ePublishing
ΕΘΝΙΚΟ ΚΕΝΤΡΟ ΤΕΚΜΗΡΙΩΣΗΣ
NATIONAL DOCUMENTATION CENTRE

Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση

Τόμ. 1 (2003)

2ο Συνέδριο Σύρου στις ΤΠΕ

ΕΝΑ ΜΟΝΤΕΛΟ ΔΙΑΔΙΚΤΥΑΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΜΕ ΥΠΟΣΤΗΡΙΞΗ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΤΗ ΔΙΔΑΣΚΑΛΙΑ ΤΩΝ ΑΡΧΩΝ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.

Βασίλειος Εφόπουλος, Γεώργιος Ευαγγελίδης, Βασίλειος Δαγδιλέλης , Θεόδωρος Κασκάλης

Βιβλιογραφική αναφορά:

Εφόπουλος Β., Ευαγγελίδης Γ., Δαγδιλέλης Β., & Κασκάλης Θ. (2025). ΕΝΑ ΜΟΝΤΕΛΟ ΔΙΑΔΙΚΤΥΑΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΜΕ ΥΠΟΣΤΗΡΙΞΗ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΤΗ ΔΙΔΑΣΚΑΛΙΑ ΤΩΝ ΑΡΧΩΝ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ . *Συνέδρια της Ελληνικής Επιστημονικής Ένωσης Τεχνολογιών Πληροφορίας & Επικοινωνιών στην Εκπαίδευση*, *1*, 733–741. ανακτήθηκε από https://eproceedings.epublishing.ekt.gr/index.php/cetpe/article/view/7460

ΕΝΑ ΜΟΝΤΕΛΟ ΔΙΑΔΙΚΤΥΑΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ ΜΕ ΥΠΟΣΤΗΡΙΞΗ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ ΓΙΑ ΤΗ ΔΙΔΑΣΚΑΛΙΑ ΤΩΝ ΑΡΧΩΝ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ.

Εφόπουλος Βασίλειος
Τμ. Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας
efop@uom.gr

Ευαγγελίδης Γεώργιος
Τμ. Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας
gevan@uom.gr

Δαγδιλέλης Βασίλειος
Τμ. Εκπαιδευτικής και Κοινωνικής Πολιτικής, Πανεπιστήμιο Μακεδονίας
dagdil@uom.gr

Κασκάλης Θεόδωρος
Τμ. Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονίας
kaskak@uom.gr

ΠΕΡΙΛΗΨΗ

Η παρούσα εισήγηση περιγράφει το ολοκληρωμένο διαδικτυακό προγραμματιστικό περιβάλλον WIPE (Web Integrated Programming Environment) και προσπαθεί να αναδείξει την εκπαιδευτική αξία του και τις διδακτικές καινοτομίες που ενσωματώνει. Το WIPE βασίζεται και επεκτείνει το περιβάλλον X-Compiler, τμήμα του εκπαιδευτικού λογισμικού «ΔΕΛΥΣ».

Το προτεινόμενο περιβάλλον ενσωματώνει εφαρμογή διαχείρισης χρηστών, μεταγλωττιστή γλώσσας Pascal σε PseudoAssembly (Web Compiler), βοηθητικά προγραμματιστικά εργαλεία και βάση δεδομένων όπου καταχωρούνται δεδομένα εκπαιδευτών και εκπαιδευομένων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: *compiler, e-learning, προγραμματιστικό περιβάλλον, interactive learning environment, programming and programming languages, education, Pascal, Assembly*

ΕΙΣΑΓΩΓΗ

Αντικείμενο της παρούσης εισήγησης είναι η παρουσίαση του WIPE, ενός πρωτότυπου ολοκληρωμένου διαδικτυακού περιβάλλοντος για την υποστήριξη της διδασκαλίας των αρχών του προγραμματισμού. Το WIPE βασίζεται και επεκτείνει το περιβάλλον X-Compiler, τμήμα του εκπαιδευτικού λογισμικού «ΔΕΛΥΣ» που έχει αναπτυχθεί στο πλαίσιο του έργου ΝΑΥΣΙΚΑ της ενέργειας «Οδύσσεια» του Υπουργείου Παιδείας. Η εμπειρία που απέκτησε η ομάδα ανάπτυξης του ΔΕΛΥΣ, από την εισαγωγή του στη δευτεροβάθμια εκπαίδευση, ήταν καθοριστική για την απόφαση επέκτασης και μεταφοράς στον παγκόσμιο ιστό του X-Compiler.

