



Ελληνική Δημοκρατία
Τεχνολογικό Εκπαιδευτικό
Ίδρυμα Ηπείρου

Αντικειμενοστραφής Προγραμματισμός

Ενότητα 7 : Προχωρημένα θέματα (2/2)

Ιωάννης Τσούλος



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Τμήμα Μηχανικών Πληροφορικής Τ.Ε

Αντικειμενοστραφής Προγραμματισμός

Ενότητα 7 : Προχωρημένα θέματα (2/2)

Ιωάννης Τσούλος

Επίκουρος Καθηγητής

Άρτα, 2015



Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.





Χρηματοδότηση

- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Δια Βίου Μάθηση**» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Ηπείρου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης





Polymorphism και Virtual Methods

- Τί γίνεται όμως αν θέλουμε να αλλάξουμε τη λειτουργία μιας μεθόδου στην νέα κλάση που υπάρχει και στην παλιά;
- Στο προηγούμενο μάθημα, η νέες κλάσεις ορίζουν μια μέθοδο την `getInfo()`, η οποία επιστρέφει πληροφορίες για τον υπάλληλο ή τον δάσκαλο αντίστοιχα.
- Όπως είναι δηλωμένη, η κάθε κλάση προσφέρει τη δική της εκδοχή αλλά η αρχική `Person`, δεν έχει ορισμένη μια τέτοια μέθοδο `getInfo()`.



Polymorphism και Virtual Methods

- Αν η getInfo() οριστεί και για την κλάση Person, π.χ. ως εξής:

```
string Person::getInfo() {  
    return (getFirstname()+” “+getLastname()  
        +”is “+getAge()+” years old”);  
}
```

τότε δημιουργείται το εξής ερώτημα:

ποια εκδοχή της getInfo() θα καλείται για κάθε αντικείμενο;



Polymorphism και Virtual Methods

- Στη συγκεκριμένη περίπτωση θα είναι αδύνατη η μεταγλώττιση των κλάσεων Clerk και Teacher.
- Η C++ δεν επιτρέπει το override των μεθόδων χωρίς ειδική άδεια.
- Η άδεια αυτή δίνεται με τη λέξη **virtual**, δηλαδή η μέθοδος ορίζεται ως δυναμική, και ο ορισμός της θα μετατραπεί ως εξής:

```
virtual string getInfo() {  
    return (getFirstname()+” “+getLastname()  
        +”is “+getAge()+” years old”);  
}
```



Polymorphism και Virtual Methods

- Με την αλλαγή αυτή, θα καλείται κάθε φορά η `getInfo()` της κλάσης που ανήκει το αντικείμενο από το οποίο καλούμε την μέθοδο.
- Για παράδειγμα, ο ακόλουθος κώδικας:

```
cout << bilbo.getInfo() << endl;  
cout << sam.getInfo() << endl;  
cout << pippin.getInfo() << endl;
```

θα παράγει

```
Bilbo Baggins is 111 years old  
Samwise Gamgee works at Baggins Inc., at Bag End, Hobbiton, The Shire.  
Email: gardener@baggins.com  
Tel: +302103456789  
Peregrin Took teaches Philosophy at King's College, Hobbiton.  
Email: pippin@theshire.net  
Tel: +30210000001
```




Polymorphism και Virtual Methods

- Καλέσαμε την ίδια μέθοδο και για τα τρία αντικείμενα (`getInfo()`) η οποία όμως ορίζεται και στις τρεις κλάσεις.
- Η γονική κλάση `Person` ορίζει την `getInfo()` με έναν απλό τρόπο (“`Bilbo Baggins is 111 years old`”) και η ίδια θα χρησιμοποιούνταν στις κλάσεις `Clerk` και `Teacher` αν δεν ορίζονταν και εκεί.
- Αν δεν ορίζαμε την `getInfo()` στην κλάση `Clerk`, η `getInfo()` για το αντικείμενο `sam` θα επέστρεφε “`Samwise Gamgee is 33 year old`”.
- Επειδή όμως η κλάση `Clerk` περιέχει περισσότερες πληροφορίες που πρέπει να απεικονιστούν με την `getInfo()`, η τελευταία επαναορίστηκε, αποκτώντας νέα λειτουργία.



