



ΑΡΧΕΣ ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΟΥΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

Ιωάννης Χατζηλυγερούδης

Καθηγητής

Τμήμα Μηχ/κών Η/Υ και Πληροφορικής
Πανεπιστήμιο Πατρών

ΜΟΡΦΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

- Διαδικασιακός ή Διαδικαστικός (Procedural) ή Προστακτικός (Imperative) Pascal, C
- Αντικειμενοστρεφής (Object-Oriented) Java, C++
- Συναρτησιακός (Functional) Lisp
- Δηλωτικός (Declarative) Prolog

ΜΟΡΦΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ

- Διαδικασιακός ή Διαδικαστικός (Procedural)
ή Προστακτικός (Imperative)
- Αντικειμενοστρεφής (Object-Oriented)
- Συναρτησιακός (Functional)
- Δηλωτικός (Declarative)



Η ΕΝΝΟΙΑ ΤΟΥ ΑΝΤΙΚΕΙΜΕΝΟΥ

Προβλήματα του διαδικασιακού προγραμματισμού.



Ενθυλάκωση (Encapsulation)

(Δεδομένα και διαδικασίες ενσωματωμένα στην ίδια υπολογιστική οντότητα-δομή δεδομένων)



Αντικείμενο (Object)

ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΑΝΤΙΚΕΙΜΕΝΟΥ

- **Απόκρυψη ή Αφαίρεση Δεδομένων (Data Hiding or Abstraction)**

(Απόκρυψη λεπτομερειών αναπαράστασης και χρήσης των δεδομένων του αντικειμένου από τον εξωτερικό κόσμο)

- **Διεπαφή Αντικειμένου (Object interface)**

(Η προσπέλαση και επεξεργασία των δεδομένων γίνεται μέσω ενός συνόλου διαδικασιών)

ΔΟΜΗ ΑΝΤΙΚΕΙΜΕΝΟΥ

➤ Στατικό μέρος
(private)

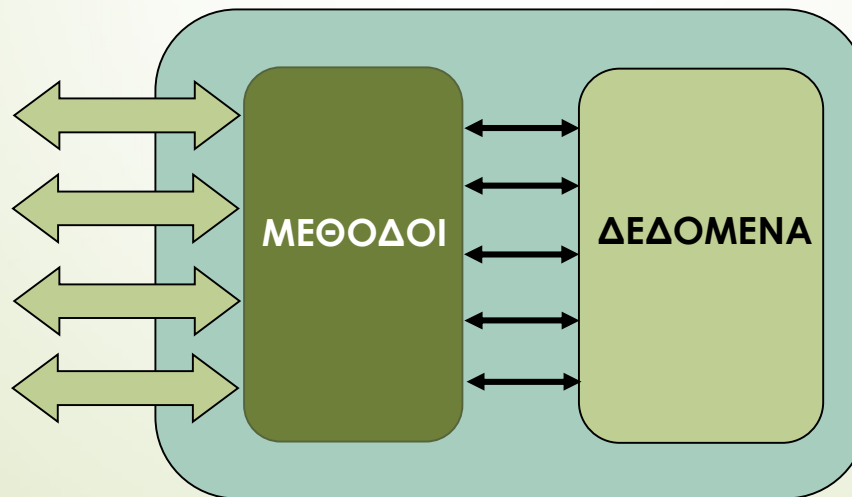


Δεδομένα
(Μεταβλητές)

➤ Δυναμικό μέρος
(public)



Διαδικασίες
(Μέθοδοι)





ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΕΦΗΣ ΠΡΟΣΕΓΓΙΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

- Το πεδίο εφαρμογής θεωρείται ως ένα σύνολο **αντικειμένων** οργανωμένων σε **ιεραρχία(ες)**, τα οποία επικοινωνούν μεταξύ τους με **μηνύματα**.
- Δύο τύποι αντικειμένων
 - * **Κλάση (class)**
 - * **Στιγμιότυπο (instance)**



ΚΛΑΣΗ

Κλάση

- Περιγράφει μια οικογένεια οντοτήτων με ίδια χαρακτηριστικά και συμπεριφορά (δηλ. μια αφηρημένη οντότητα)

Π.χ. Σκύλος, Τοστιέρα, Τρίγωνο



ΚΛΑΣΗ-ΠΑΡΑΔΕΙΓΜΑ

Article

Variables

price
quantity

Methods

```
selPrice () {  
    return (1.23 * price)}  
remove (q) {  
    quantity ← quantity - q}
```

ΜΕΤΑΒΛΗΤΕΣ-ΜΕΘΟΔΟΙ

- Οι **μεταβλητές** μιας κλάσης αναπαριστούν τα κοινά χαρακτηριστικά των οντοτήτων που αντιπροσωπεύει η κλάση ή της κλάσης ως όλον και σχετίζονται με το πρόβλημα.
- Οι **μέθοδοι** μιας κλάσης υλοποιούν διαδικασίες που σχετίζονται με το πρόβλημα και χρησιμοποιούν δεδομένα που σχετίζονται με τα χαρακτηριστικά της κλάσης.

