

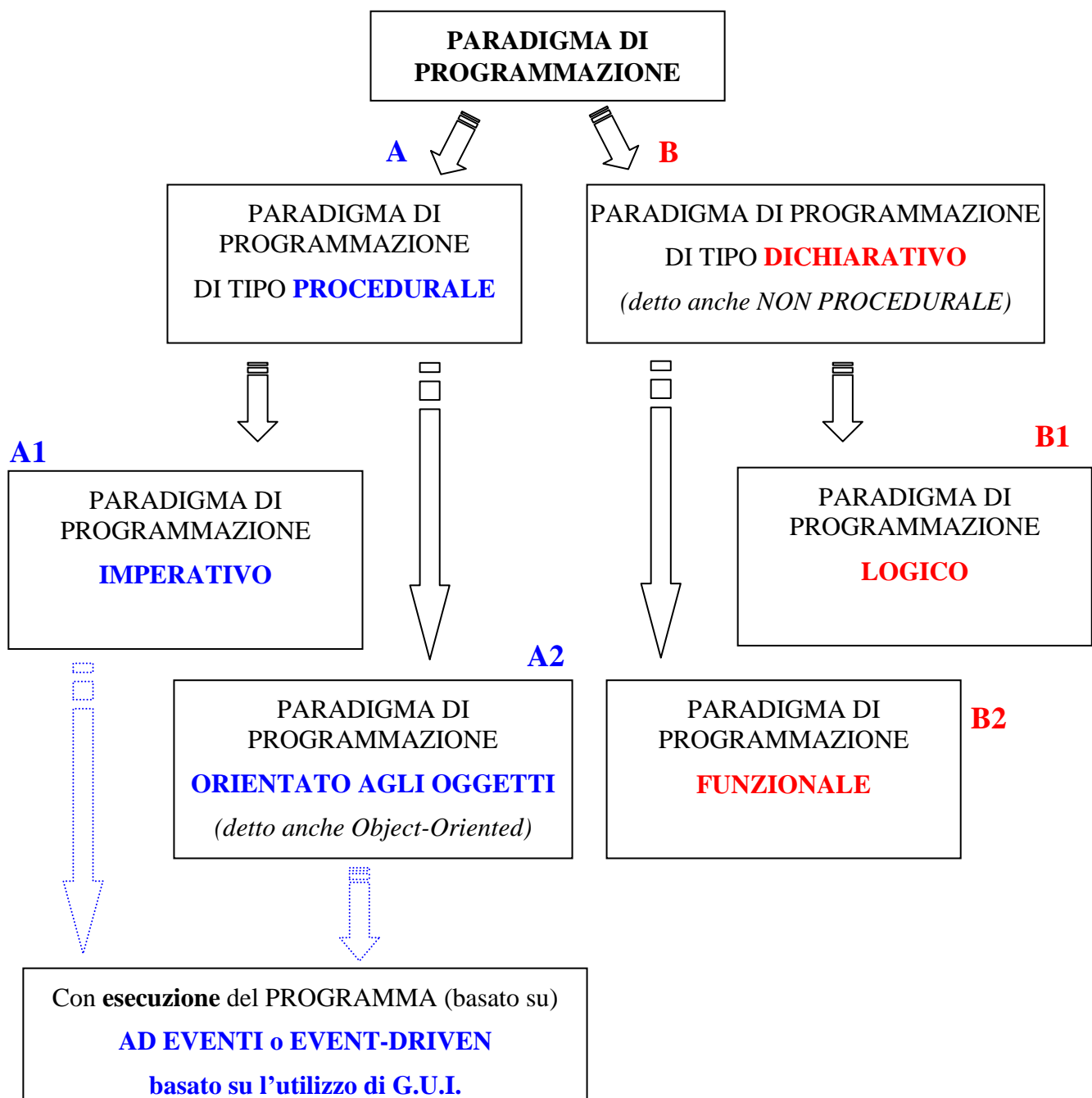
## 2. PARADIGMI DI PROGRAMMAZIONE

La disciplina che si occupa dell'automatizzazione dei processi risolutivi attraverso la codifica o l'implementazione prende il nome di programmazione di un algoritmo in un linguaggio di **programmazione**.

Un **paradigma di programmazione** (*paràdeigma*: dal greco “modello”) è un modello che permette al programmatore descrivere astrattamente l'algoritmo (cioè il metodo di soluzione di un problema).

Si ricorda che il **programmatore** è colui che traduce le istruzioni dell'algoritmo descritto in pseudolinguaggio e/o con un flow-chart, nel linguaggio di programmazione di alto livello scelto.

