

Abilità Informatiche

Luigi Catuogno

[luigi.catuogno@uniparthenope.it]

Corso di Laurea in Economia e commercio - Anno Accademico 2022-23

1

Libro di testo

[IdB]

Dennis P. Curtin, Kim Foley, Kunal Sen, Cathleen Morin

Informatica di base

VII edizione (2016), MacGraw Hill Education

ISBN: 978-88-386-1537-5

2

Altro materiale di utile consultazione

[Sli]

Slides, appunti e altro materiale distribuito dal docente

[Misc]

Altra fonte diversamente specificata di volta in volta

3

Tipi di dati e operatori

4

Tipo di dati

Insieme dei possibili valori che un dato può assumere

e.g. I numeri interi, {True, False}, ... caratteri alfabetici,...

Modalità di rappresentazione

e.g. impiego di caratteri numerici, «parole chiave», delimitatori, aggregazione di valori di altro tipo...

5

Tipo di dati

Insieme dei possibili valori che un dato può assumere

e.g. I numeri interi, {True, False}, ... caratteri alfabetici,...

Modalità di rappresentazione

e.g. impiego di caratteri numerici, «parole chiave», delimitatori, aggregazione di valori di altro tipo...

0, 1112, -9283, ... *I numeri interi sono rappresentati da sequenze di simboli numerici*

True, False *Molti linguaggi di programmazione rappresentano i valori booleani mediante parole chiave (keywords).*

«A», «5», «?», ... *I caratteri «alfanumerici» rappresentano il tipo di dato con cui si compone l'informazione testuale.*

6

Tipo di dati

Insieme dei possibili valori che un dato può assumere

e.g. I numeri interi, {True, False}, ... caratteri alfabetici,...

Modalità di rappresentazione

e.g. impiego di caratteri numerici, «parole chiave», delimitatori, aggregazione di valori di altro tipo...

**Tipi di dati
aggregati**

(1200 , «\$») , ...

Un ipotetico tipo «importo di danaro» potrebbe aggregare un valore numerico e un carattere alfanumerico che ne indica la valuta

«Gennaio Esposito» , ...

L'informazione testuale si rappresenta mediante «stringhe», cioè sequenza di elementi di tipo «carattere»

7

Tipo di dati

Insieme dei possibili valori che un dato può assumere

e.g. I numeri interi, {True, False}, ... caratteri alfabetici,...

Modalità di rappresentazione

e.g. impiego di caratteri numerici, «parole chiave», delimitatori, aggregazione di valori di altro tipo...

Codifica

In che modo il calcolatore rappresenta questi dati al suo interno...

8

Tipo di dati

Insieme dei possibili valori che un dato può assumere

e.g. I numeri interi, {True, False}, ... caratteri alfabetici,...

Le «operazioni elementari»: gli operatori definiti sul tipo

e.g. somma e prodotto per gli interi, AND e OR per i booleani...

Definizione:

Quanti e quali operandi sono richiesti/ammessi;

Di che tipo è di risultato prodotto

In che modo si determina il risultato

9

Tipi di dati: operatori

La natura e la varietà degli operatori dipendono dal tipo di dato per cui sono definiti.

Operatori aritmetici

Definiti sui tipi di dati numerici (interi, reali, ...)

Implementano le operazioni di somma, differenza, prodotto...

Operatori logici

Definiti sul tipo di dati booleano (vero/falso)

Forniscono gli operatori fondamentali dell'algebra booleana: and, or, not...

Gli operatori *relazionali* hanno delle caratteristiche comuni indipendentemente dal tipo su cui sono definiti...

10

Tipi di dati: una galleria

11

Tipi di dati numerici: interi

L'insieme dei numeri interi \mathbb{Z} è infinito, quindi, un qualsiasi sistema di calcolo è in grado di rappresentarne e gestirne solo un sottoinsieme finito.

Tale insieme è costituito dai numeri compresi tra un *massimo* e un *minimo* intero rappresentabile.

$$\mathbb{Z}' = \{n \in \mathbb{Z} \mid \text{minimo} \leq n \leq \text{massimo}\}$$

Le dimensioni dell'insieme dipendono dalle caratteristiche costruttive del calcolatore e dal modo utilizzato per la codifica dei numeri.

