

Corso di Informatica

Teoria

C1-Sequenza e selezione

M. Malatesta 1-Sequenza e selezione-15

1
05/04/2024

Prerequisiti

- Analizzare un problema
- Scrivere un algoritmo
- Istruzioni semplici
- Espressione logica

M. Malatesta 1-Sequenza e selezione-15

2
05/04/2024

Introduzione

La programmazione a salti crea diverse difficoltà:

- nella lettura dell'algoritmo
- nell'apportare modifiche
- nel mantenere la numerazione dei passi
- nel comprenderne la logica

Esistono altri metodi di rappresentazione degli algoritmi?

Lo scopo di questa Unità è introdurre le **strutture di controllo**, strumenti che servono a realizzare algoritmi, privi degli inconvenienti detti e quindi detti **algoritmi strutturati**.

Rappresentare algoritmi

Gli algoritmi possono essere rappresentati in diversi modi:

- a **passi numerati** (visto in precedenza e poco conveniente)
- mediante **pseudocodifica**
- mediante **diagrammi di flusso**

Ci concentriamo sugli ultimi due tipi di rappresentazione, che consentono di esprimere con chiarezza algoritmi strutturati.

Pseudocodifica

La **pseudocodifica** (o **linguaggio di progetto** o **NLS**, **Notazione Lineare Strutturata**) è uno strumento con il quale non sarà più necessario scrivere algoritmi a passi numerati, né usare istruzioni di salto.

Per tutto il resto del corso useremo questo strumento che permette di rappresentare algoritmi mediante un uso corretto del linguaggio naturale.

Convenzioni di scrittura

- I nomi delle costanti si scrivono in maiuscolo (es. *VALOREMAX*)
- I nomi delle variabili si scrivono in minuscolo (es. *tasso*, *totale*, *cognome*); i nomi composti si scrivono con la sola iniziale maiuscola della seconda parola (es. *calcoloMassimo*)
- Le istruzioni semplici sono: *lettura*, *stampa* e *assegnazione*. Le prime due sono: **Leggi** (*variabile*) e **Stampa** (*variabile*). L'assegnazione si indica con il simbolo “=”.
- Ogni algoritmo inizia con la parola “**Inizio**” e termina con la parola “**Fine**” seguita dal punto (“.”)
- Qualunque elemento generale non specificato va scritto in *corsivo* (es. **Leggi** (*variabile*), oppure *variabile* = $x + 10$).

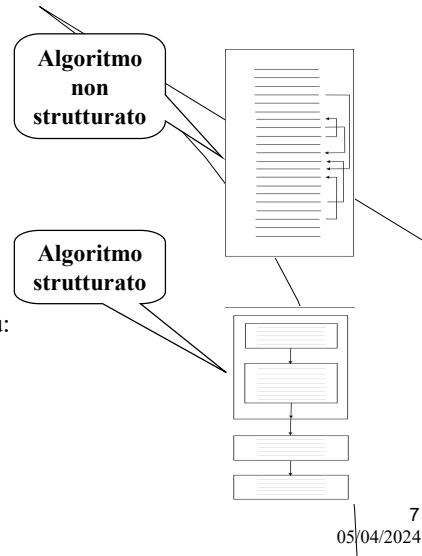
La programmazione strutturata

La **programmazione strutturata** consente di scrivere **algoritmi strutturati** nei quali

ogni blocco di istruzioni abbia un solo ingresso ed una sola uscita

La programmazione strutturata si basa su:

- **istruzioni semplici** (lettura, stampa e assegnazione)
- **strutture di controllo**



M. Malatesta 1-Sequenza e selezione-15

NLS – Le istruzioni semplici Lettura

La parola chiave **Leggi** indica all'esecutore di *acquisire un dato* ossia di svolgere un'operazione di immissione (input)

Esempio:

Algoritmo Somma

Intero valore;

Inizio

Leggi (valore);

Fine.

Pone nella variabile *valore* ciò che viene immesso con la tastiera.

M. Malatesta 1-Sequenza e selezione-15

8
05/04/2024

NLS – Le istruzioni semplici

Scrittura

La parola chiave **Scrivi** indica all'esecutore di *emettere un dato* ossia di svolgere un'operazioni di emissione (output)

Esempio:

Algoritmo Somma

Reale risultato;

Inizio

...

Scrivi (risultato);

....

