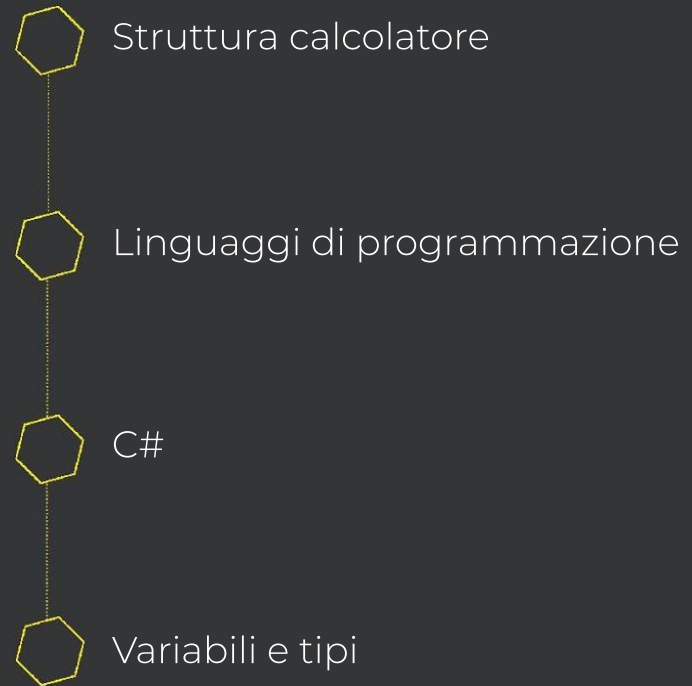


STRUTTURA CALCOLATORE TIPI C#

CORSO DI GAME PROGRAMMING
1° ANNO

Docente **Davide Caio**





COS'È UN CALCOLATORE?

Il calcolatore è:

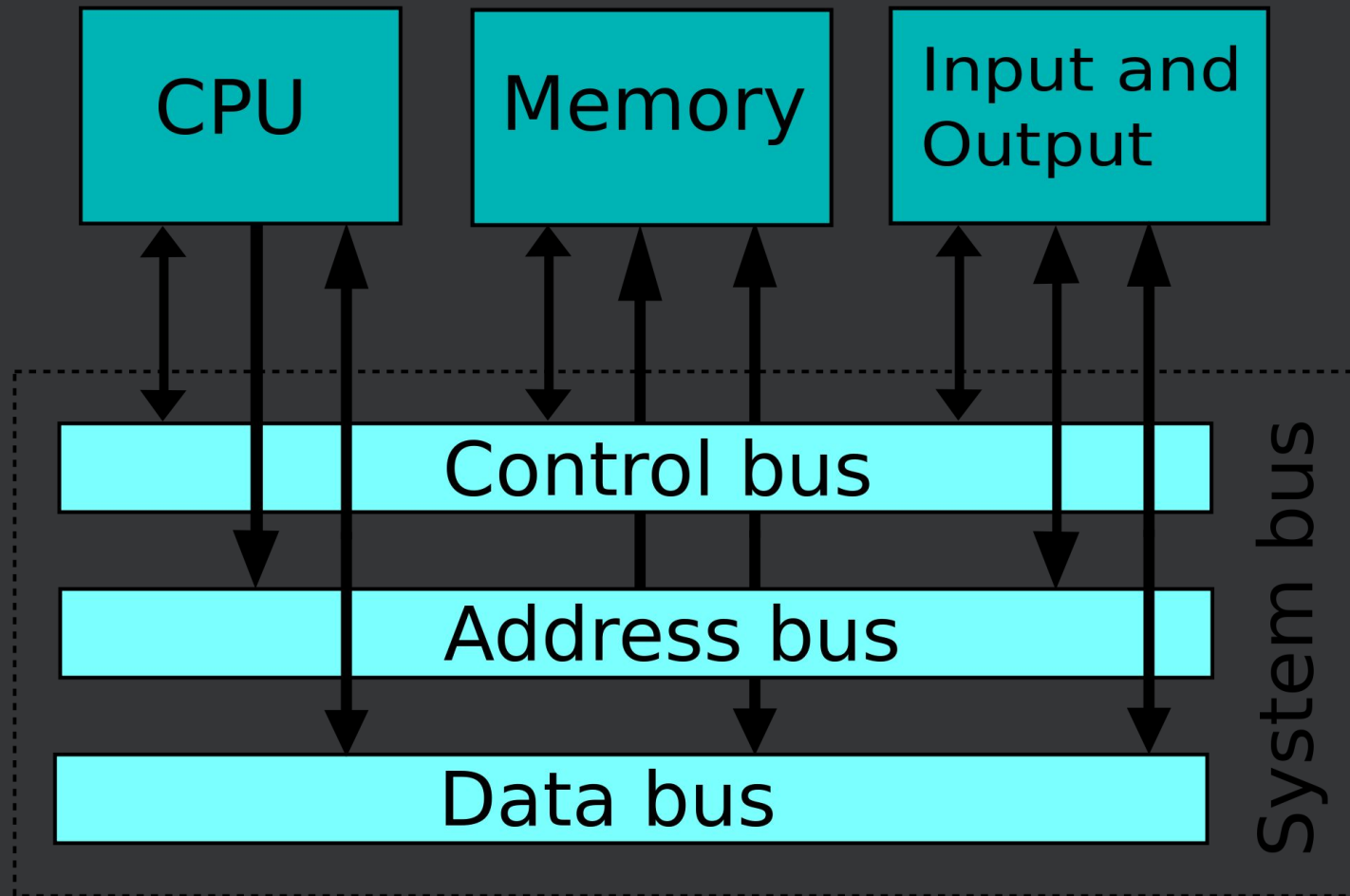
- uno **strumento programmabile** per rappresentare memorizzare ed elaborare informazioni.
- **un sistema**, costituito da molte componenti