Προγραμματιστικά περιβάλλοντα που υποστηρίζουν τη διδασκαλία του προγραμματισμού σε αρχαρίους έχουν απασχολήσει την επιστημονική κοινότητα και ως αποτέλεσμα έχουν δημιουργηθεί ενδιαφέροντα προϊόντα. Πολλά μάλιστα από αυτά επικεντρώνουν την προσοχή τους στα «λάθη» των νέων προγραμματιστών και τις

δυσκολίες κατανόησης που αυτοί συναντούν. Ενδεικτικά αναφέρουμε τα VINCE [12] (animator γλώσσας C), Karel the Robot [10] και Karel Genie [8] (προγραμματιστικοί μικρόκοσμοι για τα αρχικά βήματα των εκπαιδευομένων), το περιβάλλον Thetis C [7] (με υποστήριξη σημαντικών προγραμματιστικών εργαλείων), το visual περιβάλλον BACCII [4] (παρουσίαση βασικών προγραμματιστικών λειτουργών με διαγράμματα ροής), το Dynalab [1] (με βηματική παρουσίαση κώδικα) και το WebLab [3] (Η διαδικτυακή έκδοση του Dynalab που υποστηρίζει ένα υποσύνολο της Java). Τα τελευταία δύο περιβάλλοντα έχουν πλέον ενσωματωθεί στο WebWorks που αποτελεί μια πολύ σημαντική προσπάθεια στο τομέα αυτό με έμφαση τους τομείς του program animation, context free grammar animation και finite state automaton animation.

Το WIPE διαφοροποιείται γιατί περιλαμβάνει α) ολοκληρωμένο περιβάλλον διαχείρισης χρηστών, β) μεταγλωττιστή γλώσσας Pascal σε PseudoAssembly γ) βοηθητικά προγραμματιστικά εργαλεία και δ) βάση δεδομένων όπου καταχωρούνται δεδομένα και αποθηκεύονται αποτελέσματα των προσπαθειών των εκπαιδευομένων. Στη συνέχεια παρουσιάζεται η αρχιτεκτονική του προγραμματιστικού περιβάλλοντος, η περιγραφή του τρόπου λειτουργίας του και η διδακτική του αξία.

Σχήμα 1: *Το περιβάλλον του μεταγλωττιστή*

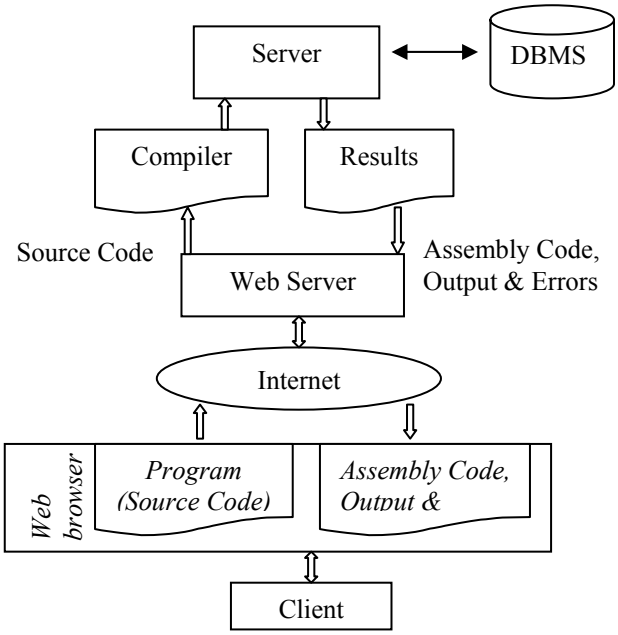
ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΠΡΟΓΡΑΜΜΑΤΙΣΤΙΚΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ

