 < 0) || (var3 < 0) || (var4 < 0) || (var5 < 0) || (var6 < 0)
    || (var7 < 0) || (var8 < 0) || (var9 < 0) || (var10 < 0) || (var11 < 0) || (var12
    < 0) || (var13 < 0)){

        JOptionPane.showMessageDialog(null, "Error: Δεν επιτρέπεται η
                εισαγωγή αρνητικών αριθμών.", "Error",
                JOptionPane.ERROR_MESSAGE);

        thetikoi = false;

        l = false;

}
```

/* Αν οι αριθμοί που έχουν εισαχθεί είναι θετικοί, καλείται η συνάρτηση Isxus με μεταβλητές εισόδου τους 13 πρώτους αριθμούς που έχει εισάγει ο χρήστης. Ο κώδικας της συνάρτησης Isxus βρίσκεται στο αρχείο με όνομα Isxus.java. Τα 8 σε πλήθος αποτελέσματα από τη συνάρτηση αποθηκεύονται. */

```
else {

        Isxus isxus= new Isxus(var1, var2, var3, var4, var5, var6, var7,
                var8, var9, var10, var11, var12, var13);

        var14 = isxus.qo;

        var15 = isxus.ho;

        var16 = isxus.hd;

        var17 = isxus.h;

        var18 = isxus.hol;

        var19 = isxus.q;

        var20 = isxus.nhpInt;

        var21 = isxus.nkwInt;
```

/* Αν έχει πατηθεί το κουμπί για υπολογισμό ισχύος, εμφανίζονται στα στα μη εγγράψιμα πεδία κειμένου από το 14ο έως το 21ο τα

```
αποτελέσματα από τη συνάρτηση Isxus. Η μεταβλητή l γίνεται
false. */

if (e.getSource().equals(jButton21)) {
    l = false;
    jTextField14.setText("" + var14);
    jTextField15.setText("" + var15);
    jTextField16.setText("" + var16);
    jTextField17.setText("" + var17);
    jTextField18.setText("" + var18);
    jTextField19.setText("" + var19);
    jTextField20.setText("" + var20);
    jTextField21.setText("" + var21);
}
}
}

/* Ο κώδικας μέσα στο block catch εκτελείται όταν κάποιο από τα πρώτα 13 πεδία
κειμένου δεν περιέχει αριθμό. Αν έχει πατηθεί κάποιο κουμπί Help το k έχει γίνει
false, οπότε ο κώδικας μέσα στο if δεν εκτελειται. Αν έχει πατηθεί ένα από τα δύο
κουμπιά υπολογισμού το k είναι true και εμφανίζεται μήνυμα λάθους. Το l γίνεται
false. */

catch (NumberFormatException nfe){
    if (k){
        l = false;
        JOptionPane.showMessageDialog(null, "Error: " +
nfe.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
}

/* Μετατραπή σε μορφή Integer των υπόλοιπων πεδίων κειμένου εγγραφής (από
το 22ο έως το 29ο) μέσα σ'ένα try block. */
```

```
try{
        double var22 = Double.parseDouble(jTextField22.getText());
        double var23 = Double.parseDouble(jTextField23.getText());
        double var24 = Double.parseDouble(jTextField24.getText());
        double var25 = Double.parseDouble(jTextField25.getText());
        double var26 = Double.parseDouble(jTextField26.getText());
        double var27 = Double.parseDouble(jTextField27.getText());
        double var28 = Double.parseDouble(jTextField28.getText());
        double var29 = Double.parseDouble(jTextField29.getText());

        /*Ελέγχεται αν οι αριθμοί αυτοί είναι θετικοί. Αν δεν είναι εμφανίζεται
        μήνυμα λάθους. */
        if (thetikoi){
            if ((var22 < 0) || (var23 < 0) || (var24 < 0) || (var25 < 0) || (var26 <
                0) || (var27 < 0) || (var28 < 0) || (var29 < 0)){
                JOptionPane.showMessageDialog(null, "Error: Δεν
                    επιτρέπεται η εισαγωγή αρνητικών αριθμών.",
                    "Error", JOptionPane.ERROR_MESSAGE);
            }

