



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Αγρονόμων Τοπογράφων Μηχανικών

Προγραμματιστικές τεχνικές

Βασίλειος Βεσκούκης
Δρ. Ηλεκτρολόγος Μηχανικός &
Μηχανικός Υπολογιστών ΕΜΠ
v.vescoukis@cs.ntua.gr

Ρωμύλος Κορακίτης
Αστροφυσικός
Αναπλ. Καθηγητής ΕΜΠ
romylos@survey.ntua.gr

**Κλάσεις: constructors, δηλώσεις,
παραδείγματα**

Κλάσεις

Σύνθετες, οριζόμενες από το χρήστη δομές, οι οποίες περιέχουν

- Δεδομένα και
- Συναρτήσεις

Τα δεδομένα αντιστοιχούν σε

- Ιδιότητες (πεδία, fields, attributes)

...ενώ οι συναρτήσεις περιγράφουν τη

- Συμπεριφορά (μεθόδους, methods, member functions)

..των οντοτήτων που παριστάνει μια κλάση, τα μέλη των οποίων λέγονται "αντικείμενα"

- object-oriented programming = αντικειμενοστρεφής προγραμματισμός

Οι κλάσεις μπορούν να ιδωθούν:

- Ως προγραμματιστικό εργαλείο
- Ως εργαλείο μοντελοποίησης δεδομένων

Παραδείγματα κλάσεων

Ένα σημείο

- Δεδομένα: x, y, id
- Συμπεριφορά: νέο_σημείο()

Μια γραμμή

- Δεδομένα: x_1, y_1, x_2, y_2, id
- Συμπεριφορά: νέα_γραμμή(), μήκος(), χάραξη()

Ένα τρίγωνο

- Δεδομένα: κορυφή 1, κορυφή 2, κορυφή 3, id
- Συμπεριφορά: νέο_τρίγωνο(), εμβαδόν(), περίμετρος()

Ένας μιγαδικός αριθμός

- Δεδομένα: πραγματικό μέρος, φανταστικό μέρος
- Συμπεριφορά: νέος_αριθμός(), πρόσθεση(), αφαίρεση()

Ένα πολύγωνο

- Δεδομένα: πλήθος κορυφών, συντεταγμένες κορυφών, id
- Συμπεριφορά: εμβαδόν(), περίμετρος(), κέντρο_βάρους()

Δρ. Βασίλειος Βεσκούκης

Ορισμός κλάσεων

Τρόπος 1

```
class [όνομα κλάσης]
{
    Προσδιοριστής ορατότητας:
    Δήλωση μεταβλητών μνήμης (πεδίων)
    Δήλωση και υλοποίηση συναρτήσεων (μεθόδων)
};

class line { //start of class declaration
public:
    float x1, y1, x2, y2;
    float len;

    float linelen() {
        len=sqrt(pow((x1-x2),2)+pow((y1-y2),2));
        return len;
    }

}; //end of class declaration
```

Δρ. Βασίλειος Βεσκούκης

Ορισμός κλάσεων

Τρόπος 1

```
class [όνομα κλάσης]
{
    Προσδιοριστής ορατότητας:
    Δήλωση μεταβλητών μνήμης (πεδίων)
    Δήλωση και υλοποίηση συναρτήσεων (μεθόδων)
};

class line { //start of class declaration
public:
    float x1, y1, x2, y2;
    float len;

    float linelen() {
        len=sqrt(pow((x1-x2),2)+pow((y1-y2),2));
        return len;
    }

}; //end of class declaration
```

Δρ. Βασίλειος Βεσκούκης

Ορισμός κλάσεων

Τρόπος 2

```
class [όνομα κλάσης]
{
    Προσδιοριστής ορατότητας:
    Δήλωση μεταβλητών μνήμης (πεδίων)
    Δήλωση συναρτήσεων (μεθόδων)
};
Υλοποίηση μεθόδων (χρήση του τελεστή ::)
class line { //start of class declaration
public:
    float x1, y1, x2, y2;
    float len;

    float linelen(); // There is some method called "linelen()"...
}; //end of class declaration
float line::linelen() { // here is "linelen()"s definition
    len=sqrt(pow((x1-x2),2)+pow((y1-y2),2));
    return len;
}
```

