



Εργαστήριο 7

Σύνθεση - Κληρονομικότητα - Ειδίκευση

(Composition - Inheritance - Specialization)

1. Εισαγωγή

Ένα από τα χαρακτηριστικά του αντικειμενοστρεφούς προγραμματισμού είναι η ευκολία με την οποία μπορεί να επανα-χρησιμοποιηθούν κλάσεις για διαφορετικούς σκοπούς. Αυτό γίνεται μέσω της σύνθεσης και της κληρονομικότητας.

Σύνθεση (Composition) είναι η δημιουργία μίας κλάσης χρησιμοποιώντας ως μέλη της χαρακτηριστικά τα οποία δεν είναι κάποιου βασικού τύπου αλλά είναι αναφορές σε αντικείμενα του τύπου μιας άλλης κλάσης.

Στην **κληρονομικότητα** (Inheritance) δημιουργείται μία κλάση επεκτείνοντας μία άλλη κλάση έτσι ώστε η νέα κλάση να κληρονομεί αυτόματα τα χαρακτηριστικά και τις μεθόδους της υπερκλάσης (γονικής κλάσης).

Η κληρονομικότητα συνεπάγεται **εξειδίκευση** (specialization). Μία παραγόμενη κλάση (subclass/derived class) εξειδικεύεται κληρονομώντας όλα τα μέλη (χαρακτηριστικά και μεθόδους) της βασικής κλάσης (superclass/base class/parent class). Τα επιπλέον μέλη κάνουν την παραγόμενη κλάση πιο περιορισμένη, πιο ειδική. Το σύνολο των αντικειμένων μιας παραγόμενης κλάσης είναι ένα υποσύνολο των αντικειμένων της βασικής κλάσης. Για παράδειγμα, το σύνολο όλων των φοιτητών είναι ένα υποσύνολο του συνόλου των ανθρώπων.

Σημειώστε ότι το πιο σημαντικό κριτήριο για να κληρονομεί μία κλάση B (υποκλάση/παραγόμενη) μία άλλη κλάση A, είναι η κλάση B να περιλαμβάνει όλα τα μέλη της κλάσης A. Αυτό σημαίνει ότι το σύνολο των μελών της παραγόμενης κλάσης (υποκλάσης) είναι υποσύνολο του συνόλου των μελών της βασικής κλάσης (υπερκλάσης). Αν αυτό σας φαίνεται αντιφατικό σκεφτείτε ότι η εξειδίκευση που οδηγεί σε ένα μικρότερο σύνολο είναι αποτέλεσμα περισσότερων κριτηρίων, αφού τα κριτήρια είναι περιορισμοί και οι περισσότεροι περιορισμοί οδηγούν σε μικρότερα σύνολα.

Κληρονομικότητα σημαίνει εξειδίκευση ενώ σύνθεση σημαίνει **ενοποίηση/ενσωμάτωση**. Για παράδειγμα, μία κλάση "Student" που περιγράφει ένα φοιτητή θα είναι εξειδίκευση μίας κλάσης "Person" η οποία περιγράφει ένα άτομο, ενώ μπορεί να ενσωματώνει μία άλλη κλάση "Name" η οποία περιγράφει το όνομα ενός ατόμου.

Συχνά χρησιμοποιούνται οι φράσεις "*είναι ένα*" και "*έχει ένα*" για να διακρίνουμε μεταξύ κληρονομικότητας και σύνθεσης. Ένας φοιτητής "*είναι ένα*" άτομο (κληρονομικότητα), ενώ ένας φοιτητής "*έχει ένα*" όνομα (σύνθεση).

Η κατανόηση της διάκρισης μεταξύ των σχέσεων "*είναι ένα*" και "*έχει ένα*" μπορεί να μας βοηθήσει να αποφύγουμε το λάθος ορισμού νέας παραγόμενης κλάσης όταν δεν χρειάζεται. Αυτό το λάθος μπορεί να συμβεί όταν νομίσουμε ότι κληρονομικότητα σημαίνει μόνο πρόσθεση νέων μελών.

