ΤΕΙ ΣΕΡΡΩΝ

ΑΝΤΙΚΕΙΜΕΝΟΣΤΡΑΦΗΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

Θεόδωρος Γ. Λάντζος

Διάλεξη Νο1

Κανόνες Ομαλής Λειτουργίας

- Ερχόμαστε στην ώρα μας
- Δεν καπνίζουμε και τρώμε εντός της αίθουσας

- Επιτρέπετε το νερό, τα αναψυκτικά και ο καφές με την προϋπόθεση να μην λερώνουμε το χώρο και πετάμε τα σκουπίδια εκτός των καλαθιών.
- Κινητά αθόρυβα και μόνον σε περίπτωση άμεσης ανάγκης.

- Σε περίπτωση συναγερμού, αποχωρούμε από την αίθουσα για το σημείο συγκέντρωσης σταδιακά, χωρίς πανικό και πιέσεις.

Συναντήσεις και Forum

- Οι διαλέξεις θα διεξάγονται κάθε Δευτέρα 11-13 στην αίθουσα 106 ΣΤΕΦ
- Ανακοινώσεις στην σελίδα μου

  http://www.teiser.gr/icd/staff/lantzos
- Ερωτήσεις δια μέσο email οι οποίες όμως θα απατούνται και αναλύονται στο μάθημα.

  lantzos@teiser.gr

Χρονοδιάγραμμα Διαλέξεων

- Έναρξη 26 Φεβρουαρίου 2008
- Δεκατρείς διαλέξεις 2 ωρών με 15 λεπτά διάλειμμα
- 1. 2/3 2. 9/3 3. 16/3 4. 23/3 5. 30/3
- 6. 6/4 7. 13/4 8. 20/4 9. 27/4
- 10. 4/5 11. 11/5 12. 18/5 13. 25/5
- 14. 1/6 15. 8/6 16. 15/6
- Λήξη μαθημάτων 15/06/2009 και ακολουθεί γραπτή εξέταση

Αντικείμενο Μαθήματος

- Εισαγωγή στην λογική του ΟΟ προγραμματισμού
- Κατανόηση των αρχών που διέπουν
- Εμπέδωση της φιλοσοφίας του
- Εκμάθηση των βασικών μηχανισμών γραφής ΟΟ προγραμμάτων
- Εφαρμογή και πειραματισμό της γνώσης σε μια γλώσσα ΟΟ προγραμματισμού υψηλού επιπέδου

