I.T.I.S. Francesco Giordani

A.S. : 2014/2015

Classe: IV D inf

Raccolta degli esercizi corretti in classe a cura di Sebastiano Fusco

Esercizio 1:

Ricerca lineare che in caso di ricerca senza successo restiusce -1 oppure la prima posizione dell'elemento cercato.

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 15/09/2014

Numero es:1

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la ricerca lineare in c++;

Obiettivo del programma: Ricerca lineare che in caso di ricerca senza successo restiusce -1 oppure la prima posizione dell'elemento cercato.

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| VET | I/O |  | INTERO | 20 BYTE |
| DIMLOG | INPUT |  | INTERO | 2 BYTE |
| K | INPUT | Chiave di ricerca | INTERO | 2 BYTE |
| I | LAVORO |  | INTERO | 2 BYTE |
| DIM | LAVORO |  | INTERO | 2 BYTE |
| TROVATO | LAVORO |  | INTERO | 2 BYTE |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

int vet[10],dimlog,k;

//Prototipi

void CaricaVet(int vetpar[10],int dimlogpar);

void StampaVet(int vetpar[10],int dimlogpar);

int OttieniDim();

int RicercaLineare(int vetpar[10],int dimlogpar,int kpar);

int main()

{

 cout<<"Quanti elementi vuoi inserire? <max 10> ";

 dimlog=OttieniDim();

 CaricaVet(vet,dimlog);

 StampaVet(vet,dimlog);

 cout<<endl;

 cout<<"Quale elemento vuoi cercare? ";

 cin>>k;

 if(RicercaLineare(vet,dimlog,k)==-1)

 cout<<"Ricerca senza successo! "<<endl;

 else

 cout<<"L'elemento si trova nella posizione: "<<RicercaLineare(vet,dimlog,k)+1<<endl;

 system("pause");

}

int OttieniDim()

{

 int dim;

 do

 {

 cin>>dim;

 if(!(dim>0 && dim<=10)) cout<<"Hai sbagliato! Riprova! ";

 }

 while(!(dim>0 && dim<=10));

 return dim;

}

void CaricaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++)

 {

 cout<<"Inserire un elemento: ";

 cin>>vetpar[i];

 }

}

void StampaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++) cout<<vetpar[i]<<" ";

}

int RicercaLineare(int vetpar[10],int dimlogpar,int kpar)

{

 int i=0,trovato=-1;

 while(i<dimlogpar && trovato==-1)

 {

 if(vetpar[i]==kpar) trovato=i;

 else i++;

 }

 return trovato;

}

Esercizio 2:

Dati due valori, scambiarli.

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 17/09/2014

Numero es:2

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L’alunno è in grado di effettuare lo scambio;

Obiettivo del programma: Dati due valori, scambiarli.

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| X | INPUT |  | INTERO | 2 BYTE |
| Y | INPUT |  | INTERO | 2 BYTE |
| APP | LAVORO |  | INTERO | 2 BYTE |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

int x,y;

//Prototipi

void Scambia(int & xpar,int & ypar);

int main()

{

 cout<<"Inserire il primo numero: ";

 cin>>x;

 cout<<"Inserire il secondo numero: ";

 cin>>y;

 Scambia(x,y);

 cout<<"I valori scambiati sono: "<<x<<" e "<<y<<endl;

 system("pause");

}

void Scambia(int & xpar,int & ypar)

{

 int app=xpar;

 xpar=ypar;

 ypar=app;

}

Esercizio 3:

Ricerca lineare in un insieme ordinato che in caso di ricerca senza successo restiusce -1 oppure la posizione dell'elemento cercato.

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 23/09/2014

Numero es:3

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la ricerca lineare in un insieme ordinato in c++;

Obiettivo del programma: Funzione ricerca lineare in un insieme ordinato che in caso di ricerca senza successo restiusce -1 oppure la posizione dell'elemento cercato.

