

# ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Χειμερινό εξάμηνο ακ. έτους 20012-2013

Α' ομάδα ασκήσεων

**Άσκηση 1:** Η παρούσα άσκηση σκοπό έχει να σας εξασκήσει στις συναρτήσεις κατασκευής-καταστροφής στην περίπτωση που έχουμε σύνθεση κλάσεων και στην προσβασιμότητα και εμβέλεια μεταξύ των μελών των κλάσεων.

Υλοποιήστε σε C++ μια σκηνή κινουμένων σχεδίων (Scene). Η σκηνή αποτελείται από τα εξής:

- Το υπόβαθρο (Background)
- Μια λίμνη (Lake)
- Μια παγοδρόμο (IceSkater)

Όταν δημιουργείται μια σκηνή εκτυπώνεται το μήνυμα "Scene was just created!". Όταν καταστρέφεται μια σκηνή εκτυπώνεται το μήνυμα "Scene is about to be destroyed". Για τη σκηνή ορίζεται η συνάρτηση-μέλος που υλοποιεί την εξέλιξή της (plot). Αυτή έχει ως εξής:

*Η λίμνη φωτίζεται.*

*Η παγοδρόμος ξεκινά την επίδειξή της.*

*Η παγοδρόμος γλιστρά έως το κέντρο της λίμνης.*

*Η παγοδρόμος περιστρέφεται.*

*Η παγοδρόμος πηδά.*

*Η παγοδρόμος γλιστρά έως την άλλη άκρη της λίμνης.*

*Η λίμνη περιστρέφεται.*

*Η λίμνη σκοτεινιάζει.*

Οι οντότητες που αποτελούν τη σκηνή περιγράφονται ως εξής:

Το υπόβαθρο αποτελείται από τα εξής:

- Μια οροσειρά (Mountains)
- Μια συστοιχία από δέντρα (Trees)
- Ένα σπιτάκι (Little house)

Αρχικοποιείται, εκτυπώνοντας το μήνυμα "Background was just created!". Καταστρέφεται εκτυπώνοντας το μήνυμα "Background is about to be destroyed".

Η οροσειρά δεν έχει δεδομένα. Αρχικοποιείται, εκτυπώνοντας το μήνυμα "Mountains were just created!". Καταστρέφεται εκτυπώνοντας το μήνυμα "Mountains are about to be destroyed".

Η συστοιχία από δέντρα δεν έχει δεδομένα. Αρχικοποιείται, εκτυπώνοντας το μήνυμα "Trees were just created!". Καταστρέφεται εκτυπώνοντας το μήνυμα "Trees are about to be destroyed".

Το σπιτάκι δεν έχει δεδομένα. Αρχικοποιείται, εκτυπώνοντας το μήνυμα "A little house was just created!". Καταστρέφεται εκτυπώνοντας το μήνυμα "A little house is about to be destroyed".

*Σημείωση:* Στα παραπάνω, δεν θεωρούμε ότι κάποια πληροφορία χρειάζεται απόκρυψη.

Η παγοδρόμος έχει την πληροφορία σχετικά με την απόσταση που διανύει σε μια ολίσθηση (distance) καθώς και την απόσταση που βρίσκεται από την αριστερή άκρη της λίμνης (position) (Σημείωση: για ευκολία θεωρείστε ότι η λίμνη έχει σχήμα ορθογώνιο και ότι παγοδρόμος κινείται κατά μήκος της βάσης του. Οπότε, ως “άκρη” εννοείται η αριστερή κορυφή, “άλλη άκρη” η δεξιά κορυφή και “κέντρο της λίμνης” το μέσο της βάσης του ορθογωνίου αυτού).

Η συμπεριφορά της παγοδρόμου είναι η εξής:

Κατά την αρχικοποίηση, βρίσκεται στην άκρη της λίμνης και ανατίθεται η τιμή της απόστασης που διανύει σε μια ολίσθηση. Επίσης, εκτυπώνεται το μήνυμα "An IceSkater participates". Κατά την καταστροφή, εκτυπώνεται το μήνυμα "An IceSkater will be discarded!". Η παγοδρόμος ξεκινά την επίδειξή της (start), εκτυπώνοντας το μήνυμα "Here I am ...". Η παγοδρόμος γλιστρά κατά μήκος μιας απόστασης (skate), μεταβάλλοντας την απόσταση από την άκρη της λίμνης κατά την απόσταση αυτή και εκτυπώνοντας το μήνυμα "An IceSkater skates" καθώς και το πλήθος των ολισθήσεων που έκανε. Η παγοδρόμος περιστρέφεται (rotate), εκτυπώνοντας το μήνυμα "An IceSkater spins on position" και την απόσταση από την άκρη της λίμνης. Η παγοδρόμος πηδά (jump) εκτυπώνοντας το μήνυμα "An IceSkater jumps on position" και την απόσταση από την άκρη της λίμνης.

Η λίμνη έχει την πληροφορία σχετικά με το μήκος της (το μήκος της βάσης της) καθώς και την απόχρωση του χρώματός της, dark ή light.

Η συμπεριφορά της λίμνης είναι η εξής:

Κατά την αρχικοποίηση, ανατίθεται το μήκος της και η απόχρωση του χρώματός της είναι dark. Επίσης, εκτυπώνεται το μήνυμα "A lake appears". Κατά την καταστροφή, εκτυπώνεται το μήνυμα "A lake will disappear!". Η λίμνη φωτίζεται (light\_up) θέτοντας την τιμή light στην απόχρωση και εκτυπώνοντας το μήνυμα "Lights on!". Η λίμνη σκοτεινιάζει (darken) θέτοντας την τιμή dark στην απόχρωση και εκτυπώνοντας το μήνυμα "Lights off!". Η λίμνη περιστρέφεται (rotate), εκτυπώνοντας το μήνυμα "Lake rotated!"

