ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

Καλές Πρακτικές Επιστήμη Πληροφορικής

Καλές πρακτικές για ένα εισαγωγικό μάθημα στον Προγραμματισμό ΗΥ

Μάγια Σατρατζέμη

Καθηγήτρια

Εφαρμοσμένης Πληροφορικής, Πανεπιστήμιο Μακεδονία

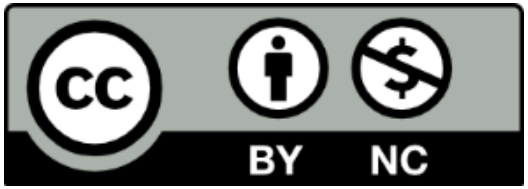
Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

Επιχειρησιακό Πρόγραμμα
Ανάπτυξη Ανθρώπινου Δυναμικού, Εκπαίδευση και Διά Βίου Μάθηση

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

ΕΣΠΑ 2014-2020
ανάπτυξη - εργασία - αλληλεγγύη

Άδεια χρήσης



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο

Επιχειρησιακό Πρόγραμμα
Ανάπτυξη Ανθρώπινου Δυναμικού,
Εκπαίδευση και Διά Βίου Μάθηση

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

ΕΣΠΑ
2014-2020
ανάπτυξη - εργασία - αλληλεγγύη

Καλές πρακτικές για ένα εισαγωγικό μάθημα στον Προγραμματισμό ΗΥ

Καλή πρακτική #1: Σύνδεση με Προγενέστερη Γνώση

- Πολλοί φοιτητές/τριες δεν κατανοούν άγνωστο ή αφηρημένο κώδικα.
- Χρησιμοποιήστε πχ ένα παιχνίδι ή κάποια επίδειξη για να εισαγάγετε νέες έννοιες κωδικοποίησης
- Οι φοιτητές/τριες πρέπει να κατανοήσουν τη λογική πίσω από την έννοια και την ανάγκη για τη χρήση της συγκεκριμένης έννοιας πριν δουν κώδικα.
- **Παραδείγματα**:
  - Conditionals: Simon Says
  - Loops: εισάγετε 1-1 ψάρι σε ενυδρείο μέχρι να εκπληρωθεί μια προϋπόθεση πχ χωρητικότητα ενυδρείου σε ψάρια
  - Arrays: διαχείριση κουτιά με χάπια
  - Random: ρίψη ζαριών ή περιστροφή τροχού
  - String functions: παίζοντας σκραμπλ

Καλή πρακτική # 2: Πρόοδος κώδικα

- **Παράδειγμα κώδικα**: Δείξτε κώδικα καλής ποιότητας και ζητήστε από τους φοιτητές/τριες να προβλέψουν την έξοδο. Δημιουργείστε κώδικα υψηλής ποιότητας.
- **Scaffolded Code**: Πάρτε κώδικα καλής ποιότητας και αφαιρέστε μερικές γραμμές. Οι μαθητές πρέπει να συμπληρώσουν τον κώδικα για να δημιουργήσουν το αναμενόμενο αποτέλεσμα. Αυτό μπορεί να χρησιμοποιηθεί σε μεγάλες ομάδες, μικρές ομάδες φοιτητών ή μεμονωμένα.
- **Κώδικας με σφάλματα**: Δώστε στους φοιτητές/τριες κώδικα που περιέχει σφάλματα. Οι φοιτητές/τριες πρέπει να περάσουν από τη διαδικασία εντοπισμού σφαλμάτων για να διορθώσουν τον κώδικα.
- **Από πρόβλημα σε κώδικα**: Τέλος, δώστε στους φοιτητές/τριες ένα πρόβλημα να γράψουν κώδικα για να το λύσουν. **Πρόβλημα > Ανάλυση** (τι σας ζητά να κάνετε το πρόβλημα, τι σας δίνεται και τι χρειάζεστε) > **Ψευκώδικας > κώδικας**

Καλή πρακτική #3: Ζωντανή κωδικοποίηση

- **1.Παρουσιάστε** ένα πρόβλημα στην τάξη.
- **2. Καταιγισμός ιδεών** για διατύπωση αλγόριθμου για την επίλυση του προβλήματος. Γράψτε ψευδοκώδικα στον πίνακα.
- **3.Κωδικοποιήστε τη λύση μαζί**. Είτε εμφανίστε τον κώδικα μέσω ενός προβολέα είτε χρησιμοποιήστε λογισμικό για να εμφανίσετε την οθόνη σας στις οθόνες των φοιτητών/τριων. Πιθανά σενάρια…
  - α. Ο δάσκαλος πληκτρολογεί τον κώδικα. Οι μαθητές δίνουν οδηγίες.
  - β. Ο δάσκαλος επιλέγει τυχαία έναν φοιτητή/τρια για να πληκτρολογήσει τον κώδικα.
- **4.Κάνε λάθη!** Κανείς δεν προγραμματίζει τέλεια. Δείξτε τον εντοπισμό σφαλμάτων (debugging), την επίλυση προβλημάτων, την κατανόηση των μηνυμάτων σφάλματος, τον τρόπο ανάγνωσης ενός API κ.λπ.