            /* Αν οι αριθμοί είναι θετικοί καλείται η συνάρτηση Kostos με
            μεταβλητές εισόδου τους αριθμούς που εισήγαγε ο χρήστης στα
            τελευταία πεδία κειμένου και ένα αποτέλεσμα από τη συνάρτηση
            Isxus. Τα 2 σε πλήθος αποτελέσματα αποθηκεύονται. */

            else{
                Kostos kostos = new Kostos(var21, var22, var23, var24,
                        var25, var26, var27, var28, var29);
                var30 = kostos.kpaInt;
                var31 = kostos.eikInt;

                /* Αν πατηθεί το κουμπί υπολογισμού κόστους
                εμφανίζονται τα αποτελέσματα σε όλα τα πεδία κειμένου
                για αποτελέσματα. */
                if (e.getSource().equals(jButton31)) {
```

```
                    jTextField14.setText("" + var14);
                    jTextField15.setText("" + var15);
                    jTextField16.setText("" + var16);
                    jTextField17.setText("" + var17);
                    jTextField18.setText("" + var18);
                    jTextField19.setText("" + var19);
                    jTextField20.setText("" + var20);
                    jTextField21.setText("" + var21);
                    jTextField30.setText("" + var30);
                    jTextField31.setText("" + var31);
                }
            }
        }
    }
```

/* Ο κώδικας μέσα στο block catch εκτελείται όταν κάποιο από τα πεδία κειμένου από το 22^{ο} έως το 29^{ο} δεν περιέχει αριθμό. Αν έχει πατηθεί κάποιο κουμπί Help το k έχει γίνει false, αν έχει πατηθεί το κουμπί για τον υπολογισμό ισχύος το l έχει γίνει false, αν κάποια εισαγωγή στα 13 πρώτα πεδία δεν είναι αριθμός ή είναι αρνητικός αρνητικός αριθμός το l έχει γίνει false. Σε όλες αυτές τις περιπτώσεις ο κώδικας μέσα στο if δεν εκτελείται. Οπότε ο κώδικας μέσα στο if εκτελειται μόνο αν έχει πατηθεί το κουμπί υπολογισμού κόστους, τα πρώτα 13 πεδία κειμένου έχουν θετικούς αριθμούς και κάποιο από τα πεδία κειμένου από το 22ο έως το 29ο δεν περιέχει αριθμό. Τότε εμφανίζεται μήνυμα λάθους. */

```
catch (NumberFormatException nfe){
    if (k && l){
        JOptionPane.showMessageDialog(null, "Error: " +
                                nfe.getMessage(), "Error",
                                JOptionPane.ERROR_MESSAGE);
    }
}

catch (Exception exc) {
```

```
                JOptionPane.showMessageDialog(null, "Error: " + exc.getMessage(),
                                                "Error",
                                                JOptionPane.ERROR_MESSAGE);

            }

        }

}
```

5.2.4.3. Αρχείο Help0.java

```
 import javax.swing.*;   //gia ti xrisi parathurwn dialogou

import java.util.*;

import java.io.*;           //gia ti xrisi stream

import java.awt.Dimension;

import java.awt.GridLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;


/*Ο κώδικας εκτελείται όταν πατηθεί το 1ο σε σειρα Help. Δημιουργείται πλαίσιο με κείμενο βοήθειας. */


public class Help0 extends JFrame{

        private JPanel jContentPane = null;

        private JTextArea jTextArea = null;


        public Help0(){

            super();

            initialize();

        }


        private void initialize() {


            /* Ορίζεται ο τίτλος του πλαισίου, το μέγεθός του, η ιδιότητα του να κλείνει το πλαίσιο από το χρήστη και τίθεται ορατό. */
```

