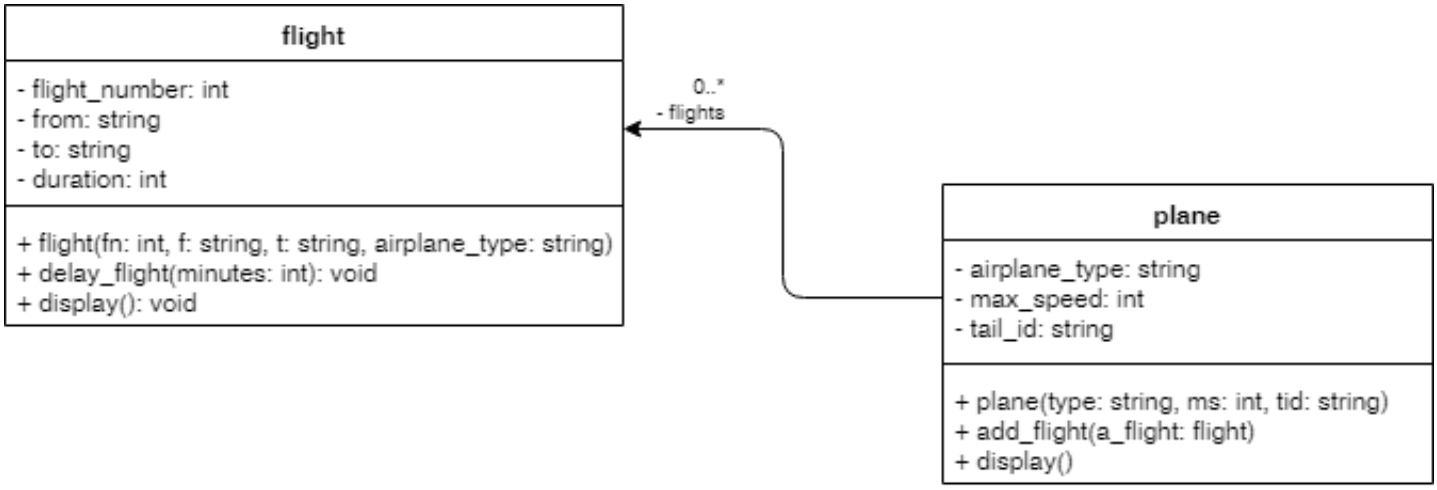
	ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΤΕΛΙΚΗ ΕΞΕΤΑΣΗ Τμήμα Πληροφορικής και Τηλεπικοινωνιών Πανεπιστήμιο Ιωαννίνων Διδάσκων: Γκόγκος Χρήστος	Άρτα 25/6/2019 **B**

Θέμα 1 [0.5 μονάδες το κάθε ερώτημα]

Δημιουργήστε μια κλάση image (εικόνα) που:

A. Να έχει ως ιδιωτικά μέλη δεδομένων τα: width (πλάτος), height (ύψος), color_depth (βάθος χρώματος σε αριθμό bytes που απαιτούνται για την αναπαράσταση των χρωμάτων της εικόνας).
B. Να έχει έναν constructor που να θέτει τα ιδιωτικά μέλη της κλάσης στις τιμές των παραμέτρων που θα δέχεται.
C. Να διαθέτει getters και setters μόνο για το πεδίο colors.
D. Να διαθέτει μια συνάρτηση get_size που να επιστρέφει το μέγεθος της εικόνας σε bytes υπολογίζοντας το γινόμενο πλάτος επί ύψος επί το βάθος χρώματος.
E. Υπερφορτώστε τον τελεστή < έτσι ώστε να διατάσσει εικόνες σε φθίνουσα σειρά μεγέθους bytes.
F. Υπερφορτώστε τον τελεστή << έτσι ώστε να επιστρέφει τα στοιχεία της κάθε εικόνας ως εξής: WIDTH=XXX HEIGHT=XXX, DEPTH=XXX, SIZE=XXX
G. Στη main
    I. Τοποθετήστε σε ένα vector τα ακόλουθα αντικείμενα[1] (640, 360, 3), (1024,768,2), (800, 600, 4), (1280,1024, 2). Ταξινομήστε τα περιεχόμενα του vector.
    II. Εμφανίστε τα περιεχόμενα του vector πρώτα σε αύξουσα και μετά σε φθίνουσα σειρά. Τα αποτελέσματα θα πρέπει να εμφανίζονται όπως παρακάτω.

```
SORTED
WIDTH:1280 HEIGHT: 1024 COLOR DEPTH: 2 SIZE: 2621440
WIDTH:800 HEIGHT: 600 COLOR DEPTH: 4 SIZE: 1920000
WIDTH:1024 HEIGHT: 768 COLOR DEPTH: 2 SIZE: 1572864
WIDTH:640 HEIGHT: 360 COLOR DEPTH: 3 SIZE: 691200
REVERSE SORTED
WIDTH:640 HEIGHT: 360 COLOR DEPTH: 3 SIZE: 691200
WIDTH:1024 HEIGHT: 768 COLOR DEPTH: 2 SIZE: 1572864
WIDTH:800 HEIGHT: 600 COLOR DEPTH: 4 SIZE: 1920000
WIDTH:1280 HEIGHT: 1024 COLOR DEPTH: 2 SIZE: 2621440
```



