

TUTTI GLI ESERCIZI, SE NON ALTRIMENTI INDICATO, RICHIEDONO LA CREAZIONE DI UN PROGETTO DI TIPO "WINDOWS FORM APPLICATION", IN AMBIENTE "VISUAL STUDIO 2022 COMMUNITY" E "LINGUAGGIO C#", CHE RISOLVA LA PROBLEMATICA INDICATA NEL TESTO.

Riepilogo su Iterazioni, Strutture Dati e Sottoprogrammi 1

1. Chiesti in input due numeri G1 e G2, indicanti i gol segnati dalle due squadre di un partita di calcio, si richiami un sottoprogramma Esito che, ricevendo G1 e G2, restituisca una stringa contenente "1" in caso di vittoria della prima squadra ($G1 > G2$); "2" in caso di vittoria della seconda squadra ($G2 > G1$); "X" in caso di pareggio. Creare due versioni, una con sottoprogramma di tipo FUNZIONE e una con sottoprogramma di tipo PROCEDURA (void).
2. Si dichiari un vettore V costituito da 5 elementi di tipo "int". Si chieda in input un numero intero P e, alla pressione di un pulsante, si richiami un sottoprogramma "Riempi", che, ricevendo P come parametro, memorizzi automaticamente nel vettore i 5 numeri a partire da P a salire (Es.: se $P = 7$ allora V conterrà 7, 8, 9, 10, 11). Quindi, richiamando un sottoprogramma "Visualizza", si copi il contenuto di V in una ListBox. Si usi il vettore V in "visibilità non locale" (non è quindi necessario passare il vettore, come parametro, nei sottoprogrammi).
3. Si chiedano in input 10 carte appartenenti a un mazzo di 40 carte napoletane. L'input di una carta si effettua inserendo una stringa di 2 caratteri composta da numero (A, 2, 3, ..., 7, D, C, R) e palo (D, C, B, S). Esempi: AC, 2S, 6B, RD, ecc. Ad ogni input, si memorizzi la nuova carta in un Vettore e si copi l'intero vettore in una ListBox per rendere subito visibile l'input effettuato. A input terminato si consenta di calcolare quanti Assi sono presenti fra le carte digitate. Sviluppare il codice utilizzando opportuni sottoprogrammi. Si usi il vettore V in "visibilità non locale". Facoltativo: si impedisca l'inserimento di una carta già inserita in precedenza.
4. Si aggiungano all'esercizio N. 2 le seguenti funzioni aggiuntive: (a) si impedisca l'inserimento di carte inesistenti (es.: 9P, KW, 8F); (b) si consenta di eliminare una carta dal vettore; (c) si impedisca di inserire più di 10 carte e non si consenta il calcolo se non ci sono almeno 10 carte; (d) si consenta di scegliere quale numero di carta si desidera conteggiare; (e) dato un palo si copi, in un'altra ListBox, tutte le carte del palo richiesto; (g) si effettui l'ordinamento del vettore secondo il numero di carta. Sviluppare il codice utilizzando opportuni sottoprogrammi.
5. Dato un Elenco di N Nominativi, lo si memorizzi in un Vettore e si utilizzi una ListBox per visualizzarne costantemente il contenuto. Ad input terminato, si usi una TextBox per richiedere la digitazione di un nominativo, se ne individui la posizione nel vettore e se ne evidenzi la posizione nella ListBox. Se il nominativo digitato non esiste si visualizzi un apposito messaggio. Si introducano i seguenti sottoprogrammi: AggiungiDatoAlVettore, CopiaVettoreInListBox, TrovaPosizione. Si usi il vettore V in "visibilità non locale".

Riepilogo su Iterazioni, Strutture Dati e Sottoprogrammi 2

6. Un alunno, durante il corso dell'anno scolastico, svolge diversi compiti scritti e, per ciascuno di essi, il professore esprime un Voto da 2 a 10. Memorizzare i voti in un vettore, visualizzarli in una ListBox e, con opportuni sottoprogrammi, calcolare: (a) la Media dei Voti; (b) il Numero di Sufficienze (voti ≥ 6); (c) il voto Massimo.
7. Su un Conto Corrente bancario si effettuano una serie di versamenti e prelievi (Movimenti). Dato il Saldo Iniziale e la Lista dei Movimenti effettuati (si tratta di numeri interi: positivi per i versamenti, negativi per i prelievi) si calcoli: (1) il saldo corrente; (2) il massimo versamento effettuato; (3) la media dei prelievi effettuati.
8. Dato un vettore V di Nomi, inizialmente vuoto, si consenta di inserire i nomi mantenendo sempre l'elenco ordinato. Per inserire un nome mantenendo l'ordinamento si effettuino due distinte operazioni: (1) si trovi anzitutto la posizione di inserimento; (2) si inserisca in tale posizione. Si risolva il problema introducendo i due relativi sottoprogrammi, oltre a quello necessario per visualizzare i dati del vettore copiandoli in una ListBox.
9. Dato un numero intero N, visualizzare l'elenco di tutti i numeri primi presenti fra 1 ed N. Si introduca un sottoprogramma che, ricevendo un intero, restituisca, come risultato, se esso è o non è un numero primo.
10. Dati due insiemi A e B i cui elementi sono Numeri Interi Positivi (in uno stesso insieme non possono essere presenti due numeri uguali), offrire la possibilità di calcolare i seguenti insiemi: (a) $A \cup B$, ossia l'unione fra A e B (un elemento fa parte dell'unione se è presente in A oppure è presente in B); (b) $A \cap B$, ossia l'Intersezione fra A e B; (l'intersezione contiene solo gli elementi presenti sia in A che in B); (c) $A - B$, ossia la differenza fra A e B; (contiene solo gli elementi presenti in A ma non in B). Memorizzare gli insiemi in oggetti di classe List<T> e visualizzarli tramite ListBox. Introdurre gli opportuni sottoprogrammi per implementare i vari sottoproblemi.

Classi: Costruttori e Metodi

11. Dato un orario (composto da Ora e Minuti - dalle 00:00 alle 23:59), memorizzarlo in un oggetto mioOrario di classe Orario (classe da creare) e offrire le seguenti opzioni: (a) verificare se mioOrario E' o NON E' un orario

antimeridiano (da 0:00 a 11:59); (b) calcolare il numero di minuti che intercorrono fra la mezzanotte precedente e mioOrario; (c) aumentare mioOrario di un'ora e (d) visualizzare nuovamente mioOrario, così aggiornato. Implementare la classe Orario in modo che preveda tutti gli attributi interni, i costruttori e i metodi, necessari alla risoluzione del problema.

12. Fornite in input due date D1 e D2, successive al 1/1/2000, si permetta di: (a) di individuare quale delle due date è più recente (impostare il metodo nella forma "D1.PrecedenteA(D2)" restituendo un bool); (b) quanti giorni intercorrono fra le due date (impostare il metodo come indicato nel punto precedente); (c) quanti giorni sono trascorsi dalla data D1 al successivo Natale. Si introduca una classe di nome MiaData e, per semplicità, si supponga che tutti i mesi sono di 30gg e che l'anno è di 360gg. (Suggerimento: per confrontare o calcolare giorni intercorsi fra date, è utile calcolare quanti giorni intercorrono dal 1/1/2000 fino alla date stesse ...)
13. Dato un Elenco di Nominativi, si consenta di: (a) Aggiungere un nominativo alla fine; (b) Eliminare un nominativo; (c) Ordinare l'elenco. Si visualizzi l'elenco in una ListBox, aggiornandola ad ogni modifica. Si introduca una classe Elenco con gli opportuni membri.

