Σύνθεση (composition)

#7

Τμήμα Πληροφορικής και Τηλεπικοινωνιών (Άρτα)

Πανεπιστήμιο Ιωαννίνων

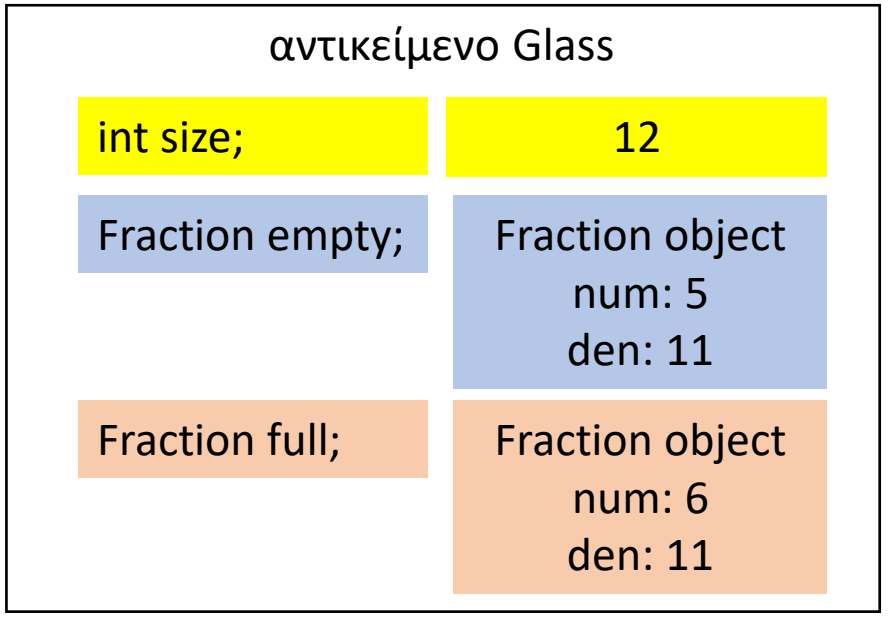
Γκόγκος Χρήστος

V 1.0

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 σχέση «αντικείμενο μέσα σε αντικείμενο» ονομάζεται σύνθεση.
  - Μπορεί να υλοποιηθεί δηλώνοντας ένα αντικείμενο ή ένα δείκτη ή μια αναφορά ως μέλος δεδομένων της κλάσης.
  - Συχνά αναφέρεται ως σχέση ‘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 συνθετότερη λειτουργικότητα.
    - Δείτε το παράδειγμα PokerHand

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
```

sample2.cpp

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
```

Ερωτήσεις σύνοψης

- Τι είναι η σύνθεση (composition);
- Πότε αρχικοποιείται ένα αντικείμενο μέλος δεδομένων;
- 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();

Αναφορές

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