

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ:
ΣΥΜΒΟΥΛΕΣ
ΚΑΙ
ΠΑΡΑΔΕΙΓΜΑΤΑ



ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ:
ΣΥΜΒΟΥΛΕΣ
ΚΑΙ
ΠΑΡΑΔΕΙΓΜΑΤΑ
ΤΟΜΟΣ Ι

ΑΛΕΞΑΝΔΡΟΣ Μ. ΓΙΑΝΝΑΚΗΣ

1. Εισαγωγή

Σχετικά με το βιβλίο

Σε αυτόν τον τόμο, θα κάνουμε μία σύντομη περιήγηση στον προγραμματισμό.

Τι είναι ο προγραμματισμός

Με τον όρο προγραμματισμός, συνήθως αναφερόμαστε στην ρύθμιση ενός μηχανισμού για την εκτέλεση μίας ή περισσότερων διαδικασιών. Παραδείγματα προγραμματισμού είναι η ρύθμιση ενός υπολογιστή για την εύρεση της λύσης μίας μαθηματικής πράξης, ο προγραμματισμός ενός ρομπότ για την εκτέλεση συγκεκριμένων κινήσεων, ο προγραμματισμός του ανθρώπινου εγκεφάλου για την εκτέλεση των κινήσεων/ανταποκρίσεων του ανθρώπου κλπ.

Τι είναι αλγόριθμος

Αλγόριθμος είναι μια πεπερασμένη σειρά εντολών, αυστηρά καθορισμένων και εκτελέσιμων σε πεπερασμένο χρόνο, οι οποίες έχουν σκοπό την επίλυση ενός προβλήματος. Με άλλα λόγια είναι μία συγκεκριμένη σειρά από συγκεκριμένες εντολές, οι οποίες έχουν σκοπό την επίλυση ενός προβλήματος μέσα σε ένα συγκεκριμένο χρονικό διάστημα. Παράδειγμα, ο αλγόριθμος του Ευκλείδη, ο οποίος είναι μια σειρά από μαθηματικές πράξεις, οι οποίες έχουν σκοπό την εύρεση του μέγιστου κοινού διαιρέτη, μεταξύ δύο αριθμών.

Γλώσσες προγραμματισμού

Με τον όρο «γλώσσα προγραμματισμού» εννοούμε μία τεχνητή γλώσσα, η οποία καθορίζει τις ενέργειες ενός μηχανισμού, συνήθως

ενός υπολογιστή. Οι γλώσσες προγραμματισμού, αποτελούνται από ένα σύνολο κανόνων, συντακτικών και εννοιολογικών που καθορίζουν τις λειτουργίες και τις χρήσεις τους. Στο συγκεκριμένο βιβλίο θα αναφερθούμε στις γλώσσες προγραμματισμού Python 3, C, C++ και Fortran 95.

Γιατί ο προγραμματισμός είναι χρήσιμος

Ο προγραμματισμός, πέρα απ' το ότι αποτελεί θεμέλιο της σημερινής κοινωνίας, είναι μία επιστήμη, η οποία φέρνει τον άνθρωπο πιο κοντά στην πραγματικότητα, καθώς τον κάνει και πιο δημιουργικό. Ο προγραμματισμός δεν είναι κάτι το καινούργιο, αντίθετα, υπήρχε από τα αρχαία χρόνια, πράγμα που αποδεικνύει ότι βρίσκεται στο γονίδιο του ανθρώπου, παρότι αρκετοί άνθρωποι σήμερα βρίσκουν δύσκολο να δημιουργήσουν και τα πιο απλά προγράμματα.

2. “Γεια σου κόσμε”

Η βασική εντολή

Στον προγραμματισμό, η πιο βασική εντολή είναι η εντολή για την εμφάνιση ενός δεδομένου(συνήθως κειμένου ή αριθμού) στην οθόνη. Η εντολή αυτή διαφέρει από την μία γλώσσα προγραμματισμού στην άλλη. Είναι επίσης μία εντολή η οποία χρησιμοποιείται πολύ στην εκσφαλμάτωση(debugging), όπως θα πούμε και αργότερα.

Παραδείγματα

Python 3:

```
print("Hello World")
```

C:

```
#include <stdio.h>

int main()
{
    printf("Hello World");
    return 0;
}
```

C++:

```
#include <iostream.h>
using namespace std;

int main()
{
    cout << "Hello World";
    return 0;
}
```

Fortran 95:

```
program hello  
  print *, "Hello World"  
end program
```

3. Δεδομένα

Τι είναι δεδομένο

Δεδομένο είναι ότι μπορεί να γίνει αντιληπτό τουλάχιστον με μία από τις πέντε αισθήσεις. Δεδομένο μπορεί να είναι ένα κείμενο, μία εικόνα, ένας ήχος κλπ. Ο υπολογιστής λαμβάνει τα δεδομένα με διαφορετικό τρόπο από τους ανθρώπους. Χρησιμοποιεί υποδοχές(inputs) όπως το πληκτρολόγιο, το ποντίκι, την κάμερα, το μικρόφωνο και άλλες υποδοχές που μπορεί να περιέχει.

