Κληρονομικότητα

#14

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Ιωαννίνων (Άρτα)
Γκόγκος Χρήστος

Σχέσεις μεταξύ αντικειμένων

- Δύο βασικές σχέσεις μεταξύ αντικειμένων είναι η σχέση HAS-A και η σχέση IS-A.
- Παραδείγματα σχέσης HAS-A
  - PlayList has-a Song
  - PokerHand has-a Card
- Σχέση IS-A
  - Ένας ComputerScienceStudent είναι ένας (IS-A) UniversityStudent.
  - Ένας UniversityStudent είναι ένας (IS-A) Student.

Εισαγωγή

- Η «κληρονομικότητα» (inheritance) στη C++ επιτρέπει στους προγραμματιστές να ορίσουν IS-A σχέσεις:
  - Μια κλάση μπορεί να οριστεί ως παραγόμενη από μια άλλη κλάση (που ονομάζεται βασική κλάση).
  - Η παραγόμενη κλάση γίνεται ένα είδος/εξειδίκευση της βασικής κλάσης.
    - Ένα αντικείμενο της παραγόμενης κλάσης είναι (IS-A) αντικείμενο της βασικής κλάσης.
  - Εναλλακτικά, μπορεί να θεωρηθεί ότι η παραγόμενη κλάση είναι ότι είναι η βασική κλάση και (πιθανά) περισσότερα.
  - Αυτό επιτρέπει, σε ορισμένες περιπτώσεις και εφόσον απαιτείται, τη χρήση της παραγόμενης κλάσης σαν να ήταν η βασική κλάση.

Κληρονομικότητα, βασική και παραγόμενη κλάση

- Η σχέση IS-A υλοποιείται μέσω της κληρονομικότητας χρησιμοποιώντας τον ακόλουθο τρόπο δήλωσης, που σημαίνει ότι τα αντικείμενα της παραγόμενης κλάσης είναι αντικείμενα της βασικής κλάσης (μπορούν να χρησιμοποιούν όλες τις συναρτήσεις της διεπαφής της βασικής κλάσης)
  - `class derivedClassName: public baseClassName`

```
class Mammal {
public:
    void PrintInfo() const;
};
class Cow : public Mammal {
public:
    void Sound() const;
};
...
int main() {
    Cow C;
    C.PrintInfo();
}
```

https://github.com/chgogos/oop/blob/master/various/COP3330/lect14/sample1.cpp

Επίπεδα προστασίας

- Μια παραγόμενη κλάση μπορεί να προσπελάσει όλα τα δημόσια μέλη της βασικής κλάσης, αλλά δεν μπορεί να προσπελάσει τα ιδιωτικά μέλη της βασικής κλάσης.
  - Αν επιθυμούμε να επιτρέψουμε σε μερικά μέλη της βασικής κλάσης να προσπελαύνονται από την παραγόμενη κλάση, αλλά να μην προσπελαύνονται από άλλες κλάσεις τότε χρησιμοποιούμε το επίπεδο προστασίας **`protected`**.
- Σύνοψη επιπέδων προστασίας:
  - **`public`**: μέλη που μπορούν να προσπελαστούν με το όνομά τους στο εσωτερικό οποιασδήποτε συνάρτησης.
  - **`private`**: Μέλη που μπορούν να προσπελαστούν στο εσωτερικό συναρτήσεων μελών και από συναρτήσεις που έχουν δηλωθεί ως φίλες συναρτήσεις (friend) της κλάσης.
  - **`protected`**: Μέλη που μπορούν να προσπελαστούν στο εσωτερικό συναρτήσεων μελών και από συναρτήσεις που έχουν δηλωθεί ως φίλες συναρτήσεις (friend) της κλάσης και επίσης μπορούν να προσπελαστούν στο εσωτερικό των συναρτήσεων μελών της παραγόμενης κλάσης και στις friend συναρτήσεις της παραγόμενης κλάσης.

