



Πανεπιστήμιο Αιγαίου

# Μεθοδολογίες και Γλώσσες Προγραμματισμού I

Δείκτες

Εργίνα Καβαλλιεράτου (kavallieratou@aegean.gr)

Μόνιμη Επίκουρος Καθηγήτρια

Τμήμα Μηχανικών Πληροφοριακών & Επικοινωνιακών Συστημάτων



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Σήμερα

---

- ✓ Επαναδέσμευση
- ✓ Δείκτες και πίνακες
- ✓ Δείκτες ως παράμετροι
- ✓ Δείκτες και strings
- ✓ Pointer σε κλάση
- ✓ Δέσμευση μνήμης σε κλάση
- ✓ Δείκτης this

# Επαναδέσμευση

---

## Λάθος

```
unsigned short int * pPointer = new unsigned short int;  
*pPointer = 72;  
pPointer = new unsigned short int;  
*pPointer = 84;
```

## Σωστό

```
unsigned short int * pPointer = new unsigned short int;  
*pPointer = 72;  
delete pPointer;  
pPointer = new unsigned short int;  
*pPointer = 84;
```

# Δείκτες και πίνακες

---

- ✓ Το όνομα ενός πίνακα είναι στη ουσία ένας *const* δείκτης.

```
int *x=0;
```

```
int a[10];
```

```
x = &a[2];
```

← Το **x** δείχνει στο **a[2]**

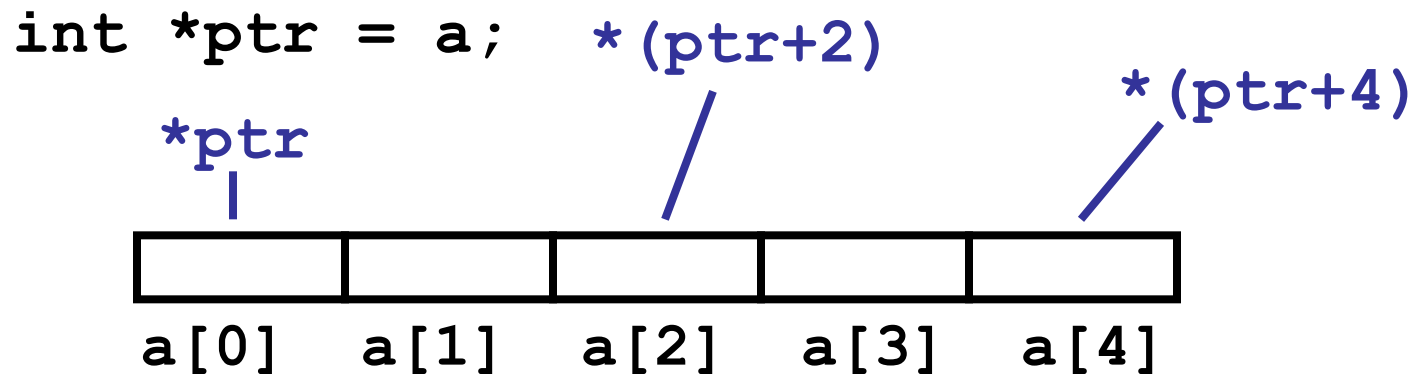
```
for (int i=0;i<3;i++)
```

```
    x[i]++;
```

← Το **x[i]** είναι το ίδιο με **a[2+i]**

# Δείκτες (Pointers)

- ✓ Οι δείκτες μπορούν να χρησιμοποιηθούν σε μαθηματικές εκφράσεις
- ✓ Όταν αυξάνουμε ένα δείκτη θα αυξηθεί σύμφωνα με το μέγεθος του αντικειμένου που δείχνει



```
int a[5];
```

# Εκτύπωση πίνακα

---

```
void print_array(int a[], int len) {  
    for (int i=0;i<len;i++)  
        cout << "[" << i << "]" = "  
            << a[i] << endl;  
}
```