Classi: Costruttori e Metodi - 2

14. Dati in input le misure di Base e Altezza di un rettangolo, creare un oggetto mioRettangolo di classe Rettangolo e fornire le seguenti opzioni: (a) calcolare l'Area; (b) calcolare il Perimetro; (c) verificare se E' o NON E' un Quadrato; (d) modificare i dati del rettangolo raddoppiando Base e Altezza; (e) visualizzare, in un'unica Label, Base e Altezza del Rettangolo. Implementare, nella classe Rettangolo, gli opportuni costruttori e metodi.
15. Per individuare una carta da gioco "napoletana" (mazzo di 40 carte) usiamo due numeri interi: il primo, da 1 a 10 indica il Numero della carta (1= Asso, ... 10=Re); il secondo, da 1 a 4, ne indica il Palo (1=Denari, 2=Coppe; 3=Bastoni; 4= Spade). Chieste in input DUE carte C1 e C2, determinare se le due carte specificate SONO o non SONO uguali: se lo sono dare un messaggio di errore; se non lo sono, determinare quale carta è vincente in una mano di Briscola, tenendo conto che C1 viene giocata per prima e che il palo di briscola (intero) è dato in input. Visualizzare le due carte usando una label per carta, esprimerle "correttamente" (non "1 2", bensì "Asso di Coppe") ed evidenziare la carta vincente con sfondo rosso. Introdurre una classe Carta, con opportuni costruttori e metodi.
16. La griglia di partenza di una gara automobilistica è costituita da una sequenza di 10 piloti. Inseriti in input i nomi dei 10 piloti (specificati secondo l'ordine di griglia), parte la gara e, per gestirla, si offrano le seguenti operazioni: (a) sorpasso: il pilota in posizione P viene superato da quello in posizione P+1; (b) perdita di posizioni (pit-stop, uscita di strada, penalità): il pilota in posizione P perde, in un colpo solo, N posizioni; (c) ritiro: il pilota in posizione P si ritira dalla gara. Dopo l'input dei piloti e dopo ogni operazione, si aggiorni la visualizzazione della classifica in una ListBox. Usare la stessa ListBox per selezionare la posizione P interessata alle varie operazioni richieste. A fine gara, visualizzare, in una label, una stringa con i nomi del 1°, 2° e 3° classificato.

Classi: Proprietà ed Elementi Privati

17. Data UNA Partita di Calcio (es. Inter-Milan 1-1), si visualizzi: (a) il relativo Segno del Totocalcio (1, X o 2); (b) il Nome della squadra vincente (in caso di pareggio si visualizzi "PAREGGIO"); (c) il numero totale di Gol segnati. Si introduca una classe Partita con il solo Costruttore Vuoto e le opportune Proprietà e Metodi (una Proprietà per ogni attributo interno e un metodo per ognuna delle elaborazioni richieste).
18. Dati due Punti del piano cartesiano P1(x1,y1) e P2(x2,y2) determinare: (a) la distanza fra i due punti; (b) se i due punti appartengono o no allo stesso quadrante; (c) il Perimetro del triangolo Origine-P1-P2; (d) verificare se entrambi i punti cadono all'interno di un cerchio con centro nell'Origine e raggio R, chiesto in input. Si introduca una classe Punto con il solo Costruttore Vuoto e le Proprietà, Metodi ed Elementi Privati necessari.
19. Due segnaposto si muovono su un percorso di 100 caselle partendo dalla prima casella e avanzando ad ogni turno secondo quanto indicato dal lancio di un dado. 20 delle 100 caselle sono "trappole" con dentro un numero da 1 a 10: se un segnaposto "capita" su una di esse viene arretrato di tanti posti quanto indicato dal numero nella casella. Si consenta a due giocatori di simulare una partita che termina solo quando uno dei due segnaposto supera la posizione 100. Si introducano le classi Dado, Segnaposto, Percorso con gli opportuni membri. Per "visualizzare" il percorso si usi una semplice ListBox oppure un DataGridView con un'unica riga, modificando il colore di sfondo per indicare la posizione dei due segnaposto.

Classi ed Elementi di Grafica in C#

20. Usando un Timer si mantenga una Pallina in movimento sull'area grafica. Inizialmente la Pallina deve procedere verso il BASSO ma tramite dei Pulsanti SU, GIU, DESTRA, SINISTRA deve essere possibile cambiarne la direzione. Se la Pallina esce da un lato, essa deve riapparire dal lato opposto dell'area grafica e continuare la sua marcia. Si introduca una classe Pallina con opportuni costruttori, proprietà e metodi.

21. Usando un Timer si mantengano due Palline in movimento sull'area grafica. Inizialmente, le palline hanno una posizione e una direzione di marcia prestabilite (SU, GIU, DESTRA o SINISTRA) ma, alla pressione di un apposito pulsante, esse devono modificare casualmente la loro direzione di marcia. Si introduca una classe Pallina con opportuni costruttori, proprietà e metodi.
22. Usando un Timer si mantenga una Pallina in movimento sull'area grafica. Tramite dei TASTI della tastiera deve essere possibile modificare la Direzione (N, NE, E, SE, S, SO, O, NO), la velocità e la dimensione. In caso di "urto" con il limite dell'area grafica, la pallina deve "rimbalzare" mutando la direzione correttamente in base al rimbalzo subito. Si introduca una classe Pallina con opportuni costruttori, proprietà e metodi.

Membri Statici

23. Chiedere in input, tramite appositi controlli (TextBox o NumericUpDown), le coordinate X e Y di due Punti P1 e P2. Tramite un pulsante, disegnare i due punti su un Panel, rappresentandoli con un cerchio di diametro piccolissimo (ad esempio 4 pixel). Se P1 e P2 sono allineati in orizzontale, tracciare fra loro una linea Verde; Se sono allineati in verticale, tracciare fra loro una linea Blu; se non sono allineati, tracciare fra loro una linea Rossa. Introdurre una classe Punto con gli opportuni attributi, costruttori, proprietà e metodi, con particolare attenzione all'individuazione e all'uso dei necessari Attributi Statici.
24. Una "Linea", tracciata in grafica su un controllo Panel, inizia da un punto grafico (X,Y), si estende in una direzione D (su, giù, destra o sinistra) per una determinata lunghezza L espressa in pixel e, infine, si muove con un passo di P pixel. Con un Timer, mantenere la Linea sempre in movimento, spostandola, ad ogni Tick, di P pixel nella propria direzione D: Consentire, con appositi pulsanti, di (a) modificare la direzione di marcia tramite l'uso di due pulsanti GiraASinistra (gira a sinistra di 90 gradi) e GiraADestra (gira a destra di 90 gradi); (b) allungare o accorciare la linea; (c) riportare la linea alla posizione (0,0) con direzione "destra" e lunghezza 10 pixel; (d) accelerare o rallentare la linea, modificando il suo passo. Infine, se la linea tocca il bordo del Panel, essa deve automaticamente girare a destra. Introdurre, in particolare, gli opportuni Attributi o Membri Statici.

