

# Εισαγωγή στον Αντικειμενοστρεφή Προγραμματισμό

#1

Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Πανεπιστήμιο Ιωαννίνων (Άρτα)

Γκόγκος Χρήστος

# Δομημένος προγραμματισμός και αντικειμενοστρεφής προγραμματισμός

- Δομημένος προγραμματισμός (διαδικασιακός προγραμματισμός)
  - Ο προγραμματισμός στηρίζεται σε καλά ορισμένες δομές ελέγχου
    - Δομή ακολουθίας, δομές ελέγχου (if, switch), δομές επανάληψης (π.χ. while), εκφράσεις και εκχωρήσεις
    - Δεδομένα (μεταβλητές, πίνακες, εγγραφές) είναι ξεχωριστά από τις λειτουργίες σε αυτά
- Αντικειμενοστρεφής προγραμματισμός
  - Αναπτύχθηκε ως μια επέκταση του δομημένου προγραμματισμού
  - Ο αντικειμενοστρεφής προγραμματισμός βασίζεται στην έννοια του **αντικειμένου**
    - Τα αντικείμενα ομαδοποιούν δεδομένα και τις λειτουργίες πάνω σε αυτά τα δεδομένα
    - Επιτρέπουν την **απόκρυψη πληροφορίας** (information hiding) που οδηγεί σε προγράμματα που είναι ευκολότερο να αναπτυχθούν και να συντηρηθούν

# Βασικές έννοιες Αντικειμενοστρεφούς Προγραμματισμού

- Μια βασική έννοια των αντικειμενοστρεφών γλωσσών είναι η **ενθυλάκωση** δεδομένων και συναρτήσεων μαζί σε μονάδες που ονομάζονται **αντικείμενα**.
- Ένα αντικείμενο αποτελείται από:
  - **Όνομα**: αποτελεί τον τρόπο με τον οποίο γίνεται η αναφορά στα αντικείμενα μέσα στο πρόγραμμα.
  - **Μέλη δεδομένα (member data)**: τα δεδομένα που περιέχονται σε ένα αντικείμενο.
  - **Μέλη συναρτήσεις (member functions)**: συναρτήσεις που επιδρούν στα δεδομένα του αντικειμένου.
  - **Διεπαφή (interface)**: Καθορίζει τους τρόπους με τους οποίους ο προγραμματιστής μπορεί απευθείας να προσπελαύνει μέλη δεδομένα και μέλη συναρτήσεις ενός αντικειμένου.

# Κλάσεις

- Η **κλάση** είναι μια ακόμα θεμελιώδης έννοια στον αντικειμενοστρεφή προγραμματισμό και μπορεί να περιγραφεί ως το «αρχιτεκτονικό σχέδιο» που προσδιορίζει ένα νέο τύπο αντικειμένων.
- Η κλάση καθορίζει:
  - τα δεδομένα, τις συναρτήσεις και τη διεπαφή των αντικειμένων της κλάσης.
  - το πως τα αντικείμενα μιας κλάσης συμπεριφέρονται παρέχοντας κώδικα που υλοποιεί τις συναρτήσεις που σχετίζονται με την κλάση.
- Ο προγραμματιστής μπορεί να δημιουργεί ένα ή περισσότερα αντικείμενα μιας κλάσης

# Πως ορίζεται και χρησιμοποιείται μια κλάση σε ένα πρόγραμμα;

- **Δήλωση της κλάσης:** επιλογή του τι θα αποθηκεύουν (μέλη δεδομένων) και πως θα συμπεριφέρονται (μέλη συναρτήσεων) τα αντικείμενα της κλάσης.
- **Ορισμός των μελών συναρτήσεων:** παρέχεται υλοποίηση για κάθε μέλος συνάρτηση της κλάσης.
- **Χρήση της κλάσης για τη δημιουργία αντικειμένων:** δήλωση νέων στιγμιοτύπων αντικειμένων της κλάσης όπως δηλώνονται και οι απλές μεταβλητές.

# Παράδειγμα: Δήλωση κλάσης

```
class Circle {  
public:                                // διεπαφή (interface)  
    void SetRadius(double r); // θέτει το μέλος δεδομένων radius στην τιμή r  
    double AreaOf();          // επιστρέφει το εμβαδόν του κύκλου  
private:  
    double radius; // ακτίνα του κύκλου  
};
```

# Ορισμός των συναρτήσεων μελών

- Υπάρχουν 2 τρόποι με τους οποίους μπορεί να καθοριστεί ο κώδικας που περιέχουν οι συναρτήσεις μέλη μιας κλάσης:
  - Μέσα στη δήλωση της κλάσης
  - Μετά τη δήλωση της κλάσης
- Η αναφορά σε μια συνάρτηση μέλος γίνεται ως εξής:  
**όνομαΚλάσης::όνομαΣυνάρτησηςΜέλους**
- Το αναγνωριστικό **όνομαΚλάσης::όνομαΣυνάρτησηςΜέλους** αναφέρεται στη συνάρτηση μέλος **όνομαΣυνάρτησηςΜέλους** της κλάσης **όνομαΚλάσης**
- Ο τελεστής :: ονομάζεται τελετής προσδιορισμού εμβέλειας (scope resolution operator)
- Μετά τη δήλωση της κλάσης, οι συναρτήσεις μέλη ορίζονται όπως οποιαδήποτε άλλη συνάρτηση.

