

ΕΡΓΑΣΙΑ ΣΤΟΝ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ

ΤΡΙΤΟ ΠΑΡΑΔΟΤΕΟ

Υλοποίηση κληρονομικότητας και συναρτήσεων δεικτών σε C++

Το τρίτο παραδοτέο της εργασίας απαιτεί τη δημιουργία των κληρονομημένων κλάσεων των τετραγώνων και των βασικότερων ενεργειών του παιχνιδιού της Φοιτητούπολης. Συγκεκριμένα, θα υλοποιηθούν οι παρακάτω κλάσεις **που βρίσκονται όλες στα αρχεία `space.h` και `space.cpp`**:

1. Κλάση Space

Η κλάση αυτή δημιουργεί τα αντικείμενα των τετραγώνων του παιχνιδιού. Θα πρέπει να περιλαμβάνει μεταβλητές για το id, το όνομα, τον τύπο και την ενέργεια του τετραγώνου, καθώς και μια μεταβλητή-δείκτη που θα δείχνει στον ιδιοκτήτη του τετραγώνου (αρχικοποιείται σε NULL ή 0 για την τράπεζα, ενώ λειτουργεί ως δείκτης για αντικείμενα της κλάσης Player), την τιμή φόρου και το κόστος αγοράς του τετραγώνου. Επίσης θα πρέπει να περιλαμβάνει κάποιες συναρτήσεις get και set για τις μεταβλητές της κλάσης.

2. Κλάση Property

Η κλάση αυτή δημιουργεί τα αντικείμενα των τετραγώνων ιδιοκτησίας. Κληρονομεί την κλάση Space. Θα πρέπει να περιλαμβάνει μεταβλητές για το κόστος αγοράς της ιδιοκτησίας, το κόστος αναβάθμισης (δηλαδή κόστος κάθε σπιτιού), την κατηγορία της ιδιοκτησίας (δηλαδή τη χρωματική της περιοχή), τον τρέχοντα αριθμό των σπιτιών της ιδιοκτησίας. Επιπλέον, θα πρέπει να έχει μια μεταβλητή-πίνακα rent που θα έχει μέγεθος έξι και θα διατηρεί το ενοίκιο για κανένα, 1, 2, ..., 5 σπίτια. Επίσης θα πρέπει να περιλαμβάνει κάποιες συναρτήσεις get και set για τις μεταβλητές της κλάσης, καθώς και μια συνάρτηση για την ανέγερση σπιτιού στην ιδιοκτησία.

3. Κλάση RailRoad

Η κλάση αυτή δημιουργεί τα αντικείμενα των τετραγώνων κυλικίων. Κληρονομεί την κλάση Space. Θα πρέπει να περιλαμβάνει μια μεταβλητή για το κόστος αγοράς του κυλικείου καθώς και μια μεταβλητή-πίνακα rent που θα έχει μέγεθος τέσσερα και θα διατηρεί το ενοίκιο για 1, 2, 3, 4 κυλικεία (**προσοχή καθώς οι αριθμοδείκτες του πίνακα θα είναι 0, 1, 2, 3**). Επίσης θα πρέπει να περιλαμβάνει κάποιες συναρτήσεις get και set για τις μεταβλητές της κλάσης, καθώς και μια συνάρτηση για τον υπολογισμό και επιστροφή του ενοικίου με βάση το συνολικό πλήθος των κυλικίων που έχει ο κάτοχος του κυλικείου.

