



ISTITUTO DI ISTRUZIONE SUPERIORE

Michele Giua

Cagliari

ANNO SCOLASTICO 2025/2026

PROGRAMMA SVOLTO

Classe: 2^a B - Ist. Tecn. Inf. Telecom. - Cagliari

Materia: Scienze e tecnologie applicate (Informatica)

Docente: Stefano Murgia

LIBRO DI TESTO

Codice Volume: 9788863646603

Autori: Barbero Alberto / Vaschetto Francesco

Titolo: Il nuovo **Dal bit al web** / libro cartaceo + ite + didastore

Volume Unico

Editore: Linx

Manuale operativo: **Impara a programmare con Flowrun** - libro digitale reperibile nella Classroom 2B Scienze e tecnologie applicate

<https://classroom.google.com/c/MjM1MTE3Nzc5Njda/m/MjIwMTQ4MzY3MzFa/detail>
s

Addendum: **documentazione integrativa** in formato digitale, reperibile nella Classroom 2B Scienze e tecnologie applicate

<https://classroom.google.com/c/MjM1MTE3Nzc5Njda?cjc=3ox2oefm>

1) DIAGRAMMI DI FLUSSO

- Cos'è un Diagramma di Flusso
- Scopo di un Diagramma di Flusso
- Componenti di un Diagramma di Flusso
 - Simboli
 - Linee di flusso
- Tipi di Diagrammi di Flusso
- Esempio di un Semplice Diagramma di Flusso: calcolare la somma di due numeri
 - Pseudocodice
 - Rappresentazione del Diagramma di Flusso
- Vantaggi dell'Uso dei Diagrammi di Flusso
- Strumenti per Creare Diagrammi di Flusso
- Diagrammi di Flusso di programmi informatici
- Scopo dei Diagrammi di Flusso di programmi informatici

- Progettazione di algoritmi
- Debugging
- Documentazione
- Educazione

2) DIAGRAMMA DI FLUSSO DI PROGRAMMI INFORMATICI

- Componenti di un Diagramma di Flusso di programma informatici
 - Linea di flusso
 - Terminale
 - Processo
 - Decisione
 - Input/Output
 - Annotazione
 - Processo Predefinito
 - Preparazione
- Passaggi per Creare un Diagramma di Flusso di Programma
 - Definire il problema
 - Identificare i passaggi
 - Scegliere i simboli
 - Organizzare il flusso
 - Verificare la logica
- Applicazioni dei Diagrammi di Flusso di programmi informatici
 - Programmazione
 - Risoluzione dei problemi
 - Sviluppo software
- Vantaggi dei Diagrammi di Flusso di programmi informatici
 - Chiarezza
 - Rilevazione degli errori
 - Efficienza
 - Documentazione
- Strumenti per Creare Diagrammi di Flusso di programmi informatici
 - Software
 - Strumenti specifici per la programmazione
 - Piattaforme online: le WebApp **FLOGO** e **FLOWRUN**

3) ALGORITMI

- Cos'è un Algoritmo
- Caratteristiche di un Algoritmo
 - Finitudine
 - Chiarezza
 - Input
 - Output
 - Efficacia

- Importanza degli Algoritmi
 - Risoluzione dei problemi
 - Efficienza
 - Riutilizzabilità
 - Base della programmazione
- Tipi di Algoritmi
 - Algoritmi Ricorsivi
- Esempi di Algoritmi
 - Algoritmo per Sommare Due Numeri
 - Algoritmo per trovare il numero maggiore tra tre
 - Calcolo del fattoriale
- Differenza tra **ALGORITMO** e **PROGRAMMA**
- Applicazioni degli algoritmi
- Proprietà chiave di un buon algoritmo
 - Correttezza
 - Efficienza
 - Generalità
 - Scalabilità

4) ALGORITMO SEQUENZIALE

- Caratteristiche di un Algoritmo Sequenziale
 - Esecuzione Lineare
 - Deterministico
 - Singolo Thread di Esecuzione
 - Semplicità
- Passaggi in un Algoritmo Sequenziale
 - Inizio
 - Input:
 - Elaborazione
 - Output
 - Fine
- Esempio di un Algoritmo Sequenziale
 - **Problema 1:** Calcolare l'area di un rettangolo
 - **Pseudocodice** per il problema 1
 - Rappresentazione con **diagramma di flusso**
 - **Problema 2:** Dati due numeri interi calcolare il MCD (per differenze successive)
 - **Pseudocodice** per il problema 2
 - Rappresentazione con **diagramma di flusso**
- Applicazioni degli Algoritmi Sequenziali
 - Calcoli matematici semplici
 - Elaborazione dei dati
 - Automazione