Ereditarietà e Polimorfismo

25. Si modifichi l'esempio pilota "PallinaCheEredita" in modo che sul Panel, oltre a una Pallina e un Triangolino, appaia in movimento anche un Quadrato. Per far questo basta creare una nuova classe Quadrato, derivata dalla classe base FiguraGrafica, la cui implementazione è del tutto simile alle altre classi derivate Pallina e Triangolino.
26. Si modifichi l'esempio pilota "PallinaCheEredita" in modo sia possibile impostare e variare il Colore delle figure, la loro Velocità di Spostamento (nell'esempio è fissata a 2 pixel per ogni Tick del Timer). Far sì, inoltre, che la Pallina sia visualizzata come una "ciambella", ossia con un ulteriore cerchio al suo interno, secondo un ulteriore diametro secondario (ovviamente minore di quello principale).
27. Si modifichi l'esempio pilota "PallinaCheEredita" in modo da generare un numero crescente di figure (Pallina, Triangolino o Quadrato o Stellina) che si muovano sul pannello variando il loro moto in modo casuale (sia come velocità che come direzione) e, inoltre, aumentando/diminuendo casualmente anche la loro dimensione. Un pulsante consente all'utente di far "apparire" una nuova figura (la cui forma è decisa, a caso, fra le quattro forme previste) che va, quindi, ad aggiungersi alle altre già presenti.

Ereditarietà e Polimorfismo 2

28. I praticanti italiani del gioco del Tennis sono identificati da un Nominativo, dalla Regione in cui giocano (Calabria, Lazio, Lombardia, ecc.) e se sono o no "mancini". Alcuni di essi sono tennisti Classificati, ossia sono iscritti alla Federazione per poter partecipare ai tornei ufficiali. I tennisti classificati hanno un Numero di Iscrizione e una Categoria (numero intero da 1 a 4). Introducendo una appropriata gerarchia di classi, scrivere un programma che, forniti i dati di un Tennista (classificato o no), calcoli, anche in base alla sua "macroarea" di appartenenza, il costo della Tessera Annuale: per i tennisti non classificati è sempre di € 50; per i tennisti classificati, invece, è di € 100 con i seguenti sconti di area: Nord=0%, Centro=10%, Sud=20%, Isole=30%.
29. Disegnare sull'area di disegno un cerchio e un quadrato, quindi, calcolarne l'area. Un numero intero indica il colore della figura: 1=Rosso, 2=Verde; 3=Blu; 4=Nero. Chiedere in input, raggio e colore del cerchio, lato e colore del quadrato; quindi, tracciare le due figure sull'area di disegno in posizioni prefissate a piacere; infine, calcolare e visualizzare l'area di entrambe. Introdurre una corretta gerarchia di classi (classe base e classi derivate) collocando opportunamente gli attributi e introducendo i necessari metodi da sottoporre ad override.

Ereditarietà e Polimorfismo 3

30. Un Atleta è definito da un Nominativo, un'Età (numero intero, in Anni) e una Nazionalità. Un Calciatore è un Atleta e, in più, gioca in una Squadra, ha un Ruolo e un Numero di Maglia. Un Atleta è considerato "Juniore" se ha un'Età inferiore ai 18 anni. Un Atleta è considerato "Over" (cioè "troppo vecchio") se ha più di 35 anni, tranne nel

caso dei Portieri, che giocano fino a 40 anni. Chiedi i dati di un Calciatore, calcolare e visualizzare: (a) un unico testo che riporti tutti i dati del Calciatore; (b) se è vero o no che il Calciatore è "Juniores"; (c) se è vero o no che il Calciatore è "Over". Introdurre una appropriata gerarchia di classi, includendo, in particolare, i metodi "eOver" e "Testo" come metodi polimorfici.

31. Una cartoleria vende prodotti caratterizzati da Codice, Descrizione, Prezzo di Listino. Alcuni di questi prodotti sono Penne caratterizzate a loro volta da un Colore, dal fatto che sono o non sono "a scatto" e dal numero di "penne" presenti in una singola confezione. Introducendo una appropriata gerarchia di classi, scrivere un programma che, forniti i dati completi di un prodotto di tipo Penne, calcoli il prezzo di una singola penna, considerato che il prezzo di listino riguarda l'intera confezione e che, in questo momento, alle penne "a scatto" viene applicato uno sconto particolare da chiedere in input.

Tecniche di Programmazione Object Oriented

32. Modificare l'esempio `AlmenoUnaCoppiaInUnaManoDiPoker` consentendo di selezionare una carta fra quelle in mano al giocatore e di sostituirla con la prima estratta dal mazzo.
33. Modificare l'esempio `AlmenoUnaCoppiaInUnaManoDiPoker` consentendo la sostituzione di più carte fra quelle in mano al giocatore. A tale scopo si prevedano due pulsanti: il primo, previa selezione di una carta del giocatore, ne provoca l'eliminazione (può essere usato più volte quindi consente di scartare più carte fino a un massimo di 5); il secondo, quando premuto, verifica quante carte sono state scartate dalla mano del giocatore e distribuisce le carte mancanti prelevandole sempre dall'inizio del mazzo.
34. Modificare l'esempio `AlmenoUnaCoppiaInUnaManoDiPoker` gestendo due giocatori (quindi due "mani" da 5 carte ciascuna) e verificare quale dei due giocatori ha, fra le proprie la carte, la carta più "alta" (l'asso è superiore a tutte le carte. A parità di carta massima, valgono i pali, secondo l'ordine C, Q, F, P).
35. Modificare l'esempio `AlmenoUnaCoppiaInUnaManoDiPoker` identificando quale punto del Poker ha in mano il giocatore (Niente, Coppia, Doppia-Coppia, Tris, Scala, Full, Colore, Scala-Reale). Consiglio: l'identificazione del punto è più semplice se le carte sono "ordinate" per numero.

Utilizzo di più Form

36. Data una partita di calcio (es. Fiorentina-Roma 2-1) si calcolino i punti in classifica che le due squadre ottengono a fine partita (es. Fiorentina = 3 e Roma = 0). L'input dei dati della partita deve essere effettuato in una form distinta.
37. Specificate due date D1 e D2, calcolare il numero di giorni che intercorrono fra D1 e D2. Si supponga che: le date siano successive al 1/1/2000; che i mesi siano tutti di 30gg; che tutti gli anni siano di 360gg. L'input di una singola data deve avvenire in una form distinta `frmInputData` e, visto che le date sono due, sono necessari due istanze di questa form. (consiglio: calcolare quanti giorni intercorrono fra il 1/1/2000 e le date D1 e D2, quindi fare la semplice differenza fra i due numeri interi così ottenuti).
38. In un Ufficio Postale i clienti versano somme di denaro per pagare le bollette. Il programma deve consentire di specificare in input l'importo X da pagare. Quindi, in una form distinta, deve consentire di specificare quante e quali banconote/monete vengono versate (es. 5 da 20€, 3 da 10€, 1 da 0,50€). Infine, nella form principale, deve essere verificato che il totale delle banconote/monete versate coincida con l'importo X richiesto.