Υπόδειξη: Θα πρέπει να χρησιμοποιήσετε τη συνάρτηση `Player::getNumberOfRailRoadsOwned()`

4. Κλάση Utility

Η κλάση αυτή δημιουργεί τα αντικείμενα των τετραγώνων γραφείων. Κληρονομεί την κλάση Space. Θα πρέπει να περιλαμβάνει μια μεταβλητή για το κόστος αγοράς του γραφείου καθώς και μια μεταβλητή-πίνακα rentMultiplier που θα έχει μέγεθος δύο και θα διατηρεί τον πολλαπλασιαστή ενοικίου για 1, 2 γραφεία (**προσοχή καθώς οι αριθμοδείκτες του πίνακα θα είναι 0, 1**). Επιπλέον, θα πρέπει να έχει μια μεταβλητή-δείκτη dice που θα δείχνει στα ζάρια του παιχνιδιού, ώστε να χρησιμοποιηθούν για τον υπολογισμό του ενοικίου. Επίσης θα πρέπει να περιλαμβάνει κάποιες συναρτήσεις get και set για τις μεταβλητές της κλάσης, καθώς και μια συνάρτηση για τον υπολογισμό και επιστροφή του ενοικίου με βάση το συνολικό πλήθος των γραφείων που έχει ο κάτοχος του γραφείου (από όπου θα βγαίνει η τιμή του πολλαπλασιαστή ενοικίου) και το άθροισμα των ζαριών.

Υπόδειξη: Θα πρέπει να χρησιμοποιήσετε τη συνάρτηση `Player::getNumberOfUtilitiesOwned()`

5. Κλάση Tax

Η κλάση αυτή δημιουργεί τα αντικείμενα των τετραγώνων φόρων. Κληρονομεί την κλάση Space. Θα πρέπει να περιλαμβάνει μια μεταβλητή για την τιμή του φόρου καθώς και μια συνάρτηση get για να είναι εφικτή η προσπέλαση αυτής της τιμής.

Όλες οι παραπάνω κλάσεις θα πρέπει να έχουν και συναρτήσεις αρχικών συνθηκών. Κατά τη δημιουργία του τετραγώνου, η μεταβλητή owner θα πρέπει να τίθεται στην τιμή NULL (ή 0). Επίσης, ο αριθμός των σπιτιών σε κάθε ιδιοκτησία θα πρέπει αρχικά να είναι 0.

Επιπλέον, θα πρέπει να υλοποιήσετε τις παρακάτω **συναρτήσεις της κλάσης Player** στο αρχείο player.cpp:

- Συνάρτηση canBuy

Η συνάρτηση αυτή αναλαμβάνει να ελέγξει αν ο παίκτης μπορεί να αγοράσει ένα συγκεκριμένο τετράγωνο του παιχνιδιού. Στην πραγματικότητα **η λειτουργικότητα χωρίζεται με πολυμορφισμό σε τρεις συναρτήσεις**, τις `bool Player::canBuy(Property* property)`, `bool Player::canBuy(RailRoad* railroad)` και `bool Player::canBuy(Utility* utility)`. **Πρέπει να υλοποιήσετε και τις τρεις συναρτήσεις**. Οι συναρτήσεις αυτές θα πρέπει να ελέγχουν αν το τετράγωνο που δίνεται ως είσοδος δεν ανήκει σε κάποιον παίκτη και αν το ποσό χρήματων του παίκτη είναι μεγαλύτερο ή ίσο από την τιμή πώλησης του τετραγώνου. Εφόσον ισχύουν αυτές οι συνθήκες, η συνάρτηση θα πρέπει να επιστρέφει true, αλλιώς αν δε ισχύει οποιαδήποτε από αυτές τις συνθήκες, η συνάρτηση θα πρέπει να επιστρέφει false.

- Συνάρτηση canUpgrade

Η συνάρτηση αυτή δέχεται ως είσοδο μια ιδιοκτησία και αναλαμβάνει να ελέγξει αν ο παίκτης μπορεί να αναβαθμίσει τη συγκεκριμένη ιδιοκτησία. Επιστρέφει true (που σημαίνει ότι ο παίκτης μπορεί να αναβαθμίσει τη συγκεκριμένη ιδιοκτησία) **αν είναι αληθείς όλες οι παρακάτω συνθήκες**:

- Το τετράγωνο-ιδιοκτησία ανήκει στον παίκτη.
- Το ποσό χρήματων του παίκτη είναι μεγαλύτερο ή ίσο από την τιμή αναβάθμισης (ανέγερσης σπιτιού) του τετραγώνου.
- Ο παίκτης έχει στην κατοχή του όλα τα τετράγωνα της χρωματικής περιοχής (category) του τετραγώνου.
- Το πλήθος των σπιτιών του τετραγώνου-ιδιοκτησίας είναι μικρότερο του 5.