- Vantaggi degli algoritmi sequenziali
 - Semplicità
 - Predicibilità
 - Efficienza (per attività semplici)
- Limitazioni degli Algoritmi Sequenziali
 - Assenza di parallelismo
 - Scalabilità
 - Dipendenza

5) GLI AMBIENTI DI SVILUPPO INTEGRATO

- **FLOGO** <https://flogo.fdossena.com/>
- **FLOWRUN** <https://flowrun.io/>
- Caratteristiche e funzionalità
 - L'interfaccia IDE - Integrated Development Environment
 - Tasti Funzione
 - Caratteristiche principali
- Grafica Tartaruga
- Interruzione con e senza condizione
- Generazione interattiva del codice
- Finestra "Visualizzazione Variabili"
- Finestra "Esecuzione"
- Specifiche
- Array monodimensionale
- Array bidimensionale (matrici)
- Cicli iterativi con controllo in testa **WHILE** e in coda **DO-WHILE**
- Cicli iterativi enumerati **FOR**
- Dichiarazione esplicita delle variabili
- Funzioni e Procedure con e senza parametri
- Funzioni ricorsive
- Ricorsione controllata
- Il **caso base** nella ricorsione
- Supporto operatori
- Supporto per funzioni e procedure create dall'utente
- Funzioni e operatori predefiniti
- Operatori
- Precedenze
- Funzioni Matematiche
- Funzioni stringa
- Altre funzioni
- Costanti predefinite
- Istruzioni
 - Input
 - Output

- Dichiarazione
- Assegnazione
- Controlli
 - IF
 - Chiamata di Funzione o Procedura: “fx Call”
- Cicli iterativi
 - WHILE
 - FOR
 - DO-WHILE
- Commento
- Interruzioni con e senza condizione
- Grafica Tartaruga con **FLOGO**
 - Ruotare
 - Avanzare
 - Punto iniziale
 - Colore penna

6) FLOWRUN e programmazione: la creazione di diagrammi di flusso (flowchart)

- Tipi di dato
 - Semplice (scalare)
 - Integer
 - Real
 - String
 - Boolean
 - Array
 - Integer[]
 - Real[]
 - String[]
 - Boolean[]
 - Matrici
 - Integer[][]
 - Real[][]
 - String[][]
 - Boolean[][]
- Identificatori: etichette univoche per variabili, funzioni, costanti
- Parole chiave: termini riservati predefiniti
- Funzioni e Procedure con e senza parametri
- La ricorsione
 - Funzione e Procedura ricorsiva
 - Esempi in FLOWRUN
 - countdown
 - calcolo del fattoriale

7) Argomenti trattati con Flowrun, organizzati per moduli logici, basati sul manuale operativo "Impara a programmare con Flowrun"

Fondamenti e Introduzione Visiva

- **Definizioni di base:** Differenza tra programma (implementazione eseguibile) e algoritmo (descrizione astratta).
- **L'ambiente Flowrun:** Utilizzo dell'editor, esecuzione di diagrammi di flusso e pannello di output.
- **Vantaggi del linguaggio visivo:** Seguire la logica attraverso le frecce e i rami del codice.

Nozioni di Base e Gestione Dati

- **Output di base:** Il primo programma "Hello World!" e l'uso delle virgolette per le stringhe.
- **Tipi di numeri:** Distinzione tra numeri interi (Integer) e numeri reali (Real) e relative operazioni matematiche (+, -, *, /).
- **Operatore Modulo (%):** Calcolo del resto della divisione per determinare la divisibilità o la parità.
- **Variabili:** Dichiarazione (nome e tipo), assegnazione di valori e modifica nel tempo.
- **Input dell'utente:** Acquisizione di dati esterni e memorizzazione in variabili.

Logica Decisionale e Booleana

- **Blocco If:** Ramificazione del codice in base a condizioni vere o false.
- **Operatori di confronto:** Uso di <, <=, ==, !=, >, >= per creare espressioni booleane.
- **Logica Booleana:** Operatori logici NOT (!), AND (&&) e OR (||) con relative tabelle della verità.
- **Gestione delle Stringhe:** Concatenazione di testo e variabili tramite l'operatore +.

