

C++ Αντικειμενοστραφής Προγραμματισμός

Εργίνα Καβαλλιεράτου

Σημερινό μάθημα

- Μειονεκτήματα Δομημένου Προγραμματισμού
- Αντικειμενοστρεφής προγραμματισμός
- Ορισμοί
- Κλάσεις
- Αντικείμενα
- Χαρακτηριστικά ΑΠ
- C++ Class
- Παράδειγμα Σχεδίασης

Δομημένος Προγραμματισμός

Ο θεμέλιος λίθος του δομημένου προγραμματισμού είναι ο τεμαχισμός μεγάλων προγραμμάτων σε επιμέρους κομμάτια (λογισμικές μονάδες), όπου το καθένα έχει συγκεκριμένη αποστολή, αλλά συνδέεται και επικοινωνεί με τα άλλα για να δημιουργήσουν ένα ολοκληρωμένο πρόγραμμα.

Μειονεκτήματα Δομημένου Προγραμματισμού

- Η πολυπλοκότητα
- Τα δεδομένα έχουν υποεκτιμηθεί
- Η τοπικότητα των μεταβλητών
Υπάρχουν οι καθολικές αλλά τότε αυξάνει η πολυπλοκότητα
- Αυξημένη πιθανότητα λάθους
- Αυξημένη πολυπλοκότητα και στη διαχείριση των δεδομένων
- Η δημιουργία νέων τύπων δεδομένων είναι δύσκολη.

Αντικειμενοστρεφής προγραμματισμός

- Στον Αντικειμενοστρεφή προγραμματισμό, δίνεται έμφαση στη δημιουργία μονάδων οι οποίες θα περιλαμβάνουν τα δεδομένα αλλά και τις εντολές που τα διαχειρίζονται.

Ορισμοί

- Ο κόσμος αποτελείται από αντικείμενα απλά ή σύνθετα, που αλληλεπιδρούν μεταξύ τους
- Για κάθε αντικείμενο του φυσικού κόσμου μπορούμε να ορίσουμε, μια αφηρημένη έννοια που περιγράφει:
 - ✓ τον τρόπο με τον οποίο το αντικείμενο επικοινωνεί με το περιβάλλον του
 - ✓ το πώς αυτό είναι οργανωμένο εσωτερικά
- Ο ΑΠ ονομάζει στιγμιότυπα (instances) τα πραγματικά αντικείμενα και κλάσεις (classes) τις αφηρημένες έννοιες που τα περιγράφουν.

Ορισμοί

- Ο όρος αντικείμενο χρησιμοποιείται και για την αφηρημένη έννοια (κλάση) αλλά και για τα στιγμιότυπά της
- Ο ΑΠ θεωρεί όλες τις οντότητες, πραγματικές και αφηρημένες, αντικείμενα (objects).
- Εδώ θα ονομάζουμε την αφηρημένη έννοια κλάση και τα στιγμιότυπα της αντικείμενα.
- Τα αντικείμενα είναι σαν μεταβλητές και οι κλάσεις σαν τύποι.

Κλάσεις

| Καθηγητής |
|-------------------------------|
| Αρ. Ταυτότητας |
| Όνομα |
| Επώνυμο |
| Διεύθυνση |
| Τηλέφωνο |
| Προσθήκη Καθηγητή() |
| Διαγραφή Καθηγητή() |
| Μεταβολή στοιχείων Καθηγητή() |

| Μάθημα |
|--------------------------------|
| Κωδικός μαθήματος |
| Θεματική ενότητα |
| Τίτλος |
| Διδάσκων |
| Προσθήκη Μαθήματος() |
| Διαγραφή Μαθήματος() |
| Μεταβολή στοιχείων Μαθήματος() |
| Ανάθεση Μαθήματος() |

Αντικείμενα ή Στιγμιότυπα

Καθηγητής 01

Αρ. Ταυτότητας: **A123456**
Όνομα: **Βασίλειος**
Επώνυμο: **Βασιλείου**
Διεύθυνση: **Αγ. Βασιλείου 1**
Τηλέφωνο: **9876543**

Προσθήκη Καθηγητή()
Διαγραφή Καθηγητή()
Μεταβολή στοιχείων Καθηγητή()

Καθηγητής 02

Αρ. Ταυτότητας: **B987654**
Όνομα: **Γεώργιος**
Επώνυμο: **Γεωργίου**
Διεύθυνση: **Αγ. Γεωργίου 1**
Τηλέφωνο: **7654321**

Προσθήκη Καθηγητή()
Διαγραφή Καθηγητή()
Μεταβολή στοιχείων Καθηγητή()

Μάθημα 01

Κωδικός μαθήματος: **ΠΛ-034**
Θεματική ενότητα: **Πληροφορική**
Τίτλος: **Προγραμματισμός Η/Υ**
Διδάσκων: **B987654**

Προσθήκη Μαθήματος()
Διαγραφή Μαθήματος()
Μεταβολή στοιχείων Μαθήματος()
Ανάθεση Μαθήματος()

Κλάση/Αντικείμενο

- Στον ΑΠ δεν ασχολούμαστε με το πώς θα χωρίσουμε το πρόβλημα σε συναρτήσεις αλλά σε κλάσεις και αντικείμενα.
- Μία κλάση μπορεί να έχει ένα ή περισσότερα αντικείμενα.
- Κάθε αντικείμενο μπορεί να περιέχει μία ή περισσότερες συναρτήσεις, **συναρτήσεις-μέλη**
- Η κλάση λειτουργεί ως πρότυπο: δηλώνουμε σε αυτή δεδομένα και συναρτήσεις που θα έχουν τα αντικείμενα της
- Όταν δημιουργήσουμε αντικείμενα θα έχουν ότι έχει η κλάση που ανήκουν

Παράδειγμα

```
#include...
```

```
class Cat {
```

```
    unsigned int itsAge;
```

```
    unsigned int itsWeight;
```

```
    Meow();
```

```
}; ...
```

```
main()
```

```
{
```

```
    unsigned int GrossWeight;
```

```
    Cat Frisky;
```

```
...}
```

Χαρακτηριστικά ΑΠ

- Διαθέτει μηχανισμούς:
 - απόκρυψης πληροφορίας
 - περάσματος μηνυμάτων
 - δημιουργίας και καταστροφής στιγμιotypών
 - αναπαράστασης ιεραρχίας
 - κληρονομικότητας κ.α.