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| VET | I/O |  | INTERO | 20 BYTE |
| DIMLOG | INPUT |  | INTERO | 2 BYTE |
| K | INPUT | Chiave di ricerca | INTERO | 2 BYTE |
| I | LAVORO |  | INTERO | 2 BYTE |
| DIM | LAVORO |  | INTERO | 2 BYTE |
| TROVATO | LAVORO |  | INTERO | 2 BYTE |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

int vet[10],dimlog,k;

//Prototipi

void CaricaVet(int vetpar[10],int dimlogpar);

void StampaVet(int vetpar[10],int dimlogpar);

int OttieniDim();

int RicercaLineareOrdinato(int vetpar[10],int dimlogpar,int kpar);

void OrdinamentoPerSelezione(int vetpar[10],int dimlogpar);

int main()

{

 cout<<"Quanti elementi vuoi inserire? <max 10> ";

 dimlog=OttieniDim();

 CaricaVet(vet,dimlog);

 cout<<"Vettore prima dell'ordinamento: "<<endl;

 StampaVet(vet,dimlog);

 cout<<endl;

 OrdinamentoPerSelezione(vet,dimlog);

 cout<<"Vettore dopo l'ordinamento: "<<endl;

 StampaVet(vet,dimlog);

 cout<<endl;

 cout<<"Inserire l'elemento da cercare: ";

 cin>>k;

 if(RicercaLineareOrdinato(vet,dimlog,k)==-1)

 cout<<"Ricerca senza successo! "<<endl;

 else

 cout<<"L'elemento e' stato trovato nella posizione: "<<RicercaLineareOrdinato(vet,dimlog,k)<<endl;

 system("pause");

}

int OttieniDim()

{

 int dim;

 do

 {

 cin>>dim;

 if(!(dim>0 && dim<=10)) cout<<"Hai sbagliato! Riprova! ";

 }

 while(!(dim>0 && dim<=10));

 return dim;

}

void CaricaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++)

 {

 cout<<"Inserire il "<<i+1<<"^ elemento: ";

 cin>>vetpar[i];

 }

}

void StampaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++) cout<<vetpar[i]<<" ";

}

void OrdinamentoPerSelezione(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar-1;i++)

 {

 for(int j=i+1;j<dimlogpar;j++)

 {

 if(vetpar[i]>vetpar[j])

 {

 int app=vetpar[i];

 vetpar[i]=vetpar[j];

 vetpar[j]=app;

 }

 }

 }

}

int RicercaLineareOrdinato(int vetpar[10],int dimlogpar,int kpar)

{

 int i=0,trovato=-1;

 while(i<dimlogpar && kpar>=vetpar[i])

 {

 i++;

 }

 if(vetpar[i-1]==kpar) trovato=i;

 return trovato;

}

Esercizio 4:

Ricerca binaria.

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 26/09/2014

Numero es:4

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L’alunno è in grado di utilizzare la ricerca binaria;

Obiettivo del programma: Ricerca binaria

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| VET | I/O |  | INTERO | 20 BYTE |
| DIMLOG | INPUT |  | INTERO | 2 BYTE |
| K | INPUT | Chiave di ricerca | INTERO | 2 BYTE |
| INF | LAVORO |  | INTERO | 2 BYTE |
| SUP | LAVORO |  | INTERO | 2 BYTE |
| CENTRO | LAVORO |  | INTERO | 2 BYTE |
| TROVATO | LAVORO |  | INTERO | 2 BYTE |
| I | LAVORO |  | INTERO | 2 BYTE |
| J | LAVORO |  | INTERO | 2 BYTE |
| APP | LAVORO |  | INTERO | 2 BYTE |

\*/

#include<iostream>

using namespace std;

int vet[10],dimlog,k;

//Prototipi

void CaricaVet(int vetpar[10], int dimlogpar);

void OrdinamentoIngenuo(int vetpar[10], int dimlogpar);

int RicercaBinaria(int vetpar[10],int dimlogpar,int k);

int main()