Δρ. Βασίλειος Βεσκούκης

Ορισμός κλάσεων

Τρόπος 2

```
class [όνομα κλάσης]
{
    Προσδιοριστής ορατότητας:
    Δήλωση μεταβλητών μνήμης (πεδίων)
    Δήλωση συναρτήσεων (μεθόδων)
};

Υλοποίηση μεθόδων (χρήση του τελεστή ::)
class line { //start of class declaration

public:
    float x1, y1, x2, y2;
    float len;

    float linelen(); // There is some method called "linelen()"...

}; //end of class declaration

float line::linelen() { // here is "linelen()" 's definition
    len=sqrt(pow((x1-x2),2)+pow((y1-y2),2));
    return len;
}
```

Δρ. Βασίλειος Βεσκούκης

Ορισμός κλάσεων

Γενικά

```
class [όνομα κλάσης]
{
    Προσδιοριστής ορατότητας 1:
    Δήλωση μεταβλητών μνήμης (πεδίων)
    Δήλωση/υλοποίηση συναρτήσεων (μεθόδων)
    Προσδιοριστής ορατότητας 2:
    Δήλωση μεταβλητών μνήμης (πεδίων)
    Δήλωση/υλοποίηση συναρτήσεων (μεθόδων)
    ...
};

Υλοποίηση μεθόδων (χρήση του τελεστή ::)
```

Δρ. Βασίλειος Βεσκούκης

Ορισμός κλάσεων

Γενικά (παράδειγμα)

```
class [όνομα κλάσης]
{
public:
    Δήλωση ορατών από έξω μεταβλητών μνήμης (πεδίων)
    Δήλωση ορατών από έξω συναρτήσεων (μεθόδων)
private:
    Δήλωση μη ορατών από έξω μεταβλητών μνήμης (πεδίων)
    Δήλωση/υλοποίηση μη ορατών από έξω συναρτήσεων (μεθόδων)
};
Υλοποίηση μεθόδων (χρήση του τελεστή ::)
```

Δρ. Βασίλειος Βεσκούκης

Ορισμός κλάσεων

```
class [όνομα κλάσης]
{
public:
    Δήλωση ορατών από έξω μεταβλητών μνήμης (πεδίων)
    Δήλωση ορατών από έξω συναρτήσεων (μεθόδων)
private:
    Δήλωση μη ορατών από έξω μεταβλητών μνήμης (πεδίων)
    Δήλωση/υλοποίηση μη ορατών από έξω συναρτήσεων (μεθόδων)
};
Υλοποίηση μεθόδων (χρήση του τελεστή ::)

class line { //start of class declaration
public:
    float x1, y1, x2, y2;
    float len;
private:
    float linelen(); // There is some method called "linelen()"...
}; //end of class declaration

float line::linelen() { // here is "linelen()"s definition
    len=sqrt(pow((x1-x2),2)+pow((y1-y2),2));
    return len;
}
```

Δρ. Βασίλειος Βεσκούκης

Χαρακτηριστικές μέθοδοι κλάσεων

Constructors

- Μέθοδοι που εκτελούνται αυτόματα με τη δήλωση ενός αντικειμένου της κλάσης
- Έχουν υποχρεωτικά το όνομα της κλάσης και ΔΕΝ έχουν τύπο
- Μπορεί να είναι περισσότερες του ενός, αλλά πρέπει να διαφέρουν ως προς τις παραμέτρους που δέχονται

Destructors

- Μέθοδοι που εκτελούνται αυτόματα με την καταστροφή ενός αντικειμένου της κλάσης
- Έχουν το όνομα της κλάσης, με το "~" στην αρχή του
- Κάθε κλάση έχει μόνο έναν destructor

Ισχυρότατο εργαλείο προγραμματισμού!