12

Tipi di dati numerici: interi

Il calcolatore fornisce un sistema per rappresentare qualsiasi intero mediante una *costante numerica*: una combinazione arbitraria delle cifre 0,1,2,3,4,5,6,7,8,9 e dei simboli + e - ,il cui valore sia compreso tra il minimo e il massimo.

Costanti numeriche:

Interi: **1, 200394, -47, 0, 592, +62**

13

Tipi di dati numerici: interi

Sul tipo intero sono definiti gli operatori aritmetici e di segno:

Operatori aritmetici:

Moltiplicativi: *,/ (*moltiplicazione e divisione*)

Additivi: +, - (*somma e sottrazione*)

Segno algebrico:

Operatori unari: + e -

14

Tipi di dati numerici: interi

L'operatore *somma* definito tra numeri interi

E' rappresentato dal simbolo $+$

A *due* numeri *interi* a e b scelti arbitrariamente,
associa il numero *intero* $c = a + b$

$$2 + 2 = 4$$

$$120 + 16 = 136$$

15

Tipi di dati numerici: interi

Osserviamo che:

La divisione tra due numeri interi, non sempre è un numero intero (e.g. $3/2$).
L'operatore di divisione tra interi produce solo la *parte intera* del risultato.

$$15/2 = 7, 2/3 = 0, 5/3 = 1, \dots$$

Infatti, spesso ci si riferisce a questo operatore come alla *divisione intera* o *operatore quoziente*.

16

Tipi di dati numerici: interi

Diversi linguaggi di programmazione forniscono un ulteriore operatore che restituisce il *resto* di una divisione tra interi. Tale operatore: il *modulo* è spesso rappresentato dal simbolo %

Operatore *modulo*:

15% 2 = 1, 5% 3 = 2, 13% 7 = 6, ...

Ricordiamo che:

$$a : b = q + r$$

dove:

$$a = q \times b + r$$

e:

$$0 \leq r < b$$

17

Tipi di dati numerici: interi

Diversi linguaggi di programmazione forniscono un ulteriore operatore che restituisce il *resto* di una divisione tra interi. Tale operatore: il *modulo* è spesso rappresentato dal simbolo %

Operatore *modulo*:

15% 2 = 1, 5% 3 = 2, 13% 7 = 6, ...

Per esempio:

44 / 6 = 7 (divisione intera): a = 44, b = 6, q = 7

44% 6 = 2 (modulo): a = 44, b = 6, r = 2

Ricordiamo che:

$$a : b = q + r$$

dove:

$$a = q \times b + r$$

e:

$$0 \leq r < b$$

18

Tipi di dati numerici: interi

Diversi linguaggi
restituisce il *resto*
rappresentato dal s

$$15\% 2 = 1, 5\% 3$$

Per esempio:

$$44 / 6 = 7 \text{ (divisione)}$$

$$44 \% 6 = 2 \text{ (modulo)}$$



$$6 \times 7 = 42$$

$$42 + 2 = 44$$

G. Casarini «Quarantaquattro gatti», cantata da
Barbara Ferigo alla X edizione dello «Zecchino
d'Oro», 1968

pre che
e spesso

Ricordiamo che:

$$a : b = q + r$$

dove:

$$a = q \times b + r$$

e:

$$0 \leq r < b$$

19

Tipi di dati numerici: reali

L'insieme dei numeri reali \mathbb{R} è infinito, e *continuo* quindi qualsiasi calcolatore ne
rappresenta una versione «approssimata» \mathbb{R}' ...

Che contiene i numeri reali compresi tra un *minimo* e un *massimo*

Nel quale due numeri che differiscono per un valore inferiore a una data
precisione, «non sono distinguibili»

Le dimensioni e la precisione dell'insieme dipendono dalle caratteristiche
costruttive del calcolatore e dal modo utilizzato per la *codifica* dei numeri.

20

Tipi di dati numerici: reali

Come per gli interi, il calcolatore fornisce un sistema per rappresentare qualsiasi reale mediante una *costante numerica*: una combinazione arbitraria delle cifre 0,1,2,3,4,5,6,7,8,9 dei simboli +, - e della «virgola» (spesso indicata dal .)