ΜΕΤΑΒΛΗΤΕΣ-ΜΕΘΟΔΟΙ

- ▶ Οι μεταβλητές διακρίνονται σε
 - **Μεταβλητές στιγμιοτύπων** (αναφέρονται σε χαρακτηριστικά των στιγμιοτύπων)
 - **Μεταβλητές κλάσης** (αναφέρονται σε χαρακτηριστικά της κλάσης ως όλον)
- ▶ Οι μέθοδοι διακρίνονται σε
 - **Μέθοδοι στιγμιοτύπων** (αναφέρονται σε διαδικασίες τα αποτελέσματα των οποίων αφορούν τα στιγμιότυπα)
 - **Μέθοδοι κλάσης** (αναφέρονται σε διαδικασίες τα αποτελέσματα των οποίων αφορούν την κλάση ως όλον)

ΜΕΤΑΒΛΗΤΕΣ/ΜΕΘΟΔΟΙ ΚΛΑΣΗΣ- ΠΑΡΑΔΕΙΓΜΑ

Article

Variables

price
quantity

mostExpensiveArticle

Methods

```
selPrice () {  
  return (1.23 * price)}
```

```
remove (q) {  
  quantity ← quantity - q}
```

```
getLargestPrice () {  
  ...}
```

Μεταβλητή
κλάσης

Μέθοδος
κλάσης

ΙΕΡΑΡΧΙΕΣ ΚΛΑΣΕΩΝ

- Οι κλάσεις που συνιστούν ένα πρόβλημα (δηλ. ένα πρόγραμμα) είναι οργανωμένες σε μια ή περισσότερες **ιεραρχίες κλάσεων** (class hierarchies)
- Στην ιεραρχία, κάθε κλάση, πλην των τερματικών, έχει μια ή περισσότερες **υποκλάσεις** (subclasses)
- Κάθε κλάση έχει μια τουλάχιστον **υπερκλάση** (superclass)

ΥΠΟ/ΥΠΕΡ-ΚΛΑΣΕΙΣ

- Μια **υποκλάση** αποτελεί **εξειδίκευση** (specialization) μιας άλλης κλάσης (υπερκλάση)

Π.χ. σκύλος → κατοικίδιο

ηλεκ_κουζίνα → ηλεκ_συσκευή

- Μια **υπερκλάση** αποτελεί **γενίκευση** (generalization) μιας κλάσης (υποκλάση)