Fine.

Emette a video il contenuto della variabile *risultato*.

M. Malatesta 1-Sequenza e selezione-15

9
05/04/2024

NLS – Le istruzioni semplici

Assegnazione

Il simbolo "=" svolge l'operazione di assegnazione e serve per porre in una variabile il risultato di un calcolo.

Esempio:

Algoritmo Somma

Intero valore1, valore2, risultato;

Inizio

Leggi (valore1);

Leggi (valore2);

risultato = valore1 + valore2;

Scrivi (risultato);

Fine.

Acquisisce due valori, ne calcola la somma e la stampa a video.

M. Malatesta 1-Sequenza e selezione-15

10
05/04/2024

NLS - Le strutture di controllo

Per programmare in modo strutturato occorre che gli algoritmi siano scritti usando esclusivamente le **strutture di controllo (SDC)** seguenti:

- **sequenza**
- **selezione**
- **iterazione**

Attraverso le strutture di controllo:

- possiamo alterare il flusso di esecuzione sequenziale del programma;.
- evitiamo la programmazione a salti.

In questa Unità vedremo le prime due strutture di controllo.

NLS – Le strutture di controllo Sequenza

Tutte le istruzioni presenti nella parte eseguibile dell'algoritmo:

- formano una **sequenza**, nel senso che si presentano e vengono eseguite una dopo l'altra;
- costituiscono un **blocco** che può essere considerato come un'unica istruzione;
- sono contenute tra le due parole "**Inizio**" e "**Fine**".

In NLS le due parole "**Inizio**" e "**Fine**" che abbiamo finora utilizzato prendono il nome di SDC **sequenziale** o **sequenza**.

NLS – Le strutture di controllo

Sequenza

La sequenza ha in NLS la seguente **sintassi**::

Algoritmo *nome*

Inizio

istruzione1;

istruzione2;

.....

Fine.

La SDC **sequenziale** o **sequenza** è sempre presente in tutti gli algoritmi

Il rientro dei margini prende il nome di **indentazione** ed ha lo scopo di facilitare la scrittura e la lettura dell'algoritmo

NLS – Le strutture di controllo

Sequenza

Problema: Trovare l'area di un rettangolo conoscendo la base e l'altezza.

ATTIVITA': scrivere l'analisi del testo

Fase 1

Analisi del testo

Il problema chiede di calcolare l'area di un rettangolo, che è una figura geometrica regolare piana, formata da quattro lati, uguali e paralleli a due a due, disposti in modo da formare quattro angoli retti. Il rettangolo è caratterizzato dalle misure della base *b* e dell'altezza *h* che consideriamo variabili intere per semplicità. Il risultato del problema sarà la variabile *area* che si calcola con la formula

$$\text{area} = b * h$$

NLS – Le strutture di controllo

Sequenza

ATTIVITA': disegnare la tabella delle variabili

Dati	Nome	Tipo	Significato
INPUT	<i>b</i>	Intero	base del rettangolo
	<i>h</i>	Intero	altezza del rettangolo
OUTPUT	<i>area</i>	Intero	area del rettangolo

ATTIVITA' : scrivere il procedimento risolutivo.

Procedimento risolutivo:

Si immettono i dati in input. Si calcola il valore di *area* con la formula precedente e si stampa.

NLS – Le strutture di controllo

Sequenza

ATTIVITA': scrivere l'algoritmo corrispondente in NLS (Fase 2)

Dichiarazione delle variabili usate

Algoritmo Rettangolo

Inizio

```
Intero b, h, area;
Leggi(b); /* lettura dati*/
Leggi(h);
area = b * h; /* calcolo area */
Stampa(area);
```

Fine.

Commenti

Passo	b	h	area
1	3		
2		2	
3			6

ATTIVITA': disegnare la tabella di traccia per l'istanza $a=3$, $b=2$.

I **commenti** scritti dal programmatore sono la documentazione dell'algoritmo e sono indispensabili per capirlo e/o modificarlo.

NLS – Le strutture di controllo

Selezione doppia

Quando si vuole che il programma segua un percorso diverso a seconda del valore di una **condizione** o **espressione logica** (vero o falso) si utilizza la SDC **selettiva doppia** che ha in NLS la sintassi seguente:

```
Se (espressione-logica)  
    istruzione1;  
Altrimenti  
    istruzione2;
```

Se *espressione-logica* ha valore **VERO** viene eseguita *istruzione1*, **altrimenti** viene eseguita *istruzione2*.
La SDC **selettiva doppia** è caratterizzata dalle parole **Se-Altrimenti**.