https://github.com/chgogos/oop/blob/master/various/COP3330/lect14/sample2.cpp

Κατασκευαστές/καταστροφείς

- Καθώς ένα στιγμιότυπο μιας παραγόμενης κλάσης είναι επίσης και στιγμιότυπο της βασικής κλάσης, τόσο ο κατασκευαστής της βασικής κλάσης όσο και ο κατασκευαστής της παραγόμενης κλάσης θα πρέπει να εκτελείται όταν ένα αντικείμενο της παραγόμενης κλάσης δημιουργείται.
- Οι κατασκευαστές εκτελούνται στη σειρά από τον πλέον γενικό προς τον πλέον ειδικό.
- Οι καταστροφείς εκτελούνται στη σειρά από τον πλέον ειδικό προς τον πλέον γενικό.
- Αν οι κατασκευαστές έχουν παραμέτρους, τότε η κλήση τους γίνεται μέσω της λίστας αρχικοποίησης.

https://github.com/chgogos/oop/blob/master/various/COP3330/lect14/sample3.cpp

https://github.com/chgogos/oop/blob/master/various/COP3330/lect14/sample4.cpp

Άσκηση #1: Απλό παράδειγμα κληρονομικότητας

- Να υλοποιηθεί η βασική κλάση Device, η οποία θα αποθηκεύει τον σειριακό αριθμό μιας συσκευής και θα εμφανίζει μήνυμα κάθε φορά που δημιουργείται ή καταστρέφεται ένα αντικείμενό της εκτυπώνοντας τις τιμές των μελών δεδομένων.
- Στη συνέχεια να υλοποιηθεί μια παράγωγη κλάση της Device, η SmartDevice που θα εμφανίζει επίσης κατάλληλα μηνύματα κατά την κατασκευή και καταστροφή των αντικειμένων της (εκτυπώνοντας τις τιμές των μελών δεδομένων) και που θα διαθέτει έναν μετρητή χρήσης που θα αυξάνεται κάθε φορά που θα καλείται μια συνάρτηση μέλος με όνομα increment.
- Στη main() να δημιουργηθούν αντικείμενα από τις δύο κλάσεις, και να κληθεί η increment()

Άσκηση #1: Λύση

```
Device constructed. Serial = 100
Device constructed. Serial = 200
SmartDevice constructed. Serial = 200, Usage = 0
increment() called. Serial = 200, Usage = 1
SmartDevice destroyed. Serial = 200, Usage = 1
Device destroyed. Serial = 200
Device destroyed. Serial = 100
```

```cpp
#include <iostream>
using namespace std;
class Device {
 protected:
  int serial;
 public:
  Device(int s) : serial(s) { cout << "Device constructed. Serial = " << serial << endl; }
  ~Device() { cout << "Device destroyed. Serial = " << serial << endl; }
};

class SmartDevice : public Device {
 private:
  int usage;
 public:
  SmartDevice(int s) : Device(s), usage(0) {
    cout << "SmartDevice constructed. Serial = " << serial << ", Usage = " << usage << endl;
  }
  ~SmartDevice() { cout << "SmartDevice destroyed. Serial = " << serial << ", Usage = " << usage << endl; }
  void increment() {
    usage++;
    cout << "increment() called. Serial = " << serial << ", Usage = " << usage << endl;
  }
};

int main() {
  Device d1(100);
  SmartDevice sd1(200);
  sd1.increment();
  return 0;
}
```

https://github.com/chgogos/oop/blob/master/various/COP3330/lect14/exercise1.cpp

Παράκαμψη (override) συναρτήσεων

- Μια από τις χρησιμότερες δυνατότητες της κληρονομικότητας είναι ότι επιτρέπει να εξειδικευτεί η συμπεριφορά των παραγόμενων αντικειμένων.
- Μια παραγόμενη κλάση μπορεί να δηλώσει τη δική της έκδοση μιας συνάρτησης που υπάρχει ήδη στη βασική κλάση από την οποία κληρονομεί.
  - Η συνάρτηση της παραγόμενης κλάσης θα αντικαταστήσει την έκδοση της συνάρτησης της βασικής κλάσης.
- Μπορούμε να έχουμε πρόσβαση και στη συνάρτηση της βασικής κλάσης με ρητή κλήση της.
  - Αυτό συμβαίνει διότι, βάσει ορισμού, η παραγόμενη κλάση θα πρέπει να είναι οτιδήποτε είναι η βασική κλάση και πιθανά περισσότερα.