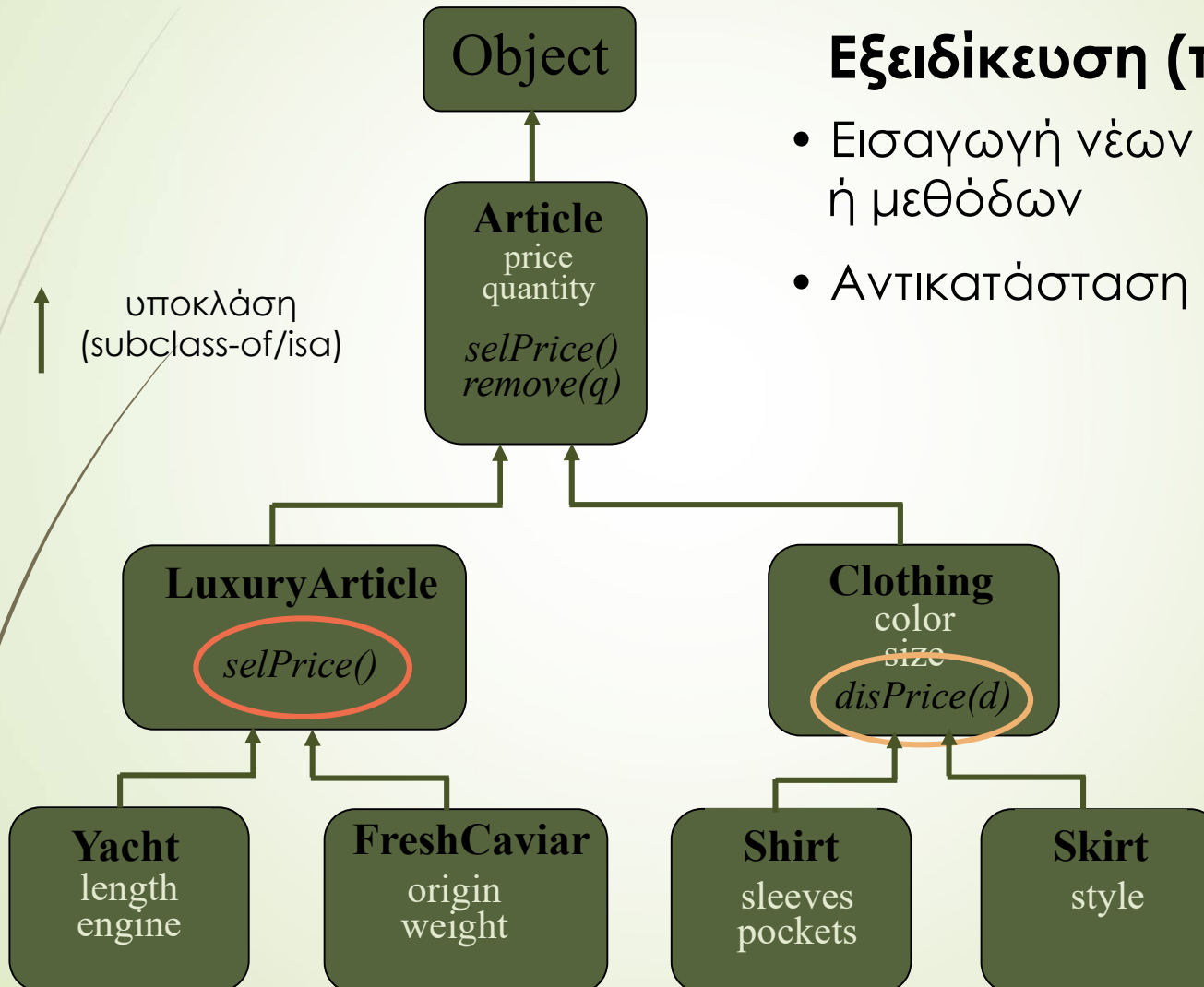
Π.χ. κατοικίδιο → σκύλος

ηλεκ_συσκευή → ηλεκ_κουζίνα

ΙΕΡΑΡΧΙΕΣ ΚΑΙ ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ

- Μια κλάση **κληρονομεί** μεταβλητές και μεθόδους από την (άμεση) υπερκλάση της και τις (έμμεσες) υπερκλάσεις αυτής.
- Η ιεραρχία (ή δέντρο) κλάσεων ονομάζεται και ιεραρχία (ή δέντρο) κληρονομικότητας.
- Η ρίζα του δέντρου ονομάζεται **βασική κλάση** (base class) ή **ρίζα** του δέντρου.
- Σ'ένα πρόγραμμα συνήθως έχουμε περισσότερες από μια βασικές κλάσεις, επομένως και δέντρα κληρονομικότητας
- Όλες οι βασικές κλάσεις είναι υποκλάσεις της κλάσης **Object** (η πιο γενική κλάση)