{

 cout<<"Quanti elementi vuoi inserire? ";

 cin>>dimlog;

 CaricaVet(vet,dimlog);

 OrdinamentoIngenuo(vet,dimlog);

 cout<<"Quale elemento vuoi cercare? ";

 cin>>k;

 if(RicercaBinaria(vet,dimlog,k)==1)

 cout<<"Elemento trovato! "<<endl;

 else

 cout<<"Elemento non trovato!"<<endl;

 system("pause");

}

void CaricaVet(int vetpar[10], int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++)

 {

 cout<<"Inserire un numero: ";

 cin>>vetpar[i];

 }

}

void OrdinamentoIngenuo(int vetpar[10], int dimlogpar)

{

 for(int i=0;i<dimlogpar-1;i++)

 {

 for(int j=i+1;j<dimlogpar;j++)

 {

 if(vetpar[i]>vetpar[j])

 {

 int app=vetpar[i];

 vetpar[i]=vetpar[j];

 vetpar[j]=app;

 }

 }

 }

}

int RicercaBinaria(int vetpar[10],int dimlogpar,int k)

{

 int inf=0,sup=dimlogpar-1,centro,trovato=0;

 while(inf<=sup && trovato==0)

 {

 centro=int((inf+sup)/2);

 if(vetpar[centro]==k)

 trovato=1;

 else

 {

 if(vetpar[centro]<k)

 inf=centro+1;

 else

 sup=centro-1;

 }

 }

 return(trovato);

}

Esercizio 5:

Caricare e leggere un file di caratteri

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 29/09/2014

Numero es:5

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di caricare e leggere un file;

Obiettivo del programma: Caricare e leggere un file di caratteri;

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| caratteriFile | I/O |  | FILE | 1 BYTE |
| car | I/O |  | CARATTERE | 1 BYTE |
| risp | INPUT |  | CARATTERE | 1 BYTE |

\*/

#include<iostream>

#include<fstream>

using namespace std;

fstream caratteriFile; //dichiarazione variabile di tipo file

char car;

//Prototipi

void CaricaFile();

void LeggiFile();

int main()

{

 CaricaFile();

 LeggiFile();

 system("pause");

}

void CaricaFile()

{

 char risp;

 caratteriFile.open ("caratteri.dat", ios::out | ios::binary); //apertura in scrittura

 do

 {

 cout<<"Inserire un carattere: ";

 cin>>car;

 caratteriFile.write((char \*) &car, sizeof(car)); //scrittura su file

 cout<<"Vuoi inserire altri caratteri? (s/n) ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 caratteriFile.close();

}

void LeggiFile()

{

 caratteriFile.open ("caratteri.dat", ios::in | ios::binary); //apertura in lettura

 if(!caratteriFile) cout<<"Errore di apertura del file! ";

 else

 {

 cout<<"I caratteri sono: "<<endl;

 caratteriFile.read((char \*) &car, sizeof(car)); //lettura da file

 while(!caratteriFile.eof())

 {

 cout<<car<<endl;

 caratteriFile.read((char \*) &car, sizeof(car));

 }

 }

 caratteriFile.close();

}

Esercizio 6:

Caricare e leggere un file di interi

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 29/09/2014

Numero es:6

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di caricare e leggere un file;

Obiettivo del programma: Caricare e leggere un file di interi;

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| intFile | I/O |  | FILE | 1 BYTE |
| num | I/O |  | INTERO | 1 BYTE |
| risp | INPUT |  | CARATTERE | 1 BYTE |

\*/

#include<iostream>

#include<fstream>

using namespace std;

fstream intFile; //dichiarazione variabile di tipo file

int num;

//Prototipi

void CaricaFile();

void LeggiFile();

int main()

{

 CaricaFile();

 LeggiFile();

 system("pause");

}

void CaricaFile()