Το προγραμματιστικό περιβάλλον λειτουργεί στο Διαδίκτυο ή στο τοπικό Intranet και η πρόσβαση σε αυτό γίνεται μέσω Web browser. Το λογισμικό εγκαθίσταται στον

εξυπηρετητή (Server) του δικτύου (τοπικού ή Διαδίκτυο) και στον σταθμό εργασίας (client) μεταφέρονται οι πληροφορίες μέσω του browser. Η διαδικασία γίνεται ως εξής:

- Ο εκπαιδευόμενος (χρήστης) συνδέεται στον κόμβο δίνοντας username και password. Επιλέγει την εργασία-άσκηση με την οποία θα ασχοληθεί.
- Αφού εισέλθει στη σελίδα της εργασίας και διαβάσει τα δεδομένα που έχει καταχωρήσει ο καθηγητής του, γράφει το πρόγραμμα που θέλει να μεταγλωττιστεί στο παράθυρο του πηγαίου κώδικα και δίνει εντολή για μεταγλώττιση.
- Ο πηγαίος κώδικας του προγράμματος μεταφέρεται στον Web Server και αποθηκεύεται στη βάση δεδομένων.
- Γίνεται η μεταγλώττιση του πηγαίου κώδικα και τα αποτελέσματα αποθηκεύονται στη βάση δεδομένων στον Server και στέλνονται στον Web browser του χρήστη.
- Αν η μεταγλώττιση γίνει επιτυχώς, τότε στο παράθυρο της Assembly εμφανίζεται ο αντίστοιχος Assembly κώδικας. Κατόπιν ο χρήστης δίνει εντολή για εκτέλεση του προγράμματος και το μεταγλωττισμένο πρόγραμμα εκτελείται στον server. Τα αποτελέσματα (output) του προγράμματος μεταφέρονται στον Web browser του χρήστη.

Στο διάγραμμα που ακολουθεί παρουσιάζεται η Αρχιτεκτονική του όλου περιβάλλοντος.



Σχήμα 2: *Αρχιτεκτονική του προγραμματιστικού περιβάλλοντος*

ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ

Το ολοκληρωμένο σύστημα θα παρέχει τρεις διαφορετικούς τρόπους πρόσβασης.

- Πρόσβαση απλού χρήστη – επισκέπτη (anonymous user)
- Πρόσβαση καθηγητή
- Πρόσβαση μαθητή (ορίζεται από τον καθηγητή)
- Πρόσβαση διαχειριστή

Στην πρώτη περίπτωση, ο χρήστης που θα συνδεθεί μέσω ενός Web browser με τη συγκεκριμένη διεύθυνση, θα μπορεί να χρησιμοποιήσει μέρος των προσφερόμενων δυνατοτήτων. Πιο συγκεκριμένα θα μπορεί να συμμετέχει σε δημόσιες (global) εργασίες που έχουν καταχωρηθεί από τους καθηγητές ή θα μπορεί να γράψει τα προγράμματα που επιθυμεί σε γλώσσα Standard Pascal (ή ένα μεγάλο υποσύνολό της) και να δει τα αποτελέσματα τους. Στην περίπτωση αυτή καμία από τις δραστηριότητες του χρήστη δεν καταγράφεται στη βάση δεδομένων του συστήματος.