Eventi

39. Dato un orario, calcolare quanti minuti (a partire da esso) mancano alla mezzanotte. Introdurre una classe "Orario" con gli opportuni membri e, in particolare, introdurre due eventi `OraErrata` e `MinutiErrati` che vengono generati quando si cerca di assegnare valori errati alle proprietà dell'orario. Nel codice della form si definiscano i due Gestori relativi ai due eventi suddetti, necessari per "gestire" l'inserimento di orari non accettabili, visualizzando opportuni messaggi di errore.
40. Su un Conto Corrente bancario, inizialmente a 0, si effettuano una serie di "movimenti", ossia di versamenti e prelievi (sono numeri Interi positivi o negativi). Ogni volta che, a seguito di un prelievo, il Saldo (ossia la somma algebrica di tutti i movimenti) risulta sotto lo 0, la banca effettua un prelievo automatico di 10 €. Ogni volta che, a seguito di un versamento, il saldo risulta sopra i 100.000 €, la banca effettua un accredito automatico pari all'1% dell'importo del versamento effettuato. Introdurre una classe "ContoCorrente" con gli opportuni membri e, in particolare, introdurre due eventi `SottoZero` e `OltreCentomila` grazie ai quali, con i relativi Gestori nel codice della form, si possano generare automaticamente i movimenti "automatici" sopra descritti.
41. Consentire a due utenti (chiamati Pallino "O" e Crocetta "X") di giocare a Tris. Utilizzare un `DataGridView` (3x3) per visualizzare lo schema e specificare le mosse. Introdurre una classe `SchemaTris` per memorizzare e gestire lo

schema 3x3 del Tris con gli opportuni membri. Introdurre in particolare un evento "TrisCompletato" che si verifica quando uno dei due giocatori completa il Tris (in orizzontale, verticale o diagonale) e, quindi, vince la partita.

Controlli Utente

42. Creare un Controllo Utente che consenta di gestire l'input/output di un Voto, ossia di un numero intero compreso fra 1 e 10. Disporre, nel controllo, una TextBox (o anche un NumericUpDown), impedendo di digitare valori errati. Introdurre una proprietà booleana "AttivaColorazione", che, se impostata a TRUE, deve provocare la costante colorazione automatica dello sfondo della TextBox, secondo la seguente semplice regola: Rosso se $Voto < 5$; Giallo se $Voto = 5$; Verde se $Voto > 5$. Se "AttivaColorazione" è FALSE, lo sfondo è sempre Bianco. Per testare il controllo, scrivere un programma che, usando il controllo, chieda in input un voto e consenta, con una CheckBox, di attivare/disattivare la suddetta funzione di colorazione.
43. Creare un Controllo Utente che permetta di gestire l'input/output di una singola Squadra di Calcio. Più precisamente, disporre nel controllo utente una singola ComboBox nel cui Elenco a Discesa, siano presenti tutte le squadre del campionato, consentendo di sceglierne una. Introdurre una proprietà "Nazione" che possa essere impostata a "ITA", "SPA" o "ENG" per scegliere di quale nazione è il campionato di cui elencare le squadre. Per testare il controllo, scrivere un programma che, specificata la nazione fra quelle previste, effettui l'input delle due squadre di una partita (usando 2 istanze del controllo utente) e il relativo risultato, visualizzando i punti acquisiti da ciascuna delle due squadre (3 vittoria, 1 pareggio, 0 sconfitta).
44. Creare un Controllo Utente per gestire l'input/output di un Numero Binario. Il controllo, oltre a consentire solo l'inserimento di 0 e 1, deve prevedere una proprietà "MaxNumCifre", che specifichi il Massimo Numero di Cifre Binarie che è possibile inserire (ad esempio: se $MaxNumCifre = 8$ allora deve essere possibile inserire al massimo un numero binario di 8 cifre) e un metodo per Convertire il numero in Decimale. Per testare il controllo si scriva un programma che chiesto un numero binario lo converta in decimale.

GLI ESERCIZI SEGUENTI, SE NON ALTRIMENTI INDICATO, RICHIEDONO
LO SVILUPPO DI **APP ANDROID** IN AMBIENTE **ANDROID STUDIO** E **LINGUAGGIO JAVA**

App Android: Primo Approccio ad Android Studio

45. Realizzare una App Android che, alla pressione di un pulsante, visualizzi, in una TextView, il messaggio "CIAO!".
46. Realizzare una App Android che consenta, tramite delle EditText, di digitare due numeri interi e, tramite un pulsante, consenta di calcolare e visualizzare la somma dei due numeri digitati.

App Android: Constraint Layout, Risorse e Stili

47. Completare l'esempio pilota "Calcolatrice" aggiungendo e posizionando correttamente i pulsanti numerici mancanti (da 4 a 9 più lo 0) e i pulsanti degli operatori mancanti (la moltiplicazione e la divisione). Applicare ai pulsanti degli operatori e al pulsante C uno stile grafico diverso da quello dei pulsanti numerici. Utilizzare, per l'icona della app e per lo sfondo della calcolatrice, immagini diverse da quelle dell'esempio.
48. Estendere le funzionalità dell'esercizio precedente aggiungendo i pulsanti per gestire la memoria: "M+" aggiunge alla memoria quello che c'è sul display; "M-": sottrae alla memoria quello che c'è sul display; "MR" pone sul display il numero presente in memoria; "MC" azzerla la memoria. Applicare, ai pulsanti della memoria, uno stile grafico diverso da quello dei pulsanti numerici e da quello dei pulsanti degli operatori.

App Android: Uso di più Activity

49. Realizzare una App Android per il calcolo di Area di un Rettangolo. L'input di Base e Altezza deve avvenire nella Activity principale, mentre il calcolo e la visualizzazione dei risultati, in una seconda Activity.
50. Realizzare una App Android per il calcolo dell'Area di un Rettangolo oppure di un Triangolo. La scelta della figura (Triangolo o Rettangolo) deve avvenire nella Activity Principale. Essa trasmette la scelta a una Seconda Activity che provvede ad effettuare l'input di Base e Altezza, a calcolare l'Area e a restituire tale risultato alla Activity principale che visualizzerà l'Area così ricevuta.

App Android: Grafica

51. Completare l'esempio pilota "DisegnareInGrafica" aggiungendo pulsanti per: (a) disegnare, al centro dell'area grafica, un CERCHIO GIALLO che ad ogni click diventa più grande; (b) disegnare, al centro dell'area grafica, un QUADRATO BIANCO che ad ogni click diventa più grande; (c) spostare il "pacman", oltre che a DESTRA, anche nelle direzioni SU, GIU', SINISTRA.
52. Su un'Area Grafica vengono disegnati 5 ostacoli (di forma quadrata e con la stessa immagine per tutti) posti in punti prefissati. Un "pacman", inizialmente posizionato al bordo sinistro dell'area, può muoversi, con appositi pulsanti, solo a DESTRA, SU e GIU'. Scopo del gioco è muovere il "pacman" e raggiungere il bordo destro, senza toccare

nessuno dei 10 ostacoli. Facoltativo: generare casualmente la posizione degli ostacoli (classe Random); gestire più livelli con un aumento del numero e delle dimensioni degli ostacoli.