Υλοποιήστε τα παραπάνω, μέσω των κατάλληλων κλάσεων, ορίζοντας τα μέλη-δεδομένα που χρειάζονται και τις συναρτήσεις-μέλη που υλοποιούν την παραπάνω συμπεριφορά.

Για να δοκιμάσετε την υλοποίησή σας, ορίστε μια συνάρτηση main που να περιέχει τον παρακάτω κώδικα

```
Scene scene(L,Dist);
scene.plot();
```

Το μήκος της (βάσης της) λίμνης,  $L$ , και η απόσταση που διανύει σε μια ολίσθηση μια παγοδρόμος,  $Dist$ , είναι παράμετροι και να δίδονται κατά την κλήση του προγράμματος από τη γραμμή εντολής, με αυτή τη σειρά.

**Άσκηση 2:** Να υλοποιηθεί σε C++ μια σκηνή κινουμένων σχεδίων επιδείξεων σχηματισμών παγοδρόμων που έχει ως εξής:

Θεωρούμε ότι υπάρχει μια λίμνη μήκους  $L$ . Για ευκολία θεωρούμε ότι αυτή έχει σχήμα ορθογώνιο παραλληλόγραμμο και το  $L$  αντιστοιχεί στο μήκος της βάσης του.

Θεωρούμε ότι έχουμε και  $N$  παγοδρόμους. Οι μισές φοράνε άσπρο κουστούμι και οι άλλες μισές μαύρο. Κάθε παγοδρόμος γεννάται με τυχαίο χρώμα κουστουμιού. Επίσης, κάθε παγοδρόμος χαρακτηρίζεται από την ταυτότητά της (τυχαίο διακριτό αριθμό) κι από την απόσταση της λίμνης που διανύει σε μια ολίσθηση (τυχαίος αριθμός μικρότερος του  $L$ ). Τέλος, κάθε παγοδρόμος έχει και μια δεξιότητα, είτε spin είτε jump, που αποκτά κατά τη γέννησή της (τυχαία). Η

παγοδρόμος επιδεικνύει (**perform**) τη δεξιότητα αυτή εκτυπώνοντάς την σε ένα μήνυμα, π.χ. **"My skill is"** ακολουθούμενο από την τιμή της δεξιότητας.

Οι παγοδρόμοι γεννώνται μία-μία. Αμέσως μόλις γεννηθεί μια παγοδρόμος, τοποθετείται στο τέλος μιας ουράς που θεωρούμε ότι θα σχηματίσουν. Σε αυτή την ουρά, τελικά θα τοποθετηθούν όλες οι παγοδρόμοι. Στην ουρά απαιτείται διαδοχικές παγοδρόμοι να φοράνε διαφορετικό χρώμα ρούχα. Αν τύχει και γεννηθούν διαδοχικά παγοδρόμοι με το ίδιο χρώμα με της τελευταίας παγοδρόμου της ουράς, αυτές προστίθενται σε μια δεύτερη ουρά και περιμένουν μέχρι να γεννηθεί και να προστεθεί παγοδρόμος του άλλου χρώματος στην πρώτη. Τότε, η πρώτη παγοδρόμος της ουράς που περιμένει αφαιρείται και προστίθεται στην βασική ουρά.

Στη συνέχεια, θεωρούμε ότι στις τέσσερις άκρες της λίμνης (κορυφές του ορθογωνίου) θα συγκεντρωθούν οι παγοδρόμοι με τη λογική στοίβας. Θεωρούμε ότι σχηματίζεται σε κάθε άκρη της λίμνης από μία στοίβα.

Οι στοίβες γεμίζουν εναλλάξ με κυκλική φορά (δηλαδή πρώτα η κάτω αριστερή, μετά η κάτω δεξιά, μετά η πάνω δεξιά και μετά η πάνω αριστερή κ.ο.κ.), αφαιρώντας μία-μία τις παγοδρόμους από την ουρά. Πάλι απαγορεύεται να εισαχθούν διαδοχικά στην ίδια στοίβα παγοδρόμοι με ίδιο χρώμα κουστουμιών (οπότε κάθε παγοδρόμος πάει στην πρώτη που επιτρέπεται να εισαχθεί).

Κατόπιν, όταν θα έχουν τοποθετηθεί όλες οι παγοδρόμοι σε στοίβες, αφαιρούνται κυκλικά πάλι από αυτές, μία μία παγοδρόμος. Κάθε παγοδρόμος όταν αφαιρείται, επιδεικνύει τη δεξιότητά της και ολισθαίνει έως ότου βγει από την απέναντι πλευρά της λίμνης. Οι παγοδρόμοι βγαίνοντας εκτυπώνουν τη ταυτότητά τους, το χρώμα τους και τον αριθμό των διασκελισμών που έκαναν για να διανύσουν το μήκος της λίμνης (για ευκολία θεωρείστε ότι όλες οι παγοδρόμοι βρίσκονται σε μηδενική απόσταση από την άκρη της λίμνης ανεξάρτητα από τη θέση τους στη στοίβα).

Κατά την προσθήκη και την αφαίρεση παγοδρόμων από τις στοίβες και τις ουρές να εκτυπώνονται σχετικά μηνύματα, π.χ. **"adding IceSkater <id> to the UpperLeftStack"**, **"deleting IceSkater <id> from the MainQueue"**.

Να υλοποιήσετε τις απαραίτητες κλάσεις σε C++. Επίσης να υλοποιηθεί και μια συνάρτηση **main** που να εκτελεί το παραπάνω σενάριο.

Το  $L$  και το  $N$  είναι παράμετροι και να δίδονται κατά την κλήση του προγράμματος από τη γραμμή εντολής με τη σειρά αυτή.