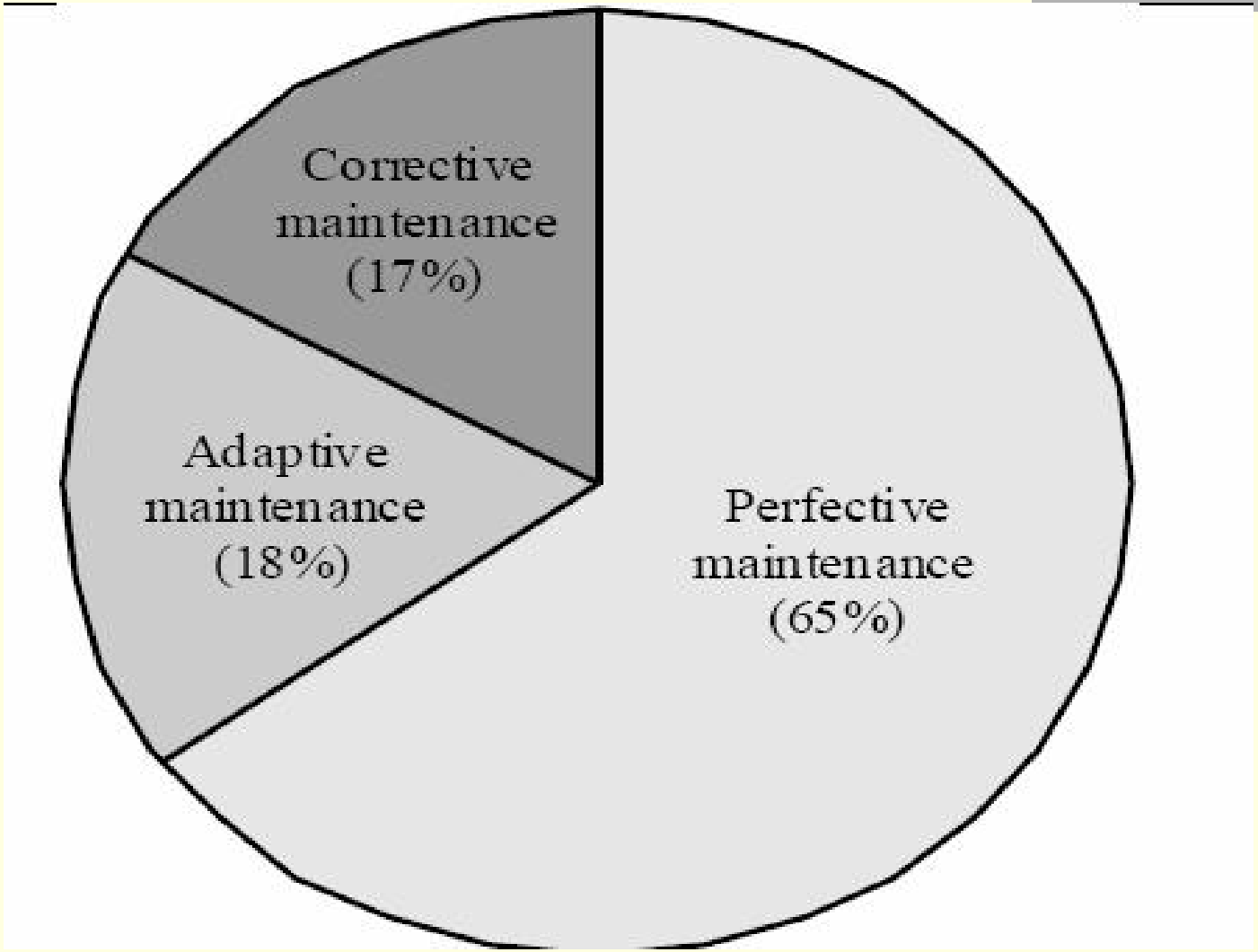
Συντήρηση Πληροφοριακού Συστήματος

Ορισμός :

«Η διαδικασία της διαχείρισης των αλλαγών του συστήματος»

- Η Συντήρηση είναι αναπόφευκτη διότι οι απαιτήσεις των συστημάτων αλλάζουν καθώς το περιβάλλον αλλάζει & εξελίσσεται
- Συντηρούμε ένα πληροφοριακό σύστημα προσπαθώντας να καλύψουμε τις νέες απαιτήσεις (adaptive), να το κάνουμε πιο αποτελεσματικό(perfective), ή να διορθώσουμε λάθη & αδυναμίες (corrective).

Κατανομή της προσπάθειας συντήρησης



Κόστος Συντήρησης

- Συνήθως μεγαλύτερο από το κόστος σχεδιασμού & δημιουργίας από 2% έως και 100%
- Επηρεάζεται από τεχνικούς παράγοντες
- Αυξάνεται όσο το σύστημα συντηρείται και καθώς μεγαλώνει η ηλικία του.

Παράγοντες Διαμόρφωσης Κόστους Συντήρησης

- Ανεξαρτησία ενοτήτων κώδικα (module independence)
- Γλώσσα Προγραμματισμού
- Ύφος προγραμμάτων (πχ δόμηση)
- Χρόνο δοκιμασίας, επικύρωσης και επαλήθευσης πριν την αποδέσμευση

Υπολογισμός Κόστους Συντήρησης

- Πολυπλοκότητα ελέγχου
- Πολυπλοκότητα δεδομένων
- Μέγεθος μεταβλητών
- Σχόλια προγράμματος
- Σύζευξη από ξένες βιβλιοθήκες
- Βαθμό επικοινωνίας με χρήστες για Είσοδο Έξοδο
- Ταχύτητα και Χώρο δημιουργίας

Γλώσσα Προγραμματισμού

- Μεθοδολογία & Τρόπο Σχεδιασμού
- Πολυπλοκότητα Ελέγχου
- Πολυπλοκότητα Δεδομένων
- Σύζευξη Ενοτήτων
- Αυτονομία Ενοτήτων
- Παραμετροποίηση & Επαναχρησιμοποίηση Ενοτήτων
- Κατανόηση Κώδικα
- Ανεξαρτησία Υλικού
- Ταχύτητα Υλοποίησης, Ελέγχου & Απελευθέρωσης Συστήματος

Γλώσσα Προγραμματισμού ως το μέσο για την επίτευξη

- Επαναχρησιμοποίηση
- Αξιοπιστίας
- Συντήρησης
- Γρήγορου & Εύκολου Σχεδιασμού

Η Γλώσσα C++

- cin >> – cout <<
- setw και setprecision
- Τύποι Δεδομένων char, int, long, float, double long double
- Τελεστές Αριθμητικοί, Λογικοί, Συσχετιστικοί
- Δομή επιλογής απλή, σύνθετη, πολλαπλή
- Δομή επανάληψης for, while, do while

Συνάρτηση με επιστροφή τιμής

```
#include <iostream.h>

int  athroisma(int x, int y);  // δήλωση συνάρτησης
main()
{
   int  a,b,c;

   cout << "Δώσε τον 1° αριθμό:";
   cin >> a;
   cout << "Δώσε τον 2° αριθμό:";
   cin >> b;
   c = athroisma(a,b);    // κλήση συνάρτησης
   cout << "Το άθροισμα των << a << " και " << b << " είναι " << c;
}
// ορισμός συνάρτησης
int  athroisma(int x, int y)
{
   int  z;    // τοπική μεταβλητή
   z = x + y;
   return  z ;
}
```

Κλήσεις Συναρτήσεων από Συναρτήσεις

Κατά την υλοποίηση της *ιεραρχικής σχεδίασης* ενός προγράμματος με τον *τμηματικό προγραμματισμό*, προκύπτει ένα πρόγραμμα με αυτόνομες λειτουργικές μονάδες (modules) σε διάφορα επίπεδα.