NLS – Le strutture di controllo

Selezione doppia

Problema: Dati due valori interi, stampare il minimo.

Fase 1 **ATTIVITA'**: scrivere l'analisi del testo

Analisi del testo

Indichiamo con *a* e *b* i numeri interi da confrontare e con *min* il valore intero del minimo. È sufficiente saper confrontare valori interi per trovare il minimo; il problema è solubile e il procedimento può essere facilmente verificato.

ATTIVITA': disegnare la tabella delle variabili

	Nome	Tipo	Significato
INPUT	a	Intero	Primo valore
	b	Intero	Secondo valore
OUPUT	min	Intero	Valore minimo

NLS – Le strutture di controllo

Selezione doppia

Algoritmo StampaMinimo

Inizio

Intero a,b,min;

Leggi(a);

Leggi(b);

Se (a<b)

min=a;

Altrimenti

min=b;

Stampa(min);

Fine

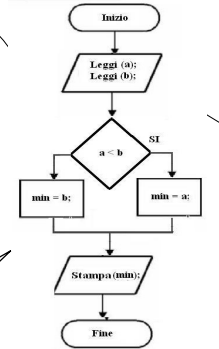
Se l'espressione logica ha valore **VERO** (*a* è minore di *b*) il *min* sarà *a*, altrimenti sarà *b*.

Passo	a	b	min
1	3		
2		2	
3			2

ATTIVITA': disegnare la tabella di traccia per l'istanza *a=3, b=2*.

ddf dell'algoritmo

ATTIVITA': scrivere l'algoritmo in NLS (Fase 2)



M. Malatesta 1-Sequenza e selezione-15

19
05/04/2024

NLS – Le strutture di controllo

Selezione semplice

In alcuni casi, la struttura selettiva si può utilizzare nella forma seguente che prende il nome di SDC **selettiva semplice**. In pseudocodice NLS la sintassi è:

Se (*espressione-logica*)
istruzione;

Se l'*espressione-logica* risulta **VERA** viene eseguita *istruzione*; se **FALSA** viene saltata *istruzione*.

Comunque, si passa ad eseguire l'*istruzione* successiva.

M. Malatesta 1-Sequenza e selezione-15

20
05/04/2024

NLS – Le strutture di controllo

Selezione semplice

Problema: Verificare la legge di annullamento del prodotto (il prodotto di due valori è zero se almeno uno dei due è zero).

ATTIVITA': disegnare la tabella delle variabili

	Nome	Tipo	Significato
INPUT	a	Intero	Primo valore
	b	Intero	Secondo valore
OUTPUT	MSG	Stringa	“Legge verificata”

Una variabile di tipo **Stringa** è utilizzata per contenere sequenze di caratteri.

NLS – Le strutture di controllo

Selezione semplice

Algoritmo VerificaAnnullamentoProdotto

Costante MSG “Legge verificata”

Inizio

Intero a, b, c;

Leggi(a);

Leggi(b);

c=a*b;

Se (((a==0) **OR** (b==0)) **AND** (c==0))

Stampa(MSG);

Fine

MSG viene definita come costante di tipo stringa contenente un messaggio

ATTIVITA': scrivere l'algoritmo in NLS (**Fase 2**)

Passo	a	b	c	msg
1	3			“Legge verificata”
2		0		“Legge verificata”
3			0	“Legge verificata”

L'operatore logico **OR** consente di verificare se *almeno una delle due condizioni* è vera. L'operatore **AND** verifica che *siano vere entrambe le condizioni*.

ATTIVITA': disegnare la tabella di traccia per l'istanza a=3, b=0

NLS – Le strutture di controllo

Selezione multipla

Quando la scelta deve essere fatta tra molti valori si usa una struttura di controllo, che prende il nome di SDC **selettiva multipla** e che in pseudocodice ha la sintassi:

Nel caso che (*espressione*) **sia**

valore_1: *istruzione1*;

valore_2: *istruzione2*;

.....

Altrimenti: *istruzione*;

Fine;

A seconda che il valore di *espressione* sia *valore_1* o *valore_2*, o....., esegue *istruzione1*, o *istruzione2*, ecc. ecc. altrimenti esegue *istruzione*.

