



Ελληνική Δημοκρατία
Τεχνολογικό Εκπαιδευτικό
Ίδρυμα Ηπείρου

Προγραμματισμός Ι

Ενότητα 9 : Εγγραφές, Σύνολα

Αλέξανδρος Τζάλλας



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Τμήμα Μηχανικών Πληροφορικής Τ.Ε

Προγραμματισμός Ι

Ενότητα 9 : Εγγραφές, Σύνολα

Αλέξανδρος Τζάλλας

Λέκτορας

Άρτα, 2015





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 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.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο ΤΕΙ Ηπείρου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Σκοποί ενότητας

- Να γίνει σαφής η αναγκαιότητα ορισμού των εγγραφών και συνόλων.
- Να περιγραφεί ο τρόπος ορισμού μια εγγραφής/συνόλου.
- Να αναλυθεί ο σκοπός/χρησιμότητα των εγγραφών και συνόλων στην pascal.



Περιεχόμενα ενότητας

- Εγγραφές – Records
- Σύνολα
- Ορισμός Τύπων Συνόλων
- Εκχώρηση Στοιχείων Συνόλων σε Μεταβλητές



Εγγραφές - Records

- Η **εγγραφή (record)** είναι ένα σύνολο στοιχείων, τα οποία δεν είναι (υποχρεωτικά) του ίδιου τύπου μεταξύ τους. Τα στοιχεία του record λέγονται **πεδία (fields)**. Η δήλωση ενός record είναι η ακόλουθη:

```
TYPE <ΟΝΟΜΑ ΤΥΠΟΥ> = RECORD
```

```
<ΟΝΟΜΑ ΠΕΔΙΟΥ 1>: <ΤΥΠΟΣ ΠΕΔΙΟΥ 1>;
```

```
<ΟΝΟΜΑ ΠΕΔΙΟΥ 2>: <ΤΥΠΟΣ ΠΕΔΙΟΥ 2>;
```

```
...
```

```
<ΟΝΟΜΑ ΠΕΔΙΟΥ N>: <ΤΥΠΟΣ ΠΕΔΙΟΥ N>;
```

```
END;
```



Εγγραφές - Records

- **Παράδειγμα 1**

Ας υποθέσουμε ότι θέλουμε να επεξεργαστούμε τα στοιχεία των σπουδαστών του Τμήματος Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών. Τα στοιχεία που θέλουμε να έχουμε είναι: Επώνυμο, Όνομα, Αριθμός Μητρώου, Χρονολογία Γέννησης, Φύλο και Βαθμολογία ενός Μαθήματος.

| SURNAME | NAME | AM | BIRTHDAY | SEX | GRADE |
|-----------|----------|-----|----------|---------|-------|
| ΣΩΤΗΡΙΟΥ | ΓΕΩΡΓΙΟΣ | 145 | 1982 | ΑΓΟΡΙ | 7,5 |
| ΓΕΩΡΓΙΟΥ | ΜΑΡΙΑ | 210 | 1981 | ΚΟΡΙΤΣΙ | 8,2 |
| ΣΤΑΜΑΤΙΟΥ | ΝΙΚΟΛΑΟΣ | 172 | 1982 | ΑΓΟΡΙ | 6,7 |



Εγγραφές - Records

Στη παραπάνω δήλωση τύπου (type) ορίζουμε τον τύπο STUDENT με στοιχεία:

Το επώνυμο(SUR) και το όνομα (NAME) του σπουδαστή ως συμβολοσειρές μήκους 20 χαρακτήρων, τον αριθμό μητρώου (AM) και την χρονολογία γέννησης (BIRTH) ως ακέραιους αριθμούς, το φύλο (SEX) ως χαρακτήρα και τη βαθμολογία (GRADE) ως πραγματικό αριθμό.

```
Type student = record
  Sur,name:string[20];
  Am,birth:integer;
  Sex:char;
  Grade:real;
End;
Var stud:student;
```



Εγγραφές - Records

Στη VAR δηλώνουμε μια μεταβλητή STUD η οποία είναι τύπου STUDENT. Το όνομα STUDENT δεν μπορεί να χρησιμοποιηθεί στις εντολές του προγράμματος, αφού έχει δηλωθεί ως τύπος δεδομένων και όχι ως μεταβλητή. Η αναφορά των πεδίων ενός RECORD γίνεται με τον τελεστή.

```
TYPE STUDENT = RECORD  
  SUR,NAME:STRING[20];  
  AM,BIRTH:INTEGER;  
  SEX:CHAR;  
  GRADE:REAL;  
END;  
VAR STUD:STUDENT;
```



Εγγραφές - Records

Η εντολή `STUD.SUR := 'ΣΩΤΗΡΙΟΥ'`; δίνει στο πεδίο `SUR` του `RECORD STUD` την τιμή `ΣΩΤΗΡΙΟΥ`.