Χαρακτηριστικά ΑΠ

- Οι μηχανισμοί αυτοί βελτιώνουν την ποιότητα του λογισμικού ως προς τα χαρακτηριστικά του:
 - επαναχρησιμοποίηση (reusability)
 - επεκτασιμότητα (extendibility)
 - ορθότητα (correctness)
 - ευρωστία (robustness)
 - συντηρησιμότητα (maintainability)

C++ Class

```
class Classic_Example {  
    public:  
        // Δεδομένα και διαδικασίες προσβάσιμα από παντού  
  
    protected:  
        // Δεδομένα και διαδικασίες προσβάσιμα από την κλάση  
        // τις παραγόμενες κλάσεις και friends κλάσεις  
  
    private:  
        // Δεδομένα και διαδικασίες προσβάσιμα από την κλάση  
        // και friends κλάσεις  
  
};
```


C++ Class - Συμβουλή

- Δηλώστε τις συναρτήσεις στο PUBLIC
- Δηλώστε τα πεδία δεδομένων στο PRIVATE

C++ Classes

```
class Mailbox
{
private:
    int day;
    int month;
    int year;
public:
    void add(Message);
    Message get_current();
    void delete_current();
};
```

Πρόσβαση σε συναρτήσεις κλάσεων

- **Εκτός** κλάσης, public συναρτήσεις μπορούν να κληθούν μόνο με αναφορά σε αντικείμενο της κλάσης.
- **Μέσα** στην κλάση, οι διάφορες συναρτήσεις μπορούν να κληθούν από άλλες συναρτήσεις ως απλές συναρτήσεις

C++ Class

- Τα δεδομένα μιας κλάσης καλούνται ΔΕΔΟΜΕΝΑ ΜΕΛΗ ενώ οι διαδικασίες λέγονται ΣΥΝΑΡΤΗΣΕΙΣ ΜΕΛΗ.
- Ο τελεστής ‘.’ χρησιμοποιείται για πρόσβαση και στα δεδομένα και στις συναρτήσεις.
- Η συνάρτηση αλλάζει τις τιμές που ανήκουν στο αντικείμενο που καλεί τη διαδικασία

