

Βασικές έννοιες 2ο Μέρος

Εισαγωγή στον Προγραμματισμό

(CEID_NY131)

Βασικές έννοιες
Διαδικαστικού Προστακτικού
Προγραμματισμού
με τη χρήση παραδειγμάτων
Υπολογιστικών διεργασιών

Kleanthis Thramboulidis
Prof. of Software and System Engineering
University of Patras
<https://sites.google.com/site/thramboulidiskleanthis/>



Βασικές έννοιες

Imperative programming
Data Abstraction Statement
Procedural Abstraction
Selection Statement

...τίποτε παραπάνω από
αντιγραφή βασικών εννοιών
από την καθημερινή ζωή

Προγραμματισμός – Βασικές έννοιες

- Διεργασία
- Αλγόριθμος
- Λεκτική περιγραφή
- Πρόγραμμα
- Μεταγλωττιστής (Compiler)
- IDE (Integrated Development Environment)
- Αφαιρετικότητα (Abstraction)
 - Στις διεργασίες (procedural abstraction)
 - Στα δεδομένα (data abstraction)



Αλγόριθμος: ένα σύνολο βημάτων τα οποία καθορίζουν τον τρόπο εκτέλεσης μίας εργασίας.

Οργάνωση Διάλεξης

- **Αφαιρετικότητα**
 - Ορισμός - Είδη
 - Στοιχεία Διεργασίας
- Ανάθεση έργου
 - Η Hailstone sequence (Λεκτική περιγραφή)
 - Η επικοινωνία με την μηχανή
 - Fibonacci numbers

Αφαιρετικότητα

- μια απλοποιημένη περιγραφή ή τεκμηρίωση που δίνει έμφαση σε ορισμένα χαρακτηριστικά ενώ ταυτόχρονα αποσιώπα άλλα.
- μια από τις βασικότερες τεχνικές με την οποία ο άνθρωπος **αντιμετωπίζει την πολυπλοκότητα**.

Είδη αφαιρετικότητας

- αφαιρετικότητα στις διεργασίες (**Procedural abstraction**)
 - "φτιάξε ένα καφέ"
 - «πάρε με τηλέφωνο»
- αφαιρετικότητα στα δεδομένα (**data abstraction**)
 - Δώσε μου ένα "βιογραφικό"
 - Δώσε μου την "αναλυτική καρτέλα σου"

Αφαιρετικότητα στις διεργασίες

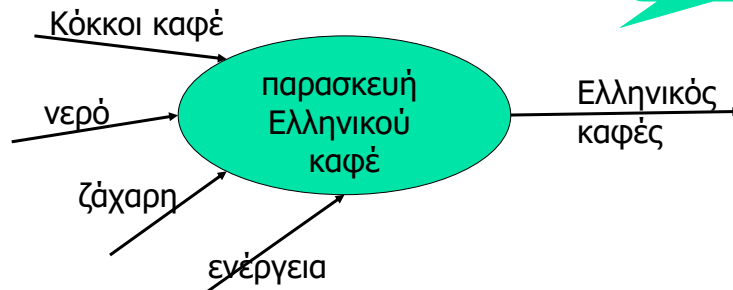
- επιτρέπει στο χρήστη να **εστιάζει τη προσοχή** του
 - στο **τι** γίνεται και όχι
 - στο **πώς** αυτό γίνεται
- επιτυγχάνεται ορίζοντας δύο τμήματα για μια διεργασία
 - τον τρόπο αναφοράς στην διεργασία - **interface**
 - την υλοποίηση της διεργασίας - **implementation**

Θα **γράψεις** 500 φορές «Δεν θα πετάω αεροπλανάκια στην τάξη»



Η διεργασία παρασκευής καφέ

Ένα Ελληνικό
μέτριο
παρακαλώ



- Είναι μια **σύνθετη** (παραγόμενη) **διεργασία** που η υλοποίηση της δίνεται με την χρήση άλλων διεργασιών

- Άλεσε τους κόκκους του καφέ
- Ανάμειξε τα συνθετικά του καφέ
- Βράσε το μείγμα
- ...



Η διεργασία παρασκευής καφέ



- Η διεργασία έχει :

- **Όνομα** (Ορίζει το **ΤΙ ΚΑΝΕΙ** η διεργασία)
- **Περιγραφή** (Ορίζει το **ΠΩΣ ΤΟ ΚΑΝΕΙ**)
- Δέχεται **εισόδους** (τις οποίες επεξεργάζεται)
- Παράγει **έξοδο/δους** (τις οποίες διαμορφώνει με βάση τις εισόδους)

Οργάνωση Διάλεξης

- Αφαιρετικότητα
 - Ορισμός - Είδη
 - Στοιχεία Διεργασίας
- **Ανάθεση έργου**
 - Η Hailstone sequence (Λεκτική περιγραφή)
 - Η επικοινωνία με την μηχανή
 - Fibonacci numbers

Ανάθεση έργου I

Θα γράψεις 500 φορές
«Δεν θα πετάω
αεροπλανάκια στην τάξη»



Ανάθεση έργου II

Γράψε μας την **hailstone sequence** για τον αριθμό 25



The Collatz problem: an unsolved conjecture in mathematics



Hailstone sequence

The Collatz problem: one of the world's unsolved problems in mathematics. It involves a sequence of numbers called a Hailstone sequence. It is called this because the numbers go up and down again. The conjecture is that **these sequences always reach 1**, no matter which positive integer is chosen to start the sequence.