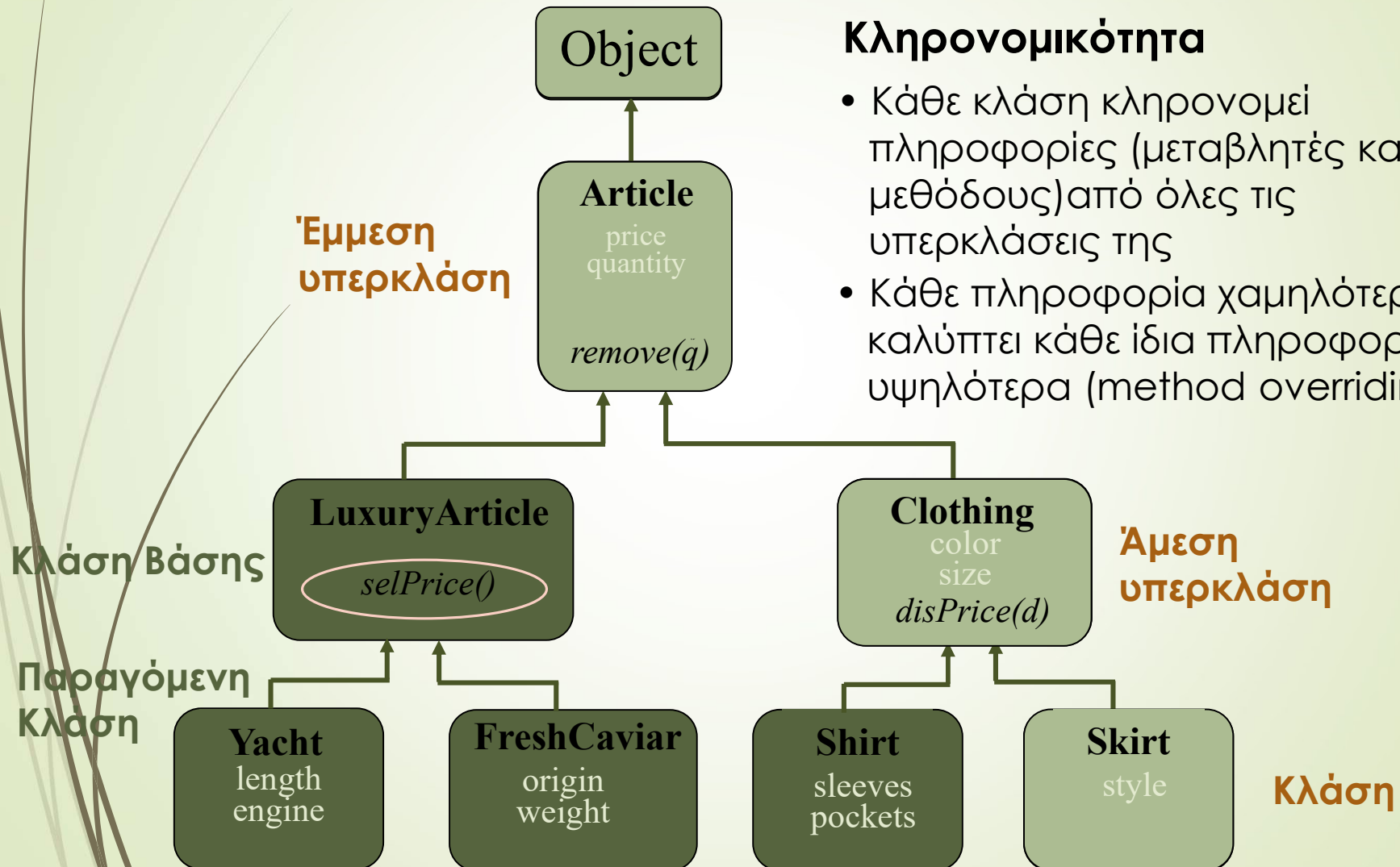
ΙΕΡΑΡΧΙΑ ΚΛΑΣΕΩΝ-ΠΑΡΑΔΕΙΓΜΑ



Εξειδίκευση (τεχνικές)

- Εισαγωγή νέων μεταβλητών ή μεθόδων
- Αντικατάσταση μεθόδων

ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ



Κληρονομικότητα

- Κάθε κλάση κληρονομεί πληροφορίες (μεταβλητές και μεθόδους) από όλες τις υπερκλάσεις της
- Κάθε πληροφορία χαμηλότερα καλύπτει κάθε ίδια πληροφορία υψηλότερα (method overriding)

ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ

- Διακρίνεται σε
 - **Απλή** (κάθε κλάση είναι υποκλάση μόνο μιας κλάσης)
 - **Πολλαπλή** (μια κλάση μπορεί να είναι υποκλάση περισσότερων της μιας κλάσης)
- Η κληρονομικότητα σχετίζεται με τη σχεδίαση του προγράμματος.
- Στοχεύει στην αναπαράσταση της ίδιας πληροφορίας μόνο μια φορά.