`b.advance(30);`

- Μια διαδικασία κλάσης μπορεί να έχει και ανεξάρτητες με την κλάση παραμέτρους

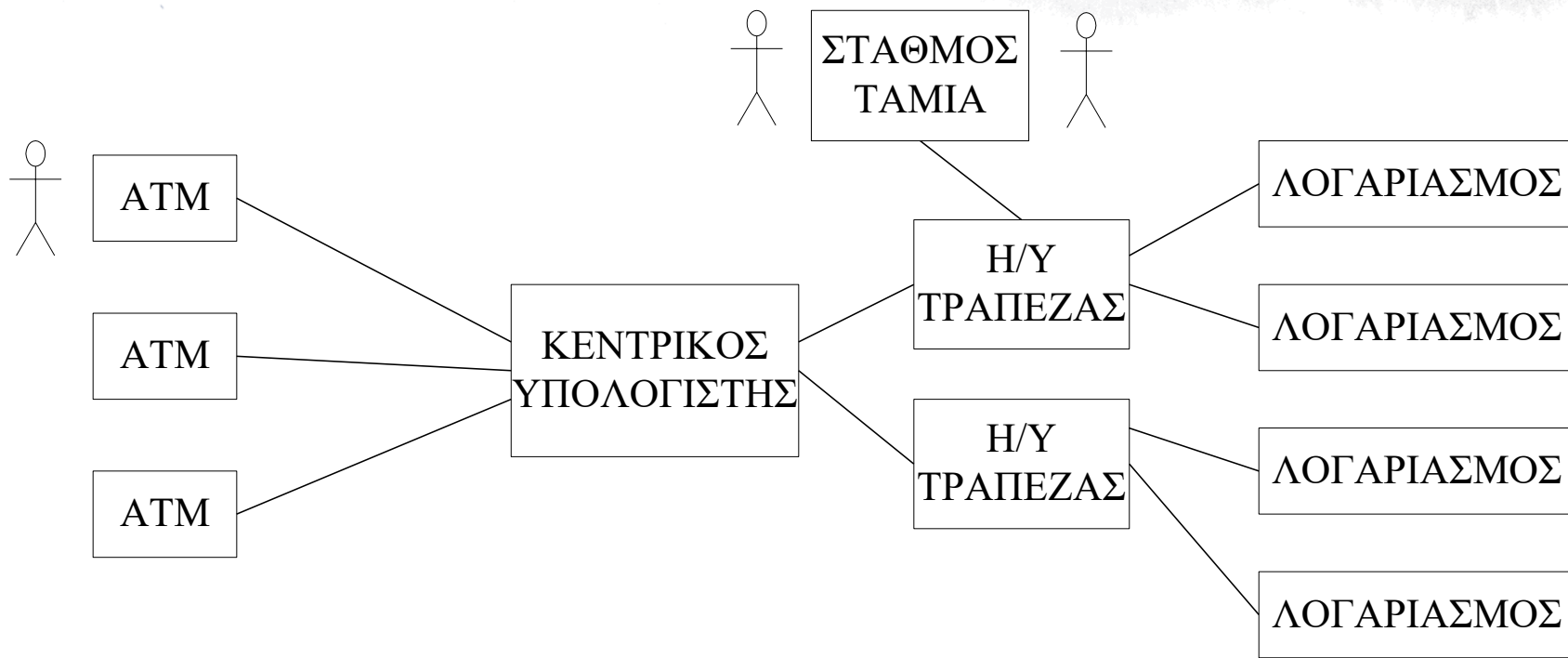
Πρόβλημα

- Σχεδιάστε το **λογισμικό** υποστήριξης ενός **τραπεζικού συστήματος** που θα εξυπηρετεί τους **ταμίες** και τις **Αυτόματες Ταμειολογιστικές Μηχανές (ATM)** ενός **τραπεζικού ομίλου**.
- Κάθε **τράπεζα** διαθέτει το **δικό της υπολογιστή** για να συντηρεί τους **λογαριασμούς** και να επεξεργάζεται τις **συναλλαγές** τους.
- Οι **τερματικοί σταθμοί των ταμείων** ανήκουν στην τράπεζα όπου βρίσκονται και επικοινωνούν απευθείας με τον υπολογιστή της τράπεζας.
- Οι ταμίες εισάγουν **δεδομένα λογαριασμών** και **δεδομένα συναλλαγών**.
- Οι ATM επικοινωνούν με τον **κεντρικό υπολογιστή** που διενεργεί τις συναλλαγές με τις αντίστοιχες τράπεζες.

Πρόβλημα (συνέχεια)

- Η ΑΤΜ όταν δέχεται **τραπεζική κάρτα**, αλληλεπιδρά με το **χρήστη**, επικοινωνεί με τον κεντρικό υπολογιστή για την έγκριση της εξ'αποστάσεως συναλλαγής, αποδίδει **μετρητά** και εκτυπώνει **αποδείξεις**.
- Το **σύστημα** απαιτεί κατάλληλη **καταγραφή στοιχείων** και **πρόνοιες ασφάλειας**.
- Το σύστημα διαχειρίζεται με σωστό τρόπο ταυτόχρονες **προσπελάσεις** στον ίδιο λογαριασμό.
- Εσείς πρέπει να σχεδιάσετε το λογισμικό των ΑΤΜ και του συστήματος.
- Το **κόστος** του διαμοιραζόμενου συστήματος επιμερίζεται στις τράπεζες σύμφωνα με τον αριθμό των **πελατών** που διαθέτουν τραπεζικές κάρτες.

Πρόβλημα



Αντικειμενοστρεφής Ανάλυση & Σχεδίαση

- Αναγνώριση αντικειμένων και κλάσεων
- Αναγνώριση συνδέσεων μεταξύ των αντικειμένων
- Αναγνώριση ιδιοτήτων των αντικειμένων
- Οργάνωση και απλοποίηση των κλάσεων αντικειμένων με τη χρήση κληρονομικότητας

Αναγνώριση αντικειμένων

- Στα αντικείμενα περιλαμβάνονται φυσικές οντότητες, όπως σπίτια, εργαζόμενοι κ.α. αλλά και λογικές οντότητες όπως κρατήσεις θέσεων, πρόγραμμα πληρωμών κ.α.
- Όλες οι κλάσεις πρέπει να έχουν νόημα στο πεδίο της εφαρμογής: αποφεύγονται αντικείμενα όπως συνδετικές λίστες, υπορουτίνες κ.α.
- Δεν αναφέρονται πάντα όλες οι κλάσεις στις προδιαγραφές της εφαρμογής: κάποιες υπονοούνται ή προκύπτουν από τη γενικότερη γνώση του προβλήματος.

Αναγνώριση αντικειμένων

- Οι κλάσεις αντιστοιχούν συνήθως σε ουσιαστικά. Γράψτε μία λίστα από υποψήφιας κλάσεις που προκύπτουν από την περιγραφή του προβλήματος.
- Στη φάση αυτή δεν μας ενδιαφέρει η κληρονομικότητα ή οι υψηλού επιπέδου abstract κλάσεις.



