

ΜΑΘΗΜΑ 22

ΚΛΑΣΕΙΣ: ΠΙΝΑΚΕΣ ΔΟΜΕΣ ΚΑΙ ΑΝΤΙΚΕΙΜΕΝΑ

Γιώργος Διάκος - Full Stack Developer



5. Κλάσεις που περιέχουν αντικείμενα

1. Θέματα μοντελοποίησης

- Είναι πάρα πολύ συχνό να έχουμε μία κλάση που περιέχει αντικείμενα άλλων κλάσεων.
- Όταν μοντελοποιούμε προβλήματα του πραγματικού κόσμου, αυτό θα είναι ιδιαίτερα συχνό.
- Παράδειγμα:
 - Μοντελοποιούμε μία τριόρωφη πολυκατοικία.
 - Η πολυκατοικία έχει 3 ορόφους
 - Κάθε όροφος έχει 2 διαμερίσματα
 - Κάθε διαμέρισμα έχει ένα πλήθος από ανθρώπους που είναι μέσα σε αυτό.
 - Θα κατασκευάσουμε ένα «χαφιε» που να μας λέει πόσα άτομα είναι στην πολυκατοικία.

- Δίνουμε έμφαση στη σχέση «έχει»
 - Στην μοντελοποίηση αυτό θα σχετίζεται με μία κλάση που έχει αντικείμενα.
- π.χ.:
 - Ένας σκύλος έχει 4 πόδια
 - Ένα αυτοκίνητο έχει τιμόνι, 4 ρόδες, μηχανή
 - κ.λπ.

5. Κλάσεις που περιέχουν αντικείμενα

2. Παράδειγμα

- Ξεκινάμε με το διαμέρισμα (flat):

```
class flat {  
public:  
    flat();  
    ~flat();  
    void set_people(int in_people);  
    int get_people() const;  
private:  
    int people;  
};
```

```
flat::flat()  
{  
    people=0;  
    cout<<"Constructing flat..."<<endl;  
}  
flat::~~flat()  
{  
    cout<<"Destructing flat..."<<endl;  
}  
void flat::set_people(int in_people)  
{  
    people = in_people;  
}  
int flat::get_people() const  
{  
    return people;  
}
```

5. Κλάσεις που περιέχουν αντικείμενα

2. Παράδειγμα

- Συνεχίζουμε με τον όροφο που περιέχει 2 διαμερίσματα:

```
class storey {  
public:  
    storey();  
    ~storey();  
    void set_people(int in_flat, int in_people);  
    int get_people() const;  
private:  
    flat flats[2];  
};
```

```
storey::storey()  
{  
    cout<<"Constructing storey..."<<endl;  
}  
storey::~storey()  
{  
    cout<<"Destructing storey..."<<endl;  
}  
void storey::set_people(int in_flat, int in_people)  
{  
    flats[in_flat].set_people(in_people);  
}  
int storey::get_people() const  
{  
    return flats[0].get_people()+flats[1].get_people();  
}
```

5. Κλάσεις που περιέχουν αντικείμενα

2. Παράδειγμα

- Και με την πολυκατοικία (building) που περιέχει 2 όροφους:

```
class building {  
    public:  
        building();  
        ~building();  
        void set_people(int in_floor, int in_flat, int  
in_people);  
        int get_people() const;  
    private:  
        storey storeys[2];  
};
```

```
building::building()  
{  
    cout<<"Constructing building..."<<endl;  
}  
building::~~building()  
{  
    cout<<"Destructing building..."<<endl;  
}  
void building::set_people(int in_storey, int in_flat,  
int in_people)  
{  
    storeys[in_storey].set_people(in_flat, in_people);  
}  
int building::get_people() const  
{  
    return  
storeys[0].get_people()+storeys[1].get_people();  
}
```

5. Κλάσεις που περιέχουν αντικείμενα

3. Σειρά εκτέλεσης constructors - destructors

- Η σειρά εκτέλεσης των κατασκευαστών είναι «από μέσα προς τα έξω»
 - Πρώτα το αντικείμενο που περιέχεται,
 - μετά το αντικείμενο που το περιέχει κ.ο.κ.
- Η σειρά εκτέλεσης των καταστροφών είναι αντίστροφη της σειράς των κατασκευαστών.
- Βλέπουμε και τη σειρά εκτέλεσης στο παράδειγμά μας:

```
int main()
{
    building b;
    b.set_people(0,0,3);
    b.set_people(0,1,2);
    b.set_people(1,0,2);
    b.set_people(1,1,1);
    cout<<endl;
    cout<<"People in building: "<<b.get_people();
    cout<<endl<<endl;
    return 0;
}
```

Το πρόγραμμα είναι CPP5.class_with_objects.cpp

```
Constructing flat...
Constructing flat...
Constructing storey...
Constructing flat...
Constructing flat...
Constructing storey...
Constructing building...

People in building: 8

Destructing building...
Destructing storey...
Destructing flat...
Destructing flat...
Destructing storey...
Destructing flat...
Destructing flat...
```

