

Σύνθεση (composition) και συνάθροιση (aggregation)

#7

Τμήμα Πληροφορικής και Τηλεπικοινωνιών (Άρτα)

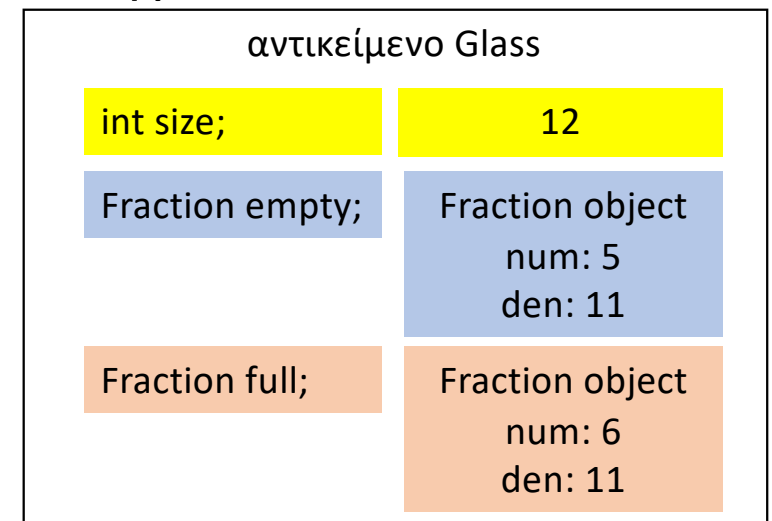
Πανεπιστήμιο Ιωαννίνων

Γκόγκος Χρήστος

Αντικείμενα ως μέλη δεδομένων

- Τα αντικείμενα είναι ένας συνδυασμός από μέλη δεδομένων, μέλη συναρτήσεων και μιας διεπαφής (interface).
- Μέλη δεδομένων μιας κλάσης μπορούν να είναι και αντικείμενα **(αντικείμενα μέσα σε αντικείμενα)**. Για παράδειγμα:

```
class Glass {  
    int size;  
    Fraction empty;  
    Fraction full;  
};
```



Σύνθεση

- Η συσχέτιση «αντικείμενο μέσα σε αντικείμενο» ονομάζεται σύνθεση (composition).
 - Μπορεί να υλοποιηθεί δηλώνοντας ένα αντικείμενο ως μέλος δεδομένων της κλάσης.
 - Η σύνθεση συχνά αναφέρεται ως συσχέτιση 'has-a'.
 - Για παράδειγμα:
 - **Glass 'has-a' Fraction** (το αντικείμενο Fraction είναι μέλος δεδομένων της κλάσης Glass).
 - **Car 'has-a(n)' Engine** (το αντικείμενο Engine είναι μέλος δεδομένων της κλάσης Car).
 - **Deck 'has-a' collection of Cards.** (το αντικείμενο «συλλογή 52 αντικειμένων Card (τραπουλόχαρτο)» είναι μέλος δεδομένων της κλάσης Deck (τράπουλα).
 - Η σύνθεση επιτρέπει στον κώδικα να είναι περισσότερο τμηματικός.
 - Μπορούμε να δημιουργούμε μικρότερες κλάσεις και να τις συνδυάζουμε έτσι ώστε να πετύχουμε την απαιτούμενη λειτουργικότητα.
 - Δείτε το παράδειγμα PokerHand
 - Αν δηλωθεί ένα δείκτης προς ένα ένα αντικείμενο ή μια αναφορά αντικειμένου ως μέλος δεδομένων μιας κλάσης τότε πρόκειται για συσχέτιση τύπου συνάθροισης (aggregation) που είναι μια ασθενέστερη μορφή συσχέτισης σε σχέση με τη σύνθεση.

