

Αντικειμενοστραφής Προγραμματισμός

Μέρος 1ο: Δομές και κλάσεις

Εξάμηνο Σπουδών: 6ο

Κωδικός Μαθήματος: 647

Τμήμα Μαθηματικών
Πανεπιστήμιο Ιωαννίνων

Μιχάλης Α. Μπέκος

bekos@uoi.gr

Περιεχόμενα

- Δομές
 - Ορισμός
 - Δομές ως ορίσματα συναρτήσεων
 - Αρχικοποίηση δομών
- Κλάσεις
 - Ορισμός, συναρτήσεις μέλη
 - Δημόσια και ιδιωτικά μέλη
 - Συναρτήσεις μετάλλαξης και προσπέλασης
 - Δομές έναντι κλάσεων

Μέρος 1^ο:
Δομές

Δομές

- 2ος συγκεντρωτικός τύπος δεδομένων: `struct`
- Συγκεντρωτικός με την έννοια της “ομαδοποίησης”
 - Π.χ. ο πίνακας είναι συλλογή τιμών του ίδιου τύπου
 - Η δομή είναι συλλογή τιμών διαφορετικών τύπων
- Αντιμετωπίζεται ως ένα μεμονωμένο στοιχείο, όπως οι πίνακες
- Σημαντική διαφορά: Η δομή πρέπει προηγουμένως να “οριστεί”
 - Πριν από τη δήλωση οποιωνδήποτε μεταβλητών

Ορισμός δομής

- Ο ορισμός μιας δομής γίνεται καθολικά (συνήθως).
- Δεν δεσμεύεται μνήμη
 - Περιγράφει απλά το πώς θα “μοιάζει” η δομή
- Παράδειγμα:
 - Όνομα της δομής: CDAccount
 - Ονόματα των μελών της: balance, interestRate, term

```
//Structure for a certificate of deposit account
struct CDAccount
{
    double balance;           //balance
    double interestRate;     //interest rate
    int term;                 //months until maturity
};
```

Δήλωση μεταβλητών της δομής

- Μετά τον ορισμό της δομής, μπορούμε να δηλώσουμε μεταβλητές αυτού του νέου τύπου
- Ακριβώς όπως με τη δήλωση απλών τύπων δεδομένων
- Παράδειγμα:
 - Η μεταβλητή `account` είναι τύπου `CDAccount`
 - Περιέχει “τιμές μελών”
 - Μία τιμή για καθένα από τα μέλη της δομής

```
int main( )
{
    CDAccount account;           //Setup an account
    account.balance = 100;       //Initialize the balance
    account.interestRate = 1.32; //Fix the interest rate
    account.term = 48;           //Set 48 months till maturity
}
```

Πρόσβαση στα μέλη της δομής

- Ο τελεστής τελεία `.` παρέχει πρόσβαση στα μέλη της δομής
 - `account.balance`
 - `account.interestRate`
 - `account.term`
- Τα μέλη της δομής ονομάζονται **μεταβλητές μέλη**
 - και αποτελούν τα “μέρη” της μεταβλητής δομής
- Διαφορετικές δομές μπορούν να έχουν μεταβλητές μέλους ίδιου ονόματος
 - Χωρίς συγκρούσεις

Hands-on ενότητα 1
Υλοποίηση της δομής CDAccount

Ένα σύνηθες σφάλμα

- Ερωτηματικό μετά τον ορισμό της δομής
 - Πρέπει απαραίτητα να υπάρχει
 - Απαιτείται δεδομένου ότι μπορείτε να δηλώσετε μεταβλητές δομής σε αυτό το σημείο
 - Αν ξεχαστεί τότε εμφανίζεται σφάλμα στον μεταγλωττιστή

```
//Structure for weather data
struct WeatherData
{
    double latitude;
    double longitude;
    double temperature;
    double windVelocity;
}; // <- required semicolon!
```

Αναθέσεις

- Δοθείσας μιας δομής με όνομα `CropYield`
- Δηλώνουμε δύο μεταβλητές:
 - `CropYield apples, oranges;`
- Και οι δύο είναι μεταβλητές είναι τύπου `CropYield`
- Απλές αναθέσεις, όπως η παρακάτω, είναι νόμιμες:
 - `apples = oranges;`
- Αντιγράφεται κάθε μεταβλητή μέλος από τη μια μεταβλητή στην άλλη.