Τύποι δεδομένων

Οι τύποι δεδομένων, διαφέρουν από την μία γλώσσα προγραμματισμού στην άλλη, οι πιο συνηθισμένοι όμως είναι τα αριθμητικά, τα αλφαριθμητικά και τα λογικά. Αριθμητικά δεδομένα είναι αριθμοί, συνήθως πραγματικοί (π.χ. 4, 3.14, 2.71828182845904, 0, -10, 1000000 και πολλοί άλλοι). Αλφαριθμητικά δεδομένα είναι σύνολα που αποτελούνται από κανέναν ή περισσότερους χαρακτήρες και βρίσκονται μέσα σε αποστρόφους (‘α’, “522x”, “Αλέξανδρος”, “2101234567”, “a325”, “123!@#” και άλλα). Λογικά δεδομένα είναι τύποι δεδομένων που εκφράζουν εάν κάτι (συνήθως μία συνθήκη) ισχύει ή όχι.

Παράδειγμα

Python 3:

```
print("Αριθμητικά δεδομένα:", 1, 3.14, -5, 3.0, 0)  
print("Αλφαριθμητικά δεδομένα:", 'α', "12345", "Alexander")  
print("Λογικά δεδομένα:", True, False)
```

Output:

```
Αριθμητικά δεδομένα: 1 3.14 -5 3.0 0  
Αλφαριθμητικά δεδομένα: α 12345 Alexander  
Λογικά δεδομένα: True False
```

Όπως βλέπουμε στην παραπάνω εικόνα, τα αλφαριθμητικά δεδομένα, εμφανίζονται στο αποτέλεσμα δίχως τις αποστρόφους όπως ισχύει συνήθως.

4. Μεταβλητές

Τι είναι μεταβλητή

Στην πληροφορική, με τον όρο μεταβλητή εννοούμε ένα κομμάτι μνήμης στο οποίο αποθηκεύονται δεδομένα και μπορεί να τροποποιηθεί κατά την εκτέλεση του προγράμματος. Η χρήση των

μεταβλητών μπορεί να διαφέρει ανάλογα με το ποια γλώσσα προγραμματισμού χρησιμοποιούμε και ανάλογα με το τι δεδομένα αποθηκεύονται σε αυτήν.

Παραδείγματα

Python 3:

```
a = 1
b = "Alexander"
c = True
print(c, b, a)
```

Output:

True Alexander 1

C++:

```
#include <iostream.h>
#include <string.h>
using namespace std;
int main() {
    int a = 1;
    string b = "Alexander";
    bool c = true;
    cout << c << " " << b << " " << a;
    return 0;
}
```

Output:

1 Alexander 1

Όπως βλέπουμε στις εικόνες, η μεταβλητή *c* εμφανίζεται διαφορετικά στην Python και διαφορετικά στην C++.

5. Σταθερές

Τι είναι σταθερή

Στην πληροφορική, με τον όρο σταθερή εννοούμε ένα κομμάτι μνήμης στο οποίο αποθηκεύονται δεδομένα, όμως δεν μπορεί να τροποποιηθεί κατά την εκτέλεση του προγράμματος, σε αντίθεση με τις μεταβλητές.

Παράδειγμα

C++:

```
#include <iostream.h>
using namespace std;
int main() {
    const double PI = 3.14;
    cout << PI;
    return 0;

}
```

Output:

3.14

Όπως βλέπουμε και στην εικόνα, οι σταθερές ορίζονται όπως και οι μεταβλητές. Ο τρόπος με τον ορίζονται διαφέρει από την μία γλώσσα προγραμματισμού στην άλλη, καθώς και το αν ορίζονται. Στην Python 3 για παράδειγμα δεν υπάρχουν σταθερές.

6. Πράξεις

Αριθμητικές πράξεις

Αριθμητικές πράξεις είναι οι πράξεις, οι οποίες γίνονται με την χρήση αριθμητικών τελεστών (+, -, *, / κτλ.). Οι αριθμητικοί τελεστές μπορεί να διαφέρουν ανάλογα με την γλώσσα προγραμματισμού.

Συγκρίσεις

Συγκρίσεις είναι οι πράξεις, οι οποίες γίνονται με την χρήση συγκριτικών τελεστών (>, <, =, ≠, ≥, ≤). Στην Python 3, στην C και στην C++ είναι >, <, ==, !=, >=, <=, ενώ στην Fortran 95 είναι > ή .GT., < ή .LT., == ή .EQ., /= ή .NE., >= ή .GE., <= ή .LE.