Καλή πρακτική #4: Προγραμματισμός σε ζευγάρια

- **Τι είναι ο προγραμματισμός σε ζευγάρια (pair)**; Οι φοιτητές/τριες συνεργάζονται σε ζευγάρια για να κωδικοποιήσουν τη λύση ενός προβλήματος. Ο ένας είναι ο **οδηγός (driver)** και ο άλλος **ο πλοηγός (navigator)**.
- **Οδηγός**: αυτός που πληκτρολογεί τον κώδικα σύμφωνα με τις οδηγίες του πλοηγού.
- **Πλοηγός**: αυτός που καθορίζει την προσέγγιση/κατεύθυνση.
- **Εναλλαγή ρόλων κάθε 10 λεπτά**.
- **Γιατί**; Ο προγραμματισμός σε ζευγάρια αναγκάζει τους φοιτητές/τριες να μιλήσουν για την προσέγγιση επίλυσης προβλημάτων και τις λύσεις κωδικοποίησης. Όταν οι φοιτητές/τριες εκφράζουν λεκτικά τις διαδικασίες σκέψης τους, συνεργάζονται, παρατηρούν περισσότερες λεπτομέρειες και ενισχύουν τις ικανότητές τους κωδικοποίησης.

- How Pair Programming Really Works

Καλή πρακτική # 5: Βελτιώστε υπάρχοντα κώδικα

- Όταν συζητάτε μια λύση κωδικοποίησης για ένα συγκεκριμένο πρόβλημα, ζητείστε από τους φοιτητές/τριες να βρουν έναν εναλλακτικό τρόπο κωδικοποίησης του ίδιου προβλήματος.

Ζητείται να εμφανίζετε τη λέξη Excellent 5 φορές

```
printf("Excellent\n");
printf("Excellent\n");
printf("Excellent\n");
printf("Excellent\n");
printf("Excellent\n");
```

```
//μια καλύτερη λύση

for (i=1;i<6;i++
        printf("Excellent\n");

//και αυτή είναι ισοδύναμη
i=1;
while (i<6) {
        printf("Excellent\n");
        i++;
}
```

Καλή πρακτική #6: Δώστε Άμεσα, Στοχευμένη ανατροφοδότηση

Κακή ανατροφοδότηση:

- Καλή δουλειά!
- Αυτό είναι λάθος.
- Δοκιμάστε ξανά.
- Cool!

Καλή ανατροφοδότηση:

- Ήταν πολύ έξυπνο να χρησιμοποιήσει ένα βρόχο και μια συνθήκη υπό συνθήκη για να εναλλάσσεται ο τύπος εξόδου.
- Πώς μπορείτε να αλλάξετε τον κώδικας σας ώστε να εξάγει μόνο τους ζυγούς αριθμούς;
- Επαναλαμβάνετε αυτό το ίδιο μπλοκ κώδικα πολλές φορές. Πώς μπορείτε να το κάνετε αυτό πιο αποτελεσματικό;
- Τι θα γινόταν αν…;

