

Προγραμματιστικές Ασκήσεις, Φυλλάδιο 2
Να ολοκληρωθούν μέχρι 06/12/11 στις 12 μ.μ.

Οι παρακάτω ασκήσεις πρέπει να παραδοθούν μέχρι τις 6 Δεκεμβρίου τα μεσάνυχτα. Για την υλοποίηση τους θα πρέπει να χρησιμοποιήσετε κλάσεις και αντικείμενα. Στις κλάσεις δεν πρέπει να χρησιμοποιείτε public μεταβλητές. Εκτός από σταθερές, δεν πρέπει να χρησιμοποιήσετε καθολικές μεταβλητές (global variables). Τα προγράμματα σας πρέπει να εξασφαλίζουν ότι αποδεσμεύτηκε όλη η μνήμη η οποία είχε δεσμευτεί. Στα σχόλια του κώδικα σας, αναφέρετε για κάθε άσκηση ποιες κλάσεις σχετίζονται μεταξύ τους και τι είδους σχέση έχουν (composition ή aggregation).

1. Στην τάξη κατασκευάσαμε ένα πρόγραμμα το οποίο προσομοίωνε δύο οχήματα τα οποία κινούνται τυχαία σε ένα μονοδιάστατο χώρο ακεραίων μέχρι να συγκρουστούν. Τροποποιήστε το πρόγραμμα ώστε να μπορεί να κάνει την προσομοίωση σε πολυδιάστατους χώρους ακεραίων. Στον πολυδιάστατο χώρο για κάθε κίνηση του οχήματος, για κάθε διάσταση η τιμή της είτε αυξάνεται κατά ένα, είτε μένει ως έχει, είτε μειώνεται κατά ένα. Η τιμή οποιαδήποτε διάστασης δεν μπορεί να είναι μεγαλύτερη από 2, η μικρότερη από -2. Αν έχουμε d διαστάσεις αυτό σημαίνει ότι το αυτοκίνητο κινείται στον d-διάστατο κύβο $[-2,2]^d$. Αν σε μία διάσταση το όχημα είναι στα όρια του κύβου (δηλαδή η διάσταση έχει τιμή 2 ή -2), τότε η τυχαία επιλογή θα είναι είτε να μην αλλάξει η τιμή της διάστασης, είτε να αλλάξει με τον επιτρεπτό τρόπο.

Υλοποιήστε την παραπάνω προσομοίωση. Τα οχήματα ξεκινάνε από το σημείο μηδέν και συνεχίζουν μέχρι να συγκρουστούν. Στο τέλος τυπώνεται η θέση στην οποία έγινε η σύγκρουση και ο αριθμός των επαναλήψεων μέχρι να γίνει η σύγκρουση. Δώστε την επιλογή να μπορεί το πρόγραμμα να τυπώνει και τη θέση των δύο οχημάτων σε κάθε γύρο. Ο αριθμός των διαστάσεων d θα δίνεται ως είσοδος από το χρήστη. Το πρόγραμμα σας θα υλοποιεί μια κλάση Car, και η υλοποίησή σας θα πρέπει να είναι τέτοια ώστε ο κώδικας της main και της κλάσης Car να είναι ακριβώς ο ίδιος ανεξάρτητα από τον αριθμό των διαστάσεων. (Υπόδειξη: ορίστε μια κλάση Position, και μία κλάση Dimension).