Δομές ως ορίσματα συναρτήσεων

- Υποστηρίζονται όπως κάθε απλός τύπος δεδομένων
 - με τιμή
 - με αναφορά
 - ή συνδυασμός τους
- Μία δομή μπορεί επίσης να επιστραφεί από συνάρτηση
 - Ο τύπος επιστροφής είναι ο τύπος της δομής
 - Η επιστροφή (return) “στέλνει” τη δομή στο σημείο από το οποίο κλήθηκε η συνάρτηση.

```
void getData(CDAccount& theAccount) {  
    cout << "Enter account balance: $";  
    cin >> theAccount.balance;  
    cout << "Enter account interest rate: ";  
    cin >> theAccount.interestRate;  
    cout << "Enter the number of months until maturity: ";  
    cin >> theAccount.term;  
}
```

Αρχικοποίηση δομών

- Η αρχικοποίηση μιας δομής μπορεί να γίνει με διαφορετικούς τρόπους
- Υπάρχει και η δυνατότητα αρχικοποίησης κατά τη δήλωση
- Παράδειγμα:

```
//Structure for dates
struct Date
{
    int month;
    int day;
    int year;
};

int main() {
    //Initialization with initial values
    Date dueDate = {12, 31, 2023};
    return 0;
}
```

Hands-on ενότητα 2
Επέκταση της δομής CDAccount

Μέρος 2^ο:
Κλάσεις

Κλάσεις

- Οι κλάσεις ορίζονται όμοια με τις δομές
 - Υποστηρίζουν επιπρόσθετα συναρτήσεις μέλη
 - Όχι μόνο δεδομένα μέλη
- Αποτελούν αναπόσπαστο μέρος του αντικειμενοστρεφούς προγραμματισμού
 - Στη C++, οι μεταβλητές τύπου κλάσης λέγονται **αντικείμενα**
 - Αντικείμενο: δεδομένα + λειτουργίες
- Στο παράδειγμα δίνεται το πρότυπο της συνάρτησης μέλους (η υλοποίηση είναι αλλού)

```
class DayOfYear // Name of new class type
{
public:
    int month;    // member variables
    int day;
    void output(); // member function
};
```

Δήλωση αντικειμένων

- Η δήλωση αντικειμένων γίνεται όπως με όλες τις μεταβλητές
 - Π.χ. βασικοί τύποι δεδομένων, τύποι δομής
- Παράδειγμα: `DayOfYear today, birthday;`
 - Δηλώνει δύο αντικείμενα τύπου `DayOfYear`
- Τα αντικείμενα περιλαμβάνουν:
 - Δεδομένα (των μεταβλητών μελών): `month`, `day`
 - Λειτουργίες (συναρτήσεις μέλη): `output()`

```
class DayOfYear // Name of new class type
{
public:
    int month;    // member variables
    int day;
    void output(); // member function
};
```


Πρόσβαση στα μέλη της κλάσης

- Η πρόσβαση στα μέλη μιας κλάσης γίνεται όπως και στις δομές
- Παραδείγματα:
 - `today.month`
 - `today.day`
- Για την πρόσβαση σε συνάρτηση μέλος:
 - `today.output()`; → Κλήση στη συνάρτηση μέλους

```
int main() {  
    DayOfYear bachBirthday;  
    bachBirthday.month = 03;  
    bachBirthday.day = 21;  
    cout << "J. S. Bach's birthday is "  
    bachBirthday.output();  
}
```

Υλοποίηση συναρτήσεων μελών

- Για κάθε συνάρτηση μέλος απαιτείται η δήλωση και η υλοποίηση της
- Η υλοποίηση γίνεται όμοια με αυτή των απλών συναρτήσεων
 - Μπορεί να γίνει μετά τον ορισμό της `main()`
 - Πρέπει να καθορίζεται η κλάση: `void DayOfYear::output() {...}`
- Ο τελεστής `::` ονομάζεται **τελεστής εμβέλειας**
 - Καθορίζει στον μεταγλωττιστή “από ποια κλάση” προέρχεται το μέλος

```
/**
 * Outputs the day of the year
 */
void DayOfYear::output()
{
    cout << day << ", " << month << endl;
}
```