Αναγνώριση αντικειμένων

ΛΟΓΙΣΜΙΚΟ

ΤΡΑΠΕΖΙΚΟ
ΣΥΣΤΗΜΑ

ΤΑΜΙΑΣ

ΑΤΜ

ΤΡΑΠΕΖΙΚΟΣ
ΟΜΙΛΟΣ

ΤΡΑΠΕΖΑ

Η/Υ
ΤΡΑΠΕΖΑΣ

ΛΟΓΑΡΙΑΣΜΟΣ

ΣΥΝΑΛΛΑΓΗ

ΤΕΡΜΑΤΙΚΟ
ΤΑΜΕΙΟΥ

ΔΕΔΟΜΕΝΑ
ΛΟΓΑΡΙΑΣΜΑ
ΩΝ

ΔΕΔΟΜΕΝΑ
ΣΥΝΑΛΛΑΓΩΝ

ΚΕΝΤΡΙΚΟΣ
ΥΠΟΛΟΓΙΣΤΗΣ

ΤΡΑΠΕΖΙΚΗ
ΚΑΡΤΑ

ΧΡΗΣΤΗΣ

ΜΕΤΡΗΤΑ

ΑΠΟΔΕΙΞΗ

ΣΥΣΤΗΜΑ

ΚΑΤΑΓΑΦΗ
ΣΤΟΙΧΕΙΩΝ

ΠΡΟΝΟΙΑ
ΑΣΦΑΛΕΙΑΣ

ΠΡΟΣΠΕΛΑΣΗ

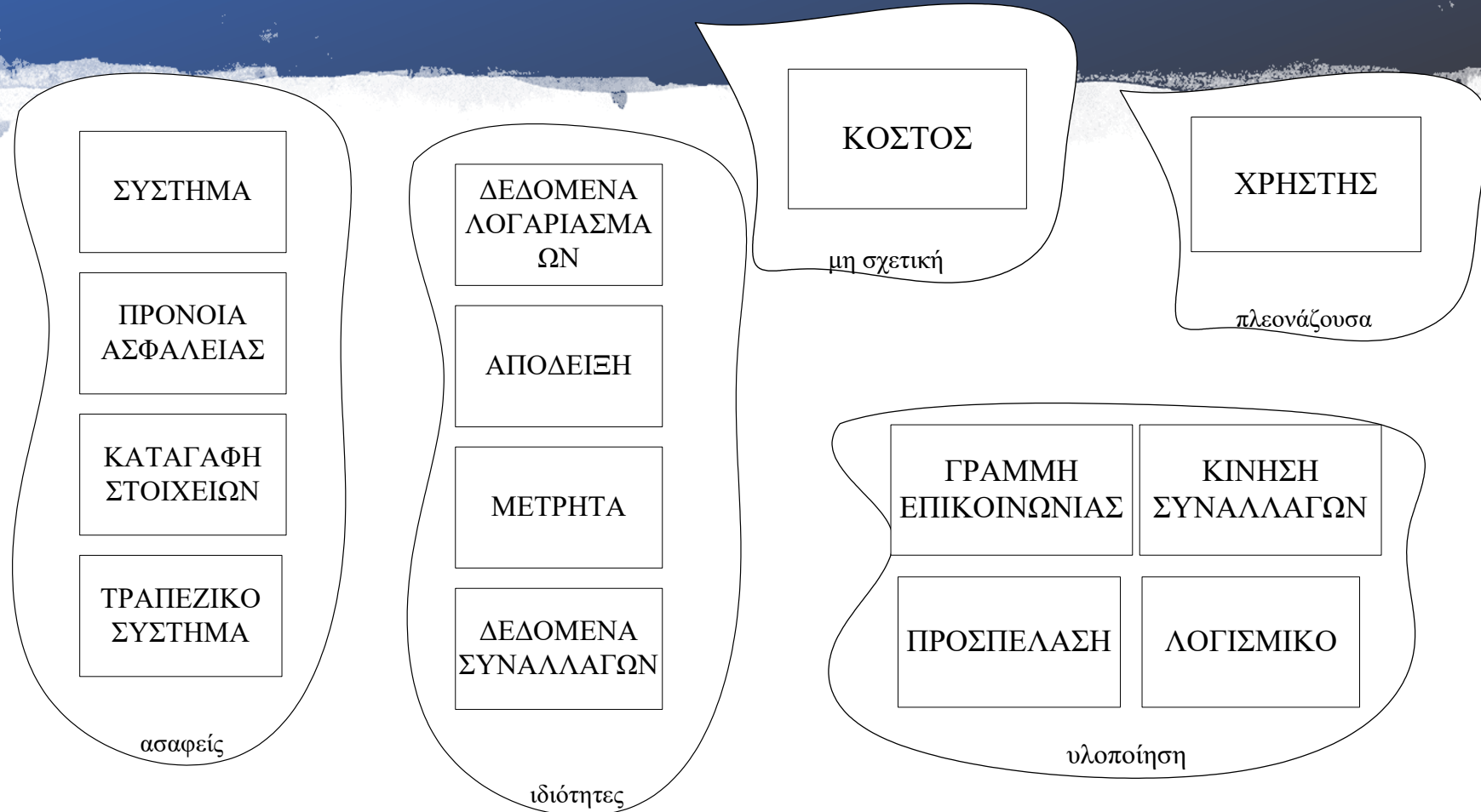
ΚΟΣΤΟΣ

ΠΕΛΑΤΗΣ

Αναγνώριση αντικειμένων

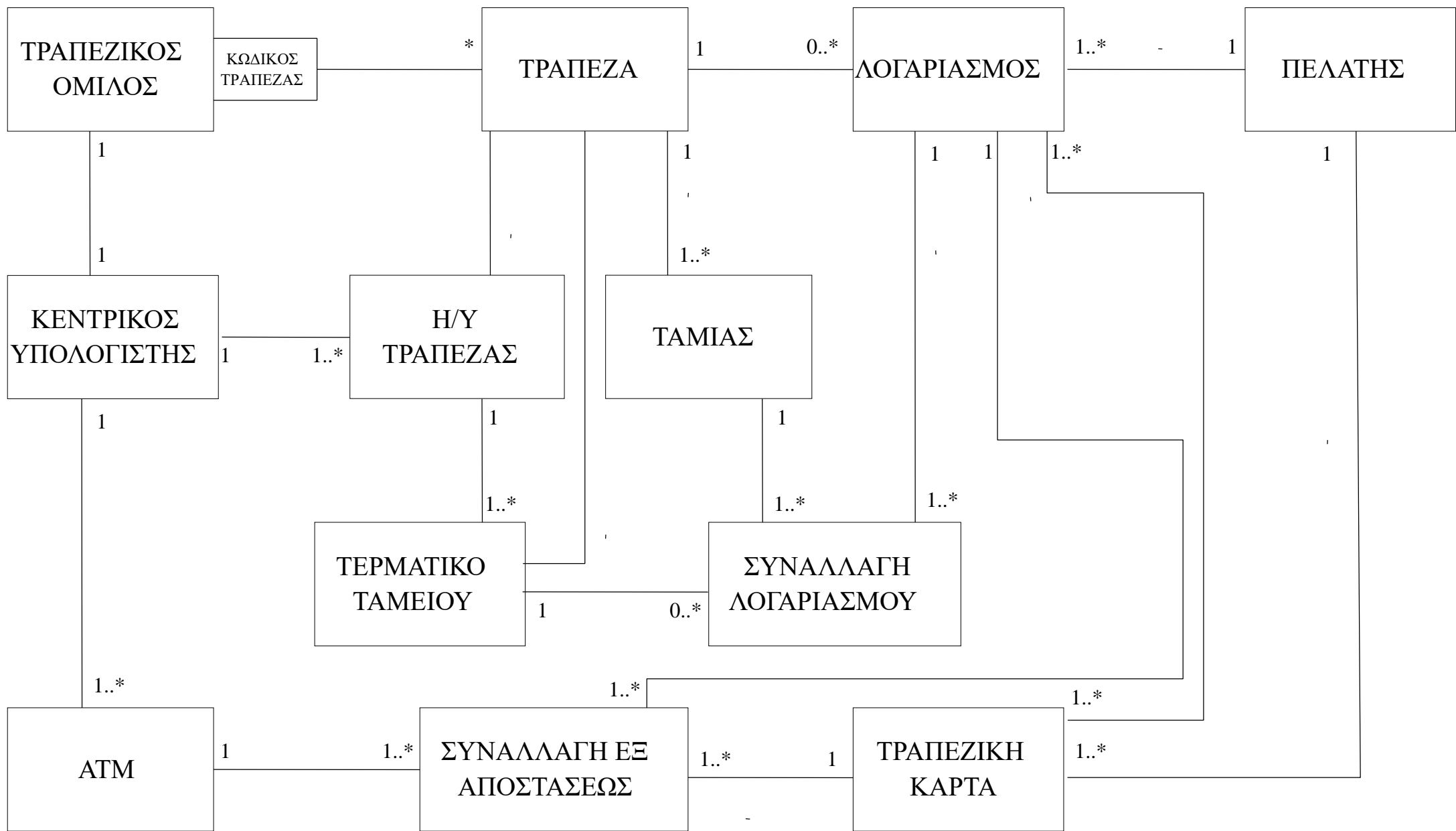
- Απαλείφονται οι κλάσεις που:
 - είναι πλεονάζουσες
 - είναι μη σχετικές
 - είναι ασαφείς
 - αποτελούν ιδιότητες άλλων κλάσεων
 - αποτελούν λειτουργίες αντικειμένων
 - αποτελούν ρόλο που η κλάση παίζει σε μία σύνδεση π.χ. κάτοχος.
Μία αποδεκτή κλάση θα ήταν π.χ. ο πελάτης
 - έχουν σχέση με την υλοποίηση του συστήματος