ΣΤΙΓΜΙΟΤΥΠΟ

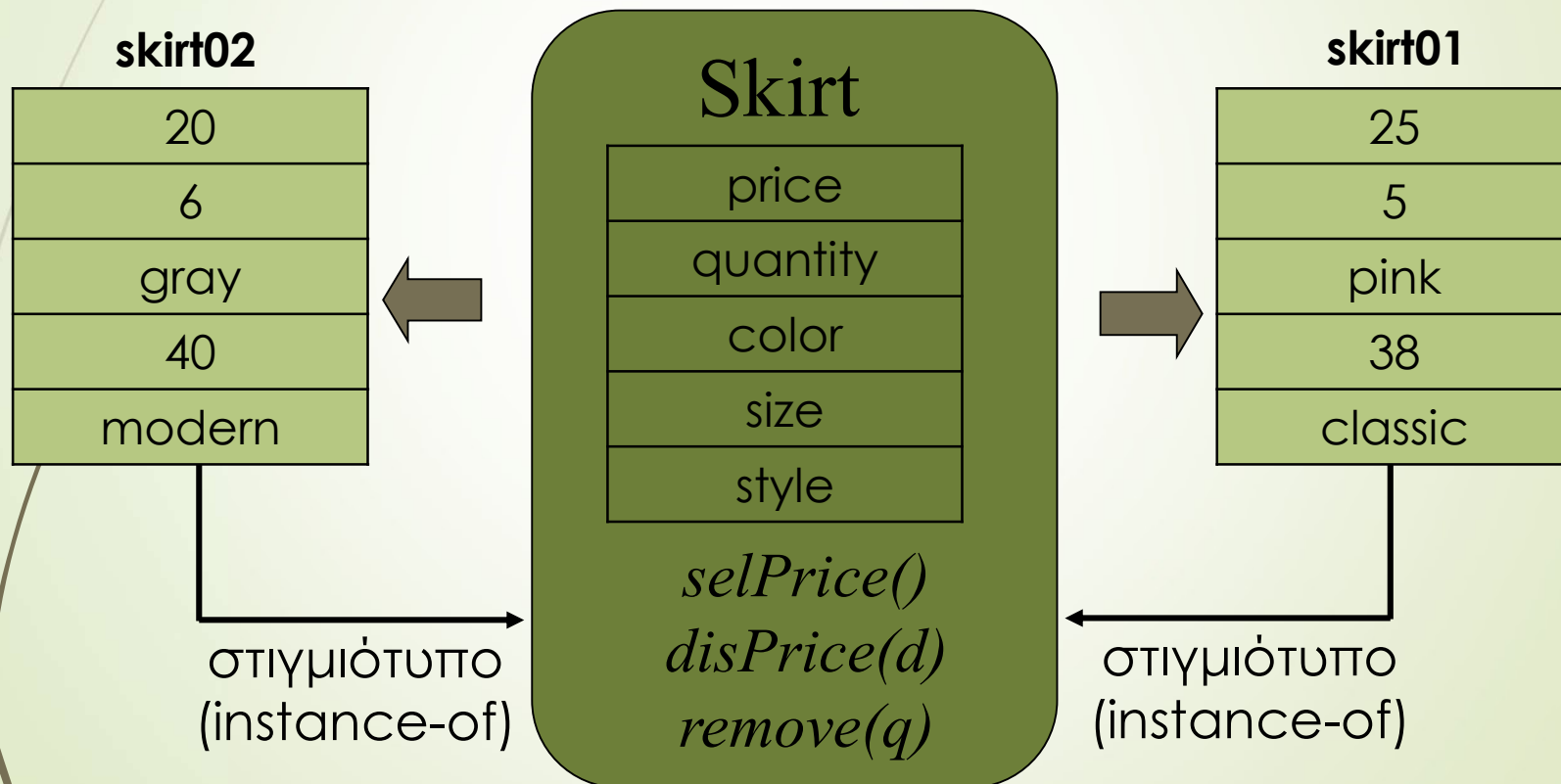
Στιγμιότυπο

- Περιγράφει μια συγκεκριμένη οντότητα, που **παράγεται** από μια αντίστοιχη κλάση, στην οποία λέμε ότι **ανήκει** ή ότι είναι **στιγμιότυπό** της.

Π.χ. Πλούτο, Τοστιέρα_ge50,
Τρίγωνο_ABΔ

ΔΗΜΙΟΥΡΓΙΑ ΣΤΙΓΜΙΟΤΥΠΩΝ


Στιγμιότυπα δημιουργούνται από μια κλάση με την **ανάθεση** συγκεκριμένων **τιμών στις μεταβλητές** της κλάσης.





ΔΗΜΙΟΥΡΓΟΙ

- Τα στιγμιότυπα δημιουργούνται με την βοήθεια ειδικού τύπου συναρτήσεων που ονομάζονται δημιουργοί ή κατασκευαστές (constructors).
- Οι δημιουργοί είναι συναρτήσεις που έχουν το ίδιο όνομα με την κλάση τους και αποδίδουν τιμές στις μεταβλητές της κλάσης.
- Μια κλάση μπορεί να έχει παραπάνω από ένα δημιουργούς.
- Ο δημιουργός μιας κλάσης μπορεί να χρησιμοποιεί τους δημιουργούς της υπερκλάσης της (π.χ. μέσω της λέξης-κλειδί super).



ΚΛΑΣΗ-ΔΟΜΗ

<ClassName>

Variables

var1

var2

...

Constructors

constructor1

constructor2

...

Methods

method1

method2

...