# Παράδειγμα: ορισμός υλοποιήσεων συναρτήσεων μελών εκτός της κλάσης

```
class Circle
{
public:                                // διεπαφή (interface)
    void SetRadius(double r); // θέτει το μέλος δεδομένων radius στην τιμή r
    double AreaOf();          // επιστρέφει το εμβαδόν του κύκλου
private:
    double radius; // ακτίνα του κύκλου
};

// υλοποιήσεις συναρτήσεων μελών
void Circle::SetRadius(double r) {
    radius = r;
}
double Circle::AreaOf() {
    return (3.14 * radius * radius);
}
```



# Αντικείμενα

- Από τη στιγμή που μια κλάση έχει δηλωθεί και οριστεί, μπορούν να δηλώνονται και να χρησιμοποιούνται αντικείμενα της κλάσης όπως οποιοσδήποτε άλλος τύπος δεδομένων
- Ο προγραμματιστής μπορεί να δηλώνει ένα αντικείμενο με τον ακόλουθο τρόπο:
  - **ΌνομαΚλάσης όνομαΑντικειμένου;**
- Η παραπάνω δήλωση δημιουργεί ένα αντικείμενο βάσει των «οδηγιών» που περιέχονται στην κλάση **ΌνομαΚλάσης** και το αντικείμενο μπορεί να αναφερθεί με το αναγνωριστικό **όνομαΑντικειμένου**
- Ο τελεστής . (dot operator) μπορεί να χρησιμοποιηθεί για να προσπελαστούν τα δημόσια μέλη ενός αντικειμένου.
- Η μορφή με την οποία γίνεται η αναφορά στα μέλη ενός αντικειμένου είναι:
  - **όνομαΑντικειμένου.ΜέλοςΣυνάρτηση()**
  - **όνομαΑντικειμένου.ΜέλοςΔεδομένων**

# Αντικείμενα

```
int main() {  
    Circle C1;  
    Circle C2;  
  
    C1.SetRadius(1);  
    C2.SetRadius(10);  
    cout<<"The area of C1 is "<<C1.AreaOf()<<endl;  
    cout<<"The area of C2 is "<<C2.AreaOf()<<endl;  
  
    C1.SetRadius(2);  
    C2.SetRadius(3);  
    cout<<"The area of C1 is "<<C1.AreaOf()<<endl;  
    cout<<"The area of C2 is "<<C2.AreaOf()<<endl;  
    return 0;  
}
```

The area of C1 is 3.14  
The area of C2 is 314  
The area of C1 is 12.56  
The area of C2 is 28.26

# Ο κώδικας συνολικά

- Ο κώδικας του `sample1.cpp`
  - Δηλώνει την κλάση `Circle` και ορίζει τα μέλη της και τη διεπαφή της.
  - Ορίζει την υλοποίηση των συναρτήσεων μελών της κλάσης `Circle`.
  - Δηλώνει 2 αντικείμενα της κλάσης `Circle` με ονόματα `C1` και `C2`.
  - Χρησιμοποιεί τις διεπαφές των `C1` και `C2` για να αποθηκεύσει τις ακτίνες των 2 κύκλων και στη συνέχεια για να υπολογίσει το εμβαδό τους.

<https://github.com/chgogos/oop/blob/master/variou/COP3330/lect1/sample1.cpp>

<https://github.com/chgogos/oop/tree/master/variou/COP3330/lect1/sample1> έκδοση με διαμέριση κώδικα

# Σύνοψη

- Ένα αντικείμενο είναι μια μονάδα που ενθυλακώνει δεδομένα και συναρτήσεις. Έχει 4 στοιχεία: όνομα, μέλη δεδομένα, μέλη συναρτήσεις και διεπαφή.
- Μια κλάση καθορίζει την ορισμένη από το χρήστη μορφή των αντικειμένων.
- Η χρήση των αντικειμένων σε ένα C++ πρόγραμμα ακολουθεί τη σειρά: δήλωση, ορισμός και χρήση.
- Ο τελεστής :: χρησιμοποιείται έτσι ώστε να οριστούν οι συναρτήσεις μιας κλάσης εκτός της κλάσης.
- Ο τελεστής . χρησιμοποιείται για να κληθεί μια συνάρτηση μέλος ή να προσπελαστούν τα μέλη δεδομένων ενός αντικειμένου.

# Αναφορές

- <http://www.cs.fsu.edu/~xyuan/cop3330/>