{

 char risp;

 intFile.open ("num.dat", ios::out | ios::binary); //apertura in scrittura

 do

 {

 cout<<"Inserire un numero: ";

 cin>>num;

 intFile.write((char \*) &num, sizeof(num)); //scrittura su file

 cout<<"Vuoi inserire altri numeri? (s/n) ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 intFile.close();

}

void LeggiFile()

{

 intFile.open ("num.dat", ios::in | ios::binary); //apertura in lettura

 if(!intFile) cout<<"Errore di apertura del file! ";

 else

 {

 cout<<"I numeri sono: "<<endl;

 intFile.read((char \*) &num, sizeof(num));

 while(!intFile.eof())

 {

 cout<<num<<endl;

 intFile.read((char \*) &num, sizeof(num)); //lettura da file

 }

 }

 intFile.close();

}

Esercizio 7:

Caricare,leggere ed effuttare la somma di un file di interi

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 29/09/2014

Numero es:7

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di caricare e leggere un file;

Obiettivo del programma: Caricare,leggere ed effuttare la somma di un file di interi;

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| intFile | I/O |  | FILE | 1 BYTE |
| num | I/O |  | INTERO | 1 BYTE |
| risp | INPUT |  | CARATTERE | 1 BYTE |
| somma | LAVORO |  | INTERO | 2 BYTE |

\*/

#include<iostream>

#include<fstream>

using namespace std;

//Prototipi

void CaricaFile();

void LeggiFile();

int SommaFile();

int main()

{

 CaricaFile();

 LeggiFile();

 cout<<"La somma dei numeri del file e': "<<SommaFile()<<endl;

 system("pause");

}

void CaricaFile()

{

 fstream intFile;

 int num;

 char risp;

 intFile.open ("somma.dat", ios::out | ios::binary); //apertura in scrittura

 do

 {

 cout<<"Inserire un numero: ";

 cin>>num;

 intFile.write((char \*) &num, sizeof(num)); //scrittura su file

 cout<<"Vuoi inserire altri numeri? (s/n) ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 intFile.close();

}

void LeggiFile()

{

 fstream intFile;

 int num;

 intFile.open ("somma.dat", ios::in | ios::binary); //apertura in lettura

 if(!intFile) cout<<"Errore di apertura del file! ";

 else

 {

 cout<<"I numeri sono: "<<endl;

 intFile.read((char \*) &num, sizeof(num));

 while(!intFile.eof())

 {

 cout<<num<<endl;

 intFile.read((char \*) &num, sizeof(num)); //lettura da file

 }

 }

 intFile.close();

}

int SommaFile()

{

 fstream intFile;

 int num,somma=0;

 intFile.open ("somma.dat", ios::in | ios::binary);

 if(!intFile) cout<<"Errore di apertura del file! ";

 else

 {

 intFile.read((char \*) &num, sizeof(num));

 while(!intFile.eof())

 {

 somma=somma+num;

 intFile.read((char \*) &num, sizeof(num));

 }

 }

 return somma;

 intFile.close();

}

Esercizio 8:

Caricare e leggere un file di caratteri contando il numero di questi

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 30/09/2014

Numero es:8

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di caricare e leggere un file;

Obiettivo del programma: Caricare e leggere un file di caratteri contando il numero di questi;

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| caratteriFile | I/O |  | FILE | 1 BYTE |
| car | I/O |  | CARATTERE | 1 BYTE |
| risp | INPUT |  | CARATTERE | 1 BYTE |
| cont | LAVORO |  | INTERO | 2 BYTE |

\*/

#include<iostream>

#include<fstream>

using namespace std;

//Prototipi

void CaricaFile();

void LeggiFile();

int ContaCarFile();

int main()

{

 CaricaFile();

 LeggiFile();

 cout<<"Il numero di caratteri e': "<<ContaCarFile()<<endl;

 system("pause");

}

void CaricaFile()