ΚΛΑΣΗ-ΠΑΡΑΔΕΙΓΜΑ

Skirt

Variables

style

Constructors

Skirt (p, q, c, sz, st) { ... }

Skirt (p, q, c, sz) { ... }

Skirt () { ... }

Methods

ΚΛΑΣΗ-ΠΑΡΑΔΕΙΓΜΑ

Skirt

Variables

price
quantity
color
size
style

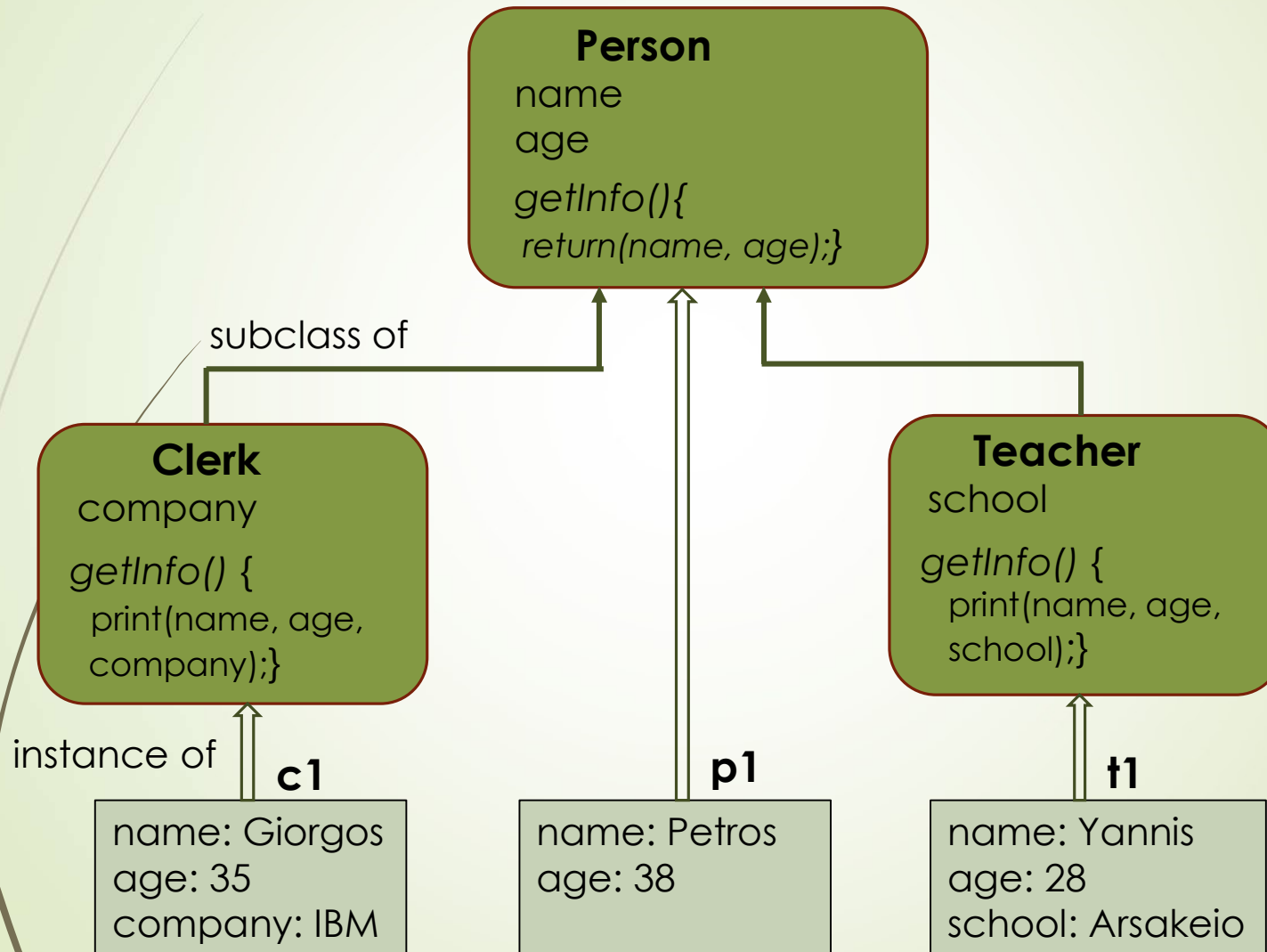
Constructors

Skirt (p, q, c, sz, st) { ... }
Skirt (p, q, c, sz) { ... }
Skirt () { ... }

Methods

selPrice() { ... }
disPrice(d) { ... }
remove(q) { ... }

ΙΕΡΑΡΧΙΑ ΚΛΑΣΕΩΝ-ΣΤΙΓΜΙΟΤΥΠΩΝ



ΑΠΟΣΤΟΛΗ ΜΗΝΥΜΑΤΩΝ

- Ο μόνος τρόπος επικοινωνίας μεταξύ των αντικειμένων
- Καταλήγει σε ενεργοποίηση αντίστοιχης μεθόδου
- Στοιχεία μηνύματος
 - Παραλήπτης
 - Όνομα μεθόδου
 - Παράμετροι μεθόδου (πραγματικοί)