*array version*

```
void print_array(int *a, int len) {  
    for (int i=0;i<len;i++)  
        cout << "[" << i << "]" = "  
            << *a++ << endl;  
}
```

*pointer version*



# Παράδειγμα

---

```
//===== swap_ref
```

```
void swap_ref(int& a, int& b)
```

```
{
```

```
    int temp = a;
```

```
    a = b;
```

```
    b = temp;
```

```
}
```

```
//===== swap_ptr
```

```
void swap_ptr(int* a, int* b)
```

```
{
```

```
    int temp = *a;
```

```
    *a = *b;
```

```
    *b = temp;
```

```
}
```

# Δείκτες ως παράμετροι

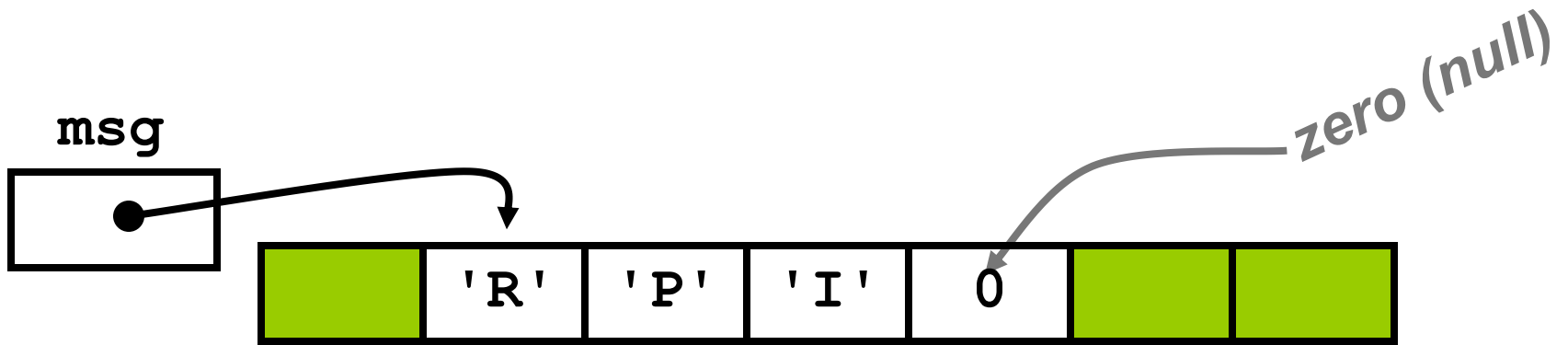
---

- ✓ Οι δείκτες περνούν με τιμή(?)
- ✓ Αν αλλάξουμε αυτό που δείχνει θα επιστραφεί η νέα τιμή

# Δείκτες και strings

- ✓ Οι δείκτες χρησιμοποιούνται συχνά με strings:

```
char *msg = "RPI";
```



# strcpy

---

```
void strcpy(char a[], const char b[])
{
    int i = 0;
    while (b[i] != '\0') {
        a[i] = b[i]; i++;
    }
    a[i] = '\0';
}
```

# strcpy

---

```
void strcpy(char* a, char* b)
{
    while (*b != '\0') {
        *a = *b;
        a++;
        b++;
    }
    *a = '\0';
}
```

**H**

```
void strcpy(char* a, char* b) {while (*a++ = *b++);}
```

# Pointer σε κλάση

---

- ✓ Όταν ορίσουμε pointer σε κλάση, θα πρέπει να χρησιμοποιείται μετά ως:

```
(*pRags).GetAge();
```

- ✓ Για ευκολία όμως η C++ επιτρέπει ισοδύναμα:

```
pRags->GetAge();
```

# Δέσμευση μνήμης σε κλάση/1

---

```
#include <iostream.h>
class SimpleCat
{
public:
    SimpleCat();
    ~SimpleCat();
    int GetAge() const { return *itsAge; }
    void SetAge(int age) { *itsAge = age; }
    int GetWeight() const { return *itsWeight; }
    void setWeight (int weight) { *itsWeight = weight; }
private:
    int * itsAge;
    int * itsWeight;
};
```