Costanti reali:

Reali: **0.1, 200.394, -4.7, 0.000000592, 0 ...**

21

Tipi di dati numerici: reali

In molti casi, è resa disponibile un modo alternativo per rappresentare i reali: la notazione esponenziale (a volte detta «notazione scientifica»)

Notazione esponenziale:

Reali: **21E6, 5E - 6, -1233E - 2**

Ricordiamo che:

$$n E b = n \times 10^b$$

per esempio:

$$21E6 = 21 \times 10^6 = 21000000$$

e:

$$5E - 6 = 5 \times 10^{-6} = \frac{5}{10^6} = \frac{1}{200000} = 0.000005$$

22

Tipi di dati numerici: reali

Sul tipo reale sono definiti gli operatori aritmetici e di segno:

Operatori aritmetici:

Moltiplicativi: *, / (*moltiplicazione e divisione*)

Additivi: +, - (*somma e sottrazione*)

Segno algebrico:

Operatori unari: + e -

23

Tipo di dati booleano

Il tipo *Booleano* è basato su un insieme che contiene due elementi

$$\mathbb{B} = \{vero, falso\}$$

E' utilizzato per esprimere misure che possono avere solo due valori (e.g. *acceso/spento, vero/falso, presente/assente, si/no...*)

E' utilizzato per esprimere il risultato di operatori che analizzano il verificarsi o meno di una determinata condizione, il soddisfacimento di una proprietà...

24

Tipo di dati booleano

I valori booleani possono essere rappresentati mediante le costanti simboliche usuali (o a loro sinonimi in base alla convenzione adottata) oppure utilizzando le costanti numeriche 0 e 1 rispettivamente per *falso* e *vero*

Costanti booleane:

Notazione simbolica: ***True, False***

Notazione numerica: **1, 0**

In alcuni casi, la costante *false* è espressa con lo 0 mentre, il *true* è rappresentato da qualsiasi valore numerico diverso da zero.

25

Tipo di dati booleano

Sul tipo booleano sono definiti alcuni operatori generalmente detti «logici»:

Operatori logici:

AND: $\mathbb{B}^2 \rightarrow \mathbb{B}$

OR: $\mathbb{B}^2 \rightarrow \mathbb{B}$

NOT: $\mathbb{B} \rightarrow \mathbb{B}$

x_1	x_2	OR (x_1, x_2)	AND (x_1, x_2)	NOT (x_1)
<i>Falso</i>	<i>Falso</i>	<i>Falso</i>	<i>Falso</i>	<i>Vero</i>
<i>Falso</i>	<i>Vero</i>	<i>Vero</i>	<i>Falso</i>	<i>Vero</i>
<i>Vero</i>	<i>Falso</i>	<i>Vero</i>	<i>Falso</i>	<i>Falso</i>
<i>Vero</i>	<i>Vero</i>	<i>Vero</i>	<i>Vero</i>	<i>Falso</i>

26

Tipo di dati booleano

Sul tipo booleano sono definiti alcuni operatori generalmente detti «logici»:

$OR(vero, falso) \rightarrow vero;$

$AND(falso, vero) \rightarrow falso;$

$NOT(falso) \rightarrow vero;$

$AND(vero, vero) \rightarrow ?;$

x_1	x_2	$OR(x_1, x_2)$	$AND(x_1, x_2)$	$NOT(x_1)$
Falso	Falso	Falso	Falso	Vero
Falso	Vero	Vero	Falso	Vero
Vero	Falso	Vero	Falso	Falso
Vero	Vero	Vero	Vero	Falso

27

Operatori relazionali

28

Relazioni

Informalmente, in un generico insieme, una *relazione* è un criterio arbitrario che permette di aggregarne gli elementi in gruppetti di grandezza fissa n (n -ple).

Per esempio, nell'insieme $\{1,2,3\}$ (i numeri interi compresi tra 1 e 3) la relazione *binaria* \leq (minore o uguale) determina le seguenti coppie:

$(1,1)$, $(1,2)$, $(1,3)$, $(2,2)$, $(2,3)$ e $(3,3)$.