Εικόνα 1. Διάγραμμα κλάσεων για το θέμα 2

[1] Οι τιμές δίνονται με τη σειρά πλάτος, ύψος, βάθος

Θέμα 2 [Α: 4 μονάδες, Β: 1 μονάδα, C: 1 μονάδα]

A. Κατασκευάστε τις κλάσεις που δείχνει το UML διάγραμμα κλάσεων της Εικόνας 1 και οι οποίες αναπαριστούν αεροπλάνα και πτήσεις. Αναλυτικότερα κατασκευάστε τα ακόλουθα:
   I. Κλάσεις – σχέση (association) από την κλάση plane προς την κλάση flight. [1 μονάδα]
   II. Κατασκευαστές[2] [1 μονάδα]
   III. Συναρτήσεις display[3] και delay_flight[4] [1 μονάδα]
   IV. Συνάρτηση add_flight [1 μονάδα]

B. Στη main, δημιουργήστε τα ακόλουθα αντικείμενα:
   - Αντικείμενο αεροπλάνο: airplane type = “T1”, max speed = 450, tail id = “AB100”
   - Αντικείμενο αεροπλάνο: airplane type = “T2”, max speed = 500, tail id = “ZX900”
   - Αντικείμενο πτήση: flight number = 1001, from: “A”, to: “B” με τύπο αεροπλάνου “T1”
   - Αντικείμενο πτήση: flight number = 1002, from: “A”, to: “C” με τύπο αεροπλάνου “T1”
   - Αντικείμενο πτήση: flight number = 1003, from: “A”, to: “D” με τύπο αεροπλάνου “T2”

C. Στη main
   - Ορίστε ότι πτήση 1003 είχε καθυστέρηση 5 λεπτών.
   - Ορίστε ότι το αεροπλάνο AB100 πραγματοποίησε τις πτήσεις 1001 και 1002.
   - Ορίστε ότι το αεροπλάνο ZX900 πραγματοποίησε την πτήση 1003.
   - Καλέστε τη συνάρτηση display για τα δύο αεροπλάνα. Τα αποτελέσματα θα πρέπει να εμφανίζονται όπως παρακάτω.

```
Plane [type: T1 maximum speed: 450 tail id: AB100]
Flight [flight number: 1001 from: A to: B duration: 200
Flight [flight number: 1002 from: A to: C duration: 150
Plane [type: T2 maximum speed: 500 tail id: ZX900]
Flight [flight number: 1003 from: A to: D duration: 245
```

```
int get_flight_duration(std::string from, std::string to, std::string airplane_type)
{
    std::string data[5][4] = {
        {"A", "B", "T1", "200"},
        {"A", "B", "T2", "180"},
        {"A", "C", "T1", "150"},
        {"A", "C", "T2", "140"},
        {"A", "D", "T2", "240"},
    };
    for (int i = 0; i < 5; i++)
    {
        if ((data[i][0] == from) && (data[i][1] == to) && (data[i][2] == airplane_type))
        {
            return stoi(data[i][3]);
        }
    }
    return -1;
}
```

[2] Ο κατασκευαστής της flight δέχεται ως τελευταία παράμετρο τον τύπο του αεροπλάνου και με τη χρήση της συνάρτησης get_flight_duration, που δίνεται, επιστρέφει τη διάρκεια της πτήσης.

[3] Η συνάρτηση display για τα αντικείμενα flight εμφανίζει όλα τα πεδία της πτήσης, ενώ για τα αεροπλάνα εμφανίζει όλα τα πεδία του αεροπλάνου συμπεριλαμβανομένων των πτήσεων που του έχουν ανατεθεί.

[4] Η συνάρτηση delay_flight δέχεται ως παράμετρο την καθυστέρηση μιας πτήσης και την προσθέτει στη διάρκεια (duration) της πτήσης.