https://github.com/chgogos/oop/blob/master/various/COP3330/lect7/poker_hand.cpp

Κατασκευαστές

- Όταν δημιουργείται ένα αντικείμενο, εκτελείται ο κατασκευαστής του, και επιπλέον καλείται ο κατασκευαστής κάθε ενός αντικειμένου που υπάρχει ως μέλος δεδομένων σε αυτό.
- Η τυπική συμπεριφορά είναι ότι καλείται ο προκαθορισμένος κατασκευαστής για το μέλος δεδομένων.
 - Ποιος κατασκευαστής θα κληθεί νωρίτερα;
 - Πρώτα καλείται ο κατασκευαστής του αντικειμένου που περιέχεται ως μέλος δεδομένων στην κλάση και στη συνέχεια ο κατασκευαστής της κλάσης.

```
#include <iostream>
using namespace std;

class small_class {
public:
    small_class();
private:
    int data;
};
small_class::small_class() {
    cout << "small_class default constructor called"
    << endl;
}

class large_class {
public:
    large_class();
private:
    small_class sc;
};
large_class::large_class() {
    cout << "large_class constructor called" << endl;
}

int main(){
    large_class obj;
}
```

```
small_class default constructor called
large_class constructor called
```

<https://github.com/chgogos/oop/blob/master/various/COP3330/lect7/sample1.cpp>

Κατασκευαστές

- Πως αρχικοποιείται ένα αντικείμενο μέσα σε ένα αντικείμενο;
- Τι θα πρέπει να γίνει έτσι ώστε να χρησιμοποιηθεί ένας κατασκευαστής (εκτός από τον προκαθορισμένο) για ένα μέλος δεδομένων;
 - Μπορεί να χρησιμοποιηθεί λίστα αρχικοποίησης:
 - Η λύση αυτή έχει περιορισμούς. Μπορεί να χρειαστεί να καλέσουμε τον κατασκευαστή μέσα στον κατασκευαστή της large κλάσης.

<https://github.com/chgogos/oop/blob/master/various/COP3330/lect7/sample2.cpp>

<https://github.com/chgogos/oop/blob/master/various/COP3330/lect7/sample3.cpp>

```
#include <iostream>
using namespace std;
class small_class {
public:
    small_class();
    small_class(int);
private:
    int data;
};
small_class::small_class() {
    cout << "small_class default constructor called" <<
endl;
}
small_class::small_class(int d) {
    data = d;
    cout << "small_class parameter constructor called" <<
endl;
}
class large_class {
public:
    large_class();
    large_class(int);
private:
    small_class sc;
};
large_class::large_class() {
    cout << "large_class default constructor called" <<
endl;
}
large_class::large_class(int d) : sc(d) {
    cout << "large_class parameter constructor called" <<
endl;
}
int main() {
    large_class obj(1);
}
```

small_class parameter constructor called
large_class parameter constructor called

Τελεστής τελεία (dot operator)

- Αν ένα αντικείμενο που είναι μέλος δεδομένων ενός άλλου αντικειμένου έχει δημόσια μέλη (δεδομένα ή συναρτήσεις), τότε μπορούμε να τα προσπελάσουμε μέσω του τελεστή τελείας.

<https://github.com/chgogos/oop/blob/master/various/COP3330/lect7/sample4.cpp>

```
#include <iostream>
using namespace std;
class Circle {
public:
    Circle(int);
    float radius;
};
Circle::Circle(int r) { radius = r; }

class Square {
public:
    Square(int);
    float getlength();
private:
    float length;
};
Square::Square(int l) { length = l; }
float Square::getlength() {return length;}

class ShapeHolder {
public:
    ShapeHolder();
    Circle c1; Square s1;
};
ShapeHolder::ShapeHolder() : c1(1), s1(2) {}

int main() {
    ShapeHolder s1;
    cout << "SH Circle Radius: " << s1.c1.radius << endl;
    cout << "SH Square Length: " << s1.s1.getlength() << endl;
    return 0;
}
```

SH Circle Radius: 1
SH Square Length: 2

Άσκηση #1 (σύνθεση): Εκφώνηση

- Να γράψετε ένα πρόγραμμα που να αναπαριστά τη συσχέτιση σύνθεσης (composition) μέσα από την αλληλεπίδραση μεταξύ ενός υπολογιστή και του επεξεργαστή του.
- Δημιουργήστε μια κλάση CPU με ιδιότητες όπως model και speed, καθώς και μια μέθοδο process() που να εμφανίζει μήνυμα επεξεργασίας δεδομένων.
- Δημιουργήστε μια κλάση Computer που να περιέχει ένα αντικείμενο τύπου CPU (σύνθεση) και να διαθέτει ιδιότητες όπως brand και ram, καθώς και μια μέθοδο start() που να εμφανίζει μήνυμα εκκίνησης του υπολογιστή και να καλεί τη μέθοδο process() της CPU.
- Δημιουργήστε αντικείμενα των δύο κλάσεων και δείξτε την αλληλεπίδραση τους μέσω της μεθόδου start().

