

CIVIFORM

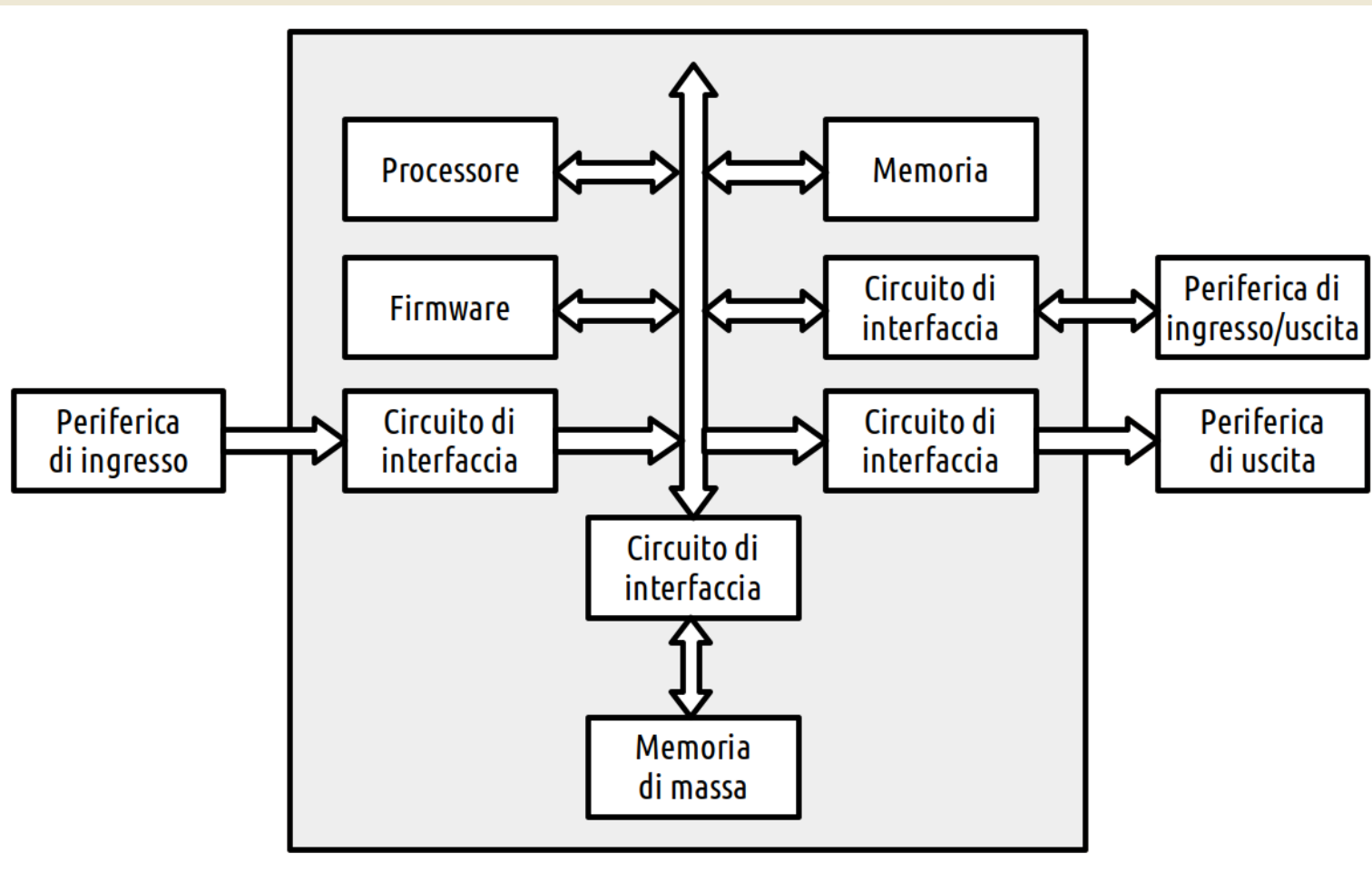
CODING E ROBOTICA

PER L'INNOVAZIONE SOCIALE

Cividale, gennaio-febbraio 2022

gzuliani.github.io/civiform/corso12

ARCHITETTURA DEI CALCOLATORI



Struttura dell'unità centrale

ESTENSIONE “PENNA”

POLIGONI

POLIGONI

```
/*  
  QUADRATO:  
    ripeti (4) volte  
      fai (50) passi  
        ruota σ di (90) gradi
```

POLIGONI

```
/*  
  QUADRATO:  
  ripeti (4) volte  
    fai (50) passi  
    ruota ⌚ di (90) gradi  
  
  TRIANGOLO:  
  ripeti (3) volte  
    fai (50) passi  
    ruota ⌚ di (120) gradi
```


POLIGONI

```
/*  
  QUADRATO:  
    ripeti (4) volte  
      fai (50) passi  
      ruota ⌚ di (90) gradi  
  
  TRIANGOLO:  
    ripeti (3) volte  
      fai (50) passi  
      ruota ⌚ di (120) gradi  
  
  PENTAGONO:  
    ripeti (5) volte  
      fai (50) passi  
      ruota ⌚ di (72) gradi
```