Hailstone Sequences **follow these rules:**

- If a number is even, divide it by 2
- If a number is odd, multiply it by 3 and add 1

For 25: 25 – 76 – 38 – 19 – 58 – 29 – 88 – 44 ...



Hailstone sequence - Λεκτική περιγραφή V1

■ Το έργο

Γράψε μας την **hailstone sequence** για τον αριθμό 25

■ Ανάθεση σε άνθρωπο. **Πώς;**

- A1. Πάρε έναν ακέραιο αριθμό
- A2. Εάν ο αριθμός είναι 1
τερμάτισε
- A3. Εάν ο αριθμός είναι άρτιος
τότε διαίρεσε τον με το 2
διαφορετικά
πολλαπλασίασε τον με το 3 και
πρόσθεσε 1
- A4. Τύπωσε τον n
- A5. Επανάλαβε από το βήμα A2.

Βρες το
λάθος στον
αλγόριθμο

Hailstone sequence - Ψευδοκώδικας

- Δεν αποτελεί λεκτική περιγραφή

- A1. [Input n] Πάρε έναν ακέραιο αριθμό
- A2. [Termination] If $n = 1$
exit
- A3. [Check n] If n is even
 $n := n / 2$
else
 $n := n * 3 + 1$
- A4. [Print n] Τύπωσε τον n
- A5. [Next number] Go to A2.