```
            this.setContentPane(getJContentPane());
            this.setTitle("HELP 1");
            this.setVisible(true);
            this.setBounds(830, 47, 0, 0);
            this.pack();
        }

        private JPanel getJContentPane() {
            if (jContentPane == null) {

                /* Δημιουργείται το πάνελ του πλαισίου και μία περιοχή κειμένου. Στην
                περιοχή κειμένου εισάγεται το κείμενο βοήθειας και αυτή προστίθεται στο
                πανελ. */
                GridLayout gridLayout = new GridLayout();
                    jTextArea = new JTextArea("Κατά τον υπολογισμό της
                    αντίστασης και κατά συνέπεια των απωλειών πίεσης που
                    συμβαίνουν σε έναν αγωγό, το πραγματικό μήκος του είναι
                    απαραίτητο να προσαυξηθεί (σύμφωνα με την έννοια του
                    ισοδύναμου μήκους) προκειμένου να αντιμετωπισθούν οι
                    αυξημένες αντιστάσεις στην περιοχή των συνδέσμων και οι
                    απώλειες πιέσεως εκ κρούσεων των συνδέσμων, στις αλλαγές
                    διεύθυνσης εξαιτίας των βελών κάμψης που παρουσιάζει ο
                    αγωγός, κ.λπ. Προσεγγιστικά, είθισται να προσαυξάνεται το
                    πραγματικό μήκος του αγωγού κατά 15% - 20% προκειμένου
                    να συμπεριληφθούν αυτά τα μεγέθη.");

                jTextArea.setLineWrap(true);
                jTextArea.setWrapStyleWord(true);
                jTextArea.setPreferredSize(new Dimension(360, 170));
                jContentPane = new JPanel();
                jContentPane.setLayout(gridLayout);
                jContentPane.add(jTextArea, null);
            }
            return jContentPane;
        }
}
```

5.2.4.4. Αρχείο Isxus.java

```
public class Isxus {

//Ο κώδικας εκτελέιται όταν πατηθεί το πρώτο κουμπί υπολογισμού.

    public double qo, ho, hd, h, hol, q, nhp, nkw;

    public int nhpInt, nkwInt;

    /* Η συνάρτηση Isxus παίρνει σα μεταβλητές εισόδου τα δεδομένα που εισάγει ο χρήστης
    στα 13 πρώτα πεδία κειμένου. */

    Isxus(double s, double l, double d, int numErgatwn, double ape, double hp, double aphp,
        double eta, double sa, double a, double b, double g, double sl){

        double q1, q2, v;

        q1 = (ape * numErgatwn + aphp * hp)/60;

        q2 = eta + d;

        qo = Math.max(q1, q2);

        ho = 0.1 * sl * l * Math.pow(qo, a) / Math.pow(d, 5);

        q = qo + (g * Math.pow(ho, b) * l / (b + 1));

        v = qo * 4 / ( 3.14 * Math.pow(d, 2));

        hd = 0.06 * Math.pow(v, 2);

        h = ho * (1 + a * (q - qo) / ((b + 2) * qo));

        hol = hd + h;

        nhp = q * (hd + h) / (75 * sa / 100);

        nkw = nhp * 0.745;

        //Στρογγυλοποίηση των μεταβλητών εξόδου στο ψηφίο των χιλιοστών.

        qo = Math.round(qo * 1000);

        qo = qo / 1000;

        ho = Math.round(ho * 1000);
```

```
            ho = ho / 1000;
            hd = Math.round(hd * 1000);
            hd = hd / 1000;
            h = Math.round(h * 1000);
            h = h / 1000;
            hol = Math.round(hol * 1000);
            hol = hol / 1000;
            q = Math.round(q * 1000);
            q = q / 1000;
            nhpInt = (int)Math.round(nhp);
            nkwInt = (int)Math.round(nkw);
        }
}
```

5.2.4.5. Αρχείο Kostos.java

```
public class Kostos {

//Ο κώδικας εκτελείται όταν πατηθεί το δεύτερο κουμπί υπολογισμού.

        public double kpa, eik;
        public int kpaInt, eikInt;

        /* Η συνάρτηση Kostos παίρνει σα μεταβλητές εισόδου τα δεδομένα που εισάγει ο χρήστης στα πεδία κειμένου από το 22ο έως το 29ο και ένα αποτέλεσμα από τη συνάρτηση Isxus. */