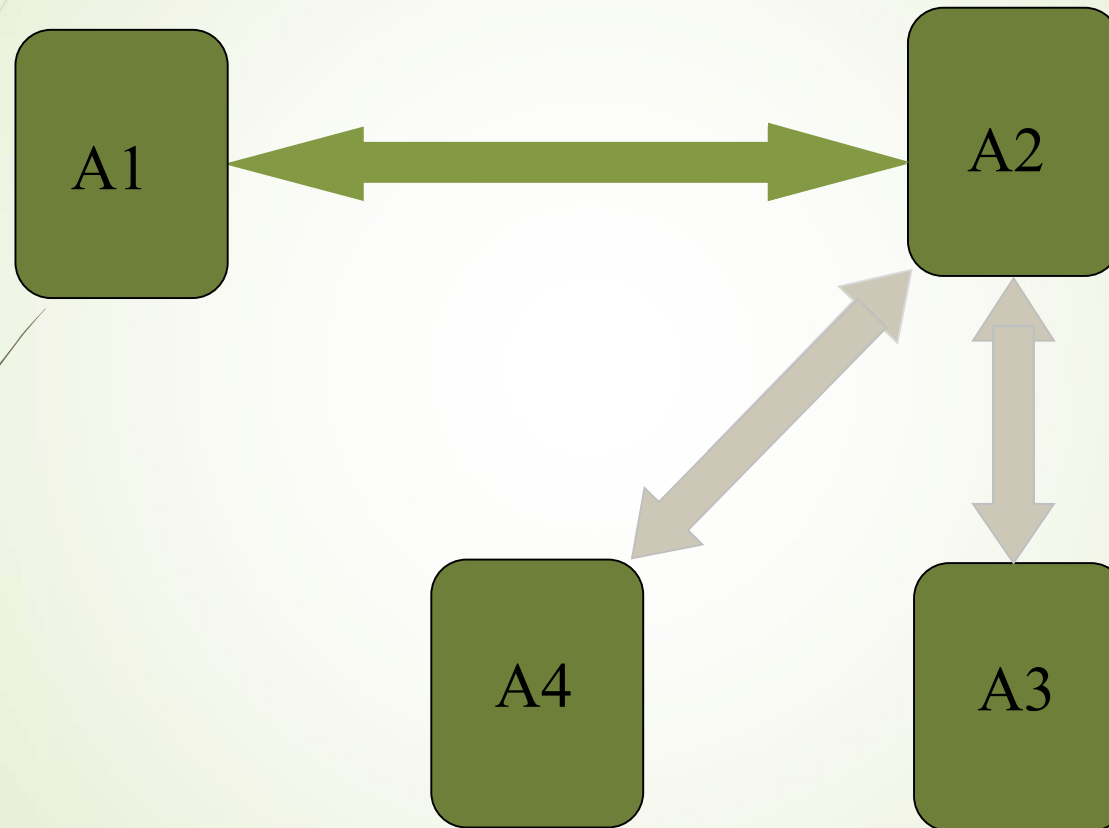
ΑΠΟΣΤΟΛΗ ΜΗΝΥΜΑΤΩΝ

- Μηνύματα αποστέλλονται κυρίως προς στιγμιότυπα.
- Μπορεί να αποστέλλονται και σε κλάσεις όταν αφορούν μεθόδους κλάσεις.
- Στη Java η αποστολή μηνύματος γίνεται ως εξής:

<αντικείμενο>.<μέθοδος>(<παράμετροι>)

Π.χ. c1.getInfo()