Άσκηση #1 (σύνθεση): Λύση

```
class CPU {
private:
    string model;
    double speed;

public:
    CPU(string m, double s) : model(m), speed(s) {}

    void process() const {
        cout << "CPU " << model << " running at "
              << speed << " GHz is processing data...\n";
    }
};
```

```
int main() {
    Computer myPC("Lenovo", 16, "Intel Core i7", 3.5);
    myPC.start();

    return 0;
}
```

```
class Computer {
private:
    string brand;
    int ram;
    CPU cpu;

public:
    Computer(string b, int r, string cpuModel, double cpuSpeed)
        : brand(b), ram(r), cpu(cpuModel, cpuSpeed) {}

    void start() const {
        cout << "Starting computer " << brand
              << " with " << ram << "GB RAM..." << endl;
        cpu.process();
    }
};
```

Starting computer Lenovo with 16GB RAM...
CPU Intel Core i7 running at 3.5 GHz is processing data...

<https://github.com/chgogos/oop/blob/master/various/COP3330/lect7/exercise1.cpp>

Aggregation (συνάθροιση)

- Η συσχέτιση συνάθροισης σημαίνει ότι το αντικείμενο A που περιέχει ένα άλλο αντικείμενο B δεν κατέχει το άλλο αντικείμενο, το οποίο, αντικείμενο B, μπορεί να συνεχίζει να υπάρχει ανεξάρτητα από το αντικείμενο A
 - Στην περίπτωση της συνάθροισης χρησιμοποιείται για το αντικείμενο B μια αναφορά (ή ένας δείκτης) κατά τη δήλωσή του ως μέλος δεδομένων της A

Άσκηση #2 (συνάθροιση)

- Μετατρέψτε στην άσκηση 1 τη συσχέτιση σύνθεσης μεταξύ CPU και Computer σε συσχέτιση συνάθροισης, δείχνοντας τη διαφορά σχετικά με τη διάρκεια ζωής του αντικειμένου που περιέχεται (CPU) στο αντικείμενο που το περιέχει.
- Εντοπίστε πως απεικονίζεται σχηματικά η συσχέτιση στην περίπτωση της σύνθεσης και πως στην περίπτωση της συνάθροισης.
- Σχεδιάστε τη συσχέτιση (διάγραμμα κλάσεων) και για τις δύο περιπτώσεις (σύνθεση, συνάθροιση) με το mermaid live <https://mermaid.live/> ή με κάποιο άλλο λογισμικό σχεδίασης.

Άσκηση #2 (συνάθροιση): Λύση (1/2)

```
class CPU {
private:
    string model;
    double speed;

public:
    CPU(string m, double s) : model(m), speed(s) {}

    void process() const {
        cout << "CPU " << model << " running at "
              << speed << " GHz is processing data...\n";
    }
};
```

```
int main() {
    CPU cpu1("Intel Core i7", 3.5);
    {
        Computer myPC("Lenovo", 16, cpu1);
        myPC.start();
    } // Το myPC καταστρέφεται εδώ, αλλά η cpu1 παραμένει

    cout << "Computer object destroyed, but the CPU still exists and works:" << endl;
    cpu1.process();
    return 0;
}
```

```
class Computer {
private:
    string brand;
    int ram;
    CPU& cpu;

public:
    Computer(string b, int r, CPU& c) : brand(b), ram(r), cpu(c) {}

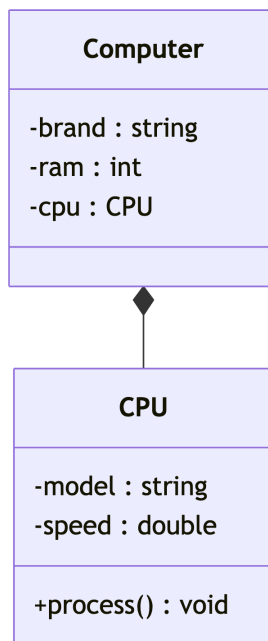
    void start() const {
        cout << "Starting computer " << brand
              << " with " << ram << "GB RAM..." << endl;
        cpu.process();
    }
};
```

Starting computer Lenovo with 16GB RAM...
CPU Intel Core i7 running at 3.5 GHz is processing data...
Computer object destroyed, but the CPU still exists and works:
CPU Intel Core i7 running at 3.5 GHz is processing data...