Υλοποίηση συναρτήσεων μελών

- Η υλοποίηση μιας συνάρτησης μέλους συνήθως χρησιμοποιεί τις μεταβλητές μέλη της κλάσης
- Κάθε συνάρτηση μέλος είναι διαθέσιμη σε όλα τα αντικείμενα της κλάσης
 - Όταν γίνεται κλήση της, αναφέρεται στα δεδομένα του αντικειμένου
 - ... ανάλογα με την υλοποίηση
- Παράδειγμα: `today.output()`;
 - Εκτυπώνει στο τερματικό τα δεδομένα του αντικειμένου `today`

```
/**  
 * Outputs the day of the year  
 */  
void DayOfYear::output()  
{  
    cout << day << ", " << month << endl;  
}
```

Hands-on ενότητα 3

Μια πρώτη υλοποίηση της κλάσης `DayOfYear`

Ο τελεστής τελεία (.) και ο τελεστής εμβέλειας (::)

- Χρησιμοποιούνται για τον καθορισμό της “κλάσης” που είναι μέλη.
- Ο τελεστής τελεία (.):
 - Καθορίζει το μέλος ενός συγκεκριμένου αντικειμένου
 - Π.χ. `today.output()` ;
- Ο τελεστής εμβέλειας (::):
 - Καθορίζει από ποια κλάση προέρχεται ο ορισμός της συνάρτησης
 - Π.χ. `void DayOfYear::output() {...}`

Συμπερασματικά για τις κλάσεις

- Κάθε κλάση είναι ένας πλήρης τύπος
 - ακριβώς όπως οι βασικοί τύποι δεδομένων `int`, `double` κ.λ.π.
- Δημιουργούμε μεταβλητές τύπου κλάσης
 - τις οποίες ονομάζουμε απλά **αντικείμενα**
- Τα αντικείμενα μπορεί να είναι παράμετροι συναρτήσεων
 - με τιμή
 - με αναφορά
 - ή συνδυασμός τους
- Με άλλα λόγια, ο τύπος κλάσης χρησιμοποιείται όπως κάθε άλλος τύπος

Ενθυλάκωση

Σημαίνει συνένωση σε ένα

- Οποιοσδήποτε τύπος δεδομένων περιλαμβάνει:
 - Δεδομένα (τα οποία έχουν ένα εύρος)
 - Λειτουργίες (που μπορούν να εκτελεστούν σε αυτά)
- Παράδειγμα: Ο τύπος δεδομένων `int` περιλαμβάνει:
 - Δεδομένα το εύρος των οποίων είναι -2147483648 έως 2147483647 (για 32 bit ακεραίους).
 - Λειτουργίες: `+`, `-`, `*`, `/`, `%`, λογικές, κλπ.
- Το ίδιο και οι κλάσεις, αλλά εμείς ορίζουμε τα δεδομένα και τις λειτουργίες που υποστηρίζουν
- Το αντικείμενο ουσιαστικά είναι “ενθυλάκωση” συνενώνοντας σε ένα:
 - τιμές δεδομένων
 - λειτουργίες επί των δεδομένων (συναρτήσεις μέλη)

Αφηρημένοι τύποι δεδομένων

- Συντομογραφία: ADT (Abstract Data Type)
 - Συλλογή δεδομένων και βασικών λειτουργιών που ορίζονται επί των δεδομένων
 - Οι προγραμματιστές δεν γνωρίζουν τις λεπτομέρειες της υλοποίησης τους
- Οι ADTs συχνά είναι ανεξάρτητοι της γλώσσας προγραμματισμού
 - Στην C++ υλοποιούνται με κλάσεις
 - Μια κλάση στη C++ ουσιαστικά ορίζει έναν ADT
 - Και άλλες γλώσσες προγραμματισμού υλοποιούν ADTs.