La prima decomposizione di un calcolatore è relativa a due macro-componenti:

- **Hardware**
- **Software**

L'architettura dell'hardware di un calcolatore reale è molto complessa, ma un modello semplificato dei calcolatori è la cosiddetta **macchina di Von Neumann**

MACCHINA DI VON NEUMANN





TRASFERIMENTO

Obiettivo: permettere lo **scambio di informazioni** tra le varie componenti funzionali del calcolatore □

- trasferimento dei dati e delle informazioni di controllo

Due possibili soluzioni □

- collegare ciascun componente con ogni altro componente
- □ collegare tutti i componenti a un unico canale (bus)

L'utilizzo di un bus favorisce la **modularità** e l'espandibilità del calcolatore



PERIFERICHE E INTERFACCE I/O

Un calcolatore può essere collegato a dispositivi di ingresso e/o uscita, le **periferiche**.

Attenzione, nella macchina di Von Neumann le periferiche non fanno parte del calcolatore e sono controllate con un'opportuna **interfaccia**. (traduce segnali interni del calcolatore in un formato comprensibile alla periferica e viceversa).



MEMORIZZAZIONE

Un calcolatore memorizza:

- **i dati**, che rappresentano informazioni di interesse
- **i programmi** per l'elaborazione dei dati.

La **memoria centrale** è organizzata in **celle**, identificate da un **indirizzo** e in grado di memorizzare un singolo dato o istruzione.

L'unità di memoria fornisce due operazioni:

- **scrittura**
- **lettura**



CPU

L'unità centrale di elaborazione o processore presiede all'**esecuzione di un programma**:

- Il programma è memorizzato in memoria centrale come **sequenza di istruzioni** in linguaggio macchina
- **Linguaggio macchina** o assembly è il linguaggio per cui la CPU si comporta da esecutore

La CPU è composta da

- unità di controllo (**CU**)
- unità aritmetico-logica (**ALU**)



CU & ALU

La **CU** coordina l'**esecuzione temporale** delle operazioni. Il controllo avviene in modo sincrono rispetto alla scansione temporale imposta dall'orologio di sistema (**clock**)

La **ALU** esegue le **istruzioni del linguaggio macchina** che possono essere:

- operazioni aritmetiche
- operazioni relazionali (confronto tra dati)
- operazioni su caratteri e valori di verità
- altre operazioni numeriche.

Un calcolatore sa svolgere poche tipologie di operazioni elementari ma in modo molto efficiente (centinaia di milioni di istruzioni del linguaggio macchina al secondo).