Αναγνώριση αντικειμένων



Αποδεκτές κλάσεις

| | | | | | |
|-------------|----------------------|--------------------------|----------------------|--------------------|-----------|
| ΛΟΓΑΡΙΑΣΜΟΣ | ΑΤΜ | ΤΡΑΠΕΖΑ | Η/Υ ΤΡΑΠΕΖΑΣ | ΤΡΑΠΕΖΙΚΗ ΚΑΡΤΑ | |
| ΤΑΜΙΑΣ | ΤΕΡΜΑΤΙΚΟ ΤΑΜΕΙΟΥ | ΚΕΝΤΡΙΚΟΣ ΥΠΟΛΟΓΙΣΤΗΣ | ΤΡΑΠΕΖΙΚΟΣ ΟΜΙΛΟΣ | ΠΕΛΑΤΗΣ | ΣΥΝΑΛΛΑΓΗ |



Αντικειμενοστρεφής Ανάλυση & Σχεδίαση

- Αναγνώριση αντικειμένων και κλάσεων
- Αναγνώριση συνδέσεων μεταξύ των αντικειμένων
- Αναγνώριση ιδιοτήτων των αντικειμένων
- Οργάνωση και απλοποίηση των κλάσεων αντικειμένων με τη χρήση κληρονομικότητας

Αναγνώριση συνδέσεων

- Οποιασδήποτε μορφής εξάρτηση μεταξύ δύο ή περισσότερων κλάσεων είναι σύνδεση.
- Επίσης, σύνδεση είναι και η αναφορά μιας κλάσης σε μία άλλη.
- Οι ιδιότητες κλάσεων δεν θα πρέπει να αναφέρονται ως κλάσεις, αλλά πρέπει να χρησιμοποιούνται συνδέσεις.
- Όταν σε αυτό το στάδιο της σχεδίασης ανατίθενται σε ιδιότητες αντικειμένων τιμές άλλων αντικειμένων, τότε γίνεται εμφανής η σύνδεση μεταξύ των αντικειμένων σε ένα διάγραμμα συνδέσεων.

Αναγνώριση συνδέσεων

- Οι συνδέσεις υλοποιούνται με διάφορους τρόπους, αλλά σε αυτή τη φάση της σχεδίασης δεν μας απασχολεί η υλοποίηση.
- Οι συνδέσεις εκφράζονται συνήθως με **ρήματα** ή με **φράσεις με ρήματα**.
- Μπορεί να εκφράζουν θέση (δίπλα στο, μέρος του, περιέχεται στο), ενέργειες, επικοινωνία (μιλά στο), κατοχή (έχει, μέρος του) ή κάποια συνθήκη (δουλεύει για το, είναι παντρεμένος στο, διευθύνει) κ.α.

Αναγνώριση συνδέσεων

Φράσεις με ρήματα

- Το τραπεζικό σύστημα **εξυπηρετεί** τους ταμίες και τις ΑΤΜ.
- Οι ΑΤΜ **ανήκουν** στον τραπεζικό όμιλο.
- Η τράπεζα **διαθέτει** έναν Η/Υ τράπεζας.
- Ο Η/Υ της τράπεζας **συντηρεί** λογαριασμούς.
- Ο Η/Υ της τράπεζας **επεξεργάζεται** συναλλαγές λογαριασμών.
- Η τράπεζα **διαθέτει** τερματικό σταθμό ταμείου.
- Ο τερματικός σταθμός του ταμείου **επικοινωνεί** με τον Η/Υ της τράπεζας.
- Ο ταμίας **εισάγει** συναλλαγές λογαριασμών.

Αναγνώριση συνδέσεων

- Οι ATM **επικοινωνούν** με τον κεντρικό υπολογιστή
- Ο κεντρικός υπολογιστής **διενεργεί** τη συναλλαγή με την τράπεζα
- Η ATM **δέχεται** τραπεζική κάρτα.
- Η ATM **αλληλεπιδρά** με το χρήστη.
- Η τραπεζική κάρτα **εγκρίνει** την εξ'αποστάσεως συναλλαγή
- Η ATM **αποδίδει** μετρητά.
- Η ATM **εκτυπώνει** αποδείξεις.
- Το σύστημα **διαχειρίζεται** προσπελάσεις.
- Η τράπεζα **παρέχει** το λογισμικό.
- Το κόστος **επιμερίζεται** στις τράπεζες.