29

Esempio

L'operatore relazionale \leq , prende una coppia di numeri interi n_1, n_2 e restituisce il valore booleano *vero* se n_1 è minore o uguale a n_2 oppure *falso* in caso contrario.

$$\leq (n_1, n_2) = \begin{cases} \textit{vero} & \text{se } n_1, n_2 \text{ sono in relazione} \\ \textit{falso} & \text{altrimenti} \end{cases}$$

(generalmente: $n_1 \leq n_2$)

30

Esempio

Supponiamo di avere la sequenza di dati di tipo intero:

$$n_1 = 7, n_2 = 41, n_3 = 7, n_4 = 11, n_5 = 58 \text{ e } n_6 = 41,$$

L'operatore di uguaglianza $==$ darà i seguenti risultati:

$$\begin{aligned} n_1 == n_3 &\rightarrow \textit{vero}; \\ n_6 == n_2 &\rightarrow \textit{vero}; \\ n_2 == n_3 &\rightarrow \textit{falso}; \\ \dots & \\ n_3 == n_3 &\rightarrow \textit{vero}; \\ \dots & \end{aligned}$$

31

Operatori relazionali

Un *operatore relazionale* Op_{rel} «indica» se tra gli elementi di una n -pla x_1, \dots, x_n dell'insieme X sussista la relazione rel .

$$Op_{rel}: X^n \rightarrow \{\textit{vero}, \textit{falso}\}$$

Restituisce il valore booleano *vero* se x_1, \dots, x_n sono associati dalla relazione rel oppure *falso* in caso contrario.

$$Op_{rel}(x_1, \dots, x_n) = \begin{cases} \textit{vero} & \text{se } x_1, \dots, x_n \text{ sono in relazione} \\ \textit{falso} & \text{altrimenti} \end{cases}$$

33

Operatori relazionali

Tipici operatori relazionali sono gli operatori di uguaglianza:

Uguale	=	==
Diverso	≠	!=, <>

e gli operatori di confronto, che sottendono all'esistenza di una *relazione d'ordine* tra gli elementi del loro dominio:

Minore	<	<
Maggiore	>	>
Minore o uguale	≤	<=
Maggiore o uguale	≥	>=

34

Operatori relazionali (ordinamento)

Il significato di operatore d'ordine dipende dalla natura dell'insieme di elementi che costituisce il tipo su cui è definita

Può derivare dalle proprietà *«algebriche»* dell'insieme stesso:

Per esempio, nei numeri interi, gli operatori ricalcano l'usuale ordinamento

$$\{-10, -99, \dots, 0, 1, 2, 3, 4, \dots\}: -100 \leq -99 \leq \dots \leq 0 \leq 1 \leq 2 \leq 3 \leq 4 \dots$$

Può essere determinato arbitrariamente in fase di definizione:

Tra gli elementi del tipo *carattere* che comprende i simboli «tipografici» che compongono il testo, $\{ '1', '2', \dots, 'a', 'b', \dots \}$: è definita l'ordinamento *lessicografico* $'1' \leq '2' \leq \dots \leq 'a' \leq 'b' \dots$ che dipende dall'ordine in cui i simboli sono enumerati.

35

Operatori relazionali (ordinamento)

Nei tipi in cui è definito un ordinamento, sono spesso definiti degli operatori come:

Predecessore e successore:

Nell'insieme dei numeri interi \mathbb{Z} , per qualsiasi x :

$$pre(x) \rightarrow x - 1;$$

$$succ(x) \rightarrow x + 1;$$

36

Operatori relazionali (ordinamento)

Nei tipi in cui è definito un ordinamento, sono spesso definiti degli operatori come:

Ordine (o posizione):

Nell'insieme ordinato $X = \{x_1, \dots, x_n\}$, per qualsiasi x :

$$ord(x_i) \rightarrow i;$$

Indica la posizione che l'elemento ha nell'insieme.