https://github.com/chgogos/oop/blob/master/various/COP3330/lect14/sample5.cpp

https://github.com/chgogos/oop/blob/master/various/COP3330/lect14/sample6.cpp

Άσκηση #2: Απλή περίπτωση override

- Να υλοποιηθούν δύο κλάσεις:
  - Η βασική κλάση Message, με το προστατευμένο μέλος δεδομένων message και με τη δημόσια συνάρτηση μέλος show() η οποία εμφανίζει ένα γενικό μήνυμα της μορφής "General message: <κείμενο μηνύματος>".
  - Η παράγωγη κλάση της Message, WarningMessage η οποία δηλώνει μια συνάρτηση show() με την ίδια υπογραφή με τη show() της βασικής κλάσης, αλλά η οποία εμφανίζει μήνυμα της μορφής "Warning message: <κείμενο μηνύματος>".
- Στη main() να δημιουργηθούν αντικείμενα και από τις δύο κλάσεις και να κληθούν οι αντίστοιχες συναρτήσεις show().

Άσκηση #2: Λύση

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

class Message {
 protected:
  string message;

 public:
  Message(const string& msg) : message(msg) {}

  void show() { cout << "General message: " << message << endl; }
};

class WarningMessage : public Message {
 public:
  WarningMessage(const string& msg) : Message(msg) {}

  void show() { cout << "Warning message: " << message << endl; }
};

int main() {
  Message m("System running normally.");
  WarningMessage wm("Low battery level.");
  m.show();
  wm.show();
  return 0;
}
```

```
General message: System running normally.
Warning message: Low battery level.
```

https://github.com/chgogos/oop/blob/master/various/COP3330/lect14/exercise2.cpp

Ερωτήσεις σύνοψης

- Πως δηλώνεται ότι μια νέα κλάση A έχει ως βασική κλάση την κλάση B;
- Ποια είναι η διαφορά των `private` και `protected` επιπέδων προστασίας;
- Πως μπορεί μια παραγόμενη κλάση να κάνει παράκαμψη (override) μιας συνάρτησης που ορίζεται στη βασική κλάση;
- Ποια είναι η σειρά με την οποία καλούνται κατασκευαστές και καταστροφείς στην περίπτωση αντικειμένων παραγόμενης κλάσης;

Απαντήσεις στις ερωτήσεις σύνοψης

- Πως δηλώνεται ότι μια νέα κλάση A έχει ως βασική κλάση την κλάση B;
  - Δηλώνεται ως: class A : public B { ... };
- Ποια είναι η διαφορά των `private` και `protected` επιπέδων προστασίας;
  - Το private επίπεδο προστασίας επιτρέπει προσπέλαση από μεθόδους της ίδιας κλάσης.
  - Το protected επίπεδο προστασίας επιτρέπει προσπέλαση από μεθόδους της ίδιας κλάσης και από παράγωγες κλάσεις.
- Πως μπορεί μια παραγόμενη κλάση να κάνει παράκαμψη (override) μιας συνάρτησης που ορίζεται στη βασική κλάση;
  - Στη παράγωγη κλάση ορίζεται μια συνάρτηση με την ίδια υπογραφή όπως η συνάρτηση μέλος της βασικής κλάσης για την οποία γίνεται παράκαμψη (υπάρχει και η έννοια του virtual για την οποία θα μιλήσουμε στη συνέχεια).
- Ποια είναι η σειρά με την οποία καλούνται κατασκευαστές και καταστροφείς στην περίπτωση αντικειμένων παραγόμενης κλάσης;
  - Για τους κατασκευαστές η σειρά είναι πρώτα ο κατασκευαστής της βασικής κλάσης και μετά ο κατασκευαστής της παραγόμενης κλάσης.
  - Για τους καταστροφείς πρώτα ο καταστροφέας της παραγόμενης κλάση και μετά ο καταστροφέας της βασικής κλάσης.

Αναφορές

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