Αναγνώριση συνδέσεων

Ρήματα που εννοούνται

- Ο όμιλος **αποτελείται** από τράπεζες.
- Η τράπεζα **διατηρεί** τους λογαριασμούς.
- Ο όμιλος **έχει** έναν κεντρικό υπολογιστή.
- Το σύστημα **κάνει** καταγραφή στοιχείων.
- Το σύστημα **παρέχει** ασφάλεια.
- Οι πελάτες **έχουν** τραπεζικές κάρτες.

Αναγνώριση συνδέσεων

Ρήματα που προκύπτουν από τη γνώση του πεδίου της εφαρμογής

- Η τραπεζική κάρτα **έχει προσπέλαση** σε λογαριασμούς.
- Η τράπεζα **απασχολεί** ταμίες.

Αναγνώριση συνδέσεων

Απαλείφονται οι συνδέσεις που:

- είναι συνδέσεις μεταξύ κλάσεων που έχουν απαλειφθεί
- δεν είναι σχετικές ή αφορούν την υλοποίηση του συστήματος
- εκφράζουν ενέργειες. Μία σύνδεση πρέπει να εκφράζει δομική ιδιότητα της εφαρμογής και όχι παροδικά γεγονότα.
- τριπλές συνδέσεις: οι τριπλές συνδέσεις συνήθως αποσυντίθενται σε διπλές
- συναγόμενες συνδέσεις: δεν περιλαμβάνουμε συνδέσεις που προκύπτουν από άλλες συνδέσεις γιατί θεωρούνται πλεονασματικές συνδέσεις

Αναγνώριση συνδέσεων

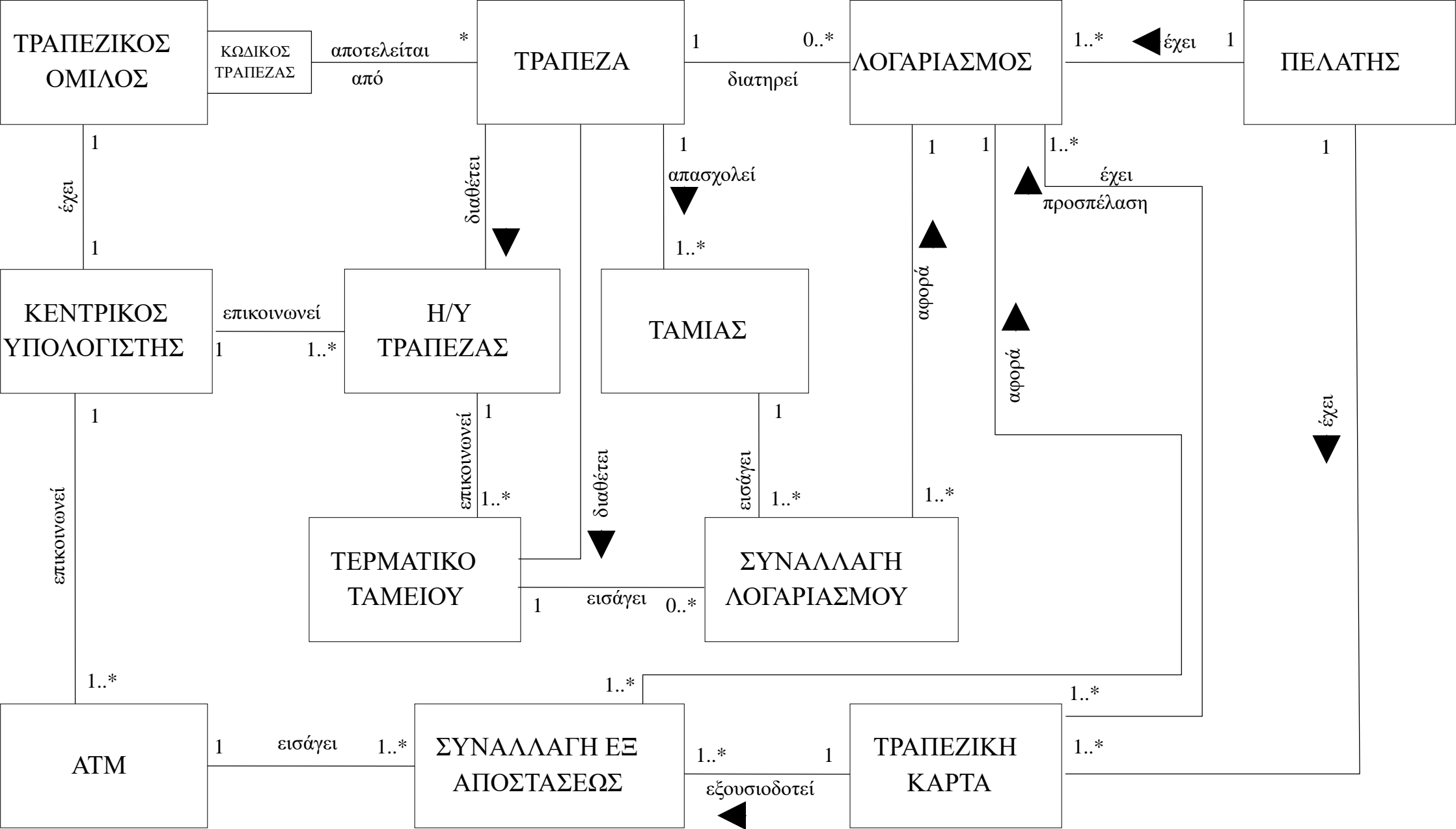
Προσδιορίζουμε τη σημασία των συνδέσεων ως εξής:

- αποφεύγουμε την περιγραφή του πως ή του γιατί προκύπτει μία κατάσταση.
- Περιγράφουμε το τι είναι μία κατάσταση.

Π.χ. «Η τράπεζα συντηρεί λογαριασμούς.» είναι μία ενέργεια.

Το αναδιατυπώνουμε ως «Η τράπεζα διατηρεί λογαριασμούς.»