Schematizzando, i principali paradigmi di programmazione sono così suddivisibili:



## A PARADIGMA DI PROGRAMMAZIONE PROCEDURALE

La programmazione basata sul **paradigma procedurale** descrive **COME** deve essere risolto un problema.

Segue le tra fasi classiche per la risoluzione di un problema:

- conoscenza del problema ossia studio della realtà;
- comprensione del problema;
- ricerca della soluzione che consiste nello stabilire *come* risolvere il problema

### A1 PARADIGMA DI PROGRAMMAZIONE PROCEDURALE IMPERATIVO

Si basa sul concetto di “*imperio*” o *comando* ossia su di una serie di esplicite richieste inviate all’esecutore del programma.

In questo contesto il programmatore rappresenta il processo di calcolo come una sequenza di azioni

Esempi di linguaggi imperativi: l’**Assembler**, il **COBOL**, il **C**, il **PASCAL**, il **FORTRAN**, etc.

### A2 PARADIGMA DI PROGRAMMAZIONE PROCEDURALE ORIENTATA AGLI OGGETTI (o Object-Oriented Programming o OOP)

Si basa sul concetto di **classe** che descrive “*proprietà*” e “*azioni*” che un **oggetto** può compiere.

In questo contesto il programmatore rappresenta il processo di calcolo come l’interazione di due o più oggetti della stessa classe o di classi differenti.

Esempi di linguaggi ad oggetti: il linguaggio **C++**, il linguaggio **JAVA**, etc.

## PROGRAMMAZIONE BASATA SUGLI EVENTI

Sia nella **programmazione imperativa**, che nella **programmazione ad oggetti**, l’esecuzione del programma **può** (non è obbligatorio!) essere guidata da “*eventi*”, che sono in grado, una volta che si sono verificati, di causare l’esecuzione di determinate “*azioni*”.

Un evento potrebbe essere il *click* sul pulsante destro del mouse piuttosto che lo *schiacciamento di un tasto* da parte dell’utente che, se opportunamente programmato, potrebbe dare origine, ad esempio, alla visualizzazione di un particolare menù di scelta piuttosto che all’esecuzione di un determinato sottoprogramma.

Di solito la programmazione guidata dagli eventi è legata all’implementazione di una **G.U.I. (Graphic User Interface)** ossia una serie di interfacce grafiche che sono spesso realizzate tramite oggetti.

Laddove ciò non fosse, viene utilizzata una più spartana, ma comunque efficace, **C.L.I. (Command Line Interface)** ossia una interfaccia a caratteri basata su una finestra di comandi

## B PARADIGMA DI PROGRAMMAZIONE DICHIARATIVO

La programmazione basata sul **paradigma dichiarativo** si concentra esclusivamente su che **COSA** si vuole ottenere come soluzione richiesta il problema e non si occupa minimamente sul COME fare per raggiungere i risultati.

L'idea base è quella di pervenire alla soluzione del problema dettagliando sia la conoscenza della realtà che il problema, mediante opportuno "linguaggio di specifica".

Il principale svantaggio di tale paradigma di programmazione è rappresentato dall'efficienza: ai risultati si perviene in tempi nettamente superiori ai casi modellati secondo il paradigma procedurale.

### B1 PARADIGMA DI PROGRAMMAZIONE LOGICO

Il programmatore giunge alla soluzione del problema esprimendo una serie di "**assiomi**", formalizzati opportunamente e comunicati al sistema, che rappresentano il complesso dei fatti che devono essere veri all'interno della realtà esaminata.

Il risultato scaturisce dal conseguente processo di calcolo.

L'esecuzione di un programma si ottiene ponendo un **quesito** al sistema: la risposta consiste nella verifica di verità o di falsità dell'affermazione fatta dal quesito sulla base delle conoscenze specificate all'interno del sistema.

Esempi di linguaggi logici: il linguaggio **PROLOG**.

### B2 PARADIGMA DI PROGRAMMAZIONE FUNZIONALE

Un programma scritto in accordo a questo paradigma è costituito da un insieme di "**funzioni**", nel senso matematico del termine, che elaborano liste di simboli.

L'esecuzione del programma avviene in seguito alla richiesta fatta al sistema di valutare una espressione.

Il valore di una funzione è nota una volta noto il valore dei suoi argomenti.

La coppia (funzione, lista di argomenti) prende il nome di applicazione.

Esempi di linguaggi funzionali: il linguaggio LISP.

Schematizzazione del processo evolutivo dei linguaggi di programmazione

I linguaggi di programmazione si sono evoluti di pari passo con l'evoluzione tecnologica delle macchine e per questo si parla di **generazione di linguaggi di programmazione** (vedi schema successivo)