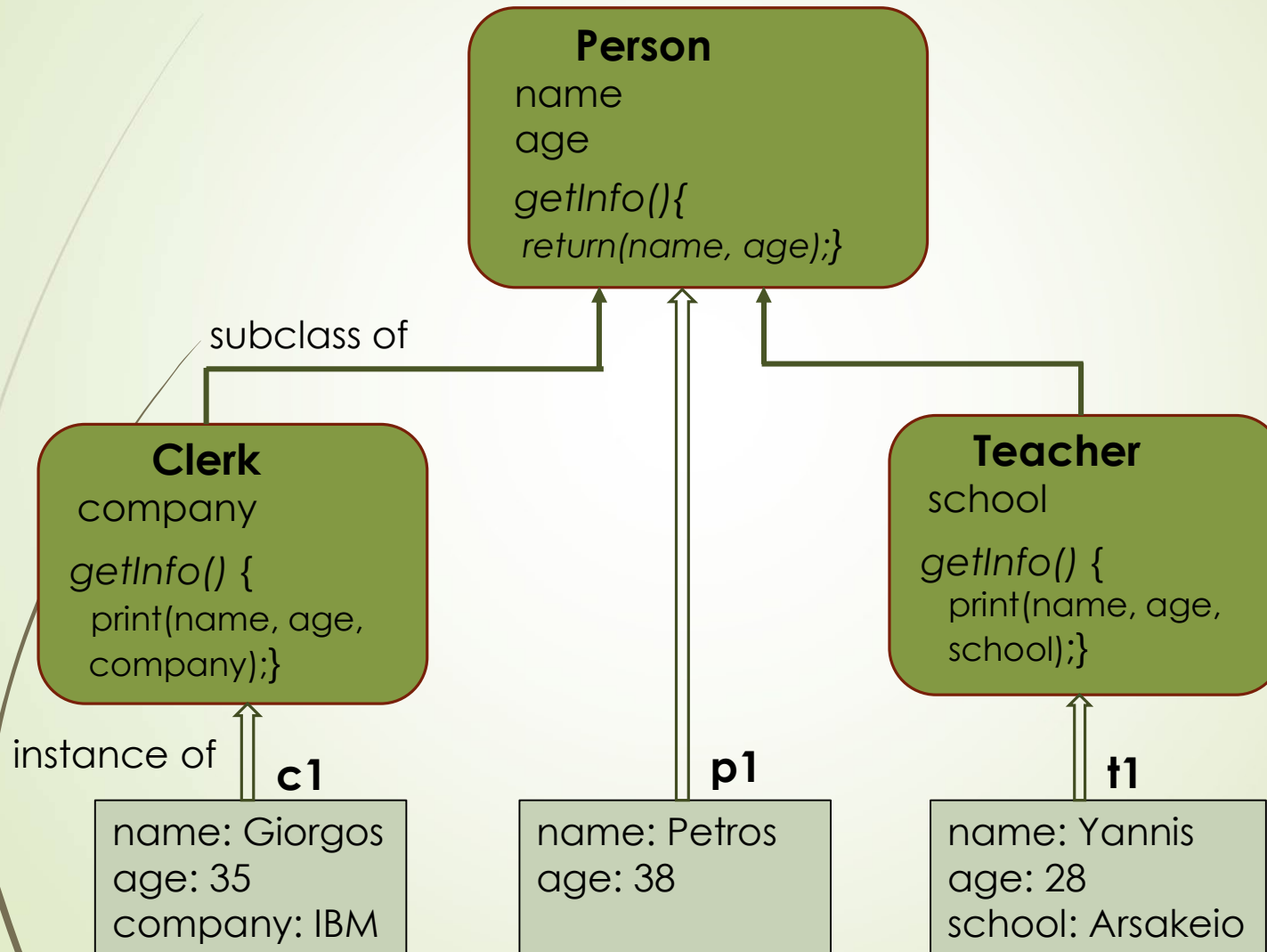
ΑΛΥΣΙΔΕΣ ΜΗΝΥΜΑΤΩΝ



ΠΟΛΥΜΟΡΦΙΣΜΟΣ

- Ο πολυμορφισμός (polymorphism) συνίσταται στη δυνατότητα εφαρμογής ενός υπολογιστικού προτύπου (pattern) με διαφορετικούς τύπους δεδομένων.
- Υπάρχουν διάφοροι τρόποι υλοποίησης του πολυμορφισμού στον αντικειμενοστρεφή προγραμματισμό.
 - Πολυμορφισμός στιγμιοτύπου
 - Πολυμορφισμός υποκλάσης
 - Υπερφόρτωση μεθόδου
 - Επικάλυψη μεθόδου

ΙΕΡΑΡΧΙΑ ΚΛΑΣΕΩΝ-ΣΤΙΓΜΙΟΤΥΠΩΝ



ΠΟΛΥΜΟΡΦΙΣΜΟΣ

Πολυμορφισμός στιγμιότυπου

Η αποστολή μηνύματος σ' ένα στιγμιότυπο δεν προϋποθέτει γνώση της κλάσης του παραλήπτη.

Π.χ. `c1.getInfo()`, `p1.getInfo()`, `t1.getInfo()`

Πολυμορφισμός υποκλάσης

Ο τύπος μιας κλάσης καλύπτει τις υποκλάσεις της.

Π.χ. Ορισμός: `presentInfo(Person p) {p.getInfo();}`

Κλήση: `presentInfo(p1)`, `presentInfo(c1)`,
`presentInfo(t1)`

ΠΟΛΥΜΟΡΦΙΣΜΟΣ

Υπερφόρτωση μεθόδου

Επιτρέπεται η ύπαρξη μεθόδων με το ίδιο όνομα και διαφορετικά ορίσματα στην ίδια ή σε διαφορετικές κλάσεις.

Π.χ. Περίπτωση δημιουργών

```
sum(int x, int y), sum(int x, int y, int z)
```

Επικάλυψη μεθόδου

Επιτρέπεται η ύπαρξη μεθόδων με το ίδιο όνομα, ίδια ορίσματα και ίδιο τύπο αποτελέσματος σε διαφορετικές κλάσεις στην ίδια διαδρομή. Αυτή που βρίσκεται χαμηλότερα στην ιεραρχία επικαλύπτει αυτές που βρίσκονται υψηλότερα.

Π.χ. `getInfo()`, `selPrice()`

▶ ΠΛΕΟΝΕΚΤΗΜΑ ΑΣΠ (ΟΟΡ)

- ▶ Αφαίρεση δεδομένων (απόκρυψη δεδομένων-ενθυλάκωση, πολυμορφισμός)
- ▶ Τμηματικότητα (κλάσεις, υποκλάσεις)
- ▶ Επεκτασιμότητα (αλλαγές σ' ένα αντικείμενο ελάχιστη επίπτωση έχουν στα άλλα)
- ▶ Επαναχρησιμοποίηση (κληρονομικότητα, βιβλιοθήκες αντικειμένων)
- ▶ Κατανοησιμότητα (απόρροια των δύο πρώτων)