- περιορισμένες συνδέσεις (μη μοναδικά ονόματα) π.χ. κωδικοί
- αναγνωρίζουμε πολλαπλότητα
- συμπληρώνουμε τις συνδέσεις που λείπουν



Αντικειμενοστρεφής Ανάλυση & Σχεδίαση

- Αναγνώριση αντικειμένων και κλάσεων
- Αναγνώριση συνδέσεων μεταξύ των αντικειμένων
- Αναγνώριση ιδιοτήτων των αντικειμένων
- Οργάνωση και απλοποίηση των κλάσεων αντικειμένων με τη χρήση κληρονομικότητας

Αναγνώριση ιδιοτήτων

- Οι ιδιότητες συνήθως αντιστοιχούν σε ουσιαστικά που ακολουθούνται από **κτητικές φράσεις** του τύπου «το χρώμα του αυτοκινήτου». Επίθετα όπως «κόκκινο» εκφράζουν συνήθως τιμές ιδιοτήτων.
- Σε αντίθεση με τις συνδέσεις και τις κλάσεις οι ιδιότητες των αντικειμένων είναι λιγότερο πιθανό να περιγράφονται πλήρως στις προδιαγραφές της εφαρμογής.

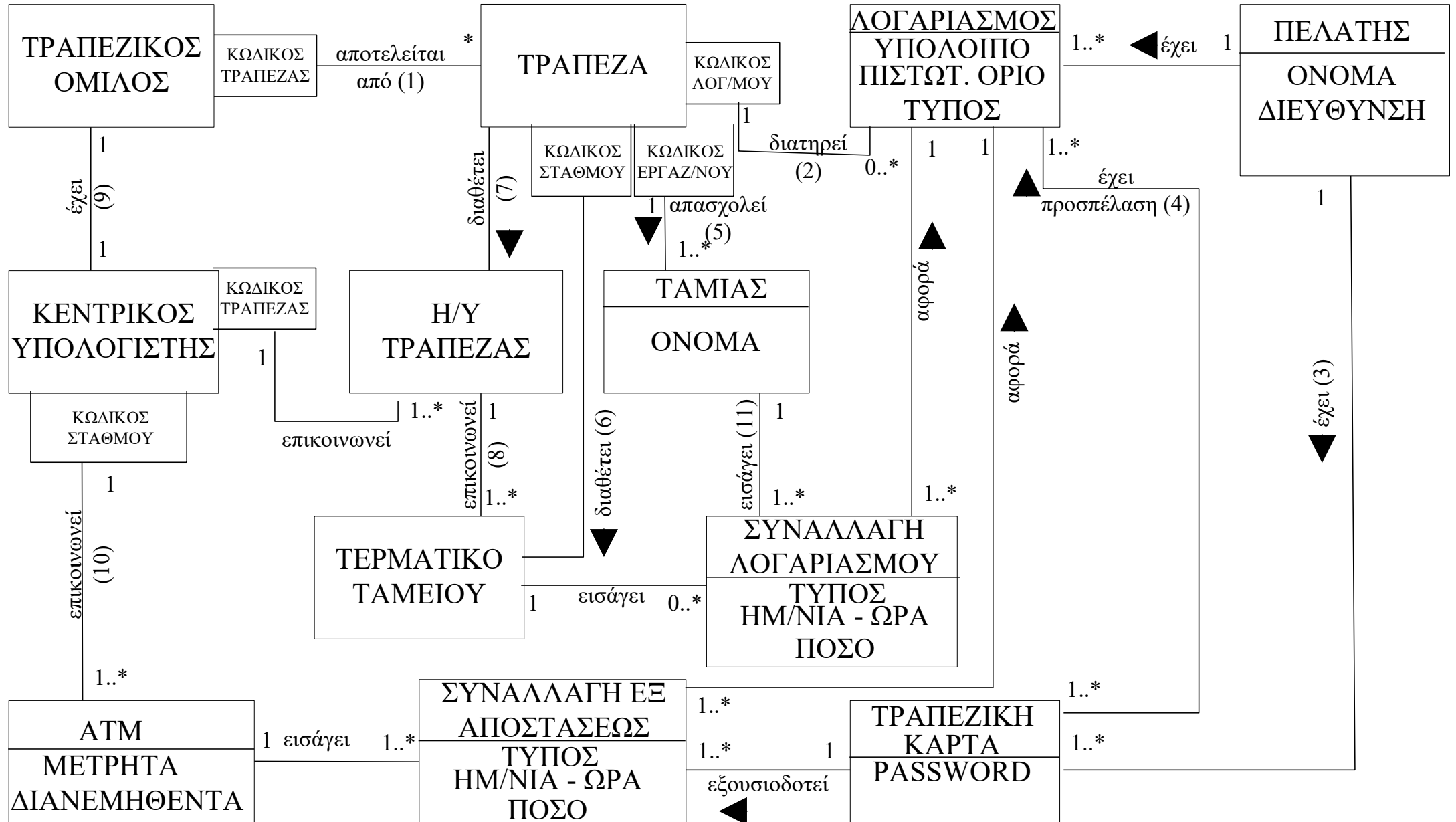
Αναγνώριση ιδιοτήτων

- **παραγόμενες ιδιότητες** πρέπει να παραλείπονται π.χ. η «ηλικία» προκύπτει από τη «ημερομηνία γέννησης»
- συχνά επιβάλλεται η αναγνώριση **ιδιοτήτων σύνδεσης**. π.χ. σε μία πολλά προς πολλά σύνδεση μεταξύ των ΜΕΤΟΧΟΣ και ΕΤΑΙΡΕΙΑ υπάρχει η ιδιότητας σύνδεσης «αριθμός μετοχών»

Αναγνώριση ιδιοτήτων

Απαλείφουμε μη απαραίτητες και λάθος ιδιότητες με βάση τα εξής κριτήρια:

- αν έχουν επιλεγεί ως αντικείμενα
- αν είναι ιδιότητες σύνδεσης σε μία πολλά προς πολλά σύνδεση
- μη σχετική ιδιότητα: μία ιδιότητα που φαίνεται να είναι εντελώς διαφορετική και ότι δε σχετίζεται με τις άλλες ιδιότητες της κλάσης. Μπορεί να είναι ένδειξη ότι η κλάση πρέπει να χωριστεί σε δύο διακριτές κλάσεις.