Λογικές πράξεις

Λογικές πράξεις είναι οι πράξεις, οι οποίες γίνονται με την χρήση λογικών τελεστών (Οι λογικοί τελεστές διαφέρουν από την μία γλώσσα προγραμματισμού. Παράδειγμα στην Python 3 είναι & ή and, | ή or, not, ενώ στην C και στην C++ είναι &&, ||, !. Στην Fortran 95 είναι .AND., .OR., .NOT.).

7.Input: Είσοδος/Υποδοχή

Τι είναι είσοδος

Στην πληροφορική είσοδος είναι το μέσο με το οποίο ένας μηχανισμός (συνήθως υπολογιστής) δέχεται δεδομένα. Η πιο συνηθισμένη μορφή εισόδου είναι το πληκτρολόγιο του υπολογιστή. Στον προγραμματισμό συγκεκριμένα είναι η εντολή ή το σύνολο των εντολών με το οποίο ο μηχανισμός μπορεί να δέχεται δεδομένα. Η πιο βασική εντολή εισόδου είναι αυτή που δέχεται δεδομένα από το πληκτρολόγιο και διαφέρει ανάλογα με την γλώσσα προγραμματισμού.

Παραδείγματα

Python 3:

```
something = input("Enter Something: ")  
print("You entered:", something)
```

C++:

```
#include <iostream.h>  
#include <string.h>  
using namespace std;  
int main() {  
    string something;  
    cout << "Enter something: ";  
    cin >> something;  
    cout << "You entered: " << something;  
    return 0;  
}
```

8. Δομές επιλογής

Τι είναι η δομή επιλογής

Δομή επιλογής ονομάζουμε το μέρος του προγράμματος, το οποίο καθορίζει εάν ένα μέρος του προγράμματος θα εκτελεστεί ή όχι, ελέγχοντας μία ή περισσότερες συνθήκες. Ένα πρόγραμμα μπορεί να περιέχει καμία ή περισσότερες δομές επιλογής. Η δομές επιλογής μπορεί να διαφέρουν από την μία γλώσσα προγραμματισμού στην άλλη.

Παράδειγμα

Python 3:

```
age = input("Εισάγετε ηλικία σε έτη: ")
if age < 0:
    print("Η ηλικία δεν μπορεί να είναι μικρότερη του 0.")
elif age < 18:
    print("Λυπάμαι, πρέπει να είστε τουλάχιστον 18.")
elif age < 100:
    print("Ηλικία:", age)
else:
    print("Μας δουλεύετε;")
```

Στο παραπάνω πρόγραμμα, ο χρήστης εισάγει την ηλικία του. Εάν η ηλικία είναι μικρότερη του 0, εμφανίζει «Η ηλικία δεν μπορεί να είναι μικρότερη του 0.», εάν η ηλικία είναι μικρότερη του 18, εμφανίζει «Λυπάμαι, πρέπει να είστε τουλάχιστον 18.», εάν η ηλικία είναι μικρότερη του 100, εμφανίζει «Ηλικία: » και δίπλα την ηλικία που

πληκτρολόγησε ο χρήστης, εναλλακτικά, αν η ηλικία είναι μεγαλύτερη ή ίση του 100, εμφανίζει «Μας δουλεύετε;».

9. Δομές επανάληψης

Τι είναι η δομή επανάληψης

Η δομή επανάληψης είναι το κομμάτι του προγράμματος που καθορίζει πόσες φορές θα τρέξει ένα μέρος του προγράμματος, ελέγχοντας εάν ισχύει μία συνθήκη. Ένα πρόγραμμα μπορεί να περιέχει καμία ή περισσότερες δομές επανάληψης. Η δομές επανάληψης διαφέρουν από την μία γλώσσα προγραμματισμού στην άλλη.

Παράδειγμα

C++:

```
#include <iostream.h>
using namespace std;

int main(){
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++){
        cout << i << endl;
    }
    return 0;
}
```

Στην παραπάνω εικόνα βλέπουμε ένα πρόγραμμα που παίρνει έναν αριθμό από τον χρήστη και αφού ορίζει μία τιμή i ίση με 1, εμφανίζει το i , προσθέτοντας το με 1 στο τέλος κάθε επανάληψης, μέχρι το i να πάψει να είναι μικρότερο ή ίσο του αριθμού που του δώθηκε από τον χρήστη.

10. Δομές δεδομένων

Τι είναι η δομή δεδομένων

Δομή δεδομένων είναι μια μορφή οργάνωσης, διαχείρισης και αποθήκευσης δεδομένων που συνήθως επιλέγεται για αποτελεσματική

πρόσβαση στα δεδομένα. Οι δομές δεδομένων μπορεί να είναι διαφορετικές ανάλογα με την γλώσσα προγραμματισμού.