

Abstract

- Εφαρμόζεται σε τάξεις και μεθόδους
- Τα interfaces είναι abstract εξ ορισμού (οπότε δεν το γράφουμε, το προσθέτει ο compiler, ωστόσο δεν είναι λάθος και να το γράψουμε)
- Τάξεις:
 - Δε μπορούμε να δημιουργήσουμε instance (αντικείμενο) αυτής
 - Χρησιμοποιείται κυρίως για να δηλώσουμε τάξεις που θέλουμε να επεκτείνουμε (extend) μέσω κληρονομικότητας
- Μέθοδοι:
 - Είναι οι μέθοδοι που έχουν υπογραφή αλλά όχι σώμα
 - Το σώμα των μεθόδων το παρέχουν συνήθως οι υπο-τάξεις
 - Μια μέθοδος που έχει σώμα αλλά κενό (...) δεν είναι abstract method
- Μια τάξη μπορεί να δηλωθεί abstract και να μην έχει abstract methods
- Αν μια μέθοδος δηλωθεί abstract τότε πρέπει να δηλωθεί abstract και η τάξη που την περιέχει

Σημειώσεις για abstract

- A class cannot be declared both abstract and final
- If a class contains one or more abstract methods then the class should be declared abstract, or else a compile error will be thrown
- An abstract class may contain both abstract methods as well normal methods
- An abstract class does not need to contain abstract methods
- Abstract methods can never be final
- Any class that extends an abstract class must implement all the abstract methods of the super class unless the subclass is also an abstract class
- Abstract methods end with a semicolon

Final

- Εφαρμόζεται σε τάξεις, μεθόδους και μεταβλητές
- Τα interfaces είναι abstract εξ ορισμού, οπότε δεν μπορούν να δηλωθούν final
- Τάξεις:
 - Μια final class δεν μπορεί να την επεκτείνει (μέσω κληρονομικότητας) καμία τάξη
- Μέθοδοι:
 - Μια final method δεν μπορεί να υποσκελιστεί (override) από μια τάξη κληρονόμο (sub class) (υπό τάξη)
- Μεταβλητές:
 - Σε μια final μεταβλητή δεν μπορούμε να ξάνα-δώσουμε τιμή. Δηλαδή θα πάρει τιμή μόνο μια φορά (μόνο μια φορά αρχικοποίηση) (Προσοχή: Τι συμβαίνει με τις reference variables?)

Static 1/2

- Εφαρμόζεται σε nested τάξεις, nested interfaces, μεθόδους και μεταβλητές
- Εμφωλευμένες τάξεις:
 - Δηλώνοντες ως στατικές εντός μιας top level class
 - Δεν μπορούν να προσπελάσουν μη στατικά μέλη (non static members)
 - Αναφερόμαστε σε αυτές χρησιμοποιώντας το όνομα της outer class
- Εμφωλευμένα interfaces:
 - Είναι by default δηλωμένα ως static, οπότε δεν έχει σημασία αν θα τα δηλώσουμε static ή όχι

Static 1/2

- Μέθοδοι:
 - Δηλώνονται ως στατικές εντός μιας class και ανήκουν σε αυτή (όχι στα στιγμιότυπά της)
 - Δεν μπορούν να προσπελάσουν μη στατικά μέλη (non static members) (προσοχή)
 - Αναφερόμαστε σε αυτές χρησιμοποιώντας το όνομα της outer class
 - Static methods μπορούν να υπάρχουν τόσο σε τάξεις, όσο και σε interfaces (Java 8+)
- Μεταβλητές:
 - Δηλώνονται ως στατικές εντός μιας class και ανήκουν σε αυτή (όχι στα στιγμιότυπά της)
 - Είναι κοινές για όλα τα instances της τάξης. Δεν χρειάζεται object instantiation για την ύπαρξή τους
 - Όλες οι μεταβλητές ενός interface είναι εξ ορισμού static
 - Στη Java για να δηλώσουμε μια σταθερά (δεν υπάρχει const) συνήθως χρησιμοποιούμε final μεταβλητές και πολλές φορές και static
 - Η προσπέλαση γίνεται τόσο από την τάξη, αλλά και από τα αντικείμενα (προσοχή στον 2^ο τρόπο, δεν ενδείκνυται)

Προσπέλαση μεταξύ static και non-static

- Μια στατική μέθοδος και μια στατική μεταβλητή δεν μπορούν να προσπελάσουν μη στατικές μεθόδους και μεταβλητές (Προσοχή: Τα στατικά μέλη προϋπάρχουν με την τάξη. Τα μη στατικά όμως;)
- Το αντίθετο ισχύει: Μη στατικές μέθοδοι και μεταβλητές μπορούν να προσπελάσουν στατικές μεθόδους και μεταβλητές
- Πολύ μεγάλη Προσοχή!:
 - Ακόμα και αν ένα αντικείμενο είναι null, μπορεί να προσπελάσει μια στατική μεταβλητή ή/και μέθοδο. Αυτό συμβαίνει διότι τα εν λόγω μέλη δεν ανήκουν σε αυτό, αλλά στην τάξη. Για το λόγο αυτό, δεν θα έχουμε null pointer exception. Ωστόσο, τέτοιος κώδικας πρέπει να αποφεύγεται!..

Static Vs Non-static

Member type	Can access <i>static</i> attribute or method?	Can access <i>non-static</i> attribute or method?
<code>static</code>	Yes	No
<code>Non-static</code>	Yes	Yes