Η σύνθεση είναι αρκετά ευέλικτη. Τα αντικείμενα-μέλη της νέας κλάσης είναι συνήθως ιδιωτικά και έτσι δεν έχουν σε αυτά πρόσβαση οι προγραμματιστές που χρησιμοποιούν αυτή την κλάση. Αυτό μας επιτρέπει να μεταβάλλουμε αυτά τα μέλη χωρίς να επηρεάσουμε τον υπόλοιπο κώδικα του προγράμματος. Μπορούμε επίσης να αλλάξουμε τα αντικείμενα-μέλη κατά την διάρκεια της εκτέλεσης του προγράμματος, δηλαδή να μεταβάλλουμε δυναμικά την συμπεριφορά του προγράμματος. Η κληρονομικότητα δεν έχει αυτή την ευελιξία αφού ο μεταγλωττιστής πρέπει να θέσει κάποιους περιορισμούς, κατά το στάδιο της μεταγλώττισης, στις κλάσεις που δημιουργήθηκαν με κληρονομικότητα.

Η κληρονομικότητα είναι μεν σημαντική αλλά επειδή είναι πιο δυσνόητη, της αφιερώνεται συνήθως περισσότερος χρόνος και της δίνεται μεγαλύτερη έμφαση κατά την διδασκαλία της. Έτσι, οι νέοι προγραμματιστές δημιουργούν την λανθασμένη αντίληψη ότι πρέπει να την χρησιμοποιούν παντού. Αντιθέτως, όταν δημιουργούμε νέες κλάσεις, θα πρέπει πρώτα να ελέγξουμε την δυνατότητα χρήσης σύνθεσης, η οποία είναι απλούστερη και πιο ευέλικτη. Αφού αποκτηθεί κάποια εμπειρία, τότε είναι σχετικά εύκολο να κρίνουμε εάν χρειάζεται κληρονομικότητα.

2. Σύνθεση (Composition)

Δηλώστε την κλάση "**Vehicle**" με τα εξής χαρακτηριστικά:

- Ονοματεπώνυμο ιδιοκτήτη (*idioktitis* τύπου String),
- Αριθμός κυκλοφορίας (*arKykloforias* τύπου String),
- Έτος κατασκευής (*etosKat* τύπου int),
- Σύστημα διεύθυνσης (*sysdief* τύπου "SystimaDieythynsis"),
- Μηχανή αυτοκινήτου (*eng* τύπου "Mihani")

Φυσικά, για να δηλώσετε τα χαρακτηριστικά που περιγράφουν το σύστημα διεύθυνσης καθώς και τη μηχανή του οχήματος θα πρέπει να δημιουργήσετε τις αντίστοιχες δομές. Έτσι, θα πρέπει να δηλώσετε τις κλάσεις "SystimaDieythynsis" και "Mihani".

- Η κλάση "SystimaDieythynsis", έχει 4 μεθόδους: (*vazeiEmpros()*, *svynei()*, *striveiAristera()*, *striveiDexia()*).
- Η κλάση "Mihani", έχει 2 πεδία (*kyvismos:int*, *ippodynami:int*), και 3 μεθόδους (*leitourgia()*, *kinisi()*, *svysimo()*).

2.1. Δημιουργία αντικειμένων

Να γραφεί πρόγραμμα σε Java, το οποίο θα δημιουργεί 3 σύνθετα αντικείμενα σε πίνακα, της κλάσης “Vehicle”. Η κλάση αυτή θα ενσωματώνει δύο αναφορές σε αντικείμενα, μία της κλάσης “SystimaDieythynsis” και μία της κλάσης “Mihani”

Η δημιουργία των αντικειμένων θα γίνεται με κλήση των κατάλληλων δομητών και πέρασμα των τιμών παραμετρικά, εκτός των 2 πεδίων *idioktitis* και *arKykloforias* της κλάσης “Vehicle”.