Η εντολή `STUD.NAME := 'ΓΕΩΡΓΙΟΣ'`; δίνει στο πεδίο `NAME` του `RECORD STUD` την τιμή `ΓΕΩΡΓΙΟΣ`.

Η εντολή `STUD.AM := 145`; δίνει στο πεδίο `AM` του `RECORD STUD` την τιμή `145`.

Η εντολή `STUD.BIRTH := 1982`; δίνει στο πεδίο `BIRTH` του `RECORD STUD` την τιμή `1982`.

Η εντολή `STUD.SEX := 'M'`; δίνει στο πεδίο `SEX` του `RECORD STUD` την τιμή `M` (Male -> Αγόρι).

Η εντολή `STUD.GRADE := 7.5`; δίνει στο πεδίο `GRADE` του `RECORD STUD` την τιμή `7.5`.

```
TYPE STUDENT = RECORD
  SUR,NAME:STRING[20];
  AM,BIRTH:INTEGER;
  SEX:CHAR;
  GRADE:REAL;
END;
VAR STUD:STUDENT;
```



Εγγραφές - Records

Πεδία του RECORD μπορεί να είναι και πίνακες.

Στο πιο πάνω παράδειγμα θα μπορούσαμε να είχαμε τους βαθμούς (GRADE) 5 μαθημάτων οπότε η δήλωση του RECORD θα γινόταν ως εξής:

```
TYPE STUDENT = RECORD
  SUR, NAME: STRING[20];
  AM, BIRTH: INTEGER;
  SEX: CHAR;
  GRADE: ARRAY[1..5] OF REAL;
END;
VAR STUD: STUDENT;
```

Η εντολή `STUD.GRADE[1] := 7,5` δίνει στο πεδίο `GRADE[1]` του RECORD `STUD` την τιμή 7.5. (Βαθμό για το 1ο μάθημα).



Εγγραφές - Records

Επίσης στοιχεία ενός πίνακα μπορεί να είναι RECORDS.

Στο πρώτο παράδειγμα που χρησιμοποιήσαμε θα μπορούσαμε να είχαμε τα στοιχεία για 50 μαθητές οπότε η δήλωση του RECORD θα γινόταν ως εξής:

```
TYPE STUDENT = RECORD  
  SUR,NAME:STRING[20];  
  AM,BIRTH:INTEGER;  
  SEX:CHAR;  
  GRADE:REAL;  
END;  
VAR STUD:ARRAY[1..50] OF STUDENT;
```



Εγγραφές - Records

Η εντολή `STUD[1].SUR := 'ΣΩΤΗΡΙΟΥ'`; δίνει στο πεδίο `SUR` του `RECORD STUD[1]` την τιμή `ΣΩΤΗΡΙΟΥ`. (Για το 1ο Σπουδαστή)

Η εντολή `STUD[1].NAME := 'ΓΕΩΡΓΙΟΣ'`; δίνει στο πεδίο `NAME` του `RECORD STUD[1]` την τιμή `ΓΕΩΡΓΙΟΣ`. (Για το 1ο Σπουδαστή)

Η εντολή `STUD[1].AM := 145`; δίνει στο πεδίο `AM` του `RECORD STUD[1]` την τιμή `145`. (Για το 1ο Σπουδαστή)

Η εντολή `STUD.BIRTH := 1982`; δίνει στο πεδίο `BIRTH` του `RECORD STUD[1]` την τιμή `1982`. (Για το 1ο Σπουδαστή)

```
TYPE STUDENT = RECORD
  SUR,NAME:STRING[20];
  AM,BIRTH:INTEGER;
  SEX:CHAR;
  GRADE:REAL;
END;
VAR STUD:ARRAY[1..50] OF STUDENT;
```