Αρχές του Αντικειμενοστρεφή Προγραμματισμού


- Απόκρυψη πληροφοριών (Data hiding)
 - Λεπτομέρειες σχετικά με τον τρόπο υλοποίησης των συναρτήσεων μελών της κλάσης δεν γίνονται γνωστές ή διαθέσιμες στο “χρήστη” της κλάσης
- Αφαίρεση δεδομένων (Data abstraction)
 - Λεπτομέρειες σχετικά με τον τρόπο χειρισμού των δεδομένων εντός της κλάσης δεν είναι γνωστές ή διαθέσιμες στο “χρήστη” της κλάσης
- Ενθυλάκωση (Encapsulation)
 - Συγκέντρωση δεδομένων και λειτουργιών, αλλά διατήρηση των “λεπτομερειών” κρυφών

Δημόσια και ιδιωτικά μέλη

- Οι μεταβλητές μέλη μιας κλάσης σχεδόν πάντα είναι ιδιωτικές (εξ ορισμού) καθώς έτσι
 - αποκρύπτονται τα δεδομένα από το χρήστη (τα αντικείμενα δεν έχουν πρόσβαση)
 - επιτρέπεται ο χειρισμός τους μόνο μέσω λειτουργιών (συναρτήσεις μέλη)
 - υποστηρίζονται έτσι οι αρχές του Αντικειμενοστρεφή Προγραμματισμού
- Τα δημόσια στοιχεία είναι προσβάσιμα στο χρήστη
 - συνήθως συναρτήσεις μέλη

```
class DayOfYear
{
private:           // Private members
    int month;
    int day;
public:           // Public members
    void input();
    void output();
};
```

```
int main()
{
    DayOfYear today;
    today.month = 13; //error
    today.day = 32; //error
    cout << "Today is ";
    today.output();
}
```



Hands-on ενότητα 4

Μια πιο σωστή υλοποίηση της κλάσης DayOfYear

Συναρτήσεις μετάλλαξης και προσπέλασης

- Συναρτήσεις προσπέλασης (ή getter)
 - επιστρέφουν τις τιμές των μεταβλητών μελών
 - άρα επιτρέπουν την πρόσβαση σε αυτές
- Συναρτήσεις μετάλλαξης (ή setter)
 - αλλάζουν τις τιμές των μεταβλητών μελών
 - διαφέρουν ανάλογα με την εφαρμογή

```
class DayOfYear {
private:
    int month;
    int day;
public:
    void setDay(int newDay);           //Precondition: 1 <= newDay <= 31
    void setMonth(int newMonth);      //Precondition: 1 <= newMonth <= 12
    int getDay();                     //Returns the day
    int getMonthNumber();             //Returns 1 for January, 2 for February, etc.
};
```

Δομές έναντι κλάσεων

- Δομές
 - Συνήθως όλα τα μέλη είναι δημόσια
 - Δεν υπάρχουν συναρτήσεις μέλη
- Κλάσεις
 - Συνήθως, οι μεταβλητές μέλη είναι ιδιωτικές
 - Οι συναρτήσεις μέλη (που αποτελούν την **διεπαφή της κλάσης**) είναι δημόσιες
- Τεχνικά, αντιστοιχούν στο ίδιο concept
 - Πρακτικά, πολύ διαφορετικοί μηχανισμοί

Σκεπτόμενοι με αντικείμενα

- Η εισαγωγή των αντικειμένων άλλαξε τον προγραμματισμό
 - Πριν την εισαγωγή τους → Οι αλγόριθμοι ήταν στο επίκεντρο
 - Μετά την εισαγωγή τους → Τα δεδομένα είναι στο επίκεντρο
- Οι αλγόριθμοι εξακολουθούν να υπάρχουν
 - Απλώς επικεντρώνονται στα δεδομένα
 - Προσαρμόζονται για να “ταιριάξουν” στα δεδομένα
- Ο σχεδιασμός λύσεων λογισμικού περιλαμβάνει
 - Τον ορισμό ενός συνόλου αντικειμένων και του τρόπου αλληλεπίδρασής τους

Σύνοψη

- Η δομή είναι μια συλλογή διαφορετικών τύπων
- Οι κλάσεις συνδυάζουν δεδομένα και συναρτήσεις σε μία μονάδα → αντικείμενο
- Δεδομένα μέλη και συναρτήσεις μέλη
 - Μπορεί να είναι δημόσια → Προσβάσιμα εκτός της κλάσης
 - Μπορεί να είναι ιδιωτικά → Προσβάσιμα μόνο μέσω συναρτήσεων μελών
- Τύποι κλάσης και δομής μπορεί να είναι τυπικές παράμετροι σε συναρτήσεις