Δρ. Βασίλειος Βεσκούκης

Constructors: παράδειγμα

Μια γραμμή ως κλάση

Δεδομένα:

- x_1, y_1, x_2, y_2, len : πραγματικοί αριθμοί

Συμπεριφορά:

- `νέα_γραμμή()`
 - Διάβασε όλα τα δεδομένα της γραμμής από το πληκτρολόγιο
 - Υπολόγισε το μήκος
- `μήκος()`
 - Υπολόγισε το μήκος και ενημέρωσε το πεδίο `len`
- `νέα_γραμμή(σΧ1, σΥ1, σΧ2, σΥ2)`
 - Τα δεδομένα της γραμμής δίνονται ως παράμετροι
 - Υπολόγισε το μήκος
- `νέα_γραμμή(σΧ2, σΥ2)`
 - Δίνεται το δεύτερο άκρο ως παράμετροι, και ΘΕΩΡΟΥΜΕ ότι το πρώτο είναι το (0,0)
 - Υπολόγισε το μήκος

Δρ. Βασίλειος Βεσκούκης

Constructors: παράδειγμα

```
class line { // *** Start of class declaration
public: // "public" means that everything is visible from outside
    // Fields declaration
    float x1, y1, x2, y2;
    float len;

    // Method to calculate the length
    void linelen() {
        len=sqrt(pow((x1-x2),2)+pow((y1-y2),2));
    }
    // Constructor 1: all four coordinates are given
    line(float px1, float py1, float px2, float py2) {
        x1=px1;
        y1=py1;
        x2=px2;
        y2=py2;
        linelen();
    }
    // Constructor 2: point 1 coordinates ASSUMED to be (0,0),
    // point 2 coordinates are given
    line(float px2, float py2) {
        x1=0;
        y1=0;
        x2=px2;
        y2=py2;
        linelen();
    }
    // Constructor 3: No coordinates are given.
    // Will be read from the console
    line() {
        cout<<endl<<"New line"<<endl;
        cout<<"X1:"<< cin>>x1;
        cout<<"Y1:"<< cin>>y1;
        cout<<"X2:"<< cin>>x2;
        cout<<"Y2:"<< cin>>y2;
        linelen();
    }
}; // *** End of class declaration
```

Δρ. Βασίλειος Βεσκούκης

Παράδειγμα

```
class line { // *** Start of class declaration
public: // Fields declaration
    float x1, y1, x2, y2;
    float len;
    // Methods declaration. Will be implemented outside the class.
    void linelen();
    line(float px1, float py1, float px2, float py2);
    line(float px2, float py2);
    line();
}; // *** End of class declaration

// *** definition of methods. notice the use of "class:." ***

void line::linelen() { // Method to calculate the length
    len=sqrt(pow((x1-x2),2)+pow((y1-y2),2));
}

line::line(float px1, float py1, float px2, float py2) { // Constructor 1
    x1=px1;
    y1=py1;
    x2=px2;
    y2=py2;
    linelen();
}

line::line(float px2, float py2) { // Constructor 2
    x1=0;
    y1=0;
    x2=px2;
    y2=py2;
    linelen();
}

line::line() { // Constructor 3
    cout<<endl<<"New line"<<endl;
    cout<<"X1:"<< cin>>x1;
    cout<<"Y1:"<< cin>>y1;
    cout<<"X2:"<< cin>>x2;
    cout<<"Y2:"<< cin>>y2;
    linelen();
}
```

Δρ. Βασίλειος Βεσκούκης

Γιατί είναι χρήσιμες οι κλάσεις;;;

Στην ανάλυση προβλημάτων

- Κατάλληλες για παράσταση σύνθετων εννοιών

Στην υλοποίηση

- Όταν ολοκληρωθεί ο ορισμός μιας κλάσης, η προσοχή μας αφιερώνεται στο πραγματικό πρόβλημα και όχι στα τεχνικά σημεία