Αντικειμενοστρεφής Ανάλυση & Σχεδίαση

- Αναγνώριση αντικειμένων και κλάσεων
- Αναγνώριση συνδέσεων μεταξύ των αντικειμένων
- Αναγνώριση ιδιοτήτων των αντικειμένων
- Οργάνωση και απλοποίηση των κλάσεων αντικειμένων με τη χρήση κληρονομικότητας

Απλοποίηση των κλάσεων αντικειμένων

Η κληρονομικότητα μπορεί να ενσωματωθεί στα διάγραμμα κλάσεων με δύο τρόπους:

- **από κάτω προς τα πάνω** με τη γενίκευση των κοινών στοιχείων δύο ή περισσότερων κλάσεων σε μία υπερκλάση.
- Ψάχνουμε για κλάσεις με παρόμοιες ιδιότητες, συνδέσεις ή λειτουργίες.

Για παράδειγμα οι κλάσεις ΣΥΝΑΛΛΑΓΗ ΛΟΓΑΡΙΑΣΜΟΥ και ΣΥΝΑΛΛΑΓΗ ΕΞ ΑΠΟΣΤΑΣΕΩΣ.

Η κληρονομικότητα μπορεί να ενσωματωθεί στα διάγραμμα κλάσεων με δύο τρόπους:

- **από κάτω προς τα πάνω** με τη γενίκευση των κοινών στοιχείων δύο ή περισσότερων κλάσεων σε μία υπερκλάση.
- Ψάχνουμε για κλάσεις με παρόμοιες ιδιότητες, συνδέσεις ή λειτουργίες.

Για παράδειγμα οι κλάσεις ΣΥΝΑΛΛΑΓΗ ΛΟΓΑΡΙΑΣΜΟΥ και ΣΥΝΑΛΛΑΓΗ ΕΞ ΑΠΟΣΤΑΣΕΩΣ.

Απλοποίηση των κλάσεων αντικειμένων

- Επιτρέπεται να προβούμε σε επανακαθορισμό κάποιων ιδιοτήτων ή ακόμη και κλάσεων αλλά δεν πρέπει αυτό να γίνεται άκριτα.
- **από πάνω προς τα κάτω** με την εξειδίκευση μιας κλάσης σε περισσότερο ειδικές κλάσεις
- Όταν υπάρχουν ιεραρχίες κλάσεων, οι συνδέσεις και οι ιδιότητες θα ορίζονται στις ειδικές κλάσεις και όχι στις υπερκλάσεις.

Άσκηση

Ο Γιώργος αποφασίζει να ανοίξει ένα μαγαζί πρόχειρου φαγητού (fast food). Γνωρίζει πως χρειάζεται ένα φούρνο μικροκυμάτων, μια τοστιέρα, ένα ψυγείο, ένα πλυντήριο πιάτων, και ορισμένα άλλα αντικείμενα, τα οποία είναι απαραίτητα για να παρέχει σωστές υπηρεσίες το μαγαζί του. Ο Γιώργος αρχίζει με την ετοιμασία του χώρου και όταν αυτός είναι έτοιμος, ειδοποιεί να του φέρουν και εγκαταστήσουν τα επί μέρους αντικείμενα συσκευές που απαρτίζουν το fast food. Μια ανακοίνωση στην είσοδο κάτω από την πινακίδα *fast food* ο Γιώργος γράφει «Ζητούνται υπάλληλοι».

Μετά από μερικές μέρες το μαγαζί είναι έτοιμο, και ο Γιώργος με τους δύο υπαλλήλους που ήδη προσέλαβε, τη Μαίρη και το Νίκο, είναι έτοιμοι να υποδεχτούν τον πρώτο τους πελάτη.

Ο Χρήστος, μπαίνει στο μαγαζί. «Ένα τοστ με τυρί ζαμπόν παρακαλώ, μια τυρόπιτα και μια παγωμένη ΗΒΗ». Ο Γιώργος λέει στη Μαίρη «Φτιάξε σε παρακαλώ το τοστ», και αυτός αναλαμβάνει την τυρόπιτα. Διαπιστώνει όμως πως είναι κρύα και αποφασίζει να χρησιμοποιήσει το φούρνο μικροκυμάτων.

Μετά από λίγο ο Χρήστος πήρε μια σακούλα και αφού πλήρωσε βγήκε από το μαγαζί.

Πιθανή λύση

| Κλάση | Στιγμιότυπο |
|----------------------|--|
| Ιδιοκτήτης | Γιώργος |
| fast food | «fast food ο Γιώργος» |
| Υπάλληλος | Μαίρη, Νίκος |
| Πελάτης | Χρήστος, Κώστας |
| Τοστιέρα | οι δύο τοστιέρες του μαγαζιού |
| Φούρνος μικροκυμάτων | ο φούρνος του μαγαζιού |
| Πρόσωπο | Γιώργος, Μαίρη, Νίκος, Χρήστος, Κώστας |

Παράδειγμα

```
#include <iostream.h>

class Cat
{ public:
    int itsAge;
    int itsWeight; };

void main() {
    Cat Frisky;
    Frisky.itsAge = 5;
    cout << "Frisky is a cat who is " ;
    cout << Frisky.itsAge << " years old.\n";}
```


Παράδειγμα

```
#include <iostream>

class CRectangle {
    int x, y;
public:
    void set_values (int,int);
    int area () {return (x*y);}
};

void CRectangle::set_values (int a, int b) {
    x = a;
    y = b;
}
```

Παράδειγμα

```
int main () {  
    CRectangle rect;  
    rect.set_values (3,4);  
    cout << "area: " << rect.area();  
    return 0;  
}
```