Εγγραφές - Records

Η εντολή `STUD[1].GRADE := 7.5;` δίνει στο πεδίο `GRADE` του `RECORD STUD[1]` την τιμή 7.5. (Για το 1ο Σπουδαστή)

Επίσης θα μπορούσαμε να είχαμε συνδυασμό ή συνδυασμούς των πιο πάνω περιπτώσεων. π.χ. να είχαμε τα στοιχεία για 50 μαθητές και τους βαθμούς (`GRADE`) 5 μαθημάτων οπότε η δήλωση του `RECORD` θα γινόταν ως εξής:

```
TYPE STUDENT = RECORD
  SUR,NAME:STRING[20];
  AM,BIRTH:INTEGER;
  SEX:CHAR;
  GRADE:ARRAY[1..5] OF REAL;
END;
VAR STUD:ARRAY[1..50] OF STUDENT;
```



Εγγραφές - Records

Η εντολή `STUD[1].GRADE := 7.5;` δίνει στο πεδίο `GRADE` του `RECORD STUD[1]` την τιμή 7.5. (Για το 1ο Σπουδαστή)

Επίσης θα μπορούσαμε να είχαμε συνδυασμό ή συνδυασμούς των πιο πάνω περιπτώσεων. π.χ. να είχαμε τα στοιχεία για 50 μαθητές και τους βαθμούς (`GRADE`) 5 μαθημάτων οπότε η δήλωση του `RECORD` θα γινόταν ως εξής:

Η εντολή `STUD[1].GRADE[1] := 7.5;` δίνει στο πεδίο `GRADE[1]` του `RECORD STUD[1]` την τιμή 7.5. (Το πρώτο μάθημα για το 1ο Σπουδαστή)

```
TYPE STUDENT = RECORD
  SUR,NAME:STRING[20];
  AM,BIRTH:INTEGER;
  SEX:CHAR;
  GRADE:ARRAY[1..5] OF REAL;
END;
VAR STUD:ARRAY[1..50] OF STUDENT;
```




Εγγραφές - Records

- **Παράδειγμα 2**

Να γραφεί πρόγραμμα σε γλώσσα Pascal το οποίο να διαβάζει τα στοιχεία 5 σπουδαστών (επώνυμο, αριθμό μητρώου, χρονολογία γέννησης και βαθμό στο μάθημα Προγραμματισμός) του Τμήματος Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών.

Να υπολογίζει το μέσο όρο στο μάθημα Δομές Δεδομένων και να τυπώνει τα στοιχεία των σπουδαστών που έχουν βαθμό μεγαλύτερο από το μέσο όρο.



Εγγραφές - Records

```

Program paradeigma_21;
Uses crt;
Type student=record
Sur:string[20];
Am,birth:integer;
Dd:real;
End;
Var stud:array[1..5] of student;
l:integer;sum,mo:real;
Begin
Clrscr;
Sum:=0;
For i:=1 to 5 do
Begin
Write('dose eponymo ',i,' ou spoudasth : ');
Readln(stud[i].sur);
Write('dose ar. Mhtroou ',i,' ou spoudasth : ');
Readln(stud[i].am);
Write('dose hm. Gennhshs ',i,' ou spoudasth : ');
Readln(stud[i].birth);
Write('dose bathmo ',i,' ou spoudasth : ');
Readln(stud[i].dd);
Sum := sum + stud[i].dd;
End;
Writeln;
Mo := sum / 5;
Writeln('o mesos oros einai : ',mo:0:2);
Writeln;
For i:=1 to 5 do
If stud[i].dd > mo then
Writeln('eponymo:',stud[i].sur,'a.m.':,stud[i].am,'
Hm.gennhshs:',stud[i].birth,'bathmos:',stud[i].dd:0:2);
End.
    
```



Σύνολα

- Τα σύνολα είναι τύποι δεδομένων που ορίζονται από τον χρήστη και οι οποίοι έχουν κάποιο βαθμό δόμησης.
- Όπως και στα μαθηματικά, έτσι και στην Pascal, ένα *σύνολο (set)* είναι μια διατεταγμένη συλλογή στοιχείων. Ένα σύνολο αποτελείται από μια πεπερασμένη συλλογή (μοναδικών) στοιχείων τα οποία βρίσκονται στον ίδιο προκαθορισμένο τύπο, ο οποίος καλείται *βασικός*.
- Η γενική μορφή δήλωσης ενός συνόλου είναι η εξής :