Στη δεύτερη περίπτωση, όπου ο χρήστης ζητά πρόσβαση στο σύστημα ως καθηγητής, συμπληρώνει μια φόρμα-αίτηση που είναι διαθέσιμη (μέσω συνδέσμου) από την κεντρική σελίδα του ιστοχώρου. Η αίτηση αυτή αποστέλλεται στο διαχειριστή (administrator) του συστήματος. Ο διαχειριστής αφού μελετήσει την αίτηση και την εγκρίνει, αποστέλλει με e-mail στον αιτούντα χρήστη, τον κωδικό πρόσβασης (password) στο σύστημα ως καθηγητής. Ο καθηγητής από τη στιγμή που εισέλθει στο περιβάλλον έχει τη δυνατότητα:

- Να ορίσει ομάδες χρηστών (τμήματα-τάξεις) που διδάσκει καθώς και τα στοιχεία της σχολικής μονάδας όπου ανήκει.
- Να καταχωρήσει τους μαθητές κάθε ομάδας-τμήματος. Σημειώνουμε ότι για κάθε μαθητή θα πρέπει να ορίσει τα στοιχεία εισόδου στο σύστημα (username και password).
- Να καταχωρήσει εργασίες-ασκήσεις για τις συγκεκριμένες ομάδες. Η εργασία-άσκηση είναι ο δομικός λίθος του project. Διαχειρίζεται από τον καθηγητή και αφορά τους χρήστες (μαθητές και επισκέπτες) που αυτός ορίζει ότι μπορούν να την παρακολουθήσουν. Σε κάθε εργασία-άσκηση ο καθηγητής καταχωρεί τα ακόλουθα στοιχεία: την εκφώνηση, τους στόχους, το χρόνο παράδοσης, βοηθητικά στοιχεία, πηγές πληροφόρησης (λ.χ. links), τους χρήστες που έχουν δικαίωμα να την παρακολουθήσουν και (προαιρετικά) τα δεδομένα εισόδου με τα αντίστοιχα δεδομένα εξόδου με τα οποία θα γίνει ο αυτόματος έλεγχος ορθότητας των προγραμμάτων. Παράλληλα ο καθηγητής μπορεί να εισάγει σημεία βοήθειας (hints) σε συγκεκριμένες περιπτώσεις λογικών λαθών στις ασκήσεις των μαθητών.
- Να ορίσει μια εργασία-άσκηση ως καθολική (global) έτσι ώστε να έχουν δικαίωμα να δουν τα περιεχόμενα της όλοι οι χρήστες (καθηγητές, μαθητές, επισκέπτες).
- Να δημιουργήσει ένα ή περισσότερα σύνολα χρηστών που θα μπορούν να εργαστούν από κοινού για τη διεκπεραίωση της εργασίας. Στον φάκελο του συνόλου αυτού έχουν πρόσβαση μόνο οι χρήστες που ανήκουν σε αυτό.
- Να παρακολουθεί τις επιδόσεις των μαθητών του.

Συνοπτικά ο καθηγητής είναι ο διαχειριστής (administrator) των τμημάτων και των μαθητών που ο ίδιος ορίζει. Καταχωρεί και αρχειοθετεί υλικό (ενδεικτικά προγράμματα, εργασίες, ασκήσεις) και το διαθέτει στους μαθητές του όποτε ο ίδιος κρίνει.

Η τρίτη περίπτωση πρόσβασης, αναφέρεται στο μαθητή ο οποίος προηγουμένως θα πρέπει να έχει καταχωρηθεί στη βάση δεδομένων από τον καθηγητή του. Όταν συνδεθεί και συμπληρώσει σωστά τα στοιχεία του θα έχει πρόσβαση στο προγραμματιστικό περιβάλλον και θα μπορεί:

- Να εξερευνήσει το προγραμματιστικό περιβάλλον και να γράψει τα δικά του προγράμματα (που καταγράφονται κάθε φορά που μεταγλωττίζονται)
- Να εκτελέσει τα προγράμματά του με διάφορους τρόπους (εξ' ολοκλήρου εκτέλεση, βηματική εκτέλεση με ταυτόχρονη παρακολούθηση των τιμών των μεταβλητών)
- Να λύσει τις προγραμματιστικές ασκήσεις που έχει καταχωρήσει ο καθηγητής του.
- Να στείλει τα αποτελέσματα των ασκήσεων στον καθηγητή του.