ESEMPIO: nel tipo carattere, i simboli alfanumerici si susseguono secondo l'ordine in cui appaiono nella tabella definita dal codice ASCII. In questo tipo, l'elemento 'A' occupa la 66-esima riga, quindi:

$$ord('A') = 65;$$

37

Tipi di dati aggregati

38

Ricordiamo che:

- Una **struttura dati** rappresenta la «*realizzazione*» di un tipo di dati
 - Definizione dell'insieme degli elementi da cui sono scelti i dati
 - Definizione del metodo di rappresentazione degli elementi
 - Definizione delle operazioni definite sul dato stesso
- In base alla natura dell'elaborazione da effettuare
- Alle caratteristiche del sistema di elaborazione (del computer)

39

Tipo di dati: atomico vs. aggregato

- **Atomici**

- Valori non scomponibili in elementi più semplici.
 - Possono essere definiti
 - Mediante l'enumerazione delle costanti appartenenti al tipo
 - Definizione delle proprietà di cui le costanti godono
- Gli operatori possono essere definiti:
 - Mediante una tabella
 - Mediante assiomi

40

Tipi di dati atomici: esempi

- **Tipo booleano**

- Valori: $D = \{Vero, Falso\}$

- **Operatori:**

- $OR(x_1, x_2)$
- $AND(x_1, x_2)$
- $NOT(x_1)$

x_1	x_2	$OR(x_1, x_2)$	$AND(x_1, x_2)$	$NOT(x_1)$
<i>Falso</i>	<i>Falso</i>	<i>Falso</i>	<i>Falso</i>	<i>Vero</i>
<i>Falso</i>	<i>Vero</i>	<i>Vero</i>	<i>Falso</i>	<i>Vero</i>
<i>Vero</i>	<i>Falso</i>	<i>Vero</i>	<i>Falso</i>	<i>Falso</i>
<i>Vero</i>	<i>Vero</i>	<i>Vero</i>	<i>Vero</i>	<i>Falso</i>

41

Tipo di dati: atomico vs. aggregato

- **Aggregato (o strutturato)**
 - Valori composti da elementi più semplici, in altre parole, definiti «aggregando» dati di altro tipo
 - **Array e stringhe**: sequenze n-dimensionali (lunghezza arbitraria) di dati di tipo omogeneo (array e matrici)
 - **Record**: aggregato di lunghezza costante di dati anche di tipo disomogeneo

42

Tipi di dati aggregati: array

- Un array ha:
 - Un nome;
 - Una lunghezza (o cardinalità) finita;
 - Un *tipo base* (il tipo di tutti gli elementi che contiene);
 - Un operatore di accesso [...] (parentesi quadre)



43

Tipi di dati aggregati: array

Si può immaginare un array come una scaffalatura in cui ciascuno scomparto:

- 1) contiene un solo *elemento* (un dato) del tipo base;
- 2) è contrassegnato da un numero (indice) che ne indica la posizione nella scaffalatura



44

Tipi di dati aggregati: array

In un array:

- 1) Per leggere o memorizzare un dato nell'array è necessario indicare la sua posizione con l'*operatore di accesso* (le `[]`);



45

Esempi di Array

Rappresentiamo nell'array *estrazioni* i 6 numeri dell'ultima estrazione del Superenalotto

<i>indice</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
<i>elementi</i>	78	11	53	2	36	26

46

Esempi di Array

Rappresentiamo nell'array *estrazioni* i 6 numeri dell'ultima estrazione del Superenalotto

<i>indici</i>	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
<i>elementi</i>	78	11	53	2	36	26

Nome: *estrazioni*
 Cardinalità: 6
 Tipo base: *numeri interi*

Il primo numero estratto, è contenuto nel primo scomparto (indice 0), il secondo nello scomparto con indice 1 e così via...

```
Primo_estratto = estrazioni[0];
Ultimo_estratto = estrazioni[5];
```

Operatore di accesso:
nome_array [*indice*]

47

Tipi di dati aggregati : stringhe

- Il tipo di dati «*carattere*» è un tipo di dati *atomico* cui appartengono i simboli alfanumerici (e.g. i simboli del codice ASCII).
- In alcuni linguaggi di programmazione, sono delimitati da *singoli apici* (l'apostrofo)

$$\mathbb{A} = \{'!', '\$', \dots, '1', '2', \dots, 'a', 'b', \dots\}$$

48

Tipi di dati aggregati : stringhe

Le «*stringhe*» sono una sequenza di *caratteri* di lunghezza arbitraria

- 1) Sono generalmente delimitate dai *doppi apici* (le virgolette)

sono stringhe:

«*a*», «*aa*», «*12b*», «*kI\$*», ... , «» (stringa vuota)

49

Tipi di dati aggregati : stringhe

- Tipicamente una stringa può essere vista come un array di caratteri
 - In una stringa s , $s[i]$ indica il carattere che ne occupa l' i -esima posizione;

Mod. F24

DELEGA IRREVOCABILE A:

AGENZIA PROV.