Polymorphism και Virtual Methods

- Η τεχνική αυτή καλείται **Method Overriding** και είναι το θεμελιώδες χαρακτηριστικό του πολυμορφισμού.
- Η κλάσεις στις οποίες ορίζουμε **virtual methods** λέγονται virtual (virtual classes).
- Ο πολυμορφισμός μας δίνει τη δυνατότητα δημιουργίας αφηρημένων εννοιών/κλάσεων/μεθόδων στις οποίες θα κολλάμε κάθε φορά τη σωστή υλοποίηση αναλόγως τις ανάγκες μας.



Polymorphism και Virtual Methods

- Το ακόλουθο παράδειγμα έχει σκοπό να δείξει πώς ακριβώς χρησιμοποιούμε τον πολυμορφισμό. Θεωρούμε τα αντικείμενα που έχουμε ορίσει πριν (`bilbo`, `sam`, `pippin`).

```
Person* who[3];  
who[0] = &bilbo;  
who[1] = &sam;  
who[2] = &pippin;  
for (int i=0; i < who.length; i++)  
    cout << who[i]->getInfo() << endl;
```

- Το παραπάνω παράδειγμα θα τυπώσει ό,τι και το προηγούμενο μόνο που τώρα χρησιμοποιήσαμε έναν πίνακα αντικειμένων `Person` για να οργανώσουμε την πληροφορία. Αλλά το αντικείμενο `sam` είναι τύπου `Clerk` και το `pippin` ανήκει στην κλάση `Teacher`. Πώς τα καταχωρήσαμε στον πίνακα `who`;



Polymorphism και Virtual Methods

- **Απάντηση:** Ένα αντικείμενο Clerk είναι ταυτόχρονα και αντικείμενο Person, όπως και ένα αντικείμενο Teacher είναι επίσης Person. Ένας Clerk δεν είναι όμως ταυτόχρονα Teacher.
- Έτσι μπορούμε να τα αντιμετωπίζουμε κατά βούληση ως Clerk και Teacher ή Person, αναλόγως τις ανάγκες μας.
- Ακόμη όμως και αν τα προσπελάσουμε ως Person, η getInfo() του καθενός θα είναι αυτή που ορίζει η κλάση του.
- Τί γίνεται όμως αν χρειάζεται να καλέσουμε και την γονική μέθοδο (δηλαδή αυτή που επαναορίζουμε στη νέα κλάση); Στην περίπτωση αυτή χρησιμοποιούμε την αναφορά στη γονική κλάση καλώντας τον δημιουργό της.



Polymorphism και Virtual Methods

- **Παράδειγμα:** Μετασχηματίζουμε το δημιουργό και τη μέθοδο getInfo() της κλάσης Teacher στα εξής:

```
Teacher(string fname, string lname, int age, string tel,
string email, string title, string school,
string schooladdress, string schooltel,
string coursename, string coursedescription)
{
    Person(fname, lname, age, tel, email);
    Title_ = title;
    School_ = school;
    SchoolAddress_ = schooladdress;
    SchoolTel_ = jobtel;
    CourseName_ = coursename;
    CourseDescription_ = coursedescription;
}
string getInfo() {
    return (Person.getInfo()
        + " and teaches "+CourseName_+" at "+School_
        +", "+SchoolAddress_+".\n"
        +"Email: "+getEmail()+"\n"
        +"Tel: "+SchoolTel_);
}
```



Polymorphism και Virtual Methods

- Στον δημιουργό αρκεί πλέον να αρχικοποιούμε (*initialize*) μόνο τα χαρακτηριστικά που είναι νέα στην κλάση χωρίς να κάνουμε διπλό κόπο που έχει ήδη γίνει στην γονική κλάση.
- Στην `getInfo()`, χρησιμοποιούμε την πληροφορία που επιστρέφεται από τη γονική κλάση και συμπληρώνουμε με τα νέα στοιχεία.

- **Παράδειγμα**

Αν αλλάζαμε το κείμενο που επιστρέφει η `getInfo()` της κλάσης `Person`, θα άλλαζε αυτόματα και το κείμενο που επέστρεφε η `getInfo()` της κλάσης `Teacher`, εφόσον χρησιμοποιούμε την αναφορά στη κλάση `Person`.



Polymorphism και Virtual Methods

- ΣΗΜΕΙΩΣΗ

- Έχουμε `method overriding` **μόνο όταν** η δήλωση της μεθόδου είναι η ίδια στη γονική και στην θυγατρική κλάση. Δηλαδή ίδιο όνομα, επιστρεφόμενα δεδομένα και παραμέτρους.
- Αν ένα από αυτά είναι διαφορετικά (εκτός από το όνομα φυσικά) τότε έχουμε υπερφόρτωση μεθόδου (`method overloading`).