        Kostos(int nkw, double kan, double kag, double khe, double wl, double hl, double de,
                double ea, double uan){
            double k = 0, paua;

            ea = ea / 100;
            for(int i = 1; i <= de; i++){
                k = k + 1 / Math.pow((1 + ea), i);
```

```
                }

                paua = kan * uan / (100 * Math.pow((1 + ea), de));
                kpa = kan + kag - paua + nkw * khe * wl * hl * k;
                eik = (kan + kag - paua) * ea * Math.pow((1 + ea), de)/(Math.pow((1 + ea), de) -
1) + nkw * khe * wl * hl;

                //Στρογγυλοποίηση των μεταβλητών εξόδου στο ψηφίο των εκατοστών.

                kpa = Math.round(kpa / 100);
                kpa = kpa * 100;
                kpaInt = (int)kpa;
                eik = Math.round(eik / 100);
                eik = eik * 100;
                eikInt = (int)eik;
        }
}
```

5.3. Βασικές οδηγίες χρήσης του προγράμματος aux_v1.0

Το πρόγραμμα aux_v1.0 είναι ένα χρήσιμο εργαλείο για την επιλογή του κατάλληλου ανεμιστήρα και το πιθανό κόστος λειτουργίας του στο έργο.

Το πρόγραμμα ξεκινάει με διπλό κλικ πάνω στο εικονίδιο (εικόνα 5.7.) και ανοίγει το παράθυρο του προγράμματος (εικόνα 5.2.). Ο χρήστης πρέπει να εισάγει τα αριθμητικά δεδομένα που του ζητούνται στα κενά πεδία (λευκά κελιά). Τα αποτελέσματα των υπολογισμών εμφανίζονται στα γαλάζια κελιά.

Για την εκτίμηση των χαρακτηριστικών του ανεμιστήρα συναρτήσει των χαρακτηριστικών του αγωγού του αερισμού και του έργου (π.χ. προσωπικό, ιπποδύναμη εξοπλισμού με κινητήρες εσωτερικής καύσης, κ.λπ.), ο χρήστης πρέπει να εισάγει τα δεδομένα στα δεκατρία πρώτα πεδία:

- Διατομή σήραγγας (m^2)
- Μήκος σήραγγας/αγωγού (m)
- Διάμετρος αγωγού (m)
- Αριθμός εργαζομένων εντός σήραγγας
- Απαιτούμενη παροχή ανά εργαζόμενο (m^3/min)
- Συνολική ιπποδύναμη οχημάτων (HP)
- Απαιτούμενη παροχή ανά HP
- Ελάχιστη ταχύτητα αέρα (m/sec)
- Συντελεστής απόδοσης ανεμιστήρα (%)
- Συντελεστής α
- Συντελεστής β
- Συντελεστής γ
- Συντελεστής λ΄

Ακολούθως, πατώντας το πρώτο κουμπί “Υπολογισμός” εμφανίζονται τα οχτώ πρώτα αποτελέσματα, τα οποία αφορούν στα ακόλουθα:

- Απαιτούμενη παροχή στο μέτωπο (m^3/s)
- Στατικό ύψος υπό την παραδοχή μηδενικών διαφυγών - ho (mmH_2O)
- Το δυναμικό ύψος - h δυναμικό (mmH2O)
- Το στατικό ύψος - h στατ.ανεμιστήρα (mmH2O)
- Η πίεση ανεμιστήρα h και δυναμικό ύψος h_δ - h ολικό ανεμιστήρα (mmH2O)
- Παροχή ανεμιστήρα - Q ανεμιστήρα (m^3/sec)
- N: ισχύς ανεμιστήρα σε HP
- N: ισχύς ανεμιστήρα σε kW

Τα έξι πρώτα αποτελέσματα εμφανίζονται με στρογγυλοποίηση στο ψηφίο των χιλιοστών, η ισχύς (σε HP και σε kWh) εμφανίζεται με στρογγυλοποίηση στο ψηφίο των μονάδων (ακέραιος αριθμός).