App Android: Grafica e Touch

53. Modificare l'esempio pilota "SpostareConUnDito" in modo che il "pacman", normalmente fermo, si sposti nella direzione indicata con il movimento del dito sul touch-screen dello smartphone. Per semplicità: (a) il movimento è possibile solo nelle direzioni su, giù, destra, sinistra (non in diagonale); (b) ad ogni tocco, il "pacman" deve solo spostarsi, nella direzione indicata, di 20px e restare fermo in quella posizione.
54. Modificare l'esempio pilota "SpostareConUnDito" in modo che il "pacman" si muova costantemente in una certa direzione e velocità. Il movimento del dito sul touch-screen dello smartphone, consente di imporre una nuova direzione e una nuova velocità al "pacman". Direzione e velocità possono essere qualsiasi.

App Android: Grid Layout e Matrici di Controlli

55. Estendere l'esempio pilota "GiocoGriglia" in modo che, anziché 4, siano visualizzate 6 faccine (3 righe e 2 colonne).
56. Modificare l'esempio pilota "GiocoGriglia" in modo che: (a) come nell'esercizio precedente, anziché 4, siano visualizzate 6 faccine (3 righe e 2 colonne); (b) un tocco su una faccina "visibile" ne provoca la scomparsa e un ulteriore tocco la rende nuovamente visibile.

App Android: ListView

57. Realizzare una App Android che visualizzi, tramite una ListView, l'elenco dei giocatori della tua squadra di calcio preferita. Visualizzare, nella parte superiore dell'activity, il nome della squadra e, di seguito, l'elenco dei giocatori nel seguente formato: <numero maglia> - <nome> <cognome> (<ruolo(solo 3 lettere)>). Esempio: 7 - Cristiano Ronaldo (ATT).
58. Modificare l'esercizio precedente in modo che siano elencati solo i cognomi dei giocatori. Al tocco di uno di essi, visualizzare, in una finestrina (Toast), tutti i dati del giocatore selezionato, nella forma: <numero maglia> - <nome> <cognome> (<ruolo(solo 3 lettere)>). Esempio: 7 - Cristiano Ronaldo (ATT).

Strutture Dinamiche

59. Gestire una Lista di Numeri Interi memorizzandola in una Struttura Dinamica. La Lista è inizialmente vuota. Si consenta di: (a) aggiungere nuovi Numeri Interi all'inizio della lista; (b) calcolare la Somma dei dati nella lista. Ad ogni modifica una ListBox deve mostrare automaticamente la lista aggiornata.
60. Gestire una Lista di Numeri Interi memorizzandola in una Struttura Dinamica. La Lista è inizialmente vuota. Si consenta di: (a) aggiungere nuovi Numeri Interi all'inizio della lista; (b) calcolare la Media Aritmetica dei dati nella lista. Ad ogni modifica una ListBox deve mostrare automaticamente la lista aggiornata (consiglio: creare un sottoprogramma "FUNCTION Conta (ByVal Testa as Nodo) as Integer" che restituisca il numero di dati presenti nella lista).
61. Gestire una Lista di Numeri Interi memorizzandola in una Struttura Dinamica. La Lista è inizialmente vuota. Si consenta di: (a) aggiungere nuovi Numeri Interi all'inizio della lista; (b) calcolare il Minimo dei dati nella lista; Ad ogni modifica una ListBox deve mostrare automaticamente la lista aggiornata. Si implementi una classe LISTA che incapsuli la Struttura Dinamica e offra tutti i membri (proprietà e metodi) necessari per risolvere il problema.

Pila, Coda e Lista Generica

62. Si Gestisca una PILA di Nominativi utilizzando una Struttura Dinamica. Sulla Pila, inizialmente vuota, si offrano le seguenti opzioni: (a) Dato un Nuovo Nominativo, Inserirlo in Testa; (b) Eliminare il Nominativo in Testa; (c) Dato un Nominativo, verificare se esso "E' Presente" oppure "Non è Presente" nella Pila; (d) Svotare la Pila. Dopo ogni modifica, l'intera Pila deve essere copiata (e quindi visualizzata) in una ListBox.
63. Si Gestisca una CODA di Nominativi utilizzando una Struttura Dinamica. Sulla Coda, inizialmente vuota, si offrano le seguenti opzioni: (a) Dato un Nuovo Nominativo, Inserirlo in Fondo alla Coda; (b) Eliminare il Primo Nominativo in Coda; (c) Dato un Nominativo, determinare la Posizione del Nominativo nella Coda (se non è presente visualizzare un appropriato messaggio); (d) Svotare la Coda. Dopo ogni modifica, l'intera Coda deve essere copiata (e quindi visualizzata) in una ListBox. Suggerimento: oltre al puntatore d'entrata TESTA, mantenere anche un secondo puntatore FONDO e far sì che esso "punti" sempre all'ultimo elemento.
64. Si Gestisca una LISTA GENERICA di Nominativi utilizzando una Struttura Dinamica. Sulla Lista, inizialmente vuota, si offrano le seguenti opzioni: (a) Inserimento di un Nuovo Nominativo in Testa, in Fondo o in una data Posizione "intermedia"; (b) Data una Posizione (0 = primo), Eliminare il Nominativo nella posizione data; (c) Dato un Nominativo, eliminarlo dalla Lista; (d) Data una Posizione, evidenziare il Nominativo corrispondente (0 = primo); (e) Svotare la Lista. Dopo ogni modifica, l'intera Lista deve essere copiata (e quindi visualizzata) in una ListBox.

Suggerimenti: oltre al puntatore d'entrata TESTA, mantenere anche un secondo puntatore FONDO e far sì che esso "punti" sempre all'ultimo elemento. Si ricordi che per inserire/eliminare, è necessario "trovare" il Puntatore all'Elemento Precedente rispetto a quello da inserire/eliminare.

Ricorsione

65. Realizzare un sottoprogramma ricorsivo che, ricevuto un numero intero positivo N, determini la somma Sp dei primi N numeri pari (es. se N=6 allora $Sp = 12+10+8+6+4+2 = 42$). E' possibile definire Sp(N), ossia la Somma dei primi N numeri pari, in modo ricorsivo, come segue: (a) caso ricorsivo - se $N > 1$ allora $Sp(N) = 2*N + Sp(N-1)$; (b) caso base - se, invece, $N = 1$ allora $Sp(N) = 2$. Scrivere un progetto per testare il sottoprogramma.
66. Realizzare un sottoprogramma ricorsivo che, dato un numero intero N, restituisca il valore della potenza 2^N . Scrivere un progetto per testare il sottoprogramma.
67. Realizzare una Function ricorsiva che, dato un numero intero N, restituisca una stringa contenente la sua conversione in binario. Si ricordi che, per convertire N in binario, è possibile convertire in binario il numero $(N / 2)$ e aggiungere alla fine il resto della divisione $(N / 2)$. Ad esempio, se $N = 25$ allora la sua conversione in binario è: la conversione di $(25 / 2) = 12$ ("1100") con, in aggiunta, alla fine, il resto di $(25 / 2) = 1$, quindi "11001".
68. Dato un Elenco ordinato di N nominativi, memorizzarlo in un vettore Vet. Dato un nominativo NomeDaCercare, determinare la posizione (in base 0) di NomeDaCercare nel vettore Vet (se non è presente, restituire -1). Per effettuare la ricerca implementare un sottoprogramma ricorsivo TrovaPosizione che, ricevendo Vet, NomeDaCercare e la zona di ricerca LimiteInferiore, LimiteSuperiore, restituisca la posizione del nominativo nella zona di ricerca specificata. Inizialmente la zona di ricerca è estesa a tutto il vettore, quindi LimiteInferiore = 0 e LimiteSuperiore = N-1.