Στον προγραμματισμό περιβαλλόντων με γραφικά

- ΟΛΑ ανεξαιρέτως απαιτούν αντικειμενοστρεφή προγραμματισμό

Στην επαναχρησιμοποίηση έτοιμων λύσεων

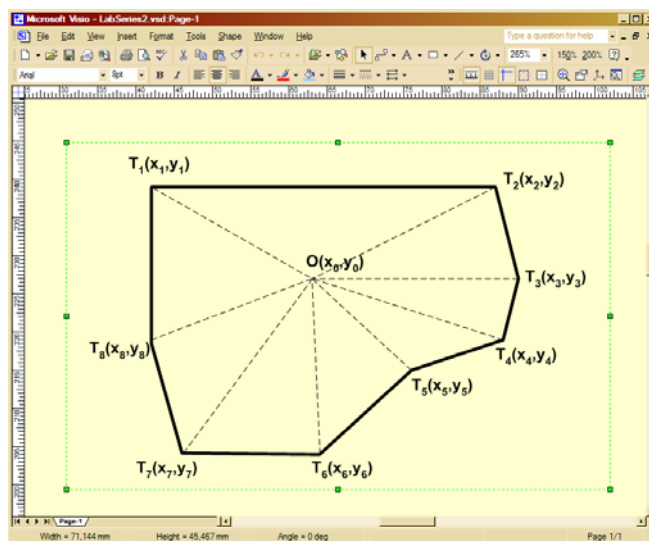
- Βρίσκουμε έτοιμες βιβλιοθήκες για "τα πάντα" και δεν χρειάζεται να ξέρουμε να προγραμματίζουμε αυτά που υλοποιούν οι βιβλιοθήκες: απλά τα χρησιμοποιούμε αμέσως

Στην επέκταση έτοιμων λύσεων

- Προσθέτουμε δικά μας στοιχεία συμπεριφοράς σε έτοιμα προγράμματα (πχ: arcview, autocad, msword, κλπ)

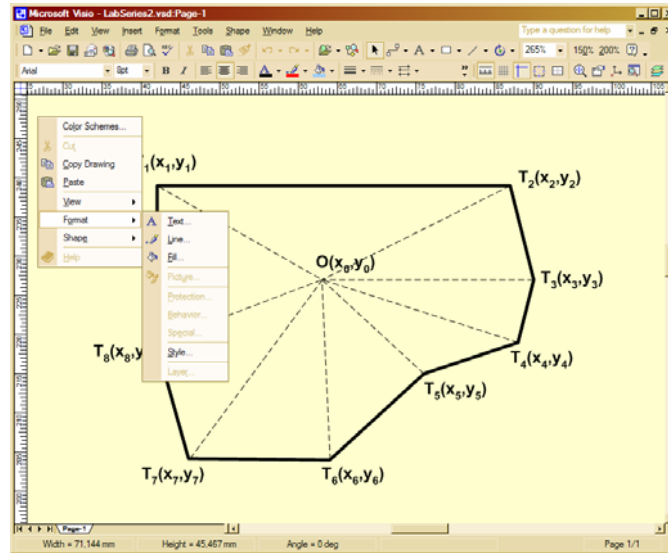
Δρ. Βασίλειος Βεσκούκης

Παράδειγμα



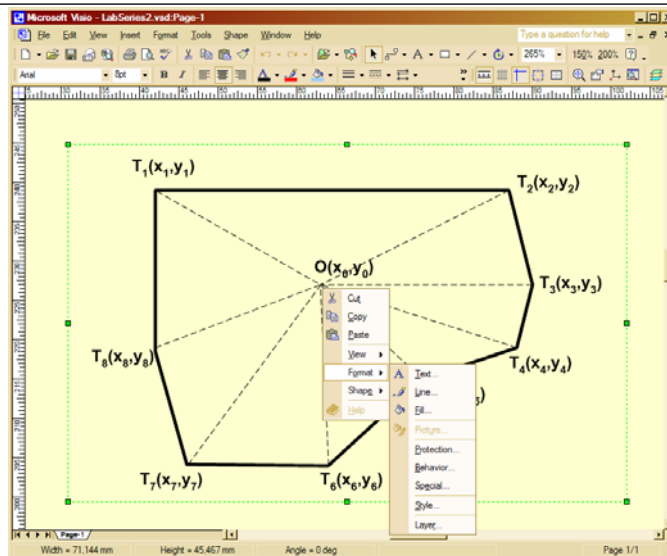
Δρ. Βασίλειος Βεσκούκης

Παράδειγμα



Δρ. Βασίλειος Βεσκούκης

Παράδειγμα



Δρ. Βασίλειος Βεσκούκης

Παραδείγματα struct και κλάσεων

Ορισμός struct που παριστάνει πολύγωνο

- α) για το πλήθος των κορυφών του πολυγώνου (ακέραιος αριθμός που δεν μπορεί να είναι μεγαλύτερος από έναν αριθμό MAXK που δηλώνεται ως σταθερά στην αρχή του προγράμματος).
- β) για τις συντεταγμένες x και y των κορυφών του πολυγώνου (δύο μονοδιάστατοι πίνακες πραγματικών αριθμών με διάσταση MAXK)
- γ) το εμβαδό και την περίμετρο του πολυγώνου (πραγματικοί αριθμοί)
- δ) τις συντεταγμένες ενός εσωτερικού σημείου του πολυγώνου (πραγματικοί αριθμοί)

```
const int MAXK = 100;  
struct polygon {  
    int K;  
}
```