Σε κάθε άλλη περίπτωση επιστρέφει false.

Υπόδειξη: θα πρέπει να κάνετε ρητή μετατροπή τύπου ώστε να πάρετε τα τετράγωνα του παίκτη σε μορφή ιδιοκτησίας και να μπορείτε να ελέγξετε τη χρωματική περιοχή τους.

- Συνάρτηση buy

Η συνάρτηση αυτή δέχεται ως είσοδο ένα τετράγωνο και θέτει τον παίκτη ως κάτοχο του τετραγώνου. **Δεν πραγματοποιείται έλεγχος, ενώ επίσης δεν προστίθενται/αφαιρούνται χρήματα** (αυτές οι ενέργειες γίνονται εκτός της υλοποίησης αυτής της συνάρτησης).

- Συνάρτηση upgrade

Η συνάρτηση αυτή δέχεται ως είσοδο μια ιδιοκτησία και την αναβαθμίζει, προσθέτει δηλαδή ένα σπίτι στην ιδιοκτησία. **Δεν πραγματοποιείται έλεγχος, ενώ επίσης δεν προστίθενται/αφαιρούνται χρήματα** (αυτές οι ενέργειες γίνονται εκτός της υλοποίησης αυτής της συνάρτησης).

Εγκατάσταση

Για να βάλετε το project στο CodeBlocks, αρκεί να αποσυμπίεσετε το αρχείο Monopoly.zip σε μια τοποθεσία του σκληρού σας δίσκου, και στη συνέχεια από το περιβάλλον του CodeBlocks να επιλέξετε File → Open... και να επιλέξετε το Monopoly.cpb από τον φάκελο που μόλις αποσυμπίεσατε το project.

Παρατηρήσεις

Η υλοποίηση θα πρέπει να γίνει στη C++ και να μπορεί να ανοίξει με το CodeBlocks, **με τις εκδόσεις που χρησιμοποιούμε** στο πλαίσιο του μαθήματος. Ο κώδικάς σας θα πρέπει να είναι καλά τεκμηριωμένος, ώστε να είναι παντού σαφείς οι λεπτομέρειες υλοποίησης.

Για την υλοποίηση, σας δίνονται τα αρχεία κεφαλίδων .h των κλάσεων/συναρτήσεων που πρέπει να υλοποιήσετε καθώς και κάποιες βοηθητικές κλάσεις/συναρτήσεις. Επιπλέον, σας δίνεται ο κώδικας της συνάρτησης main (main.cpp). **Σε καμία περίπτωση δεν επιτρέπεται να επέμβετε στον κώδικα των κλάσεων και των συναρτήσεων αυτών. Σε περίπτωση που το κάνετε, η εργασία σας αυτομάτως θεωρείται λανθασμένη και μηδενίζεται. Θα πρέπει να γράψετε κώδικα μόνο στα αρχεία space.cpp και player.cpp.**

Παραδοτέο

Το παραδοτέο θα είναι ένα αρχείο zip με όνομα **Monopoly.zip** που θα περιλαμβάνει **όλο το project** (τα αρχεία που θα υλοποιήσετε αλλά και αυτά που σας έχουν δοθεί), δηλαδή ακριβώς ίδιο με το αρχείο **Monopoly.zip** που δίνεται, φυσικά με τον κώδικα υλοποιημένο.

Προθεσμία υποβολής

Το παραδοτέο πρέπει να παραδοθεί μέχρι τις **23:59 της Δευτέρας 13 Μαΐου**. Καμία παρέκκλιση δε θα γίνει από την παραπάνω προθεσμία.