Ζητείστε άμεση απάντηση

Ερώτηση

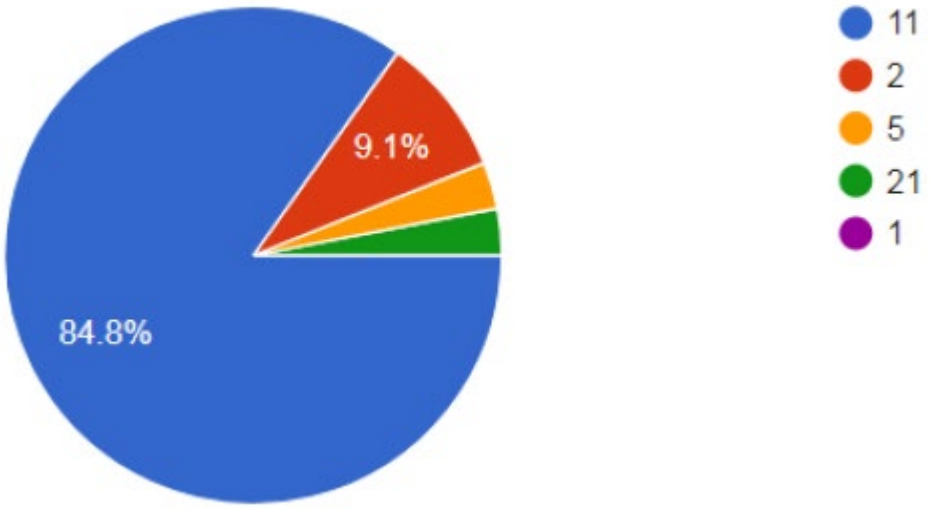
Ποιον είναι το αποτέλεσμα της παρακάτω έκφρασης

```
1 + 2 * 3 + 7 * 2 % 5
```

Απάντηση

- 11
- 2
- 5
- 21
- 1

Απαντήσεις



Καλή πρακτική #7: Διδάξτε Δεξιότητες διαχείρισης χρόνου

Πρόβλημα:

- Οι νέοι προγραμματιστές δεν έχουν ένα πλαίσιο αναφοράς για να καθορίσουν πόσο χρόνο θα διαρκέσει μια προγραμματιστική εργασία. Ο προγραμματισμός είναι συχνά μια νέα δεξιότητα για τους φοιτητές/τριες.

Λύσεις:

- Απολογισμός στο τέλος του μαθήματος. Τι σας εξέπληξε σε αυτό το έργο; Συζητήστε τον χρόνο που απαιτούνται συγκεκριμένες εργασίες.
- Δώστε στους φοιτητές/τριες συγκεκριμένο χρόνο. Πείτε τους ότι έχουν 10 λεπτά για να ολοκληρώσουν μια μικρή εργασία. Κάντε έναν γρήγορο έλεγχο αν την ολοκλήρωσαν στο τέλος του μαθήματος.
- Χωρίστε τα μεγάλα έργα σε μικρές, διαχειρήσιμες εργασίες. Κάντε ελέγχους προόδου.

Καλή πρακτική # 8: Διδάξτε τη διαφορά μεταξύ συνεργασίας και λογοκλοπής

Συζήτηση στην αρχή του μαθήματος

Λογοκλοπή

- Αντιγραφή & επικόλληση χωρίς κατανόηση του κώδικα
- Δανείζονται κώδικα για να μάθουν αλλά τον υποβάλλουν ως δικό τους.
- Υποβάλλουν κώδικα αλλά δεν μπορούν να τον εξηγήσουν ή να τον τροποποιήσουν.

Συνεργασία

- Συζήτηση σχεδίου, ιδεών για μια εργασία
- Αναζητούν βοήθεια όταν «κολλήσουν» με τον κώδικα που έχουν δημιουργήσει
- Διδάσκουν σε άλλον μια νέα τεχνική
- Οι φοιτητές/τριες είναι υπεύθυνοι για την κατανόηση του κώδικά τους.
- Οι φοιτητές/τριες μπορούν να εξηγήσουν, να αναδημιουργήσουν, να τροποποιήσουν ή να προβλέψουν οποιονδήποτε κώδικα έχουν δημιουργήσει.

Καλή πρακτική #9: Γίνε Mover, όχι Stopper

- **Movers**
  - Δεν τα παρατάνε!
  - Τα προβλήματα δημιουργούν ευκαιρία για μάθηση και βρίσκουν τρόπο (google, συνεργάτης, δάσκαλος).

Stoppers

- Σταματάν
- Ανταποκρίνονται την πρόκληση, αλλά τα παρατάν
- Νικητές- πες μου πώς
- Αναγνωρίστε τα συμπτώματα (καρδιά, κλάμα, θυμός, δεν ξέρω, δεν μπορώ)
- Καθορίστε στρατηγικές για να ξεπεραστεί η κατάσταση του προβλήματος (λίστα ελέγχου με το τι πρέπει να κάνετε)

- Learning and Teaching Programming: A Review and Discussion (εργασία με μεγάλη επίδραση)

Καλή πρακτική # 10: Βγείτε από την άνεσή σας. Θα σας βοηθήσει να δημιουργήσετε σχέσεις με τους μαθητές σας!

- Πότε ήταν η τελευταία φορά που κάνατε κάτι για πρώτη φορά;

- Προσαρμογή από το Best Practices in Computer Science Classes, GBEA, September 2017, Pam Whitlock