**8η Προγραμματιστική άσκηση για το μάθημα**
**343 Εισαγωγή στον Προγραμματισμό**
**Μιχάλης Μπέκος / Κωνσταντίνα Τζουβάρα**

**Χειμερινό εξάμηνο 2025-2026**
**Πανεπιστήμιο Ιωαννίνων**
**29/11/2025**

# 8η Προγραμματιστική Εργασία

**Στόχοι:** Εξοικείωση με τις συμβολοσειρές της C++.

## Άσκηση: **Μυστικός Κωδικός**

Να υλοποιήσετε ένα πρόγραμμα–παιχνίδι στο οποίο ο χρήστης προσπαθεί να “σπάσει” έναν μυστικό κωδικό (string) που δημιουργεί ο υπολογιστής. Ο κωδικός αποτελείται αποκλειστικά από πεζά γράμματα του αγγλικού αλφαβήτου. Ο χρήστης έχει συνολικά δέκα προσπάθειες.

### Περιγραφή παιχνιδιού

Ο υπολογιστής δημιουργεί έναν τυχαίο κωδικό, για παράδειγμα "mgrtvd". Ο χρήστης πληκτρολογεί επαναληπτικά μια λέξη–προσπάθεια (guess). Μετά από κάθε προσπάθεια, το πρόγραμμα εμφανίζει ανατροφοδότηση με βάση τους εξής κανόνες:

- Για κάθε χαρακτήρα που είναι σωστός και στη σωστή θέση, εμφανίζεται ο ίδιος ο χαρακτήρας.
- Για κάθε χαρακτήρα που υπάρχει κάπου αλλού μέσα στον κωδικό αλλά όχι στη σωστή θέση, εμφανίζεται *.
- Για κάθε χαρακτήρα που δεν υπάρχει καθόλου στον κωδικό, εμφανίζεται #.

Το παιχνίδι σταματά όταν ο χρήστης βρει σωστά τον κωδικό, ή εξαντληθούν οι 10 προσπάθειες.

**Παράδειγμα:**
*Κρυφός κωδικός:* "mgrtvd", *Προσπάθεια χρήστη:* "mdrtag", *Αποτέλεσμα:* m*rt#*

### Υλοποίηση της κλάσης Mastermind

Να υλοποιήσετε την κλάση `Mastermind`, η οποία θα διαχειρίζεται τη λειτουργία του παιχνιδιού. Η κλάση πρέπει να περιλαμβάνει:

Ιδιωτικά πεδία:

- `int codeLength` — το μήκος του μυστικού κωδικού
- `int maxAttempts` — ο μέγιστος αριθμός προσπαθειών
- `string secretCode` — ο μυστικός κωδικός

Ιδιωτική βοηθητική μέθοδο:

- `void generateCode(int length)` – Δημιουργεί έναν τυχαίο κωδικό μήκους `length`, αποτελούμενο από πεζά γράμματα του αγγλικού αλφαβήτου και τον αποθηκεύει στο `secretCode`.

Αντικείμενα της κλάσης `Mastermind` υποστηρίζουν τις παρακάτω (δημόσιες) μεθόδους:

| | |
|---|---|
| **`Mastermind(int length)`** | Ορίζει το μήκος του κωδικού ως `length`, θέτει τις προσπάθειες σε 10 και καλεί τη `generateCode`. |
| **`int getLength()`** | Επιστρέφει το μήκος του μυστικού κωδικού. |
| **`int getMaxAttempts()`** | Επιστρέφει τον μέγιστο αριθμό των προσπαθειών. |
| **`string getCode()`** | Επιστρέφει το μυστικό κωδικό. |
| **`string play(const string &guess)`** | Ελέγχει τους χαρακτήρες σύμφωνα με τους κανόνες του παιχνιδιού και επιστρέφει μια συμβολοσειρά με χαρακτήρες, * ή #, ανάλογα με τα στοιχεία της προσπάθειας. |
| **`bool isValidGuess(const string &guess)`** | Κάνει έλεγχο εγκυρότητας της προσπάθειας, δηλαδή επιστρέφει `true` αν η λέξη έχει σωστό μήκος και περιέχει μόνο πεζά γράμματα a–z. |

**Υλοποίηση main()**

Υλοποιήσετε τη συνάρτηση-οδηγό `main()` ως εξής:

Αρχικά ορίστε το μήκος του κωδικού με τιμή της επιλογής σας (π.χ. 6) και δημιουργήστε ένα αντικείμενο της κλάσης `Mastermind`. Στη συνέχεια ζητήστε επαναληπτικά από το χρήστη να εισάγει προσπάθειες μέχρι είτε να βρει τον κωδικό ή να εξαντληθούν οι προσπάθειες. Η κάθε προσπάθεια πρέπει να ελέγχεται για την εγκυρότητά της μέσω της `isValidGuess()` και το αποτέλεσμά της (από την `play()`), να εμφανίζεται στο χρήστη. Στο τέλος του παιχνιδιού να εμφανίζεται κατάλληλο μήνυμα νίκης ή ήττας.

Ενδεικτική εκτέλεση:

Μυστικός Κωδικός — Βρες τον! Έχεις 10 προσπάθειες.
Ο κωδικός έχει μήκος 6 χαρακτήρες.

Δώσε προσπάθεια: abcdef
Αποτέλεσμα: ab####

Δώσε προσπάθεια: ghijql
Αποτέλεσμα: #*#*q#

Δώσε προσπάθεια: stuvwx
Αποτέλεσμα: ####*#

Δώσε προσπάθεια: abhwqj
Αποτέλεσμα: ab*wq*

Δώσε προσπάθεια: abjwqh
Αποτέλεσμα: abjwqh

Σωστά! Το βρήκες σε 5 προσπάθειες.