

Motivation

- Most of the things we will study will *not refer to a specific* programming language.
- Things discussed during these classes can be applied when working in almost any programming language.

Pseudocode

Pseudocode

an artificial and informal language

"text-based" detail (algorithmic) design tool

helps programmers develop algorithms

- *standard* pseudocode statements

(only for this course:)

- *not standard* pseudocode statements

@ - notations for pseudocode not standard statement

ex.: @ *read the array x of length n*

Specifications of subalgorithms

Subalg. name(f.p.l.)

Description: ...

Data: ...

Prec.: ...

Rez.: ...

Post.: ...

or

Subalg. name(f.p.l.)

Prec.: ...

Post.: ...

short

Specifications of subalgorithm

Function *funcname*(f.p.l.)

Description: ...

Data: ...

Prec.: ...

Rez.: *funcname*

Post.: *funcname* = ...

Pseudocode standard statements

if *cond* then

...

else

...

endif

for *count* \leftarrow *iv* , *fv* [, *step*] do

...

endfor

repeat

...

until *cond*

while *cond* do

...

endwhile

read *lista_elem*

print /write *lista_expr*

...

- assignment operator: **:= or ←**

comments

- *// one line comment*
- */* more
 than
 one line comment
/

DS specification

- What (most) programming languages offer?
languages provide basic data types

Convention:

We suppose that we have the next data types:

- *Integer*
- *Real*
- *Boolean*
- *Char*
- *String*

DS specification – notation conventions

Progr. languages provide **data structure builders**

Convention:

We suppose that we have

→ **static one-dimensional array**

array[firstindexvalue .. lastindexvalue] of TElement

$x:array[fi..li]$ of TE means: x_{fi}, \dots, x_{li}

→ **tuple**

Example:

- Vector = Record
 - n: Integer
 - els: array[1..10] of TElement
 - end

More notation conventions

- pointers:

p: ^DT declaration of a pointer **p**
to a data type **DT**

p: ↑DT

[p] access to value pointed by pointer **p**

Conventions: special note on pointers

Mechanism for working with pointers and dynamic variable

Pointer :

domain of values: {valid variable addresses} \cup {NIL}

special value: NIL or NULL – no valid addresses

- pointer type specification: Ex: **p**:[^]Integer
- dynamic variable allocation Ex: ***new*(p)**
***new*(p[20])**
- dynamic variable deallocation Ex.: ***free*(p)**
- value pointed by a pointer Ex.: **[p]**