Liste Ricorsive

69. Si effettui l'input di una sequenza di numeri interi, memorizzandola in una Lista Dinamica gestita con un unico puntatore iniziale "Testa". Ogni nuovo numero dovrà essere aggiunto alla fine della lista. La lista aggiornata dovrà essere costantemente visualizzata copiandola in una ListBox. A input terminato, si calcoli la Somma dei numeri della lista. Per risolvere il problema si introducano i seguenti sottoprogrammi RICORSIVI: AggiungiAllaFine, VisualizzaInListBox, Somma.
70. Si gestisca una Lista di Nominativi, inizialmente vuota, memorizzandola in una Struttura Dinamica. Si offrano le seguenti opzioni, da realizzare ciascuna implementando un Sottoprogramma Ricorsivo: (a) Inserimento di un Nuovo Nominativo in una data Posizione; (b) Eliminazione di un Nominativo, data la sua Posizione; (c) Dato un Nominativo, verificarne l'Esistenza nella lista. Dopo ogni modifica visualizzare la lista in una ListBox richiamando, anche per questo, un sottoprogramma ricorsivo. Consiglio: per inserire un Nuovo Nominativo in posizione P ... se $P = 0$ si inserisce in testa; se $P > 0$ si inserisce nella "sottolista" in posizione $P-1$...
71. Si gestisca una Lista Ordinata di Nominativi, inizialmente vuota, memorizzandola in una Struttura Dinamica. Si offrano le seguenti opzioni: (a) Inserimento di un Nuovo Nominativo mantenendo l'ordinamento; (b) Dato un Nominativo, eliminarlo dalla lista; (c) Dato un Nominativo, verificarne l'Esistenza nella lista e, se esiste, determinarne la Posizione. Dopo ogni modifica visualizzare la lista in una ListBox. Utilizzare, per tutte le operazioni, dei Sottoprogrammi Ricorsivi.

GLI ESERCIZI SEGUENTI, SE NON ALTRIMENTI INDICATO, RICHIEDONO LO SVILUPPO E IL TEST DI PAGINE WEB DINAMICHE CON PROGRAMMAZIONE LATO SERVER IN LINGUAGGIO PHP E IL LORO TEST SU SERVER WEB APACHE PER WINDOWS (PACCHETTO XAMPP)

Programmazione Web - Primo Approccio al Linguaggio PHP

72. Realizzare e testare un sito web dinamico, in linguaggio PHP, costituito da una singola pagina "index.php", che consenta ad utenti remoti l'utilizzo di una semplice Calcolatrice. Si consenta di digitare, in DUE TextBox, i due dati su cui operare; si consenta, quindi, la scelta dell'operazione tramite QUATTRO RadioButton (addizione, sottrazione, moltiplicazione, divisione); infine, tramite un pulsante, si consenta il calcolo del risultato e la visualizzazione del risultato nella parte bassa della pagina stessa.
73. Realizzare e testare un sito web dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" che consenta ad utenti remoti di inserire un numero intero (da 1 a 9) e ricevere sulla pagina la rispettiva tabellina (es. inserendo 6 si ottiene 6 12 18 24 32 36 42 48 54 60).
74. Realizzare e testare un sito web dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" che consenta ad utenti remoti di inserire un numero intero (da 1 a 99) e vederlo sulla pagina rappresentato in lettere (es. inserendo 25 si ottiene "venticinque").

Programmazione Web - Elementi di Base del Linguaggio PHP

75. Realizzare un Sito Web Dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" e che, digitato il Numero di Maglia di un giocatore della propria squadra preferita, ne visualizzi il Nominativo. Memorizzare i Nominativi dei Giocatori della squadra, in un Vettore Associativo definito direttamente all'interno del codice PHP. Prevedere un secondo pulsante che permetta di visualizzare direttamente il nominativo del portiere (cioè del giocatore con maglia N. 1) e un terzo pulsante per visualizzare il numero complessivo di giocatori archiviati. Utilizzare almeno 4 sottoprogrammi. La prima volta che la pagina viene richiamata, appare un messaggio di benvenuto che sparisce dopo la prima ricerca.
76. Realizzare un Sito Web Dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" che, digitato il nome di una Regione d'Italia (es. Calabria, Lazio, Piemonte, ecc.), visualizzi, su righe separate, l'Elenco delle Province relative alla Regione specificata. Un pulsante a parte deve visualizzare il numero di abitanti della Regione stessa. Utilizzare quanti più sottoprogrammi possibili, per "incapsulare" le operazioni necessarie.

Programmazione Web - ListBox e ComboBox in PHP

77. Realizzare un Sito Web Dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" che elenchi, in una ComboBox, i nominativi dei giocatori della propria squadra preferita. Una volta selezionato un nominativo, tramite un pulsante, si visualizzi il numero di maglia del giocatore selezionato. Ove possibile, introdurre opportuni sottoprogrammi. La prima volta che la pagina viene richiamata, appare un messaggio di benvenuto che sparisce dopo la prima ricerca.
78. Realizzare un Sito Web Dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" che elenchi, in una ComboBox, i nomi di tutte le Regioni d'Italia. Alla pressione di opportuni pulsanti: (a) Si visualizzi in una ListBox l'elenco di tutte le province della regione selezionata; (b) si visualizzi il numero di abitanti complessivo della regione selezionata; (c) si visualizzi il numero di abitanti della provincia selezionata nella ListBox. Utilizzare sottoprogrammi ove opportuno.

Programmazione Web – PHP e SQLite: SELECT

79. Utilizzando il software "DB-Browser for SQLite" creare manualmente un DataBase SQLite di nome "Calcio.db" e, in esso, creare manualmente una Tabella di nome "Calciatori" con i seguenti campi: idCalciatore (chiave primaria intera di tipo "Autoincrement"), Nominativo, GolFatti. Realizzare, quindi, un Sito Web Dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" che, chiedendo in input un numero minimo M di gol fatti, acceda "lato server" al suddetto DataBase SQLite e riporti sulla pagina l'elenco dei Nominativi dei soli calciatori che abbiano realizzato più di M gol.
80. Utilizzando il software "DB-Browser for SQLite" creare manualmente un DataBase SQLite di nome "Calcio.db" e, in esso, creare manualmente una Tabella di nome "Calciatori" con i seguenti campi: idCalciatore (chiave primaria INTEGER di tipo "Autoincrement"), Nominativo, DataDiNascita (TESTO nella forma "AAAA/MM/GG"), Ruolo, Squadra. Realizzare, quindi, un Sito Web Dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" che, digitati un ruolo R e una data D (sempre nella forma "AAAA/MM/GG"), riporti, sulla pagina, l'elenco dei soli Calciatori (riportare Nominativo e Squadra) che giocano in ruolo R e, al tempo stesso, siano nati prima della data D.