Όταν ολοκληρωθεί η καταχώρηση της εργασίας από τον καθηγητή, το σύστημα δημιουργεί προσωπικό φάκελο για τον κάθε μαθητή καθώς και φάκελο υποστήριξης (Library) όπου καταχωρείται το συνοδευτικό υλικό. Ο κάθε μαθητής έχει πρόσβαση στον προσωπικό του φάκελο και στο φάκελο υποστήριξης. Σημειώνεται ότι για κάθε μαθητή το σύστημα καταχωρεί στη βάση δεδομένων ένα αντίγραφο του πηγαίου κώδικα κάθε φορά που γίνεται μεταγλώττιση, τη χρονική στιγμή που έγινε αυτή καθώς και τα συντακτικά λάθη που παρουσιάστηκαν.

Η τέταρτη περίπτωση πρόσβασης αναφέρεται στο διαχειριστή που είναι ο χρήστης που έχει πλήρη δικαιώματα. Μπορεί να δημιουργήσει, να μεταβάλλει, να διαγράψει καθηγητές, μαθητές, επισκέπτες.

ΤΟ ΓΡΑΦΙΚΟ ΕΝΔΙΑΜΕΣΟ

Το γραφικό ενδιάμεσο του Web Compiler θα ενσωματώνεται στο περιβάλλον του Web browser και θα περιέχει τα ακόλουθα στοιχεία:

- Παράθυρο συγγραφής πηγαίου κώδικα (source code), όπου ο μαθητής πληκτρολογεί το πρόγραμμα.
- Παράθυρο assembly κώδικα (assembly code), όπου ο εμφανίζεται κάθε φορά η μεταγλώττιση σε assembly του πηγαίου κώδικα.
- Παράθυρο εξόδου αποτελεσμάτων (output), όπου εμφανίζονται τα αποτελέσματα της εκτέλεσης του προγράμματος.
- Παράθυρο μηνυμάτων – βοήθειας (message), όπου εμφανίζονται τα μηνύματα της μεταγλώττισης, η εκφώνηση των ασκήσεων, η βοήθεια (hints).
- Παράθυρο ελέγχου τιμών μεταβλητών (watcher), όπου ο μαθητής ελέγχει τις τιμές μεταβλητών ή παραστάσεων.
- Μπάρα εργαλείων με τις πιο συχνές λειτουργίες (Μεταγλώττιση, Εκτέλεση, Αντιγραφή, Επικόλληση κλπ.).

ΧΡΗΣΙΜΟΠΟΙΩΝΤΑΣ ΤΟΝ WEB COMPILER

Ο Web Compiler αποτελεί ένα διαδικτυακό περιβάλλον εισαγωγής στον προγραμματισμό, σχεδιασμένο για εκπαιδευτική χρήση. Το σύστημα θα είναι διαθέσιμο στην εκπαιδευτική κοινότητα και προσβάσιμο μέσω του Διαδικτύου. Στην παρούσας φάση η ανάπτυξη του web compiler γίνεται στην ηλεκτρονική διεύθυνση:

http://lachesis.it.uom.gr/x-comp/index.html

Το περιβάλλον του Web Compiler θα επιτρέπει την έκδοση, εκσφαλμάτωση και εκτέλεση προγραμμάτων γραμμένων σε γλώσσα Pascal. Ωστόσο ο Web Compiler θα ενσωματώνει μια σειρά χαρακτηριστικών που παρουσιάζουν διδακτικό ενδιαφέρον:

- Ο χρήστης θα μπορεί να παρακολουθήσει όλα τα ενδιάμεσα στάδια μιας εκτέλεσης προγράμματος: τη μεταγλώττιση, τον assembly κώδικα, τα περιεχόμενα των καταχωρητών, τις ενδιάμεσες τιμές όλων των μεταβλητών. Επί πλέον ο χρήστης θα μπορεί να παρέμβει απευθείας στον assembly κώδικα, να τον μεταβάλλει και στη συνέχεια να τον εκτελέσει.
- ·Η ύπαρξη πολλών μηνυμάτων, όσο γίνεται ακριβέστερων και επεξηγηματικών, θα βοηθούν τον αρχάριο προγραμματιστή στην εκσφαλμάτωση, αλλά και την γενικότερη βελτίωση του προγράμματός του.