{

 fstream caratteriFile;

 char car,risp;

 caratteriFile.open ("contacar.dat", ios::out | ios::binary);

 do

 {

 cout<<"Inserire un carattere: ";

 cin>>car;

 caratteriFile.write((char \*) &car, sizeof(car));

 cout<<"Vuoi inserire altri caratteri? (s/n) ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 caratteriFile.close();

}

void LeggiFile()

{

 fstream caratteriFile;

 char car;

 caratteriFile.open ("contacar.dat", ios::in | ios::binary);

 if(!caratteriFile) cout<<"Errore di apertura del file! ";

 else

 {

 cout<<"I caratteri sono: "<<endl;

 caratteriFile.read((char \*) &car, sizeof(car));

 while(!caratteriFile.eof())

 {

 cout<<car<<endl;

 caratteriFile.read((char \*) &car, sizeof(car));

 }

 }

 caratteriFile.close();

}

int ContaCarFile()

{

 fstream caratteriFile;

 char car;

 int cont=0;

 caratteriFile.open("contacar.dat", ios::in | ios::binary);

 if(!caratteriFile) cout<<"Errore di apertura del file! ";

 else

 {

 caratteriFile.read((char \*) &car, sizeof(car));

 while(!caratteriFile.eof())

 {

 cont++;

 caratteriFile.read((char \*) &car, sizeof(car));

 }

 }

 return cont;

 caratteriFile.close();

}

Esercizio 9:

Caricare e leggere un file di records

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 02/10/2014

Numero es:9

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di caricare e leggere un file di records;

Obiettivo del programma: Caricare e leggere un file di records;

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| articolifile | I/O |  | FILE |  |
| articolirec | I/O |  | ARTICOLOTYPE |  |
| risp | INPUT |  | CARATTERE | 1 BYTE |

\*/

#include<iostream>

#include<fstream>

using namespace std;

struct articolitype

 {

 char cod[15];

 char desc[15];

 float prezzo;

 };

//Prototipi

void CaricaFile();

void LeggiFile();

int main()

{

 CaricaFile();

 LeggiFile();

 system("pause");

}

void CaricaFile()

{

 fstream articolifile;

 char risp;

 articolitype articolirec;

 articolifile.open("articoli.dat", ios::out | ios::binary);

 do

 {

 cout<<"Inserire il codice del prodotto: ";

 cin>>articolirec.cod;

 cout<<"Inserire la descrizione del prodotto: ";

 cin>>articolirec.desc;

 cout<<"Inserire il prezzo del prodotto: ";

 cin>>articolirec.prezzo;

 articolifile.write((char \*) &articolirec, sizeof(articolirec));

 cout<<"Vuoi continuare? s/n ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 articolifile.close();

}

void LeggiFile()

{

 fstream articolifile;

 articolitype articolirec;

 articolifile.open("articoli.dat", ios::in | ios::binary);

 if(!articolifile) cout<<"Errore di apertura del file! ";

 else

 {

 cout<<"Gli articoli presenti nel file sono: "<<endl;

 cout<<"Codice \t Descrizione \t Prezzo"<<endl;

 articolifile.read((char \*) &articolirec, sizeof(articolirec));

 while(!articolifile.eof())

 {

 cout<<articolirec.cod<<"\t"<<articolirec.desc<<"\t"<<articolirec.prezzo<<endl;

 articolifile.read((char \*) &articolirec, sizeof(articolirec));

 }

 }

 articolifile.close();

}

Esercizio 10:

Esempi sulle stringhe in c++

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 10/10/2014

Numero es:10

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare le stringhe in c++;

Obiettivo del programma: Esempi sulle stringhe;

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| s1 | OUTPUT |  | STRINGA |  |
| s2 | I/O |  | STRINGA |  |
| nome | INPUT |  | CARATTERE | 16 BYTE |

\*/

#include<iostream>

#include<string.h>

using namespace std;

int main()