Questa prende il nome di SDC **selettiva multipla**

NLS – Le strutture di controllo

Selezione multipla

Nel caso che (*giorno*) **sia**

1: **Stampa**("Lunedì");

2: **Stampa**("Martedì");

3: **Stampa**("Mercoledì");

4: **Stampa**("Giovedì");

5: **Stampa**("Venerdì");

6: **Stampa**("Sabato");

7: **Stampa**("Domenica");

Altrimenti

Stampa("Non so!");

Fine;

OSSERVAZIONE: In questo esempio, per brevità abbiamo mostrato solo la parte relativa alla SDC selettiva multipla.

Quando viene trovata una corrispondenza di valori tra la variabile *giorno* e uno dei valori numerici elencati, viene eseguita l'istruzione. Ad esempio, se *giorno* valesse 2, verrebbe stampato "Martedì"

La parola **Altrimenti** in questo caso, serve a trattare il caso in cui il valore di *giorno* non sia corrispondente a nessuno dei valori elencati.

Connettivi logici

Nell'attività di programmazione una scelta viene fatta sulla base di una **espressione logica** o **condizione**, ossia di una frase che può essere soltanto **VERA** o **FALSA**.

Una condizione si dice **condizione composta** se è formata da più condizioni legate con i **connettivi logici**.

Tabelle di verità

- **AND** detto **prodotto logico** (o **coniunzione** e corrisponde alla "e")

AND	0	1
0	0	0
1	0	1

- **OR** detto **somma logica** (o **disgiunzione** e corrisponde alla "o")

OR	0	1
0	0	1
1	1	1

- **NOT** detto **negazione** e corrisponde al "non"

NOT	
0	1
1	0

Condizioni composte

Attraverso i connettivi è possibile costruire e valutare **espressioni logiche composte**.

Ad esempio, se $a = 3$ e $b = 10$ le seguenti espressioni hanno i valori indicati nella **tabella di verità**.

	1° termine	2° termine	Valore
$(a=b) \text{ e } (b>0)$	0	1	0
$(a < b) \text{ o } (b=1)$	1	0	1
$b > a + 10$			0

M. Malatesta 1-Sequenza e selezione-15

27
05/04/2024

I diagrammi di flusso (ddf)

I **diagrammi di flusso (ddf)** sono un altro modo per rappresentare gli algoritmi.

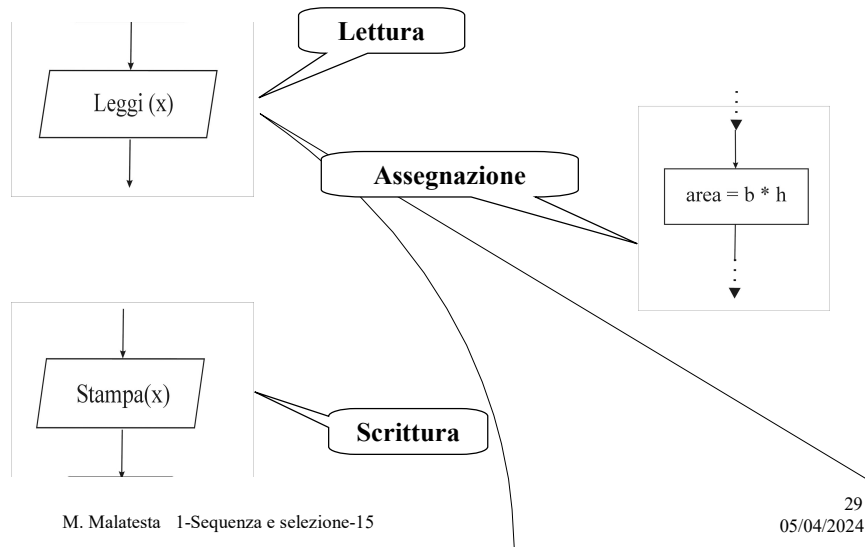
Questa tecnica fa uso di simboli grafici che ora descriviamo.

L'uso di disegni è a volte poco pratico, specialmente nel caso di algoritmi complessi o lunghi, nei quali è preferibile l'impiego della pseudocodifica.