Programmazione Web – PHP e SQLite: INSERT, UPDATE, DELETE

81. Realizzare un Sito Web Dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" e che, digitato il Nominativo di un Calciatore, ne visualizzi la Squadra di appartenenza. I dati dei Giocatori sono memorizzati in un DataBase SQLite "Calcio.db", situato nella cartella in cui il sito è pubblicato sul server. Il DataBase contiene una Tabella "Calciatori" con i campi IdCalciatore (chiave primaria), Squadra e Nominativo. Consentire, inoltre, di aggiungere un nuovo giocatore al DataBase, specificandone Nominativo e Squadra.
82. Realizzare un Sito Web Dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" che, scelta una Squadra di Calcio del Campionato di Serie A (selezionandola da una apposita Casella a Discesa), visualizzi in un tag <select> l'elenco dell'intera formazione della Squadra, indicando su ogni singola riga il Numero di Maglia e il Nominativo del Giocatore. Si offra anche la possibilità di aggiungere, modificare e eliminare Giocatori alla squadra selezionata. I dati dei Giocatori sono memorizzati in un opportuno DataBase SQLite "Calcio.db" lato server.
83. Realizzare un Sito Web Dinamico, in linguaggio PHP, che consenta ad utenti remoti di visionare e aggiornare liberamente un Elenco di Frasi "Simpatiche". Tale elenco, affinché "sopravviva" durante e dopo la navigazione, deve essere gestito sul server con un DataBase SQLite lato server. Nella sua Home Page, visualizza l'Elenco delle Frasi in un tag <select>, consentendo di: (a) aggiungere una nuova Frase; (b) selezionata una frase sulla ListBox, eliminarla; (c) se viene specificata una Parola, visualizzare solo le frasi contenenti quella parola.

TUTTI GLI ESERCIZI, SE NON ALTRIMENTI INDICATO, RICHIEDONO LA CREAZIONE DI UN PROGETTO DI TIPO "WINDOWS FORM APPLICATION", IN AMBIENTE "VISUAL STUDIO 2022 COMMUNITY" E "LINGUAGGIO C#", CHE RISOLVA LA PROBLEMATICA INDICATA NEL TESTO.

Memorizzazione su Disco - File Testo in C#

84. Un elenco di Importi in Euro è contenuto in un file testo di nome "Importi.txt". Offrire le seguenti opzioni: (a) Calcolare la somma degli importi; (b) Visualizzare in una ListBox solo gli importi maggiori di 100 Euro e salvare l'elenco risultante in un nuovo file di nome "ImportiOltre100.txt". Creare preventivamente il file "Importi.txt" utilizzando il programma Blocco Note.
85. Chiedere in input una sequenza di Numeri Interi, memorizzandoli man mano in un file testo di nome DATI.TXT. A input completato, chiudere il file e offrire le seguenti opzioni: (a) Visualizzare in una ListBox i dati contenuti nel file; (b) Aggiungere un singolo nuovo numero al file (utilizzare il metodo AppendText); (c) Leggere tutti i dati del file e calcolarne la media aritmetica; (d) dato un numero X, verificarne l'esistenza nel file e determinarne la posizione.
86. In due files testo esistenti, di nome "Studenti.txt" e "Voti.txt" sono memorizzati rispettivamente i Nomi di un gruppo di studenti e i relativi Voti. Consentire la modifica e l'elaborazione di questi files con le seguenti opzioni: (a) aggiungere, ai files, un nuovo Nominativo di studente e il relativo Voto, impedendo l'inserimento di nomi già esistenti; (b) digitato il nome di uno studente, determinare se esiste nel file "Studenti.txt" e visualizzarne il relativo voto; (c) accedendo ai due files, determinare il nome dello studente con voto maggiore; (e) visualizzare in una Listbox Nomi e Voti (ogni riga della listbox è del tipo nome-voto) dei soli studenti con voto maggiore o uguale a 6. Creare preventivamente i files utilizzando il programma Blocco Note.

Memorizzazione su Disco – SQLite in C#

87. In un DataBase SQLite "QuartaAlf.db" è presente una tabella "Alunni" che memorizza, per ogni alunno della classe: IdAlunno (chiave primaria in autoincrement); Nominativo; MediaVoti (numero decimale). Realizzare una App Windows in linguaggio C# che, visualizzando i dati in una ListBox (un record per riga), consenta l'editing completo della tabella ossia l'inserimento di nuovi record e l'eliminazione o la modifica dei record esistenti. Creare preventivamente il DataBase con "DB-Browser for SQLite".

DataTable e DataGridView

88. In un file testo di nome "Dati.txt" sono memorizzati informazioni sugli alunni che frequentano una determinata scuola. Ad inizio file sono presenti, uno per riga, i dati del primo alunno: Cognome, Nome, Sesso (F/M), Classe, Età; seguiti dai dati del secondo alunno (sempre uno per riga), del terzo, e così via. Specificata una Classe, si carichino dal file in un DataTable tutti i dati relativi ai soli alunni appartenenti alla classe richiesta e si utilizzi un DataGridView (configurato in sola lettura), per visualizzarli. Creare preventivamente il file "Dati.txt" utilizzando il Blocco Note.
89. In un file testo di nome "Libri.txt" sono memorizzati informazioni sui libri presenti in una libreria. Ad inizio file sono presenti, uno per riga, i dati del primo libro: Titolo, Autore, CasaEditrice, AnnoDiPubblicazione, Prezzo (intero); seguiti dai dati del secondo libro (sempre uno per riga), del terzo, e così via. All'avvio del programma, si carichino dal file, in un DataTable, i dati di tutti i libri e si utilizzi un DataGridView per visualizzarli e consentirne l'editing completo. Un pulsante Salva deve consentire il salvataggio nel file "Libri.txt" dell'intero contenuto del DataTable, incluse le modifiche apportate ai dati dei libri. Creare preventivamente il file "Libri.txt" utilizzando il Blocco Note.

Personalizzazione del DataGridView

90. In un file testo di nome "ArchivioAtleti.txt" sono memorizzati informazioni sugli Atleti che partecipano ad una competizione. Più precisamente, per ogni atleta, è presente: un Numero Univoco di Iscrizione; il Nominativo; la Data di Nascita; il Peso (in Kg); l'Altezza (in metri, con 2 decimali); l'importo in euro indicante il Guadagno maturato durante l'anno. All'avvio del programma, caricare l'intero archivio in un DataTable e visualizzarlo in un DataGridView personalizzato a sola lettura, con la miglior impostazione grafica possibile (larghezze di colonna regolate correttamente, formati numerici corretti per date e numeri decimali (2 cifre decimali visibili), due colori che si alternano fra le colonne, allineamenti coerenti, disattivazione di modifica, aggiunta, eliminazione, ridimensionamento, ecc.).
91. Un file testo "Catalogo.txt" contiene un Catalogo non ordinato di Prodotti. Più precisamente, per ogni prodotto, sono presenti i seguenti dati, uno per riga: Codice del Prodotto (numero intero), Nome del Prodotto, Categoria, Prezzo (in Euro, con 2 cifre decimali), Data di Uscita sul Mercato, Larghezza (in metri, con 2 cifre decimali), Infiammabile (booleano: false o true). Il campo Categoria serve a "classificare" il prodotto (cancelleria, giocattoli, alimentari,

telefonini, ecc.). All'avvio, il programma deve rilevare dal file tutte le categorie presenti nel catalogo ed elencarle, in una ComboBox. L'utente potrà così selezionare la categoria desiderata e ottenere in un DataGridView l'elenco dei soli prodotti appartenenti alla categoria richiesta. Il DataGridView, personalizzato e "a sola lettura", deve essere configurato con la miglior impostazione grafica possibile: larghezze di colonna regolate correttamente, formati numerici corretti per date e numeri decimali (sempre 2 cifre decimali visibili), due colori che si alternano fra le colonne, allineamenti coerenti, check-box per valori booleani, disattivazione di modifica, aggiunta, eliminazione, ridimensionamento, ecc.). Consentire, inoltre, dato il Nome di un Prodotto (anche solo l'inizio del nome) di evidenziare automaticamente la riga del DataGridView contenente il primo prodotto con il nome richiesto.