Η χρήση του
go to καταστρέφει
τη δόμηση του κώδικα

Hailstone sequence -Λεκτική περιγραφή V2

A.1. Ζήτα τον ακέραιο αριθμό

A.2 Πάρε τον αριθμό

A.3 Επανέλαβε τα παρακάτω

*A3.1 Αν ο αριθμός είναι 1
τερμάτισε*

*A3.2 Αν ο αριθμός είναι άρτιος
Διαίρεσε τον με το 2*

Διαφορετικά

*Πολλαπλασίασε τον με το 3 και
πρόσθεσε το 1*

A3.3 Τύπωσε τον αριθμό

A.3: Loop Statement

A3.1: Decision Statement

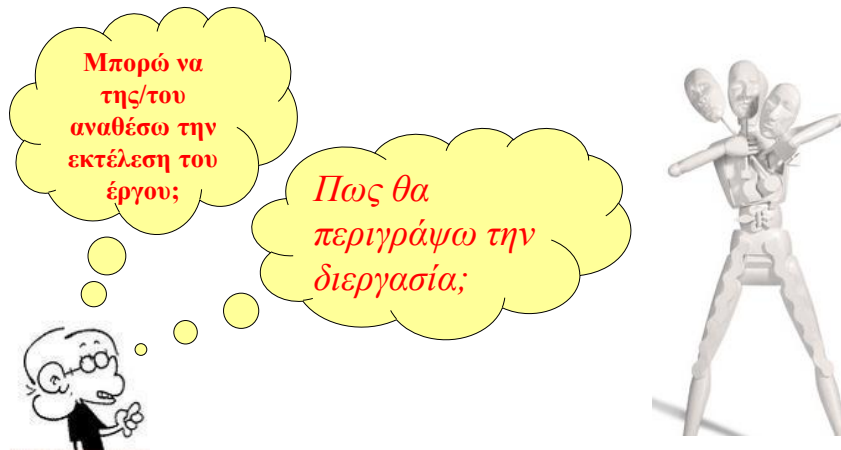
A3.2: Selection Statement

Οργάνωση Διάλεξης

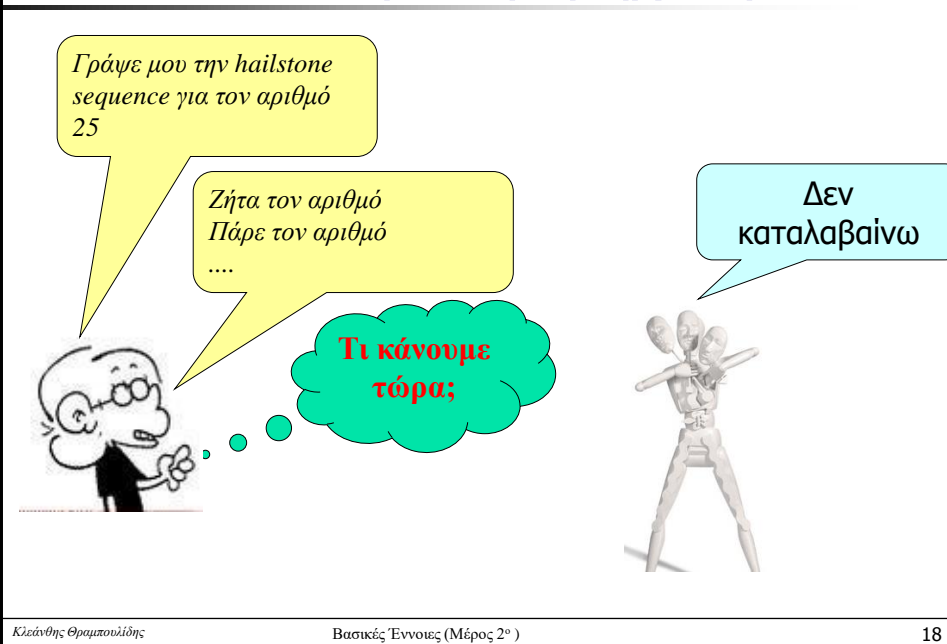
- Αφαιρετικότητα
 - Ορισμός - Είδη
 - Στοιχεία Διεργασίας
- Ανάθεση έργου
 - Η Hailstone sequence (Λεκτική περιγραφή)
 - **Η επικοινωνία με την μηχανή**
 - **Fibonacci numbers**

Η επικοινωνία με την μηχανή 1/2

Το έργο : "Θα γράψεις 500 φορές «Δεν θα πετάω αεροπλανάκια στην τάξη»"



Η επικοινωνία με την μηχανή 2/2



Τεχνικές περιγραφής διεργασιών

- φυσική γλώσσα
- δομημένα αγγλικά (structured English)
- ψευδοκώδικας
- Διαγράμματα ροής (flow-charts)
- Program design languages
- SDL – STD – DFDs – ERDs – Petri Nets ...
- **UML (Unified Modeling Language)**


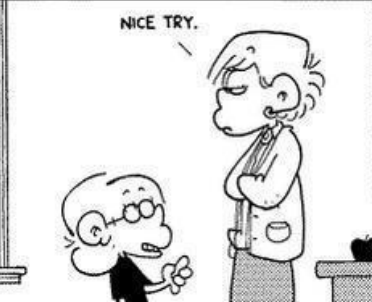
Ερωτήσεις



- Τι πρέπει να διαθέτει το Robot για να του αναθέσω το έργο;
 - **computer model**
- Ποια διαδικασία θα ακολουθήσω για την ανάθεση του έργου;
 - **development process**

Η λύση στο πρόβλημα επικοινωνίας

```
#include <stdio.h>
int main(void)
{
    int count;
    for (count = 1; count <= 500; count++)
        printf("I will not throw paper airplanes in class.");
    return 0;
}
```



Κιλάνθης Θραμπαζίδης

Βασικές Έννοιες (Μέρος 2^ο)

21

Περιγραφή διεργασίας σε C

```
#include <stdio.h> Preprocessor Statement
int main(void) Function Declaration Statement
{
    int count; Variable Declaration Statement
    for (count = 1; count <= 500; count++) Loop Statement
        printf("I will not throw paper airplanes in class."); Function call Statement
    return 0; Control flow Statement
}
```

Κιλάνθης Θραμπαζίδης

Βασικές Έννοιες (Μέρος 2^ο)

22

Fibonacci numbers

- Έργο: **Τύπωσε την ακολουθία αριθμών Fibonacci**
- **Λεκτική περιγραφή** σε ομιλούμενη γλώσσα
 - Ξεκίνα με δυο πρώτους αριθμούς, το 0 και 1
 - Για όσους όρους ζητήθηκε
 - Υπολόγισε τον επόμενο ως το άθροισμα των 2 προηγούμενων του
- Δώσε πηγαίο κώδικα

Υπολογισμός του n Fibonacci αριθμού

- **32-bit x86 machine code**
8B542408 83FA0077 06B80000
0000C383 FA027706 B8010000
00C353BB 01000000 B9010000
008D0419 83FA0376 078BD98B
C84AEBF1 5BC3

Υπολογισμός του n Fibonacci αριθμού

■ x86 assembly language using MASM syntax:

```
fib:
    mov edx, [esp+8]
    cmp edx, 0
    ja @f
    mov eax, 0
    ret

    @@:
    cmp edx, 2
    ja @f
    mov eax, 1
    ret

    @@:
    push ebx
    mov ebx, 1
    mov ecx, 1

    @@:
    lea eax, [ebx+ecx]
    cmp edx, 3
    jbe @f
    mov ebx, ecx
    mov ecx, eax
    dec edx
    jmp @b

    @@:
    pop ebx
    ret
```

Υπολογισμός του n Fibonacci αριθμού

■ C κώδικας

How I feel when my code works