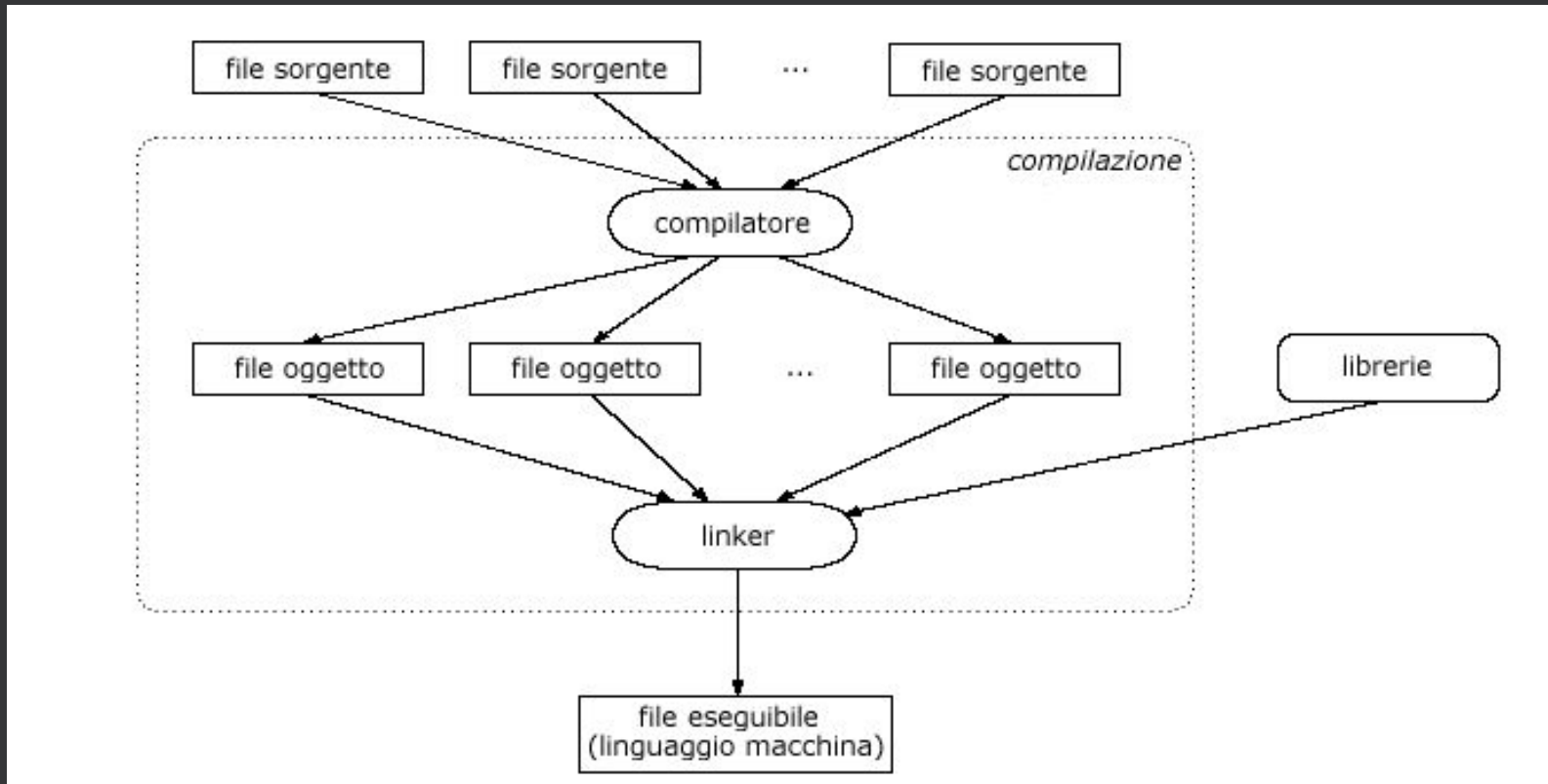
COS'È UN LINGUAGGIO DI PROGRAMMAZIONE?

Analizziamo la definizione fornita da Wikipedia:

*“Un **linguaggio di programmazione**, in informatica, è un **linguaggio formale** che specifica un insieme di **istruzioni** che possono essere usate per produrre dati in uscita: esso è utilizzabile per il controllo del comportamento di una macchina formale o di un'implementazione di essa (tipicamente, un **computer**) ovvero in fase di programmazione di questa attraverso la scrittura del **codice sorgente di un programma** ad opera di un **programmatore**: un linguaggio di programmazione è considerato a tutti gli effetti tale se è **Turing-completo**.”*

Turing-completo vuol dire avere lo stesso potere computazionale della macchina di turing.

COME SI ARRIVA AL LINGUAGGIO MACCHINA





C#

Il **C#** è un linguaggio di programmazione orientata agli oggetti basato sulla piattaforma **.NET**.

Microsoft .NET è una **piattaforma di sviluppo** general purpose di Microsoft.



COMMON LANGUAGE INFRASTRUCTURE

Il **Common Language Infrastructure** (CLI), è una specifica aperta sviluppata da Microsoft, che descrive il codice eseguibile e l'ambiente di esecuzione che costituisce il cuore del framework .NET. La specifica definisce un **ambiente che permette a più linguaggi ad alto livello di essere utilizzati su differenti piattaforme** senza la necessità di essere riscritti per specifiche architetture hardware o software. Come?

- Il codice sorgente del programma viene tradotto in **CIL** (common intermediate language).
- Quando il programma viene eseguito (JIT), a quel punto un sistema chiamato **CLR** (common language runtime), trasforma le istruzioni CIL nel codice nativo della macchina su cui sta girando il CLR.



LE VARIABILI

Le **variabili** in un linguaggio di programmazione sono contenitori di dati situati in una porzione di memoria.

Devono avere un **nome**, che utilizzerai all'interno del programma come riferimento a quel particolare dato e un **tipo**, che sostanzialmente indica lo spazio in memoria che deve essere occupato per rappresentare quel dato.



I TIPI DI C#

Keyword	Class	Range
bool	System.Boolean	true and false
byte	System.Byte	0 to 255
sbyte	System.SByte	-128 to 127
short	System.Int16	-32768 to 32767
ushort	System.UInt16	0 to 65535
int	System.Int32	-2,147,483,648 to 2,147,483,647
uint	System.UInt32	0 to 4,294,967,295
long	System.Int64	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
ulong	System.UInt64	0 to 18,446,744,073,709,551,615
decimal	System.Decimal	-79,228,162,514,264,337,593,543,950,335 to 79,228,162,514,264,337,593,543,950,335
double	System.Double	-1.79769313486232e308 to 1.79769313486232e308
float	System.Single	-3.402823e38 to 3.402823e38
char	System.Char	0 to 65535



DEFINIZIONE DI UNA VARIABILE

Le **variabili** in C# si definiscono definendo il tipo e il nome:

```
int playerEnergy;
```

```
float playerExp;
```

```
bool isJumping;
```