Ως παράδειγμα των διδακτικών χαρακτηριστικών που θα προσφέρει το σύστημα αναφέρουμε το παρακάτω:

Το παράθυρο ελέγχου τιμών μεταβλητών που υλοποιήθηκε στην αρχική έκδοση του X-compiler, θα τροποποιηθεί και θα συμπεριλάβει αλγεβρικές και Boolean παραστάσεις καθώς και τις συμβολικές τιμές τους.

Για παράδειγμα ας υποθέσουμε ότι έχουμε το παρακάτω πρόγραμμα:

A:=A+B;
B:=A–B;
A:=A-B;

Στην παρίπτωση αυτή ο χρήστης θα μπορεί να δει τις τιμές των μεταβλητών A και B μετά από κάθε εντολή. Έτσι κάθε ενδιάμεση τιμή θα μπορεί να εμφανίζεται δίπλα:

A:=A+B; [[A;B]]
B:=A–B; [[A;B]]
A:=A-B; [[A;B]]

Όπου οι αγκύλες δηλώνουν ότι εμφανίζονται οι ενδιάμεσες τιμές των μεταβλητών.

Ας υποθέσουμε ότι δίνουμε στις A και B τις αρχικές τιμές 10 και 20 αντίστοιχα. Τότε το παράθυρο ελέγχου τιμών μεταβλητών θα εμφανίσει:

Αρχικά:
A:=A+B; [[A;B]]
B:=A–B; [[A;B]]
A:=A-B; [[A;B]]

Μετά την πρώτη εντολή:
A:=A+B; [[A=30; B=20]]
B:=A–B;
A:=A-B;

Μετά τη δεύτερη εντολή:
A:=A+B;

B:=A–B; [[A=30; B=10]]
A:=A-B;
Μετά την τρίτη εντολή:
A:=A+B;
B:=A–B;
A:=A-B; [[A=20; B=10]]

Όμως το περιβάλλον θα ενσωματώνει επίσης και αλγεβρικές παραστάσεις και θα μπορεί να ελεγχθεί η ορθότητα τους (αυτός ο έλεγχος είναι απολύτως ακριβής σε περίπτωση ανίχνευσης σφαλμάτων και μερικώς ακριβής σε περίπτωση εύρεσης ορθών αλγεβρικών παραστάσεων). Σχεδόν θα υπάρξει μια αυτοματοποιημένη διαδικασία επαλήθευσης/απόρριψης που ενσωματώνεται στο σύστημα. Αυτή η διαδικασία, όσο ξέρουμε, είναι μοναδική στον κόσμο και είναι μια καινοτομία της ομάδας ανάπτυξης του WIPE.

Για να αναπαραστήσουμε την παραπάνω διαδικασία θα χρησιμοποιήσουμε το προηγούμενο παράδειγμα. Στην περίπτωση αυτή η έκφραση έχει ως εξής:

[[A=x; B=y]]
A:=A + B;
B:=A –B;
A:=A - B;
[[A=y; B=x]]

Με άλλα λόγια απευθύνουμε στο σύστημα την ερώτηση και του ζητούμε να αποφασίσει αν γίνεται η μετάθεση τιμών των δύο μεταβλητών. Στην περίπτωση αυτή το πρόγραμμα θα αποφανθεί ότι η ανταλλαγή έγινε σωστά. Στην περίπτωση που βρεθεί λάθος, θα παραχθεί ένα αντιπαράδειγμα.

Ας δούμε το παράδειγμα:

```
[[A=x; B=y]]
p:=0;
q:=A;
while q>=B do begin
q:=q-B;
p:=p+1;
end;
[[A=pB+q; q<B]]
```

που ελέγχει αν η διαίρεση είναι τέλεια.