Pure Virtual Classes

- Μερικές φορές μπορεί να θέλουμε να ορίσουμε απλώς μια γενική δομή κάποιων κλάσεων, χωρίς όμως να προσφέρουμε μια ακριβή υλοποίηση.
- Να ορίσουμε μια “αφηρημένη” κλάση, που να παρέχει απλώς ένα πλαίσιο που θα συμπληρώνουν οι θυγατρικές κλάσεις μέσω της τεχνικής `method overriding`.
- Αυτό σημαίνει ότι στον ορισμό της κλάσης απλώς θα δηλώνονται ορισμένες μέθοδοι ως “αφηρημένες” αλλά δεν θα παρέχεται κάποιος ορισμός γι' αυτές.



Pure Virtual Classes

- Στη C++ οι κλάσεις αυτές καλούνται **pure**.
- **Παράδειγμα αφηρημένης κλάσης** η κλάση του αυτοκινήτου.
 - Ο μηχανισμός που αλλάζει ταχύτητες σε κάθε αυτοκίνητο είναι διαφορετικός και διαφέρει ανάμεσα στους κατασκευαστές και στα διάφορα μοντέλα.
 - **Όμως** ο τρόπος αλληλεπίδρασης (δηλαδή μέσω του μοχλού ταχυτήτων) είναι καλά ορισμένος έστω και με κάποια περιθώρια παραλλαγών.



Pure Virtual Classes

- Τρόπος ορισμού κλάσης

```
class Car {  
    ...  
    // Αλλαγή της ταχύτητας. Επιστρέφει true αν η αλλαγή  
    ήταν επιτυχής  
    // ή false αν ήταν ανεπιτυχής (π.χ. από 5 σε όπισθεν).  
    virtual bool change_gear(int new_gear) = 0;  
    ...  
}  
class FerrariModena extends Car {  
    ...  
    bool change_gear(int new_gear) {  
        // Η συγκεκριμένη υλοποίηση βρίσκεται εδώ  
    }  
    ...  
}
```



Βιβλιογραφία

1. Εγχειρίδιο της C++, 2η Ελληνική έκδοση, Jesse Liberty, Γκιούρδας.
2. Μάθετε τη C++, 2η Ελληνική έκδοση, Jesse Liberty , Γκιούρδας.
3. Προγραμματισμός με τη γλώσσα C++ Μέρος Α, Αλεβίζος Θ., Έκδοση ΤΕΙ Καβάλας
4. C++ Αντικειμενοστραφής Προγραμματισμός Υπολογιστών Τομαράς Α., , Εκδόσεις Νέων Τεχνολογιών.
5. Ανακαλύψτε τη γλώσσα C, J. Purdum, Εκδόσεις Δίαυλος.
6. Εισαγωγή στο Συστηματικό Προγραμματισμό και στη γλώσσα C++, Σ. Μπαλτζής, εκδόσεις πανεπιστημίου Ιωαννίνων.
7. C++ From the beginning, Jan Skansholm, Addison Wesley.
8. The design and analysis of computer algorithms, A.V. AHO, J.E. HOPCROFT, J.D. ULLMANN, Addison Wesley 1974.
9. Structure and Interpretation of Computer Programs, H. ABELSON, G.J. SUSSMAN, J. SUSSMAN, MIT Press, Mc Graw Hill Book Company, 1985
10. The art of computer programming, D.E. KNUTH, Addison-Wesley.



Σημείωμα Αναφοράς

Copyright Τεχνολογικό Ίδρυμα Ηπείρου. Ιωάννης Τσούλος.

Αντικειμενοστραφής Προγραμματισμός.

Έκδοση: 1.0 Άρτα, 2015. Διαθέσιμο από τη δικτυακή
διεύθυνση:

<http://eclass.teiep.gr/courses/COMP113/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού-Μη Εμπορική Χρήση-Όχι Παράγωγα Έργα 4.0 Διεθνές [1] ή μεταγενέστερη. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, Διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.el>



Τέλος Ενότητας

Επεξεργασία: Ευάγγελος Καρβούνης
Άρτα, 2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Τέλος Ενότητας

Προχωρημένα θέματα (2/2)



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