Strutture Iterative (Cicli)

- **Ciclo While:** Esecuzione di blocchi di codice finché una condizione è vera.
- **Ciclo Do While:** Esecuzione del corpo del ciclo almeno una volta prima della verifica della condizione.
- **Ciclo For:** Iterazione con variabile di ciclo, valore iniziale, finale e passo (incremento).
- **Cicli Annidati:** Utilizzo di un ciclo all'interno di un altro (es. per tabelle di moltiplicazione o modelli grafici).

Modularità: Funzioni e Ricorsione

- **Funzioni e Procedure:** Differenza tra funzioni che restituiscono un valore e procedure che eseguono solo azioni.

- **Parametri e Argomenti:** Definizione delle variabili di input e passaggio dei valori reali.
- **Funzioni Predefinite:** Utilizzo di strumenti pronti all'uso come **sqrt**, **abs**, **pow** e **randomInt**.
- **Funzioni Personalizzate:** Creazione e chiamata di nuove funzioni e gestione del valore di ritorno (return).
- **Ricorsione:** Concetto di funzione che chiama se stessa e importanza del caso base (condizione di terminazione).
- **Ambito delle funzioni (Scope):** Visibilità e isolamento delle variabili all'interno delle funzioni.

Strutture Dati Complesse

- **Array (Vettori):** Dichiarazione, inizializzazione automatica e gestione degli indici (da 0 a lunghezza-1).
- **Algoritmi su Array:** Calcolo della media, ricerca di elementi (indexOf), frequenza dei numeri e ordinamento tramite Bubble Sort.
- **Matrici:** Strutture bidimensionali (righe e colonne), indici `m[i][j]` e operazioni come l'addizione di matrici o la verifica della matrice identitaria.

Applicazioni Reali e Professionalizzazione

- **Casi studio:** Calcolo di statistiche (media e mediana) e implementazione della logica di gioco per il Tic Tac Toe.
- **Generazione di codice:** Conversione dei diagrammi in linguaggi testuali (Java, Python, C#, etc.).
- **Best Practice:** Uso di nomi significativi, variabili temporanee per la leggibilità e gestione elegante degli errori.
- **Debugging:** Risoluzione dei problemi tramite l'esecuzione passo dopo passo e il monitoraggio delle variabili nel debugger.

Risorse Online

- **FLOGO**
 - documentazione e programmi di esempio - <https://fdossena.com/?p=flogo/index.frag>
 - IDE ambiente di sviluppo integrato - <https://flogo.fdossena.com>
 - EBook - manuale di FLOGO - nell'help, il simbolo “?” in alto a destra della [IDE](#)
- **FLORUN**

- documentazione e programmi di esempio - <https://flowrun.io/learn-programming>
- IDE ambiente di sviluppo integrato - <https://flowrun.io/scratchpad>
- EBook - manuale di FLOWRUN - Impara a programmare con Flowrun - <https://classroom.google.com/c/ODA1OTQ2NDczMTQ1/m/MTYyMzY3MTQ1MjI3/details>

4) COMPITI DI REALTÀ

- A. **Lancio casuale di due dadi:** simulare con Flowrun il lancio di due dadi a 6 facce e visualizzare il valore delle ricorrenze della somma e del prodotto delle facce per n lanci
- B. **Il gioco del Tris (Tic Tac Toe):** simulare con Flowrun, attraverso l'uso di funzioni con parametri, il gioco del Tris, alternando le giocate dei due giocatori e stabilendo la vittoria di uno dei due o il pareggio
- C. **Cubo di Rubik:** simulare con Flowrun, attraverso l'uso di funzioni con parametri, la visualizzazione 2D e facelet dello stato iniziale del cubo di Rubik, le rotazioni CW e CCW delle 6 facce, F, B, U, D, L, R, memorizzando le 9 tessere di ciascuna faccia in una matrice 3x3, utilizzando la notazione convenzionale del cubo di Rubik o sistema Singmaster - <https://kubekings.it/blog/post/notazione-cubo-di-rubik> in cui:

F (Front): Faccia frontale

B (Back): Faccia posteriore

U (Up): Faccia superiore

D (Down): Faccia inferiore

L (Left): Faccia di sinistra

R (Right): Faccia di destra

Programma:

1. caricamento dello stato iniziale delle facce in 6 matrici 3x3
2. visualizzazione 2D delle 6 facce
3. visualizzazione delle 6 facce in **notazione facelet**
4. rotazione CW e CCW delle 6 facce

Cagliari 22/06/2025

Il docente

Prof. Stefano Murgia