2.2. Εισαγωγή τιμών στα 2 πεδία

Τιμές στα 2 πεδία της κλάσης “Vehicle”, *idioktitis* και *arKykloforias*, θα εισάγονται με κλήση των μεθόδων μεταβολής (μέθοδοι set).

2.3. Εμφάνιση χαρακτηριστικών ενός αυτοκινήτου

Να απαντώνται τα ακόλουθα ερωτήματα με την εμφάνιση σχετικών δεδομένων:

- 1) Τι κυβισμό έχει ένα αυτοκίνητο; Ο χρήστης εισάγει την θέση του αυτοκινήτου στον πίνακα και εμφανίζεται ο κυβισμός του.
- 2) Τι ιπποδύναμη έχει ένα αυτοκίνητο; Ο χρήστης εισάγει την θέση του αυτοκινήτου στον πίνακα και εμφανίζεται η ιπποδύναμή του.
- 3) Ποιος είναι ο ιδιοκτήτης του αυτοκινήτου, και το έτος κατασκευής του; Ο χρήστης εισάγει την θέση του αυτοκινήτου στον πίνακα και εμφανίζονται τα στοιχεία του ιδιοκτήτη (ονοματεπώνυμο) και το έτος κατασκευής του.
- 4) Κατά την κλήση των μεθόδων να εμφανίζονται μηνύματα σχετικά με την λειτουργία τους. Πχ. *vazeiEmprios()* → *leitourgia()*, *svynei()* → *svysimo()*.

3. Κληρονομικότητα - Ειδίκευση

Μια εταιρεία απασχολεί έναν αριθμό υπαλλήλων οι οποίοι διακρίνονται στις ακόλουθες κατηγορίες:

- ❖ **Ωρομίσθιοι.** Αμοιβονται σύμφωνα με τον αριθμό ωρών που έχουν εργαστεί με 7 ευρώ ανά ώρα και παίρνουν ένα επίδομα παραγωγικότητας 6% επί του συνόλου της αμοιβής των ωρομισθίων.
- ❖ **Μισθωτοί.** Αμοιβονται με βασικό μισθό 700 ευρώ συν ένα επίδομα τέκνων 5% επί του βασικού μισθού για κάθε παιδί, συν το χρονο-επίδομα το οποίο ανέρχεται σε 10% επί του βασικού για κάθε τριετία προϋπηρεσίας.
- ❖ **Προϊστάμενοι.** Αμοιβονται με βασικό μισθό 700 ευρώ, συν ένα επίδομα τέκνων 5% επί του βασικού μισθού για κάθε παιδί, συν το χρονο-επίδομα το οποίο ανέρχεται σε 10% επί του βασικού για κάθε τριετία προϋπηρεσίας, συν ένα επίδομα προϊσταμένου 10% επί του συνόλου των αποδοχών.

Επί πλέον στοιχεία που καταχωρούνται για κάθε υπάλληλο είναι τα ακόλουθα:

- ❖ **Όνομα, αριθμό μητρώου, ειδικότητα και δ/νη κατοικίας.**

Ζητείται, να υλοποιηθούν οι ανωτέρω κλάσεις με τα προαναφερόμενα στοιχεία.

Οι κλάσεις θα πρέπει να διαθέτουν μεθόδους πρόσβασης/μεταβολής και την μέθοδο toString().

Στο κυρίως πρόγραμμα δημιουργείστε 6 υπαλλήλους (2 από κάθε κλάση) και εμφανίστε τα στοιχεία τους.

Υποδείξεις:

- ♦ Η κλάση που περιγράφει έναν Ωρομίσθιο θα πρέπει επίσης να διαθέτει τον αριθμό ωρών εργασίας
- ♦ Οι κλάσεις που περιγράφουν ένα μισθωτό/προϊστάμενο θα πρέπει επίσης να διαθέτουν αριθμό τέκνων και έτη προϋπηρεσίας
- ♦ Η «ρίζα» της ιεραρχίας που θα δημιουργήσετε θα είναι η κλάση “Υπάλληλος”