M. Malatesta 1-Sequenza e selezione-15

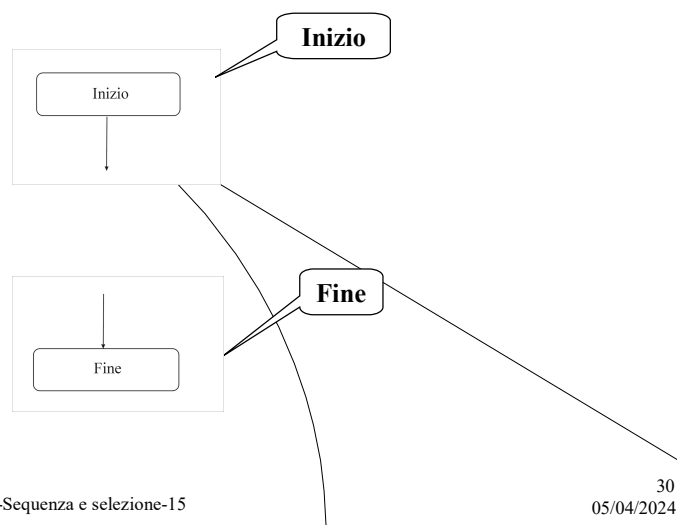
28
05/04/2024

DDF – Le istruzioni semplici



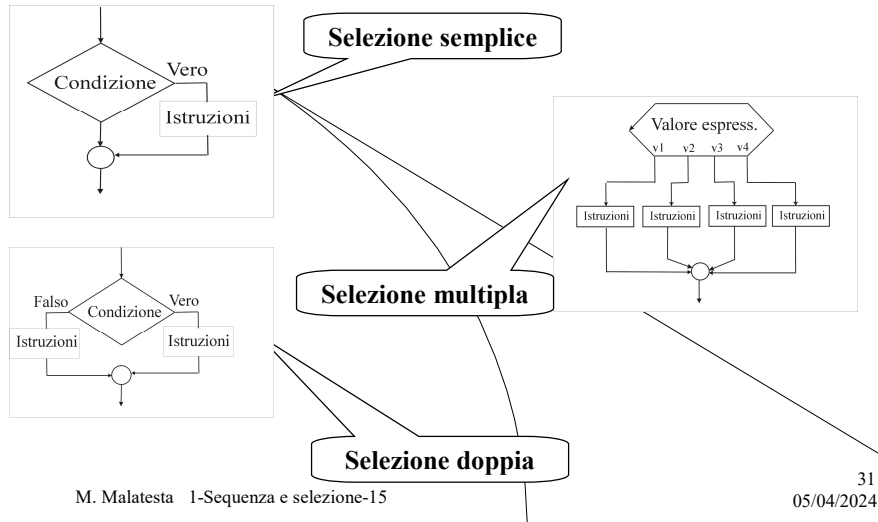
DDF – Le strutture di controllo

Sequenza



DDF – Le strutture di controllo

Selezione semplice e doppia



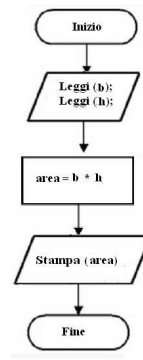
I diagrammi di flusso

Problema: Trovare l'area di un rettangolo conoscendo la base e l'altezza

ATTIVITA': scrivere il ddf corrispondente.

Problema: Dati due valori interi, stampare il minimo.

ATTIVITA': scrivere il ddf corrispondente.



Argomenti

- Rappresentazione algoritmi
- Pseudocodifica
- Convenzioni di scrittura
- La programmazione strutturata
- NLS - Le istruzioni semplici
 - Lettura
 - Scrittura
 - Assegnazione
- NLS - Le strutture di controllo
 - Sequenza
 - Selezione doppia
 - Selezione semplice
 - Selezione multipla
- Connettivi logici
- Tabelle di verità
- Condizioni composte
- I diagrammi di flusso (ddf)
- DDF – Le istruzioni semplici
- DDF – Le strutture di controllo
 - Sequenza
 - Selezione semplice e doppia

M. Malatesta 1-Sequenza e selezione-15

33
05/04/2024

Altre fonti di informazione

- P.Gallo, F.Salerno – Informatica Generale 1, ed. Minerva Italica
- G.Callegarin – Corso di Informatica 1, ed. CEDAM

M. Malatesta 1-Sequenza e selezione-15

34
05/04/2024