Για τον υπολογισμό των οικονομικών μεγεθών, ο χρήστης πρέπει να εισάγει:

- Το κόστος αγοράς του ανεμιστήρα (σε €)
- Το κόστος αγοράς του αγωγού (σε €)
- Κόστος ηλεκτρικής ενέργειας (ευρώ/kWh)

- Ώρες λειτουργίας ανά ημέρα
- Ημέρες λειτουργίας ανά έτος
- Διάρκεια έργου (έτη)
- Επιτόκιο αναγωγής (%)
- Υπολειματική αξία ανεμιστήρα (%)

Πατώντας το δεύτερο κουμπί "Υπολογισμός" υπολογίζονται η Κ.Π.Α. και ετήσιο ισοδύναμο κόστος του αερισμού, με στρογγυλοποίηση στο ψηφίο των εκατοντάδων.

Ο χρήστης πρέπει να προσέξει τα παρακάτω:

- Δεν επιτρέπεται η εισαγωγή αρνητικών τιμών και μη αριθμητικών χαρακτήρων.
- Η υποδιαστολή στους δεκαδικούς αριθμούς εισάγεται με "." (τελεία).
- Για να εκτελεστούν οι υπολογισμοί θα πρέπει να έχουν συμπληρωθεί όλα τα πεδία.

Σε περίπτωση που η εισαγωγή των δεδομένων δεν πραγματοποιηθεί σύμφωνα με τα παραπάνω εμφανίζεται μήνυμα λάθους.

Δεξιά από κάποια πεδία που συμπληρώνονται από το χρήστη υπάρχουν κουμπιά "Help". Πατώντας τα κουμπιά αυτά ανοίγουν παράθυρα με κείμενα βοήθειας. Τα κείμενα αυτά παρέχουν διευκρινιστικές πληροφορίες στο χρήστη σχετικά με τις τιμές που πρέπει να εισάγει στα αντίστοιχα πεδία. Τα παράθυρα αυτά κλείνουν πατώντας το "x" που βρίσκεται πάνω δεξιά.

Το παράθυρο του προγράμματος κλείνει και το πρόγραμμα τερματίζεται πατώντας το "x" που βρίσκεται πάνω δεξιά στο πλαίσιο του παραθύρου.

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Τα υπόγεια έργα χρονολογούνται από τα πρώτα βήματα του ανθρώπου. Μέχρι τα μέσα του 20^{ου} αιώνα οι εφαρμογές των υπογείων έργων ήταν περιορισμένες, καθώς οι υπόγειες κατασκευές ήταν περισσότερο προσανατολισμένες σε μεταλλευτικούς σκοπούς. Τα τελευταία χρόνια, ωστόσο, τα υπόγεια έργα επιδεικνύουν μια δυναμική και διαρκώς αναπτυσσόμενη πορεία σε πληθώρα εφαρμογών, π.χ. συγκοινωνιακών, περιβαλλοντικών στρατιωτικών, κ.ά., αποτελώντας πλέον μια πάγια κατασκευαστική πρακτική.

Παράλληλα με την ανάπτυξη των υπογείων έργων δημιουργήθηκε η ανάγκη για μεγαλύτερα, πιο σύνθετα και οικονομικότερα έργα, τα οποία απαιτούσαν σημαντικές μεταβολές στη χρησιμοποιούμενη τεχνολογία. Σε αυτό το πλαίσιο εξελίχθηκαν τεχνικές εξόρυξης με χρήση εκρηκτικών υλών, συστήματα φόρτωσης και μεταφοράς με χρήση ντιζελοκίνητου μηχανολογικού εξοπλισμού, κ.λπ. Η υιοθέτηση των συστημάτων αυτών, όμως, επιβάρυνε λόγω των παραγόμενων και εκπεμπόμενων ρύπων σε σημαντικό βαθμό το περιβάλλον εργασίας, με αποτέλεσμα ο βοηθητικός αερισμός να αποτελεί αναπόσπαστο κομμάτι της κατασκευής των υπογείων έργων. Η ποσότητα του αέρα που διοχετεύεται με το βοηθητικό αερισμό επιτυγχάνει την αραίωση και απαγωγή των καπνών που παράγονται από την αποσύνθεση των χρησιμοποιούμενων εκρηκτικών υλών και των καυσαερίων που προέρχονται από τις εκπομπές των μηχανών εσωτερικής καύσεως. Και συντελεί στη διαμόρφωση του κλίματος στον υπόγειο χώρο (ρύθμιση θερμοκρασίας και υγρασίας) εξασφαλίζοντας ένα ασφαλές, υγιεινό και αποδοτικό περιβάλλον.