Π.χ.

```
#include <iostream.h>

#define N 10

struct employee
{
  int am;
  char name[20];
  int hours;
  int payrate;
  int misthos;
};
typedef struct employee Ergaz;

void eisagogiDedomenon(Ergaz p[N]);
int ypologismosMisthou(int hrs, int prt);
void emfanisiApotelesmaton(Ergaz p[]);

main()
{
   Ergaz emp[N];

   eisagogiDedomenon(emp);
   emfanisiApotelesmaton(emp);
   getch();
   return 0;
}

void eisagogiDedomenon(Ergaz p[N])
{
    int i;

   for (i=0; i<N; i++)
   {
     cout << "Ergazomenos " << i+1 << endl;
```

```
    cout << "==========" << endl;
    cout << "A.M.:";
    cin >> p[i].am;
    cout << "Name:";
    cin >> p[i].name;
    cout << "Ores:;
    cin >> p[i].hours;
    cout << "Oromisthio:";
    cin >> p[i].payrate;
                    // κλήση συνάρτησης
    p[i].misthos = ypologismosMisthou(p[i].hours, p[i].payrate);
    cout << endl;
  }
}

int ypologismosMisthou(int hrs, int prt)
{
   int sal;

   sal = hrs * prt;
   return sal;
};

void emfanisiApotelesmaton(Ergaz p[N])
{
  int i;

  for (i=0; i<N; i++)
    {
      cout << endl << "   A.M. : " << p[i].am << endl;
      cout << "   Name : " << p[i].name << endl;
      cout << "   Ores: " << p[i].hours << endl;
      cout << "   Oromisthio: " << p[i].payrate << endl;
      cout << "   Misthos: " << p[i].misthos << endl;
    }
}
```

Αναδρομικές Συναρτήσεις

Συνάρτηση με επανάληψη:

```
int  factorial(int n)
{
   int i,f;

   f = 1;
   for (i=2; i<=n; i++)
        f = f*i;
   return f;
}
```

- *Αναδρομικός ορισμός:*

$$n! = 1 \quad \text{αν} \quad n = 0,$$
$$= n*(n-1)! \quad \text{αν} \quad n > 0$$

Συνάρτηση με αναδρομή:

```
int  factorial(int n)
{
   int f;

   if (n = = 0)
        f = 1;
   else
        f = n*factorial(n-1);
   return f;
}
```

Κλήση με αναφορά

```
#include <stdio.h>
#include <iostream.h>
#include <conio.h>

void swap(int *a, int *b);

main()
{
    int a=5, b=10;

    swap(&a,&b);
    cout<<"a="<<a<<"\n";
    cout<<"b="<<b;
    getch();
}

void swap(int *a, int *b)
{

  int x;

  x=*a;
  *a=*b;
  *b=x;
}
```

```
#include <iostream.h>

void repchar()
void repchar(char ch)
void repchar(char ch, int n)

main()
{
   repchar();
   repchar('+');
   repchar('$',25);
}

void repchar()
{
   int i;
   for (i=0; i<45; i++)
         cout << '*';
   cout << endl;
}
```

```
void repchar(char ch)
{
   int i;
   for (i=0; i<45; i++)
         cout << ch;
   cout << endl;
}

void repchar(char ch, int n)
{
   int i;
   for (i=0; i<n; i++)
         cout << ch;
   cout << endl;
}
```

Η έξοδος του προγράμματος είναι:

```
*********************************************
+++++++++++++++++++++++++++++++++++++++++++++
$$$$$$$$$$$$$$$$$$$$$$$$$
```

Αντικειμενοστραφής Προγραμματισμός

Βασικοί Όροι

- Δομικό στοιχείο το αντικείμενο
- Αντικείμενα & χαρακτηριστικά
- Αντικείμενα & Συμπεριφορά
- Κλάσεις η μήτρα παραγωγής αντικειμένων
- Το αντικείμενο παρουσιάζει μια συγκεκριμένη στιγμή της κλάσης
- Τα αντικείμενα είναι σαν τα κουτιά τα οποία λαμβάνουνε και στέλνου μηνύματα
- Τα κουτιά περιέχουν συναρτήσεις και δεδομένα

Ιδιότητες του Αντικειμενοστραφή

- Ταυτότητα Αντικειμένου (Object Identity)
- Encapsulation (Ενθυλάκωση)
- Κληρονομικότητα
- Επαναχρησιμοποίηση
- Πολυμορφισμός