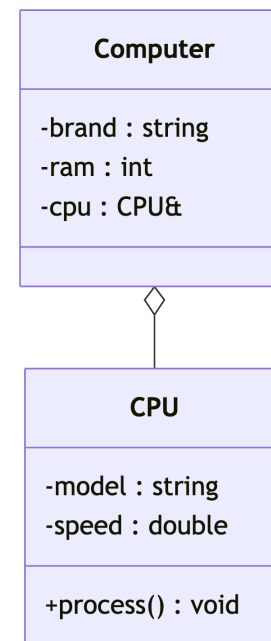
<https://github.com/chgogos/oop/blob/master/various/COP3330/lect7/exercise2.cpp>

Άσκηση #2 (συνάθροιση): Λύση (2/2)

Σύνθεση



Συνάθροιση



Ερωτήσεις σύνοψης

- Τι είναι η σύνθεση (composition);
- Πότε αρχικοποιείται ένα αντικείμενο μέλος δεδομένων στην περίπτωση της σύνθεσης αντικειμένων;
- Ποιος κατασκευαστής χρησιμοποιείται για να αρχικοποιήσει ένα αντικείμενο μέλος δεδομένων αν δεν επέμβουμε;
- Πως μπορώ να αρχικοποιήσω ένα αντικείμενο μέλος δεδομένων με ένα συγκεκριμένο κατασκευαστή;
- Τι μπορεί να σημαίνει η ακόλουθη κλήση;
`obj1.dm.fun();`
- Πως διαφέρει η σύνθεση από τη συνάθροιση;

Απαντήσεις στις ερωτήσεις σύνοψης

- Τι είναι η σύνθεση (composition);
 - Η σύνθεση είναι ένας τύπος συσχέτισης αντικειμένων, όπου ένα αντικείμενο περιέχει και είναι υπεύθυνο για τη διάρκεια ζωής ενός άλλου αντικειμένου. Πρόκειται για έναν ισχυρό τύπο συσχέτισης όπου το αντικείμενο-μέρος δεν μπορεί να υπάρξει ανεξάρτητα από το αντικείμενο που το περιέχει.
- Πότε αρχικοποιείται ένα αντικείμενο μέλος δεδομένων στην περίπτωση της σύνθεσης αντικειμένων;
 - Ένα αντικείμενο μέλος δεδομένων αρχικοποιείται πριν εκτελεστεί το σώμα του κατασκευαστή της κλάσης.
- Ποιος κατασκευαστής χρησιμοποιείται για να αρχικοποιήσει ένα αντικείμενο μέλος δεδομένων αν δεν επέμβουμε;
 - Αν δεν επέμβουμε, χρησιμοποιείται ο προεπιλεγμένος (default) κατασκευαστής του μέλους.
- Πως μπορώ να αρχικοποιήσω ένα αντικείμενο μέλος δεδομένων με ένα συγκεκριμένο κατασκευαστή;
 - Με κλήση του επιθυμητού κατασκευαστή στη λίστα αρχικοποίησης του κατασκευαστή της κλάσης που περιέχει το αντικείμενο μέλος δεδομένων.
- Τι μπορεί να σημαίνει η ακόλουθη κλήση;
`obj1.dm.fun();`
 - Σημαίνει ότι καλείται η μέθοδος `fun()` του αντικειμένου μέλους `dm` του αντικειμένου `obj1`.
- Πως διαφέρει η σύνθεση από τη συνάθροιση;
 - Στην περίπτωση της σύνθεσης υπάρχει ισχυρή εξάρτηση και κοινός κύκλος ζωής για το αντικείμενο μέλος σε σχέση με το αντικείμενο σύνολο, ενώ στη συνάθροιση υπάρχει αντίστοιχα ασθενής εξάρτηση και ανεξάρτητος κύκλος ζωής.

Αναφορές

- <http://www.cs.fsu.edu/~xyuan/cop3330/>