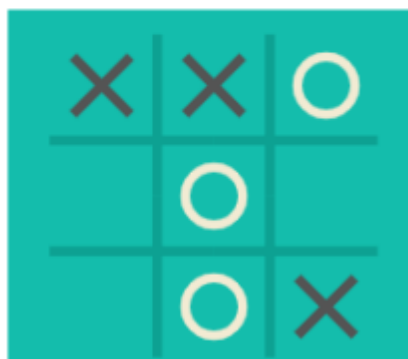
5. Κλάσεις που περιέχουν αντικείμενα

4. Γενικά περί μοντελοποίησης

- Με το τελευταίο παράδειγμα, βλέπουμε ότι με τα προγραμματιστικά εργαλεία που έχουμε μάθει ήδη, μπορούμε να μοντελοποιήσουμε περίπλοκες καταστάσεις του πραγματικού κόσμου.
 - Η μοντελοποίηση και η οργάνωση των δεδομένων μας, είναι πολύ σημαντικά θέματα σε μεγάλα προγράμματα.
- Πρέπει όμως να διαχωρίσουμε στο μυαλό μας ότι:
 - Μία γλώσσα προγραμματισμού είναι ένα εργαλείο σύνταξης (κάτι σαν την γραμματική και το συντακτικό μιας πραγματικής γλώσσας)
 - Έτσι όταν μαθαίνουμε μια γλώσσα εστιάζουμε στη γλώσσα και στα τεχνικά χαρακτηριστικά της
 - Το τι πραγματικά θα κάνουμε με μια γλώσσα προγραμματισμού είναι αντικείμενο της ανάλυσης συστημάτων
 - Π.χ. πως θα οργανώσουμε τα δεδομένα μας, τη λειτουργικότητα, πως θα αλληλεπιδρούν μεταξύ τους τα στοιχεία κ.λπ.
 - και υπάρχουν ειδικές γλώσσες περιγραφής προδιαγραφών, όπως π.χ. η UML.
- Στα μαθήματα αυτά, θα βλέπουμε θέματα μοντελοποίησης στις ασκήσεις.
 - Αλλά κάποιος θα πρέπει να μελετήσει εκτός των γλωσσών προγραμματισμού, και τις τεχνικές ανάλυσης, ώστε να μπορεί να κατασκευάσει ολοκληρωμένα προγράμματα.

Άσκηση 1: Παιχνίδι: Tic Tac Toe (γνωστό και ως XOX)

Θα υλοποιήσουμε μία κλάση που θα υλοποιεί το παιχνίδι tic-tac-toe (google it)



Συμβολική Σταθερά: N (να αρχικοποιείται σε 3)

Περιγραφή Κλάσης:

- Μέλος: Πίνακας 3x3 χαρακτήρων (να αρχικοποιούνται οι θέσεις σε '-')
- Μέθοδος `play(x,y,c)` παίρνει συντεταγμένες τετραγώνου και παίκτη (X ή O) και πραγματοποιεί την κίνηση (να γίνεται έλεγχος ότι μπορεί να γίνει η κίνηση)
- Μέθοδος `check_win()` επιστρέφει τον νικητή αν υπάρχει, και '-' αν δεν υπάρχει νικητής
- Μέθοδος `print()` εκτυπώνει το πλαίσιο

main:

- Επαναληπτικά:
 - Τυπώνει το πλαίσιο, τυπώνει ποιος παίζει και διαβάζει την επόμενη κίνηση.
 - Πραγματοποιεί την κίνηση και τυπώνει τον νικητή (αν υπάρχει)

Άσκηση 2: Παιχνίδι: Επέκταση του tic-tac-toe

Μεταφέρετε τον έλεγχο του ποιος παίκτης παίζει στην κλάση

Προσθέστε στην κλάση:

- Ορίστε ένα μέλος `player` (χαρακτήρας: X ή O)
- Μία μέθοδο `next_player` (αλλάζει το χαρακτήρα)
- Μία μέθοδο `get_player` (επιστρέφει τον παίκτη που παίζει)

Τροποποιήστε τη `main`:

- Όστε να παίρνει τον παίκτη που παίζει από την κλάση

Άσκηση 3: Stranger Things..

Ένα hive, μάθαμε πρόσφατα, ότι:

- Περιέχει το πολύ έναν demigorgon (σιχαμερό τέρας)
 - Έχει κάποια χαρακτηριστικά (ύψος = 5μ) (βάρος = 500 κιλά)
 - Έχει δείκτη υγείας (health) που αρχικοποιείται σε 10000 πόντους.
 - Επιτίθεται (attack) κάνοντας ζημιά (τυχαία από 300-500 πόντοι)
- Περιέχει πολλά demidogs (βρωμερά σκυλιά που μοιάζουν με φυτά)
 - Είναι τυχαία από 10 έως 50.
 - Κάθε ένα έχει ένα δείκτη υγείας (health) που αρχικοποιείται σε 100 πόντους.
 - Κάθε demidog επιτίθεται (attack) κάνοντας ζημιά (τυχαία από 10-30 πόντοι)

Το hive:

- Έχει κατασκευαστή, ώστε να δεσμεύει δυναμικά:
 - Τον demigorgon (εφόσον υπάρχει)
 - Τα demidogs τα οποία αποθηκεύονται σε έναν δυναμικό πίνακα (το πλήθος τους είναι όρισμα)
- Επιτίθεται, ενεργοποιώντας τις επιθέσεις των τεράτων που υπάρχουν (τυπώνει τις επιμέρους επιθέσεις και επιστρέφει το συνολικό πλήθος πόντων ζημιάς που προξένησε)

Η main:

- Κατασκευάζει ένα hive που έχει demigorgon.
- Ενεργοποιεί την επίθεση τυπώνοντας το συνολικό πλήθος των πόντων ζημιάς που προκάλεσε.

ΤΕΛΟΣ ΜΑΘΗΜΑΤΟΣ

ΚΛΑΣΕΙΣ: ΠΙΝΑΚΕΣ ΔΟΜΕΣ ΚΑΙ ΑΝΤΙΚΕΙΜΕΝΑ

Γιώργος Διάκος - Full Stack Developer