POLIGONI

```
/*  
  QUADRATO:  
    ripeti (4) volte  
      fai (50) passi  
      ruota ⌚ di (360/4) gradi  
  
  TRIANGOLO:  
    ripeti (3) volte  
      fai (50) passi  
      ruota ⌚ di (120) gradi  
  
  PENTAGONO:  
    ripeti (5) volte  
      fai (50) passi  
      ruota ⌚ di (72) gradi
```

POLIGONI

```
/*  
  QUADRATO:  
    ripeti (4) volte  
      fai (50) passi  
      ruota ⌚ di (360/4) gradi  
  
  TRIANGOLO:  
    ripeti (3) volte  
      fai (50) passi  
      ruota ⌚ di (360/3) gradi  
  
  PENTAGONO:  
    ripeti (5) volte  
      fai (50) passi  
      ruota ⌚ di (72) gradi
```

POLIGONI

```
/*  
  QUADRATO:  
  ripeti (4) volte  
    fai (50) passi  
    ruota ⌢ di (360/4) gradi  
  
  TRIANGOLO:  
  ripeti (3) volte  
    fai (50) passi  
    ruota ⌢ di (360/3) gradi  
  
  PENTAGONO:  
  ripeti (5) volte  
    fai (50) passi  
    ruota ⌢ di (360/5) gradi
```

ALGORITMO

Disegno di un poligono con un numero di lati arbitrario:

```
/*  
  POLIGONO:  
  ripeti (NUM_LATI) volte  
    fai (50) passi  
    ruota  $\sigma$  di  $(360/\text{NUM\_LATI})$  gradi
```

POLIGONI IN SEQUENZA

Disegno dal triangolo all'icosangolo:

```
/*  
  porta [NUM_LATI] a 3  
  ripeti (18) volte  
    DISEGNA_POLIGONO  
  cambia [NUM_LATI] di (1)
```

POLIGONI IN SEQUENZA

Disegno dal triangolo all'icosangolo:

```
/*  
  porta [NUM_LATI] a 3  
  ripeti (18) volte  
    ripeti (NUM_LATI) volte  
      fai (50) passi  
      ruota ⌢ di (360/NUM_LATI) gradi  
  cambia [NUM_LATI] di (1)
```

SPIRALE QUADRATA

Spirale quadrata con origine al centro dello schermo:

```
/*  
vai a x: (0) y: (0)  
punta in direzione (90)  
porta [PASSI] a 10  
ripeti fino a quando <sta toccando (bordo)>  
  ripeti (2) volte  
    fai (PASSI) passi  
    ruota ⌊ di (360/NUM_LATI) gradi  
  cambia [PASSI] di 10
```


CODING CON SCRATCH

CODING CON SCRATCH

- variabili

CODING CON SCRATCH

- variabili — attribuiscono un nome a un valore

CODING CON SCRATCH

- variabili — attribuiscono un nome a un valore
- cicli nidificati

CODING CON SCRATCH

- variabili — attribuiscono un nome a un valore
- cicli nidificati

```
/*  
  porta [PASSI] a 10  
  ripeti fino a quando <sta toccando (bordo)>  
    ripeti (2) volte  
      fai (PASSI) passi  
      ruota ⤵ di (360/NUM_LATI) gradi  
      cambia [PASSI] di 10
```

CODING CON SCRATCH

- variabili — attribuiscono un nome a un valore
- cicli nidificati

```
/*  
  porta [PASSI] a 10  
  ripeti fino a quando <sta toccando (bordo)>  
    ripeti (2) volte  
      fai (PASSI) passi  
      ruota ⤵ di (360/NUM_LATI) gradi  
      cambia [PASSI] di 10
```

CODING CON SCRATCH

- variabili — attribuiscono un nome a un valore
- cicli nidificati
- cicli condizionati

CODING CON SCRATCH

- variabili — attribuiscono un nome a un valore
- cicli nidificati
- cicli condizionati

```
/*  
  ripeti fino a quando <...>  
  ...
```


CODING CON SCRATCH

- variabili — attribuiscono un nome a un valore
- cicli nidificati
- cicli condizionati
- porta, cambia

CONSIDERAZIONI

CONSIDERAZIONI

- schemi ricorrenti confluiscono in cicli

CONSIDERAZIONI

- schemi ricorrenti confluiscono in cicli
- importanza della fase di inizializzazione del programma

CONSIDERAZIONI

- schemi ricorrenti confluiscono in cicli
- importanza della fase di inizializzazione del programma
- intrinseca difficoltà del **debug**



**KEEP
CALM**

AND

**CONTINUE
CODING**

CONSIDERAZIONI

- schemi ricorrenti confluiscono in cicli
- importanza della fase di inizializzazione del programma
- intrinseca difficoltà del **debug**
- esercitarsi nella scrittura e nella lettura del codice

ah, i bei vecchi tempi...

MONDO SOMMERSO