PER L'ACCREDITO ALLA TESORERIA COMPETENTE

CONTRIBUENTE

CODICE FISCALE

cognome, denominazione o ragione sociale nome barraire in caso di anno d'imposta non coincidente con anno solare

50

Tipi di dati aggregati : stringhe

- Tipicamente una stringa può essere vista come un array di caratteri
 - In una stringa s , $s[i]$ indica il carattere che ne occupa l' i -esima posizione;

Mod. F24

DELEGA IRREVOCABILE A:

AGENZIA PROV.

PER L'ACCREDITO ALLA TESORERIA COMPETENTE

CONTRIBUENTE

CODICE FISCALE

cognome, denominazione o ragione sociale nome barraire in caso di anno d'imposta non coincidente con anno solare

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
G	N	N	S	P	E	1	1	A	5	0	F	9	9	1	K

51

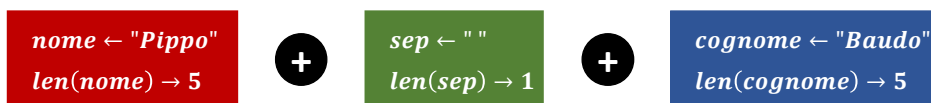
Tipi di dati aggregati : stringhe

- Alcuni operatori su stringhe:
 - L'operatore di **concatenazione**
 - date le stringhe $s_1 = \text{«ab»}$ e $s_2 = \text{«cd»}$, $s_1 + s_2 = \text{«abcd»}$
 - L'operatore **lunghezza**
 - Data $s_1 = \text{"abcd"}$, $len(s_1) = 4$
 - Una stringa $s = \text{""}$ è vuota e ha $len(s) = 0$

52

Concatenazione di stringhe: un esempio

Si considerino le tre stringhe *nome*, *sep* e *cognome* contenenti rispettivamente i dati «Pippo», « » (spazio vuoto) e «Baudo»:

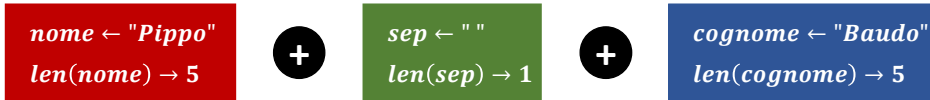


La stringa *presentatore* è ottenuta dalla concatenazione delle tre stringhe

53

Concatenazione di stringhe: un esempio

Si considerino le tre stringhe *nome*, *sep* e *cognome* contenenti rispettivamente i dati «Pippo», « » (spazio vuoto) e «Baudo»:



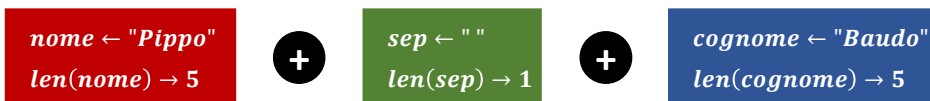
La stringa presentatore è ottenuta dalla concatenazione delle tre stringhe

$presentatore \leftarrow (nome + sep) + cognome$
 $presentatore \leftarrow ("Pippo ") + "Baudo"$

54

Concatenazione di stringhe: un esempio

Si considerino le tre stringhe *nome*, *sep* e *cognome* contenenti rispettivamente i dati «Pippo», « » (spazio vuoto) e «Baudo»:

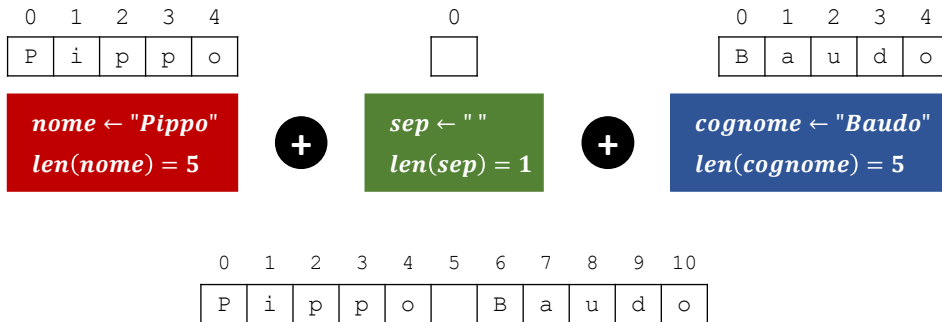


La stringa presentatore è ottenuta dalla concatenazione delle tre stringhe

$presentatore \leftarrow "Pippo Baudo"$
 $len(presentatore) \rightarrow 11 = len(nome) + len(sep) + len(cognome)$

