

ΠΡΟΤΥΠΟ ΣΧΟΛΙΑΣΜΟΥ ΑΣΚΗΣΕΩΝ/ΕΡΓΑΣΙΩΝ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ Ι

Οι προγραμματιστές βασίζονται σε σχόλια για να μπορούν να εξηγήσουν ή να καταλάβουν τι κάνει ένα κομμάτι κώδικα. Όπως σας εξηγήσαμε και στο φυλλάδιο γενικών αρχών μορφοποίησης κώδικα, τα σχόλια πρέπει να εξηγούν το σκοπό του κώδικα κι όχι απλά να περιγράφουν τι κάνει. Πέρα από αυτό, είναι σημαντικό να ακολουθούνται συγκεκριμένα πρότυπα και ως προς τη μορφοποίηση και ως προς το σχολιασμό. Στον επαγγελματικό χώρο τέτοια πρότυπα χρησιμοποιούνται πολύ συχνά, οπότε είναι καλή εξάσκηση να τα μάθετε από τώρα. Σε πιο άμεσο επίπεδο, καλός σχολιασμός βοηθά στην κατανόηση του προγράμματος και σας εξασφαλίζει καλύτερο βαθμό στις εργασίες.

Παρακάτω περιγράφουμε σε ποια σημεία του προγράμματός σας πρέπει να έχετε σχόλια και πώς αυτά πρέπει να μοιάζουν. Ακολουθείστε αυτές τις προδιαγραφές σε όλα τα προγράμματα που γράφετε.

Υπάρχουν 4 είδη σχολίων:

1. Σχόλια προγράμματος

Αυτά μπαίνουν στην αρχή του προγράμματος κι εξηγούν το γενικό σκοπό του προγράμματος. Περιλαμβάνουν:

- Όνομα προγραμματιστή/ών
- Ημερομηνία
- Σύντομη περιγραφή του σκοπού του προγράμματος
- Είσοδος προγράμματος
- Έξοδος προγράμματος
- Υποθέσεις/συνθήκες (πχ. υποθέσεις για τα δεδομένα εισόδου, συνθήκες υπό των οποίων το πρόγραμμα δε δουλεύει, κτλ.)
- Ιστορικό, δηλαδή όνομα προγραμματιστή που έχει κάνει αλλαγές στο πρόγραμμα, ημερομηνία και αιτιολογία (στην εργασία εξαμήνου, εδώ θα σημειώσετε ποια κομμάτια έκανε κάθε μέλος της ομάδας)

2. Σχόλια συναρτήσεων

Αυτά μπαίνουν πριν την υλοποίηση μιας συνάρτησης και συνήθως περιλαμβάνουν:

- Επεξήγηση των παραμέτρων
- Συνθήκες/Υποθέσεις Εισόδου (Preconditions) . Τι πρέπει να ισχύει πριν ξεκινήσει η συνάρτηση
- Συνθήκες Εξόδου (Postconditions). Τι ισχύει στο τέλος της συνάρτησης

Παράδειγμα:

```

/*****
 * void removeDuplicates(char *names[], int size)
 * Purpose: Afairei epanalambanomena onomata apo ena pinaka
 *          onomatwn
 * Parameters: names: pinakas onomatwn
 *             size: megethos tou pinaka
 * Preconditions: O pinakas einai gematos me onomata
 *               (den exei kena)
 * Postconditions: - Kathe onoma emfanizetai akribws mia fora
 *                 ston pinaka.
 *                 - Yparxonta kena deixnoun se NULL
 *****/

```

Εδώ ας αναφέρουμε κάτι σχετικά με το σχεδιασμό συναρτήσεων. Κάθε συνάρτηση πρέπει να έχει μια καλά ορισμένη λειτουργία, να κάνει αυτή τη λειτουργία και τίποτα άλλο. Για παράδειγμα, η παραπάνω συνάρτηση πρέπει μόνο να βγάζει τις επαναλαμβανόμενες λέξεις από τον πίνακα. Δεν πρέπει να κάνει άλλα πράγματα όπως για παράδειγμα να τυπώνει τις λέξεις ή να διατάσει τα ονόματα σε αλφαβητική σειρά.

3. Σχόλια μεταβλητών

Χρησιμοποιούμε σχόλια κατά τη δήλωση μεταβλητών για να επεξηγήσουμε τι αποθηκεύουν.

Παράδειγμα:

```

double price;    /* timh pizza se euro */
int numtoppings; /* plithos ulikwn panw stin pizza */

```

4. Γενικά σχόλια κώδικα

Τοποθετούμε σχόλια πριν από ένα κομμάτι κώδικα που εκτελεί μια συγκεκριμένη λειτουργία, καθώς και δεξιά από μια εντολή της οποίας η λειτουργία δεν είναι αμέσως προφανής.

Παράδειγμα:

```

int main () {

    int i;
    char *names[SIZE];

    /* gemisma pinaka onomatwn me dedomena
       apo to pliktrologio */
}

```

```
for ( i = 0; i < SIZE; i++) {
    names[i] = (char *) malloc (sizeof(char) * NAMESIZE);
    fgets( board[i], NAMESIZE, stdin);
    /* afairesi telikoy '\n' pou prosthesi to fgets sto
       onoma */
    board[ strlen(board) - 1 ] = '\0';
}
return 0;
}
```

Παρατηρείστε ο,τι το σχόλιο πάνω από το for στο παραπάνω παράδειγμα θα μπορούσαμε να το είχαμε παραλείψει γιατί ο κώδικας είναι αρκετά ξεκάθαρος. Γενικά, προσπαθείτε πάντα να γράφετε καθαρό κώδικα. Αν δείτε ότι χρειάζεται να γεμίζετε κάθε γραμμή με σχόλια, τότε κάτι δεν πάει καλά με το σχεδιασμό του προγράμματός σας.