{

 string s1,s2;

 char nome[15];

 cout<<"Inserire nome: ";

 cin>>nome;

 cout<<"Inserire la seconda stringa: ";

 cin>>s2;

 s1=nome;

 cout<<s1<<endl;

 cout<<s2<<endl;

 if(s1==s2)

 cout<<"Le stringhe sono uguali! "<<endl;

 else

 cout<<"Le stringhe non sono uguali"<<endl;

 system("pause");

}

Esercizio 11:

Caricare e leggere un file di records, sommare i prezzi degli articoli presenti nel file e contare quelli che hanno la descrizione uguale a quella inserita dall'utente

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 10/10/2014

Numero es:11

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di caricare e leggere un file di records;

Obiettivo del programma: Caricare e leggere un file di records, sommare i prezzi degli articoli presenti nel file e contare quelli che hanno la descrizione uguale a quella inserita dall'utente;

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| articolifile | I/O |  | FILE |  |
| articolirec | I/O |  | ARTICOLOTYPE |  |
| risp | INPUT |  | CARATTERE | 1 BYTE |
| kdesc | INPUT |  | CARATTERE | 1 BYTE |
| somma | LAVORO |  | REALE | 4 BYTE |
| cont | LAVORO |  | INTERO | 2 BYTE |

\*/

#include<iostream>

#include<fstream>

using namespace std;

struct articolitype

 {

 char cod[15];

 char desc[15];

 float prezzo;

 };

char kdesc[15];

//Prototipi

void CaricaFile();

void LeggiFile();

float SommaPrezzi();

int ContaDescArticoli(char kdescpar[15]);

int main()

{

 CaricaFile();

 LeggiFile();

 cout<<"La somma dei prezzi e': "<<SommaPrezzi()<<endl;

 cout<<"Inserire la descrizione da ricercare: ";

 cin>>kdesc;

 cout<<"Gli articoli con la descrizione '"<<kdesc<<"' sono: "<<ContaDescArticoli(kdesc)<<endl;

 system("pause");

}

void CaricaFile()

{

 fstream articolifile;

 char risp;

 articolitype articolirec;

 articolifile.open("articoli.dat", ios::out | ios::binary);

 do

 {

 cout<<"Inserire il codice del prodotto: ";

 cin>>articolirec.cod;

 cout<<"Inserire la descrizione del prodotto: ";

 cin>>articolirec.desc;

 cout<<"Inserire il prezzo del prodotto: ";

 cin>>articolirec.prezzo;

 articolifile.write((char \*) &articolirec, sizeof(articolirec));

 cout<<"Vuoi continuare? s/n ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 articolifile.close();

}

void LeggiFile()

{

 fstream articolifile;

 articolitype articolirec;

 articolifile.open("articoli.dat", ios::in | ios::binary);

 if(!articolifile) cout<<"Errore di apertura del file! ";

 else

 {

 cout<<"Gli articoli presenti nel file sono: "<<endl;

 cout<<"Codice \t Descrizione \t Prezzo"<<endl;

 articolifile.read((char \*) &articolirec, sizeof(articolirec));

 while(!articolifile.eof())

 {

 cout<<articolirec.cod<<"\t"<<articolirec.desc<<"\t"<<articolirec.prezzo<<endl;

 articolifile.read((char \*) &articolirec, sizeof(articolirec));

 }

 }

 articolifile.close();

}

float SommaPrezzi()

{

 fstream articolifile;

 articolitype articolirec;

 float somma=0;

 articolifile.open("articoli.dat", ios::in | ios::binary);

 if(!articolifile) cout<<"Errore di apertura del file! ";

 else

 {

 articolifile.read((char \*) &articolirec, sizeof(articolirec));

 while(!articolifile.eof())

 {

 somma=somma+articolirec.prezzo;

 articolifile.read((char \*) &articolirec, sizeof(articolirec));

 }

 }

 articolifile.close();

 return somma;

}

int ContaDescArticoli(char kdescpar[15])

{

 fstream articolifile;

 articolitype articolirec;

 int cont=0;