2. Υλοποιήστε το παραπάνω παιχνίδι αλλά με πολλά ($MAX_CARS = 5$) οχήματα. Τα οχήματα κινούνται τυχαία σε ένα δυοδιάστατο πλέγμα (grid) 5×5 . Σε κάθε γύρο του παιχνιδιού, το κάθε όχημα διαλέγει τυχαία να μετακινηθεί Βόρεια, Νότια, Ανατολικά, ή Δυτικά. Η υλοποίησή σας θα χρησιμοποιεί μια κλάση Grid, η οποία θα κρατάει ένα δισδιάστατο πίνακα από αντικείμενα της κλάσης Cell, η οποία υλοποιεί τους κόμβους του πλέγματος. Τα αντικείμενα της κλάσης Cell είναι συνδεδεμένα μεταξύ τους, δηλαδή ο κάθε κόμβος θα έχει δείκτες στους τέσσερις γείτονες του στο πλέγμα, έναν βόρειο, ένα νότιο, ένα ανατολικό, και ένα δυτικό. Οι δείκτες είναι NULL αν ο κόμβος είναι στα όρια του πλέγματος. Για το κάθε όχημα έχουμε ένα id (το οποίο θα είναι ένας ακέραιος $[0, MAX_CARS]$ ώστε να μπορεί να χρησιμοποιηθεί για να βρούμε το όχημα σε ένα πίνακα), μια ακέραια τιμή strength, η οποία θα παίρνει μια τυχαία τιμή στο διάστημα $[0, 100)$, καθώς και τον κόμβο στον οποίο βρίσκεται. Η Cell θα μπορεί να τυπώνει τις πληροφορίες για τα οχήματα που είναι σε αυτό το κόμβο. Αν έχουμε μια σύγκρουση σε ένα κόμβο (δηλαδή παραπάνω από ένα όχημα σε αυτό το κόμβο) τότε καταρχάς θα ενημερώνουμε στην έξοδο για την σύγκρουση. Μετά, το όχημα που έχει το μικρότερο strength από όσα είναι στον ίδιο κόμβο θα διαγράφεται, και το παιχνίδι θα συνεχίζεται μέχρι να μείνει μόνο ένα όχημα. Η main θα πρέπει να τυπώνει την κατάσταση του παιχνιδιού σε κάθε γύρο.

Bonus: Υλοποιήστε και ένα όχημα το οποίο ελέγχεται από το χρήστη, το οποίο τοποθετείται στη θέση (0,0) και πρέπει ο χρήστης να το πάει στη θέση (4,4) προσπαθώντας να αποφύγει να συγκρουστεί με οχήματα μεγαλύτερης δύναμης.

3. Γράψτε ένα πρόγραμμα διαχείρισης Νοσοκομείου σε C++, που να περιλαμβάνει μια κλάση department, για τις 3 υπηρεσίες (ιατρική, νοσηλευτική και διοικητική), μια κλάση employee για εργαζόμενους και μια κλάση address για τις διευθύνσεις κατοικίας των εργαζομένων. Δεχτείτε πως ένας εργαζόμενος μπορεί να εργάζεται σε ένα μόνο τμήμα και πως μένει σε μία μόνο διεύθυνση. Οι διευθύνσεις μπορούν να υπάρχουν και χωρίς κατοίκους, οι υπηρεσίες όμως καταργούνται όταν δεν έχουν εργαζομένους και δημιουργούνται όταν αποκτούν έναν τουλάχιστον εργαζόμενο.

Για τις υπηρεσίες πρέπει να καταγράφονται πληροφορίες για το όνομα τους, το όνομα του διευθυντή τους, το πλήθος των εργαζομένων τους και το σύνολο των μικτών αποδοχών τους. Για τους εργαζόμενους πρέπει να καταγράφονται το ονοματεπώνυμο, η διεύθυνση τους, ο ΑΦΜ και ο ΑΜΚΑ τους, οι μικτές αποδοχές, τα επιδόματα και ο φόρος που παρακρατείται. Για τις διευθύνσεις πρέπει να καταγράφονται οδός, αριθμός, Τ.Κ., και πόλη.

Το πρόγραμμα θα διαβάζει στοιχεία για 7 εργαζόμενους, 1 στην ιατρική, 2 στην νοσηλευτική και 4 στη διοικητική υπηρεσία, 3 από τους οποίους μένουν στην ίδια διεύθυνση. Η είσοδος θα διαβάζεται από αρχεία τα οποία θα υποβάλετε μαζί με τον κώδικα.

Μαζί με τον κώδικα σας και τα αρχεία εισόδου υποβάλλετε και τα αποτελέσματα του κώδικα.

ΓΕΝΙΚΕΣ ΟΔΗΓΙΕΣ

Κάντε turnin το πρόγραμμα σας στο lab2@cs435. Για διευκόλυνση **ονομάστε το αρχείο με τον κώδικά σας lab2_dddd.cpp όπου dddd είναι ο Α.Μ. σας.**

π.χ. turnin lab2@cs435 lab2_dddd.cpp

Στον κώδικα να αναγράφονται σε σχόλια το όνομα το login και ο ΑΜ σας.