# Δέσμευση μνήμης σε κλάση/2

---

```
SimpleCat::SimpleCat() {  
    itsAge = new int(2);  
    itsWeight = new int(5); }
```

```
SimpleCat::~~SimpleCat() {  
    delete itsAge;  
    delete itsWeight; }
```

```
int main() {  
    SimpleCat *Frisky = new SimpleCat;  
    cout << "Frisky is " << Frisky->GetAge() << " years old\n";  
    Frisky->SetAge(5);  
    cout << "Frisky is " << Frisky->GetAge() << " years old\n";  
    delete Frisky;  
    return 0;}
```



# Δείκτης this/1

---

```
#include <iostream.h>
class Rectangle
{
public:
    Rectangle();
    ~Rectangle();
    void SetLength(int length) { this->itsLength = length; }
    int GetLength() const { return this->itsLength; }
    void SetWidth(int width) { itsWidth = width; }
    int GetWidth() const { return itsWidth; }
private:
    int itsLength;
    int itsWidth;
};
```

# Δείκτης this/2

```
Rectangle::Rectangle() {  
    itsWidth = 5;  
    itsLength = 10; }  
Rectangle::~~Rectangle()  
{
```

```
int main() {  
    Rectangle theRect;  
    cout << "theRect is " << theRect.GetLength() << " feet long.\n";  
    cout << "theRect is " << theRect.GetWidth() << " feet wide.\n";  
    theRect.SetLength(20);  
    theRect.SetWidth(10);  
    cout << "theRect is " << theRect.GetLength() << " feet long.\n";  
    cout << "theRect is " << theRect.GetWidth() << " feet wide.\n";  
    return 0; }
```

```
theRect is 10 feet long.  
theRect is 5 feet wide.  
theRect is 20 feet long.  
theRect is 10 feet wide.
```

# Παράδειγμα 1.1

---

```
#include <iostream>
void increment_all (int* start, int* stop){
    int * current = start;
    while (current != stop) {
        ++(*current);
        ++current;
    }
}
void print_all (const int* start, const int* stop){
    const int * current = start;
    while (current != stop) {
```

# Παράδειγμα 1.2

---

```
    cout << *current << '\n';
    ++current;
}
}

int main ()
{
    int numbers[] = {10,20,30};
    increment_all (numbers,numbers+3);
    print_all (numbers,numbers+3);
    return 0;
}
```

# Παράδειγμα 2.1

---

```
#include <iostream>
void increase (void* data, int psize){
    if ( psize == sizeof(char) )
    { char* pchar; pchar=(char*)data; ++(*pchar); }
    else if (psize == sizeof(int) )
    { int* pint; pint=(int*)data; ++(*pint); }
}
int main (){
    char a = 'x';
    int b = 1602;
    increase (&a,sizeof(a));
    increase (&b,sizeof(b));
    cout << a << ", " << b << '\n';
    return 0;}
```

# Παράδειγμα 3.1

---

```
#include <iostream>
int addition (int a, int b)
{ return (a+b); }
int subtraction (int a, int b)
{ return (a-b); }
int operation (int x, int y, int (*functocall)(int,int))
{
    int g;
    g = (*functocall)(x,y);
    return (g);
}
```

# Παράδειγμα 3.2

---

```
int main ()
{
    int m,n;
    int (*minus)(int,int) = subtraction;

    m = operation (7, 5, addition);
    n = operation (20, m, minus);
    cout <<n;
    return 0;
}
```

???

---

```
#include <iostream.h>
int main()
{   int *pInt;
    *pInt = 9;
    cout << "The value at pInt: " << *pInt;
    return 0;
}
```



???

---

```
int main()
{
    int SomeVariable = 5;
    cout << "SomeVariable: " << SomeVariable << "\n";
    int *pVar = & SomeVariable;
    pVar = 9;
    cout << "SomeVariable: " << *pVar << "\n";
    return 0;
}
```