55

Concatenazione di stringhe: un esempio



La stringa «presentatore» è rappresentata con un nuovo array contenente tutti gli elementi degli operandi...

56

Tipi di dati aggregati: record

Un elemento di un tipo **record** è costituito da una n -pla di «sotto-elementi» detti **campi**

Ciascun campo è identificato dal suo *nome*

57

Esempio: il tipo *record* Punto2D

Definiamo il tipo *Punto2D* in cui ciascun elemento è un *record* contenente 2 campi di tipo *numero reale*.

In altre parole: ogni elemento p di *Punto2D* è composto da una coppia (x,y) di elementi del tipo numeri reali.

Al primo campo assegniamo il nome x e al secondo il nome y .

```
Punto2D = {
    x (reale)
    y (reale)
}
```

58

Esempio: il tipo *record* Punto2D

Per estrarre il valore dei singoli campi di un record, si utilizza l'*operatore di accesso .* (punto);

Dati gli elementi $p1$, $p2$ e $p3$ di tipo **Punto2D**

Il punto $p1$ ha le seguenti coordinate:

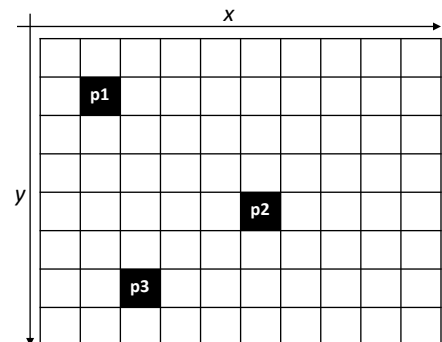
$$p1.x = 2$$

$$p1.y = 2$$

Il punto $p2$ ha le seguenti coordinate:

$$p2.x = 6$$

$$p2.y = 5$$



59

Tipi di dati aggregati: record

ESEMPIO: si consideri il tipo di dati descritto dal record `dati_anagrafici`, composto dai seguenti campi:

```
{
    nome (stringa);
    cognome (stringa);
    data_di_nascita (record data);
    luogo_di_nascita (stringa);
    indirizzo (stringa);
    recapito_telefono_fisso (stringa);
    cellulare (stringa);
    email (stringa);
}
```

I campi possono a loro volta essere espressi in un tipo aggregato. Il record `data`, può essere così composto:

```
{
    giorno (intero);
    mese (intero);
    anno (intero);
}
```

60

Array vs. record

Nell'array gli elementi aggregati sono obbligatoriamente tutti dello stesso tipo (*tipo base*)

Nel record possono essere di tipo diverso

Nell'array ciascun *elemento* è identificato univocamente da un indice numerico (la sua *posizione*)

Nel record ciascun *campo* è identificato univocamente dal suo nome

Nell'array il numero degli *elementi* è variabile.

e.g. un array di 10 e uno di 100 numeri interi sono entrambi array di interi

Nel record il numero dei campi è fisso.

e.g. un record con due campi interi e uno con tre campi interi, sono due tipi diversi

61

Tipi di dati aggregati: record

ESEMPIO: si consideri il tipo di dati descritto dal record `dati_anagrafici`, composto dai seguenti campi:

nome del campo

campi

NOME _____
 COGNOME _____
 DATA di NASCITA _____
 LUOGO di NASCITA _____
 INDIRIZZO _____
 RECAPITO TELEFONO FISSO _____ CELLULARE _____
 E-MAIL _____

62

Tipi di dati aggregati: record

ESEMPIO: si consideri il tipo di dati descritto dal record `dati_anagrafici`, composto dai seguenti campi:

