

Κανόνες μορφοποίησης κώδικα στον Προγραμματισμό 2

Τα προγράμματα που γράφετε για τις εργασίες στον Προγραμματισμό 2 πρέπει να ακολουθούν τις παρακάτω προδιαγραφές.

Δομή κώδικα

Τα βασικά στοιχεία ενός προγράμματος εμφανίζονται με την παρακάτω σειρά στο αρχείο C :

- Αρχικό σχόλιο προγράμματος
- Εντολές `#include`
- Εντολές `#define`
- Ορισμοί τύπων (`struct`, `enum`, κτλ.) εφόσον δεν περιλαμβάνονται σε `.h` αρχείο.
- Επικεφαλίδες (prototypes) συναρτήσεων
- Υλοποιήσεις συναρτήσεων (η `main` μπορεί να είναι πρώτη ή τελευταία)

Σχόλια

Τα σχόλια προγράμματος πρέπει να περιλαμβάνουν μια συνοπτική περιγραφή του προγράμματος, τυχόν οδηγίες για την εκτέλεση του και το όνομα του προγραμματιστή.

Τα σχόλια συνάρτησης πρέπει να περιλαμβάνουν μια συνοπτική περιγραφή της λειτουργίας της συνάρτησης, πληροφορίες για τις παραμέτρους και την τιμή επιστροφής. Εάν η συνάρτηση έχει παρενέργειες ή προϋποθέσεις, αυτές πρέπει να προσδιορίζονται. Τα σχόλια συνάρτησης γράφονται ακριβώς πριν την υλοποίηση της συνάρτησης.

Εάν κάποιο κομμάτι κώδικα χρειάζεται περαιτέρω επεξήγηση, τότε αυτή εισάγεται είτε ως σύντομο σχόλιο στην ίδια γραμμή, 2-3 spaces μετά τον κώδικα (και χωρίς να υπερβαίνει την 100ή στήλη) είτε ως σχόλιο πολλαπλών γραμμών πριν τον κώδικα που σχολιάζεται.

Η στοίχιση των σχολίων ακολουθεί τη στοίχιση του κώδικα.

Στοίχιση, κενά και μήκος γραμμής

Η στοίχιση πρέπει να είναι 4 κενά.

Επιτρέπεται μόνο μία εντολή ανά γραμμή.

Μια γραμμή δε μπορεί να υπερβαίνει τους 100 χαρακτήρες.

Αφήνετε κενά σε κατάλληλα σημεία ώστε να είναι διακριτά τα στοιχεία του κώδικα (π.χ. μετά από το κόμμα σε λίστα παραμέτρων, πριν και μετά από τελεστές .)

Αφήνετε μία κενή γραμμή μετά το τελευταίο άγκιστρο μιας συνάρτησης. Μην αφήνετε ποτέ περισσότερες από μία διαδοχικές κενές γραμμές.

Άγκιστρα

Το αρχικό άγκιστρο συνάρτησης τοποθετείται δεξιά της επικεφαλίδας, ή σε νέα γραμμή. Εάν τοποθετηθεί σε νέα γραμμή, να είναι στοιχισμένο όπως η επικεφαλίδα.

Το αρχικό άγκιστρο μιας εντολής τοποθετείται δεξιά της εντολής ή σε νέα γραμμή. Εάν τοποθετηθεί σε νέα γραμμή, να είναι στοιχισμένο όπως η εντολή.

Το τελικό άγκιστρο (}') τοποθετείται πάντα σε ξεχωριστή γραμμή χωρίς άλλο κώδικα σε αυτή, στοιχισμένο όπως η αρχική γραμμή της εντολής/συνάρτησης. Εξαίρεση είναι η εντολή `do while` (όπου το `while` βρίσκεται πάντα στην ίδια γραμμή με το άγκιστρο), και προαιρετικά το `else` σε αλυσιδωτές `else if`.

Συνίσταται να χρησιμοποιείτε άγκιστρα στα σώματα σύνθετων εντολών (`if`, `for`, κτλ.) ακόμη κι αν αυτά αποτελούνται από μία εντολή.

Ονόματα

Τα ονόματα μεταβλητών, σταθερών και συναρτήσεων πρέπει να είναι συνοπτικά και περιγραφικά. Δεν πρέπει να περιλαμβάνουν το όνομα του τύπου.

Τα ονόματα σταθερών γράφονται με όλα τα γράμματα κεφαλαία.

Τα ονόματα μεταβλητών και συναρτήσεων γράφονται με μικρά. Σε περίπτωση σύνθετων ονομάτων, διακρίνονται είτε με χρήση χαρακτήρα '_' ανάμεσα τους είτε ξεκινώντας το δεύτερο όνομα με κεφαλαίο γράμμα (π.χ. `num_students` ή `numStudents`).

Παραδείγματα

<pre>do { scanf("%d", &x); } while (x < 0);</pre>	NAI
<pre>do { scanf("%d", &x); } while (x < 0);</pre>	OXI. Ειδικά στις <code>do-while</code> , το <code>while</code> τοποθετείται στην ίδια γραμμή με το <code>}</code>
<pre>for (int i = 0; i < SIZE; i++) { print(student[i]); }</pre>	NAI
<pre>for (int index_for_students = 0; index_for_students < SIZE; index_for_students++) { print(student[index_for_students]); }</pre>	OXI (Το <code>index_for_students</code> είναι πολύ μεγάλο και δυσχεραίνει την ανάγνωση του κώδικα)

<pre>void add (int x, int y) { return x+y; }</pre>	OXI (Έχει μόνο 2 spaces - δε γράφουμε python!)
<pre>void add (int x, int y) { return x+y; }</pre>	NAI
<pre>void add (int x, int y) { return x+y; }</pre>	NAI
<pre>if (is_valid(name)) { insert(database, name); }</pre>	OXI
<pre>if (is_valid(name)) { insert(database, name); }</pre>	NAI
<pre>if (is_valid(name)) insert(database, name);</pre>	OXI (Η κλήση στη συνάρτηση έπρεπε να είναι σε νέα γραμμή)
<pre>if(x<0 x>10){ print(msg,x); }</pre>	OXI (Χρειάζονται κενά στη συνθήκη για να είναι ευανάγνωστη)
<pre>if (condition1) { action1; } else if (condition2) { action2; } else { action3; }</pre>	NAI
<pre>if (condition1) { action1; } else if (condition2) { action2; } else { action3; }</pre>	NAI