Δρ. Βασίλειος Βεσκούκης

Παραδείγματα struct και κλάσεων

Ορισμός struct που παριστάνει πολύγωνο

- α) για το πλήθος των κορυφών του πολυγώνου (ακέραιος αριθμός που δεν μπορεί να είναι μεγαλύτερος από έναν αριθμό MAXK που δηλώνεται ως σταθερά στην αρχή του προγράμματος).
- β) για τις συντεταγμένες x και y των κορυφών του πολυγώνου (δύο μονοδιάστατοι πίνακες πραγματικών αριθμών με διάσταση MAXK)
- γ) το εμβαδό και την περίμετρο του πολυγώνου (πραγματικοί αριθμοί)
- δ) τις συντεταγμένες ενός εσωτερικού σημείου του πολυγώνου (πραγματικοί αριθμοί)

```
const int MAXK = 100;  
struct polygon {  
    int K;  
    float x[MAXK], y[MAXK];  
}
```

Δρ. Βασίλειος Βεσκούκης

Παραδείγματα struct και κλάσεων

Ορισμός struct που παριστάνει πολύγωνο

- α) για το πλήθος των κορυφών του πολυγώνου (ακέραιος αριθμός που δεν μπορεί να είναι μεγαλύτερος από έναν αριθμό MAXK που δηλώνεται ως σταθερά στην αρχή του προγράμματος).
- β) για τις συντεταγμένες x και y των κορυφών του πολυγώνου (δύο μονοδιάστατοι πίνακες πραγματικών αριθμών με διάσταση MAXK)
- γ) το εμβαδό και την περίμετρο του πολυγώνου (πραγματικοί αριθμοί)
- δ) τις συντεταγμένες ενός εσωτερικού σημείου του πολυγώνου (πραγματικοί αριθμοί)

```
const int MAXK = 100;
struct polygon {
    int K;
    float x[MAXK], y[MAXK];
    float Area, Perimeter;
}
```

Δρ. Βασίλειος Βεσκούκης

Παραδείγματα struct και κλάσεων

Ορισμός struct που παριστάνει πολύγωνο

- α) για το πλήθος των κορυφών του πολυγώνου (ακέραιος αριθμός που δεν μπορεί να είναι μεγαλύτερος από έναν αριθμό MAXK που δηλώνεται ως σταθερά στην αρχή του προγράμματος).
- β) για τις συντεταγμένες x και y των κορυφών του πολυγώνου (δύο μονοδιάστατοι πίνακες πραγματικών αριθμών με διάσταση MAXK)
- γ) το εμβαδό και την περίμετρο του πολυγώνου (πραγματικοί αριθμοί)
- δ) τις συντεταγμένες ενός εσωτερικού σημείου του πολυγώνου (πραγματικοί αριθμοί)

```
const int MAXK = 100;
struct polygon {
    int K;
    float x[MAXK], y[MAXK];
    float Area, Perimeter;
    float Xin, Yin;
}
```

Δρ. Βασίλειος Βεσκούκης

Κατασκευάστε τη συνάρτηση **init_polygon(polygon p)** η οποία δέχεται ως όρισμα μια μεταβλητή p τύπου polygon και διαβάζει από το αρχείο polygon.txt τις συντεταγμένες του εσωτερικού σημείου και των κορυφών του πολυγώνου, τις οποίες και τοποθετεί στα αντίστοιχα πεδία του struct p.

Προσέξτε τη σωστή ενημέρωση του πεδίου που αντιστοιχεί στο πλήθος των κορυφών του πολυγώνου.

Να κάνετε την **παραδοχή** ότι στο αρχείο polygon.txt δεν περιέχονται περισσότερες από MAXK κορυφές.

Στη συνέχεια κατασκευάστε τη συνάρτηση **polygon_data(polygon p)** η οποία δέχεται επίσης ως όρισμα μια μεταβλητή p τύπου polygon και εμφανίζει στην οθόνη τις συντεταγμένες των κορυφών του πολυγώνου.