Το σύστημα θα μας επισημάνει ότι πιθανότατα το πρόγραμμα θα τρέχει ατέρμονα για τις τιμές A=5 και B=-12, που σημαίνει ότι τα A και B θα πρέπει να οριστούν ώς θετικοί αριθμοί.

Τα παραπάνω παραδείγματα παρουσιάζουν ένα τμήμα της διδακτικής αξίας του όλου περιβάλλοντος. Γενικότερα το WIPE θα επιτρέπει στον χρήστη να ελέγχει αν το πρόγραμμα του παράγει τα σωστά αποτελέσματα, θα τον καθοδηγεί στην αναζήτηση λογικών λαθών ή αδύνατων σημείων και θα τον βοηθά στην προσπάθεια του να διορθώσει και γενικότερα να βελτιώσει το πρόγραμμα του.

ΕΠΙΛΟΓΟΣ

Όπως ήδη αναφέρθηκε έχουν γίνει σημαντικές προσπάθειες για την ανάπτυξη εκπαιδευτικών περιβαλλόντων προκειμένου να υποστηριχθεί τόσο η διδασκαλία του προγραμματισμού αλλά και να βοηθηθούν οι αρχάριοι στην εκμάθηση του προγραμματισμού. Τα ερευνητικά αποτελέσματα σχετικά με τις δυσκολίες και τις λανθασμένες αντιλήψεις των σπουδαστών [14] ήταν αυτά που προκάλεσαν την ανάπτυξη εκπαιδευτικών προγραμματιστικών περιβαλλόντων. Με βάση τα ευρήματα αυτά σχεδιάσαμε το WIPE και τον Web Compiler. Στη συνέχεια συνοψίζουμε εκείνα τα χαρακτηριστικά του περιβάλλοντος που θεωρούμε ότι υποστηρίζουν σημαντικά τη διδασκαλία και εκμάθηση του προγραμματισμού.

- ·Η δυνατότητα εκτέλεσης του προγράμματος με ταυτόχρονη παρουσίαση των τιμών των μεταβλητών σε συνδυασμό με τη βηματική εκτέλεση θα βοηθήσει τον αρχάριο στην κατανόηση του μηχανισμού εκτέλεσης ενός προγράμματος των διεργασιών που συμβαίνουν στην εσωτερική κατάσταση της μηχανής και στην εκμάθηση των καθιερωμένων δομών, ζητήματα για τα οποία οι εμπειρικές μελέτες έδειξαν ότι προκαλούν δυσκολίες στους αρχάριους [5,6]
- ·Το παράθυρο μηνυμάτων βοήθειας θα υποστηρίξει την εκσφαλμάτωση καθώς τα μηνύματα των συνήθων προγραμματιστικών περιβαλλόντων δεν είναι ιδιαίτερα φιλικά προς το χρήστη και έχει διαπιστωθεί ότι αποτελούν αιτία δυσκολιών [6]
- Η καταγραφή της πορείας που ακολουθούν οι αρχάριοι όταν αναπτύσσουν ένα πρόγραμμα θα δώσει σημαντική βοήθεια στον καθηγητή [8], καθόσον θα του επιτρέψει να διαπιστώσει τα λάθη που κάνουν οι αρχάριοι και με βάση τα στοιχεία αυτά να τροποποιήσει το μάθημά του και να εισάγει ή να τροποποιήσει ασκήσεις προκειμένου να ξεπεραστούν οι δυσκολίες
- Η δυνατότητα στατιστικής επεξεργασίας των στοιχείων θα επιτρέψει στον καθηγητή να αξιολογήσει το μάθημά του και ενδεχομένως να αναθεωρήσει στοιχεία της διδασκαλίας του ή να προσθέσει επιπλέον διδακτικό υλικό για τη διευκόλυνση των μαθητών.
- Τέλος η διάθεση των αποτελεσμάτων στο Διαδίκτυο θα βοηθήσει την εκπαιδευτική κοινότητα στη διδασκαλία του προγραμματισμού αφού θα κάνει ευρύτερα γνωστά στοιχεία που αφορούν στις αντιλήψεις των μαθητών για τον προγραμματισμό.