Type Όνομα_Συνόλου = ***Set of*** Τακτικός_Τύπος

- Ο Τακτικός_Τύπος είναι ένα διατεταγμένο σύνολο τιμών, τα στοιχεία του πρέπει να είναι όλα του ίδιου τύπου και πρέπει να αποτελεί ένα απλό στοιχείο ή ένα υποδιάστημα. Ο Τακτικός_Τύπος δεν μπορεί να έχει πάνω από 256 πιθανές τιμές



Σύνολα

- **Παράδειγμα**

Const

vowels = ['A', 'E', 'I', 'O', 'U'];

upper_case = Set of 'A'..'Z' = ['A'..'Z'];

- Αφού οριστεί κάποιο σύνολο, ο τελεστής `in` χρησιμεύει για να εξετάσει την παρουσία ή την απουσία κάποιου συγκεκριμένου στοιχείου από το σύνολο, ως εξής

If `ch In upper_case` Then ...

If `ch In vowels` Then ...

- Στα παραπάνω παραδείγματα, το `In` επιστρέφει την τιμή `true` αν το `ch`



Ορισμός Τύπων Συνόλων

- Τα σύνολα μπορούν να οριστούν με πολλούς τρόπους :

- Προκαθορισμένοι τακτικοί τύποι :

Type

`boolean_set = Set of Boolean;`

`char_set = Set of Char;`

`byte_set = Set of Byte;`



Ορισμός Τύπων Συνόλων

- **Υποδιαστήματα προκαθορισμένων τύπων :**

Type

```
bits = 1..7;
byte_bits = Set of bits;
up_case = Set of 'A'..'Z';
lo_case = Set of 'a'..'z';
```

- **Απαριθμητοί τύποι :**

Type

```
transportation = (bicycle, motorcycle, car, truck, bus);
four_wheels = car..bus;
trans_set = Set of transportation;
four_wheel_set = Set of four_wheels;
```



Ορισμός Τύπων Συνόλων

- **Μεταβλητές :**

Var

```
fast_trans : four_wheels;
lower_letters : lo_case;
num1, num2 : byte_bits;
```

- **Σταθερές :**

Const

```
vowels : Set of Char = ['A', 'E', 'I', 'O', 'U'];
up_chars : Set of Char = ['A'..'Z'];
lo_chars : Set of Char = ['a'..'z'];
```



Εκχώρηση Στοιχείων Συνόλων σε Μεταβλητές

- Για την εκχώρηση στοιχείων συνόλων σε μεταβλητές χρησιμοποιούνται οι αγκύλες, ως εξής :

Μεταβλητή_Συνόλου := [Στοιχείο_Συνόλου]

- Το κενό σύνολο είναι το [] και με τη βοήθειά του μπορούμε να μηδενίσουμε μεταβλητές συνόλων. Ένα σύνολο μπορεί να αποτελείται από μια σειρά στοιχείων, ένα υποδιάστημα ή από τον συνδυασμό και των δύο, ως εξής :

```
set1 := [1, 3, 5, 7, 9]
set2 := [0..7];
set3 := [0..7, 14, 15, 16];
char_list := ['A'..'Z', 'a'..'z', '0'..'9'];
```




Εκχώρηση Στοιχείων Συνόλων σε Μεταβλητές

- **Οι Τελεστές Συνόλων**

Οι τελεστές συνόλων προσφέρουν τρόπους για την δημιουργία νέων συνόλων με στοιχεία από ήδη υπάρχοντα σύνολα. Οι τελεστές συνόλων που υποστηρίζει η Pascal είναι οι εξής :

- **Οι Σχισιακοί Τελεστές**

Οι σχισιακοί τελεστές έχουν ως σκοπό την εύρεση της σχέσης δύο ή περισσότερων συνόλων. Βασισμένος στις πιθανές καταστάσεις κάποιας σχέσης, ο τελεστής επιστρέφει την τιμή True ή False.