Στην παρούσα διπλωματική εργασία δημιουργήθηκε ένα εύχρηστο εργαλείο για την εκτίμηση των τεχνικών παραμέτρων και οικονομικών μεγεθών που αφορούν στο σχεδιασμό κυκλωμάτων βοηθητικού αερισμού (π.χ. επιλογή κατάλληλου ανεμιστήρα, αγωγού, κ.ά.). Η υλοποίηση του προγράμματος έγινε με τη χρήση της γλώσσας προγραμματισμού Java. Η συγκεκριμένη γλώσσα προγραμματισμού επιλέχθηκεδιότι προσφέρει αξιοπιστία, ασφάλεια, ευελιξία και δυνατότητα δημιουργίας γραφικού περιβάλλοντος, που καθιστά ευκολότερη και πιο άμεση τη χρήση του προγράμματος. Το πρόγραμμα έχει ελάχιστες απαιτήσεις υπολογιστικής ισχύος, είναι ιδιαίτερα μικρό (≈24Kbytes) σε μέγεθος και εξαιρετικά απλό στη χρήση του. Επιπλέον, παρέχει τις απαιτούμενες πληροφορίες για τη συμπλήρωση των αντίστοιχων πεδίων μέσω κουμπιών “Help” που μπορεί να πατήσει ο χρήστης. Το πρόγραμμα επιτρέπει στο χρήστη να εξετάζει

πολλά εναλλακτικά σενάρια αλλάζοντας τις τιμές συγκεκριμένα πεδία, παρέχοντας πολύτιμες πληροφορίες για την εξεύρεση του βέλτιστου αποτελέσματος, τη βαρύτητα των παραμέτρων που υπεισέρχονται στους υπολογισμούς, κ.λπ.

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

[1] Hartman H. L., Ramani R. V., Mutmansky J. M., Wang Y. J., “Mine ventilation and air conditioning”, Canada, 1997.

[2] McPherson M. J., “Subsurface Ventilation and Environmental Engineering”, California, 1993.

[3] Οικονομόπουλος Ι. Ν., “Ο Αερισμός των Μεταλλείων”, ΕΜΠ, Αθήνα, 1991.

[4] Προεδρικό Διάταγμα 225/1989, “Υγιεινή και ασφάλεια στα υπόγεια τεχνικά έργα”, Αθήνα, 1989.

[5] Προεδρικό Διάταγμα 90/1999, “Οριακές Τιμές Έκθεσης”, Αθήνα, 1999.

[6] Κανονισμοί Μεταλλευτικών και Λατομικών Εργασιών (ΚΜΛΕ), “Αερισμός Υπόγειων Εργασιών”, Μέρος Γ’ άρθρο 71-75, Αθήνα, 1984.

[7] Καλιαμπάκος Δ. Σημειώσεις μαθήματος Υπόγεια έργα, Ε.Μ.Π., Αθήνα, 2003.

[8] Cadenhead R., Lemay L. Πλήρες εγχειρίδιο της Java 6, Μ.Γκιούρδας, Αθήνα, 2007.

[9] Εισαγωγή στη γλωσσά προγραμματισμού Java, Ε.Μ.Π., Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών, Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής Εργαστήριο Πολυμέσων.

[10] Κακαρόντζας Γ., Αντικειμενοστραφής Προγραμματισμός ΙΙ (Java), Τμήμα Τεχνολογίας Πληροφορικής ΤΕΙ Λάρισας.

[11] Θεοτόκης, Δ. Java-Notes, Ιόνιο Πανεπιστήμιο