L'utente Gennaro Esposito, compila il modulo. Il modulo dal nostro punto di vista il dato:

- appartiene al tipo `dati_anagrafici` (l'insieme di tutti i possibili modi di compilare il modulo);
- assume lo specifico valore riprodotto qui di seguito;

Ai fini della nostra elaborazione, gli attribuiamo l'identificatore `scheda_gennaro`;

`scheda_gennaro` ←

NOME Gennaro
 COGNOME Esposito
 DATA di NASCITA 22/11/1970
 LUOGO di NASCITA Napoli
 INDIRIZZO Vico Paradiso alla Salute, 23
 RECAPITO TELEFONO FISSO 08100102003 CELLULARE 30010020300
 E-MAIL gennaro@esposito.na

63

Tipi di dati aggregati: record

Un record può anche essere visto come la riga di una tabella in cui le colonne indicano un campo:

NOME Gennaro

COGNOME Esposito

DATA di NASCITA 22/11/1970

LUOGO di NASCITA Napoli

INDIRIZZO Vico Paradiso alla Salute, 23

RECAPITO TELEFONO FISSO 08100102003 CELLULARE 30010020300

E-MAIL gennaro@esposito.na

scheda_gennaro →

nome	cognome	data_di_nascita	...	email
Gennaro	Esposito	22/11/1970	...	Gennaro@esposito.na
...

64

Tipi di dati aggregati: record

Per estrarre il valore dei singoli campi di un record, si utilizza l'operatore di accesso `.` (punto); Ciascun campo è identificato dal suo nome.

Il contenuto del campo nome è identificato dall'espressione:

scheda_gennaro.nome

Il contenuto del campo email:

scheda_gennaro.email

NOME Gennaro

COGNOME Esposito

DATA di NASCITA 22/11/1970

LUOGO di NASCITA Napoli

INDIRIZZO Vico Paradiso alla Salute, 23

RECAPITO TELEFONO FISSO 08100102003 CELLULARE 30010020300

E-MAIL gennaro@esposito.na

65

Tipi di dati aggregati: record

Per estrarre il valore dei singoli campi di un record, si utilizza l'operatore di accesso `.` (punto); Ciascun campo è identificato dal suo nome.

Il giorno di nascita del Sig. Gennaro:

`scheda_gennaro.data.giorno`

NOME	<u>Gennaro</u>		
COGNOME	<u>Esposito</u>		
DATA di NASCITA	<u>22/11/1970</u>		
LUOGO di NASCITA	<u>Napoli</u>		
INDIRIZZO	<u>Vico Paradiso alla Salute, 23</u>		
RECAPITO TELEFONO FISSO	<u>08100102003</u>	CELLULARE	<u>30010020300</u>
E-MAIL	<u>gennaro@esposito.na</u>		

66

Esercizio

Si consideri il tipo di dati descritto dal record `scarpe`, composto dai seguenti campi:

```
{
    marca (stringa);
    modello (stringa);
    colore (stringa)
    misura (intero);
}
```

Il Sig. Gennaro si reca alla cassa con la scatola di un paio di scarpe, identificata con l'attributo `scarpe_gennaro`:

Si indichino le espressioni usate dal commesso per identificare i valori della marca, del modello e della misura delle scarpe contenute nella scatola.

67

Tipi di dati aggregati: operatori

Le strutture dati basate su record sono un ingrediente fondamentale per la costruzione di *basi di dati* o *database*. In questi sistemi, un insieme ricchissimo di operatore, permette ampie possibilità di elaborazione dei dati in essi contenuti.

Quando si progetta una struttura dati basata su record (o su tipi aggregati in generale), è spesso necessario definire i relativi operatori, possibilmente «*aggregando*» gli operatori definiti sui tipi base.

NOME	Gennaro		
COGNOME	Esposito		
DATA di NASCITA	22/11/1970		
LUOGO di NASCITA	Napoli		
INDIRIZZO	Vico Paradiso alla Salute, 23		
RECAPITO TELEFONO FISSO	08100102003	CELLULARE	30010020300
E-MAIL	gennaro@esposito.na		

68

Mappa

Per lo studio e l'approfondimento degli argomenti trattati

69

Mappa

[Sli] Fare riferimento alle slide della lezione.