Εκχώρηση Στοιχείων Συνόλων σε Μεταβλητές

Ο παρακάτω πίνακας παρουσιάζει τους σχεσιακούς τελεστές που χρησιμοποιούνται στα σύνολα :

| ΤΕΛΕΣΤΗΣ | ΣΥΝΤΑΞΗ | ΕΠΙΣΤΡΕΦΕΙ TRUE AN |
|----------|-----------------------|--|
| = | Σύνολο A = Σύνολο B | Τα A και B είναι ταυτόσημα. |
| <> | Σύνολο A <> Σύνολο B | Ένα από τα σύνολα περιέχει τουλάχιστον ένα στοιχείο που δεν βρίσκεται στο άλλο. |
| <= | Σύνολο A <= Σύνολο B | Κάθε στοιχείο του A είναι και στο B. |
| < | Σύνολο A < Σύνολο B | Κάθε στοιχείο του A είναι και στο B και επιπλέον στο B υπάρχει τουλάχιστον ένα στοιχείο που δεν βρίσκεται στο A. |
| >= | Σύνολο A >= Σύνολο B | Κάθε στοιχείο του B είναι και στο A. |
| > | Σύνολο A > Σύνολο B | Κάθε στοιχείο του B είναι και στο A και επιπλέον στο A υπάρχει τουλάχιστον ένα στοιχείο που δεν βρίσκεται στο B. |
| IN | Στοιχείο στο Σύνολο A | Το στοιχείο υπάρχει στο σύνολο A. |



Εκχώρηση Στοιχείων Συνόλων σε Μεταβλητές

- **Ο Τελεστής In**

Ο τελεστής In εξετάζει την σχέση δύο ή περισσότερων συνόλων. Επιστρέφει ένα αποτέλεσμα τύπου Boolean, ανάλογα με το αν κάποιο στοιχείο είναι μέρος του συνόλου. Το εξεταζόμενο στοιχείο θα πρέπει να είναι του ίδιου τύπου ή συμβατού με τον βασικό τύπο του συνόλου.

Η σύνταξή του είναι η εξής : Όνομα_Στοιχείου In Όνομα_Συνόλου

Ακολουθούν παραδείγματα :

ch In vowels

'i' In consonants

operator In ['+', '-', '/', '*']



Βιβλιογραφία

Βλαχάβας Ι. (1994). Η γλώσσα προγραμματισμού Pascal. Εκδόσεις Γαρταγάνης Διονύσιος.

Κάβουρας Ι.Κ. (1999). Δομημένος Προγραμματισμός με Pascal. Εκδόσεις Κλειδάριθμος.

Αλεβίζου Θ., & Καμπουρέλης Α. (1995). Μαθήματα Προγραμματισμού: Εισαγωγή με τη Γλώσσα Pascal. Εκδόσεις Παπασωτηρίου.

Cooper D. (1993). Oh! Pascal!, An Introduction to Computing, του. Εκδόσεις Norton.

Larry R.N. (1998). Advanced Programming in Pascal with Data Structures. Εκδόσεις Macmillan USA.

Τσελίκης Γ.Σ., Τσελίκας Ν.Δ. (2012). C: από τη Θεωρία στην Εφαρμογή (Β' Έκδοση). Εκδόσεις Παπασωτηρίου.

Aho A.V., Hopcroft J.E., & Ullman J.D. (1974). The design and analysis of computer algorithms. Εκδόσεις Addison Wesley.

Abelson H., Sussman G.J., Sussman J. (1985). Structure and Interpretation of Computer Programs, MIT Press, McGraw Hill Book Company.



Σημείωμα Αναφοράς

Copyright Τεχνολογικό Ίδρυμα Ηπείρου. Αλέξανδρος Τζάλλας.
Προγραμματισμός Ι.

Έκδοση: 1.0 Άρτα, 2015. Διαθέσιμο από τη δικτυακή
διεύθυνση:

<http://eclass.teiep.gr/OpenClass/courses/COMP111/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά Δημιουργού-Μη Εμπορική Χρήση-Όχι Παράγωγα Έργα 4.0 Διεθνές [1] ή μεταγενέστερη. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, Διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/deed.el>



Τέλος Ενότητας

Επεξεργασία: Ευάγγελος Καρβούνης
Άρτα, 2015



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



Τέλος Ενότητας

Εγγραφές, Σύνολα



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

