

PROGRAMMA DI SISTEMI E AUTOMAZIONE
CLASSE 4^A A TIEN
Anno scolastico 2005-2006

MODULO 1

- 1.1 Introduzione alla teoria dei sistemi. Definizione di sistema. Sistema astratto orientato. Sistemi dinamici.
- 1.2 Sistemi programmabili. Stato interno di un sistema. Definizione di programma. Definizione di linguaggio
- 1.3 Modello di un sistema. Tipi di modelli. Diagramma di stato. Schema a blocchi.
- 1.4 Sistema discreto a stati finiti: automa. Modello di Moore.
- 1.5 Sistemi lineari e rappresentazione con schema a blocchi. Algebra degli schemi a blocchi. Sistemi istantanei. Concetto di funzione di trasferimento.
- 1.6 Sistemi retroazionati. Principio di sovrapposizione degli effetti.

MODULO 2

- 2.1 Descrizione generale del sistema personal computer (PC). Cenni sui dispositivi di I/O
- 2.2 Cenni sul sistema operativo.
- 2.3 Linguaggio di programmazione **Visual C# (VC#)**. Caratteristiche generali.
- 2.4 Descrizione dell'ambiente di sviluppo di Visual Studio.NET
- 2.5 Applicazioni tipo Console. I/O da console: **Console.WriteLine()**, **Console.Write()**, **Console.ReadLine()**.
- 2.6 Definizione di variabile. Tipo **int**, **byte**, **float**, **double**, **string**. Dimensione di una variabile.
- 2.7 Istruzioni di assegnazione di un valore a una variabile. Espressioni. Commenti
- 2.8 Operatori aritmetici: +, -, *, /, %.
- 2.9 Formato di stampa con marcatori all'interno di un testo. Carattere escape **\n**
- 2.10 Dati di tipo **char**. **Cast** da tipo intero a tipo char. Codice ASCII di un carattere.
- 2.11 Visibilità delle variabili, attributi **private**, **public**.
- 2.12 Istruzione condizionale **if...; if...else...; if...else if...else...;**
- 2.13 Istruzione condizionale **switch...case 1...case 2...default...;**. Istruzione **break;**
- 2.14 Istruzione iterativa **while**.
- 2.15 Istruzione iterativa **for**.
- 2.16 Istruzione iterativa **do....while**.
- 2.17 Definizione di **classe**. Definizione di una classe: componenti dati. Modalità di accesso ai componenti dati.
- 2.18 Definizione di metodi all'interno di una classe. Accesso a variabili private con metodi adeguati.
- 2.19 Definizione di **proprietà**. Metodi **get** e **set**.
- 2.20 Metodi come **costruttori di istanza**.
- 2.21 Definizione di un **array** per memorizzare i dati; indici ed elementi di un array.
- 2.22 **Polimorfismo** (overloading) dei metodi e di un costruttore.
- 2.23 Ereditarietà tra le classi. **Classe base** o **classe genitore** e **classe figlio** o **classe derivata**.

MODULO 3

- 3.1 Sistema di trasmissione delle informazioni. Significato di bus. Richiami sui concetti di informazione, segnale, codice, unità di misura dell'informazione. Codificazione in binario di messaggi.
- 3.2 Codice **ASCII**. Codice numerico in binario. Cambiamento di codice da binario a decimale e viceversa.
- 3.3 Codice esadecimale. Utilità. Cambiamento di codice da esadecimale a binario a decimale e viceversa.
- 3.4 Rappresentazione dei numeri con segno. Codifica in modulo e segno. Codifica in complemento a due. Operazioni algebriche con numeri codificati in complemento a due.
- 3.5 Segnalatori aritmetici: **C**, **N**, **Z**, **V**. Significato e attivazione in risposta ad operazioni algebriche di somma e differenza.
- 3.6 Logiche combinatorie che pilotano i segnalatori aritmetici.
- 3.7 Semplice **ALU** che realizza operazioni di somma algebrica. Bit di controllo che consentono di produrre l'operazione di somma o l'operazione di differenza.
- 3.8 Registri di memoria. Circuiti che consentono operazioni di lettura e di scrittura. Bit di controllo.
- 3.9 Architettura di un semplice processore di dati con **ALU**, registri di memoria e bus di collegamento (**Data path**).
- 3.10 Operazioni elementari eseguite su un processore attivando in modo adeguato bit di controllo.
- 3.11 **Unità di controllo** che produce sequenze di bit di controllo per risolvere una prefissata espressione algebrica. Memoria di controllo (**ROM**) pilotata da un contatore up.

LABORATORIO

- 1.1 Programma che stampa su console una frase tipo testo (*Prima Prova*)
- 1.2 Determinazione della dimensione dei tipi di variabili (*Applicazione 1 Giorno3*)
- 1.3 Stampa di caratteri introducendo il corrispondente codice Unicode corrispondente (*Applicazione 2 Giorno3*):
- 1.4 Programma che esemplifica l'istruzione **if...else...** (*Applicazione 1 Giorno4*). Debug step by step.
- 1.5 Altro programma che esemplifica l'istruzione **if...else...** (*Applicazione 1 Giorno5*) Debug step by step.
- 1.6 Programma che esemplifica l'istruzione **if...else if...else...** (*Applicazione 2 Giorno5*) Debug step by step.
- 1.7 Programma con istruzione **if** eventi più alternative **else if** (*Applicazione 3 Giorno5*) Debug step by step.
- 1.8 Programma con istruzione **switch...case1..break; case2..break;..default...break;** con alternative create con la classe **Random** (*Applicazione 4 Giorno5*).
- 1.9 Programma con istruzione **switch...case1case2..break;..default..break;** con singola soluzione per più casi e con alternative create con la classe **Random** (*Applicazione 5 Giorno5*).
- 1.10 Programma che usa l'istruzione **while**. (*Applicazione 6 Giorno5*).
- 1.11 Programma che usa l'istruzione **for**. (*Applicazione 7 Giorno5*).
- 1.12 Programma che usa l'istruzione **do...while**. (*Applicazione 8 Giorno5*).
- 1.13 Programma che esemplifica una **classe** contenente componenti dati (*Applicazione 1 Giorno6*).
- 1.14 Programma che utilizza una **classe** come componente dati (*Applicazione 2 Giorno6*).
- 1.15 Programma che utilizza metodi all'interno di una **classe**. Accesso a variabili private con metodi opportuni (*Applicazione 1 Giorno7*).
- 1.16 Programma che utilizza una **proprietà** per accesso ad una variabile privata definita all'interno di una **classe**. (*Applicazione 2 Giorno7*).
- 1.17 Programma che utilizza un **costruttore di istanza** per inizializzare componenti dati all'interno di una **classe**. (*Applicazione 3 Giorno7*).

- 1.18 Programma che utilizza un **array** per memorizzare dati. (*Applicazione 1 Giorno8*).
- 1.19 Programma che esemplifica il **poliformismo** dei metodi (*Applicazione 1 Giorno9*).
- 1.20 Programma che esemplifica il **poliformismo** dei costruttori (*Applicazione 2 Giorno9*).
- 1.21 Programma che esemplifica l'**ereditarietà** (*Applicazione 1 Giorno11*) (*Applicazione 2 Giorno11*).

Gli Insegnanti

G. Greco T. Mestriner

(Testi Utilizzati): Appunti forniti dall'insegnante.

C# Guida Completa Bradley L. Jones – Edizione Apogeo – 35.90 euro)