 //string s1,s2;

 articolifile.open("articoli.dat", ios::in | ios::binary);

 if(!articolifile) cout<<"Errore di apertura del file! ";

 else

 {

 articolifile.read((char \*) &articolirec, sizeof(articolirec));

 while(!articolifile.eof())

 {

 /\*

 s1=articolirec.desc;

 s2=kdescpar;

 if(s1==s2) cont++;

 \*/

 if(strcmp(articolirec.desc,kdescpar)==0) cont++;

 articolifile.read((char \*) &articolirec, sizeof(articolirec));

 }

 }

 articolifile.close();

 return cont;

}

Esercizio 12:

Scrivere la funzione HASH che trasforma la sigla internazionale automobilistica in un indirizzo. La funzione riceve in ingresso la chiave alfanumerica e restituisce il codice HASH;

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 13/10/2014

Numero es:12

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la funzione HASH in c++;

Obiettivo del programma: Scrivere la funzione HASH che trasforma la sigla internazionale automobilistica in un indirizzo. La funzione riceve in ingresso la chiave alfanumerica e restituisce il codice HASH;

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| k | LAVORO | VETTORE | CARATTERE | 4 BYTE |
| K | LAVORO | VETTORE | CARATTERE | 4 BYTE |
| Risp | INPUT |  | CARATTERE | 1 BYTE |
| pos1 | LAVORO |  | INTERO | 4 BYTE |
| pos2 | LAVORO |  | INTERO | 4 BYTE |
| pos3 | LAVORO |  | INTERO | 4 BYTE |
| indirizzo | LAVORO |  | INTERO  | 4 BYTE |
| ris | LAVORO |  | INTERO | 4 BYTE |

\*/

#include<iostream>

using namespace std;

char k[3];

char K[3];

char risp;

//Prototipi

int Hash(char Kpar[3]);

int main()

{

 do

 {

 cout<<"Quale sigla internazionale automobilistica vuoi trasformare in indirizzo? ";

 cin>>k;

 for(int i=0;i<3;i++)

 K[i]=toupper(k[i]); //Trasformazione di ciascun carattere della sigla in maiuscolo

 cout<<"La chiave presa in considerazione e': "<<K<<endl;

 cout<<"L'indirizzo generato dalla chiave "<<K<<" e': "<<Hash(K)<<endl;

 cout<<endl;

 cout<<"Vuoi inserire un'altra sigla automobilistica da trasformare in indirizzo? s/n ";

 cin>>risp;

 cout<<endl;

 }

 while(risp=='s' || risp=='S');

 system("pause");

}

int Hash(char Kpar[3])

{

 int pos1,pos2,pos3,indirizzo,ris;

 if(Kpar[0]==0)

 pos1=0;

 else

 {

 pos1=int(Kpar[0]); //casting esplicito che assegna a app il codice ASCII del carattere

 pos1=pos1-64; //trasforma il codice ASCII nella posizione del carattere nell'alfabeto

 }

 if(Kpar[1]==0)

 pos2=0;

 else

 {

 pos2=int(Kpar[1]);

 pos2=pos2-64;

 }

 if(Kpar[2]==0)

 pos3=0;

 else

 {

 pos3=int(Kpar[2]);

 pos3=pos3-64;

 }

 indirizzo=(pos1\*10000)+(pos2\*100)+(pos3\*1);

 ris=indirizzo-19\*(int(indirizzo/19));

 return ris;

}

Esercizio 13:

Creare un file di records con il seguente formato: codArt,descrArt,prezzoArt. Ricercare un articolo inserito da tastiera mediante il suo codice e restituire la sua posizione nel caso di ricerca con successo altrimenti -1. Dato il codice modificare il prezzo.

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 21/10/2014

Numero es:13

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di effettuare la ricerca e la modifica in un file di records;

Obiettivo del programma: Creare un file di records con il seguente formato: codArt,descrArt,prezzoArt. Ricercare un articolo inserito da tastiera mediante il suo codice e restituire la sua posizione nel caso di ricerca con successo altrimenti -1. Dato il codice modificare il prezzo.