File ad Accesso Diretto

92. Si desidera gestire l'Elenco degli Alunni di un istituto scolastico utilizzando un file ad accesso diretto "Alunni.dat". Per ogni alunno si mantengano i seguenti dati: Numero di Matricola (num. Intero univoco), Cognome (20 car.), Nome (20 car.), Data di Nascita, Classe (3 car. - Es.: "3Ai", "4Be", ecc.), Media dei Voti (con 1 decimale). Si consenta di: (a) aggiungere un nuovo alunno e i relativi dati alla fine del file; (b) data una posizione P, visualizzare i dati dell'alunno che, nel file, si trova in posizione P; (c) data una classe, caricare in una ListBox i nominativi di tutti gli alunni della classe specificata.
93. Si desidera memorizzare in un file ad accesso diretto la classifica di una gara di Formula 1. Per ogni "posizione" in gara, si gestiscano i seguenti dati: Nominativo del Pilota, Numero del Pilota, Nome della Scuderia. Il primo record memorizza i dati del pilota primo in classifica, il secondo record i dati del pilota secondo in classifica, ecc. Offrire le seguenti opzioni: (a) impostare la classifica iniziale (griglia di partenza) dando la possibilità di aggiungere i piloti (sempre a fine classifica) fino al completamento della griglia stessa; (b) Data una posizione, visualizzare i dati del pilota classificato nella posizione data; (c) data una posizione, scambiare il corrispondente pilota con quello precedente (sorpasso); (d) Caricare la classifica corrente con i relativi dati, in un DataTable e visualizzarla in un DataGridView a sola lettura.

Accesso Diretto e Calcolato

94. Un'azienda di logistica spedisce pacchi, misurandone il peso con un numero INTERO variabile fra 1 e 10 Kg. Un file ad accesso diretto "Pesi.dat" contiene 10 record, uno per ogni possibile peso (record 1 per il peso 1; record 2 per il peso 2, ecc.), strutturato con due soli campi: NumeroPacchiSpediti, indicante quanti pacchi di quel peso sono stati spediti; PrezzoUnitario, quanto costa spedire un pacco di quel peso. Inizialmente il programma chiede i 10 prezzi unitari relativi ai 10 possibili pesi e crea il file "Pesi.dat" strutturandolo con i 10 record contenenti i 10 prezzi unitari inseriti e il campo NumeroPacchiSpediti impostato inizialmente a 0. Offrire le seguenti opzioni: (a) Spedizione di un pacco di peso P: è necessario chiedere in input il peso P e incrementare di 1 il campo NumeroPacchiSpediti del record relativo al peso P; (b) Visualizzazione del numero di spedizioni effettuate finora e del prezzo unitario di spedizioni di peso P; (c) Calcolo di quanto si è finora incassato per tutte le spedizioni effettuate per pacchi di un dato peso P; (d) Calcolo dell'incasso complessivo per tutte le spedizioni finora effettuate.
95. Si desidera memorizzare in un file ad accesso diretto Nominativo (max 30 car.), Altezza (intero, in cm) e Peso (intero, in Kg) di un gruppo di circa 20 individui. Si determini una funzione Hash che consenta di determinare, a partire dal Nominativo, la posizione del record di un individuo, garantendo così un rapidissimo "accesso calcolato". Offrire le seguenti opzioni: (a) Inserimento nel file di un nuovo individuo (Nominativo, Altezza, Peso): i dati devono essere memorizzati nel record indicato dalla funzione Hash (in caso di "collisione", produrre un messaggio di errore); (b) Dato il Nominativo di un individuo, tramite la funzione Hash, determinare la posizione del suo record e visualizzare i suoi dati; (c) Dato un Nominativo, eliminarlo dall'archivio. Facoltativo: gestire le "collisioni" memorizzando i dati del nuovo Nominativo nel primo record "libero" fra quelli subito successivi a quello determinato dalla funzione Hash.
96. Data una stringa contenente una frase, si desidera memorizzare in un file ad accesso diretto "Lettere.dat" il numero di volte che, nella frase, è presente ciascuna lettera dell'alfabeto. Ad ogni lettera dell'alfabeto è associato un record del file contenente due soli campi NumeroMaiuscole e NumeroMinuscole (inizialmente posti a 0) in cui calcolare quante volte appare la lettera stessa in maiuscolo e minuscolo. Il programma, una volta acquisita la stringa, deve creare "Lettere.dat" e, scorrendo la stringa, incrementare progressivamente i campi NumeroMaiuscole e NumeroMinuscole delle varie lettere che incontra nella stringa. Al termine dell'elaborazione, offrire le opzioni: (a) data una lettera, visualizzare il relativo conteggio di maiuscole, minuscole e totale; (b) visualizzare una tabella che su ogni riga riporti: lettera; num.maiuscole; num.minuscole; totale; percentuale rispetto al totale delle lettere presenti nella stringa. Si ignorino tutti i caratteri della stringa che non sono lettere.

Programmazione Web - File Testo in PHP

97. Realizzare un Sito Web Dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" e che, digitato il Nominativo di un Calciatore, ne visualizzi la Squadra di appartenenza. I dati dei Giocatori sono memorizzati in un File Testo "Giocatori.txt" (squadra su una riga, nominativo sulla successiva, ecc.), situato nella cartella in cui il sito è pubblicato sul server. Consentire, inoltre, di aggiungere un nuovo giocatore al File Testo, specificandone Nominativo e Squadra.
98. Realizzare un Sito Web Dinamico, in linguaggio PHP, costituito da una singola pagina "index.php" che, scelta una Squadra di Calcio del Campionato di Serie A (selezionandola da una apposita Casella a Discesa), visualizzi in tag <select> l'elenco dell'intera formazione della Squadra, indicando su ogni singola riga il Numero di Maglia e il Nominativo del Giocatore. Si offra anche la possibilità di "aggiungere" un nuovo Giocatore alla squadra selezionata, inserendone Nominativo e Numero. Utilizzare un unico File Testo "Giocatori.txt" che, per ogni Giocatore contenga Cognome, Nome, Numero, Squadra (un singolo dato per riga).
99. Realizzare un Sito Web Dinamico, in linguaggio PHP, che consenta ad utenti remoti di visionare e aggiornare liberamente un Elenco di Frasi "Simpatiche". Tale elenco, affinché "sopravviva" durante e dopo la navigazione, deve essere gestito sul server in un file testo "ElencoFrase.txt" situato nella cartella in cui il sito è pubblicato. Nella sua Home Page, il sito legge dal file l'Elenco delle Frasi e lo visualizza in un tag <select>, consentendo di: (a) aggiungere una nuova Frase al file testo; (b) selezionata una frase sulla ListBox, eliminarla dal file testo; (c) se viene specificata una Parola, visualizzare solo le frasi contenenti quella parola.