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Birch M., Boroni, C., Goosey F., Patton S., Poole D., Pratt C., Ross R., “DYNALAB: A Dynamic Computer Science Laboratory Infrastructure Featuring Program Animation”, In Twenty-sixth SICGSE Technical Symposium on Computer Science Education (*SICGSE Bulletin*) vol. 27, pp. 29-33, 1995.
2. Boroni C., Goosey F., Grinder M., and Ross R., "A Paradigm Shift! The Internet, the Web, Browsers, Java, and the Future of Computer Science Education" Twenty-ninth SIGCSE Technical Symposium on Computer Science Education (SIGCSE Bulletin), 30(1), 145-149, 1998.
3. Boroni C., Goosey F., Grinder M., Ross R., and Wissenbach P., "WebLab! A Universal and Interactive Teaching, Learning, and Laboratory Environment for the

World Wide Web" SIGCSE Technical Symposium on Computer Science Education (SIGCSE Bulletin), 29(1), 199-203, March 1997.

4. Calloni B., Bagert B., "Iconic Programming in BACCII vs. Textual Programming: which is a better learning environment?", ACM, SIGSCE *'94,* Phoenix AZ, pp. 188-192, 1994.
5. Du Boulay B., O'Shea T., and Monk J., "The Black Box Inside the Glass Box: Presenting Computing Concepts to Novices", *Studying The Novice Programmer*, E. Soloway and J. Sprohrer (Eds.), Lawrence Erlbaum Associates, pp. 431-446, 1989.
6. du Boulay, B., (1989), "Some Difficulties Of Learning To Program", *In Studying The Novice Programmer*, Soloway, E., Sprohrer, J. (Eds.) Lawrence Erlbaum Associates, pp. 283-300.
7. Freund S. N. and Roberts E. S., "THETIS: An ANSI C programming environment designed for introductory use", ACM SIGSCE '96, Philadelphia, PA, USA, pp. 300-304, 1996.
8. MacGNOME Project, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213.
9. Papert, S. (1980). Mindstorms: Children, Computers and Powerful Ideas. Brighton, Sussex: Harvester Press.
10. Pattis R. E., Roberts J., and Stehlik M., "Karel - The Robot, A Gentle Introduction to the Art of Programming", 2nd edn. New York, Wiley, 1995.
11. Pierson, W. C. & Rodger, S. H. (1998). Web-based Animation of Data Structures Using JAWAA. Proceedings of the 1998 SIGCSE Symposium, Atlanta, 267-271.
12. Rowe, G. & Thorburn, G. (1999). Evaluation of VINCE - a Tool for Teaching Introductory Programming. Proceedings of the 7th Annual Conference on the Teaching of Computing, Belfast Aug 25-27, 98-102.
13. Satratzemi M., Dagdilelis V., Evangelidis G., «A System for Program Visualization and Problem-Solving Path Assessment of Novice Programmers", *SIGCE Bulletin, 33(3), 2001*
14. Δαγδιλέλης Β., (1996), "Διδακτική της πληροφορικής. Η διδασκαλία του προγραμματισμού: αντιλήψεις των σπουδαστών για την κατασκευή κι επικύρωση προγραμμάτων και διδακτικές καταστάσεις για τη διαμόρφωσή τους", Διδακτορική διατριβή, Τμήμα Εφ. Πληροφορικής Πανεπιστήμιο Μακεδονίας.
15. Σατρατζέμη Μ., Δαγδιλέλης Β., Ευαγγελίδης Γ., Εφόπουλος Β., «An Educational Programming Environment for Novices», *Πρακτικά του* 8^{ου} Πανελλήνιου Συνεδρίου στην Πληροφορική (της Ελληνικής Εταιρίας Επιστημόνων Η/Υ & Πληροφορικής ΕΠΥ*)*, Λευκωσία/Κύπρος, 8-10 Νοεμβρίου 2001.