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| ArticoliFile | I/O |  | File |  |
| ArticoliRec | I/O |  | ArticoliType | 3 BYTE |
| KcodArt | INPUT | Cod. da trovare | Carattere | 1 BYTE |
| NuovoPrezzo | INPUT |  | Carattere | 1 BYTE |
| risp | INPUT |  | Carattere | 1 BYTE |
| trovato | LAVORO |  | Intero | 4 BYTE |
| i | LAVORO |  | Intero | 4 BYTE |

\*/

#include<iostream>

#include<fstream>

#include<iomanip>

using namespace std;

struct ArticoliType

 {

 char CodArt;

 char DescArt;

 char PrezzoArt;

 };

char KCodArt;

//Prototipi

void CaricaFile();

void LeggiFile();

int RicercaLineare(char KCodArtPar);

void ModificaPrezzo(int PosizionePar);

int main()

{

 CaricaFile();

 cout<<endl;

 LeggiFile();

 cout<<endl;

 cout<<"Inserire il codice dell'articolo da ricercare: ";

 cin>>KCodArt;

 int Posizione=RicercaLineare(KCodArt);

 ModificaPrezzo(Posizione);

 cout<<endl;

 cout<<"Dopo la modifica ";

 LeggiFile();

 cout<<endl;

 system("pause");

}

void CaricaFile()

{

 fstream ArticoliFile;

 ArticoliType ArticoliRec;

 char risp;

 ArticoliFile.open("articoli.dat", ios::out | ios::binary);

 do

 {

 cout<<"Inserire il codice dell'articolo: ";

 cin>>ArticoliRec.CodArt;

 cout<<"Inserire la descrizione dell'articolo: ";

 cin>>ArticoliRec.DescArt;

 cout<<"Inserire il prezzo dell'articolo: ";

 cin>>ArticoliRec.PrezzoArt;

 ArticoliFile.write((char \*) &ArticoliRec, sizeof(ArticoliRec));

 cout<<"Vuoi inserire un altro articolo? s/n ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 ArticoliFile.close();

}

void LeggiFile()

{

 fstream ArticoliFile;

 ArticoliType ArticoliRec;

 ArticoliFile.open("articoli.dat", ios::in | ios::binary);

 if(!ArticoliFile) cout<<"Errore di apertura del file! ";

 else

 {

 cout<<"Gli articoli presenti nel file sono: "<<endl;

 cout<<setw(16)<<"Codice"<<setw(16)<<"Descrizione"<<setw(10)<<"Prezzo"<<endl;

 ArticoliFile.read((char \*) &ArticoliRec, sizeof(ArticoliRec));

 while(!ArticoliFile.eof())

 {

 cout<<setw(16)<<ArticoliRec.CodArt<<setw(16)<<ArticoliRec.DescArt<<setw(10)<<ArticoliRec.PrezzoArt<<endl;

 ArticoliFile.read((char \*) &ArticoliRec, sizeof(ArticoliRec));

 }

 }

 ArticoliFile.close();

}

int RicercaLineare(char KCodArtPar)

{

 int trovato=-1,i=0;

 fstream ArticoliFile;

 ArticoliType ArticoliRec;

 ArticoliFile.open("articoli.dat", ios::in | ios::binary);

 if(!ArticoliFile) cout<<"Errore di apertura nel file! ";

 else

 {

 ArticoliFile.read((char \*) &ArticoliRec, sizeof(ArticoliRec));

 while(!ArticoliFile.eof() && trovato==-1)

 {

 i++;

 if(ArticoliRec.CodArt==KCodArtPar) trovato=i;

 ArticoliFile.read((char \*) &ArticoliRec, sizeof(ArticoliRec));

 }

 }

 ArticoliFile.close();

 return trovato;

}

void ModificaPrezzo(int PosizionePar)

{

 fstream ArticoliFile;

 ArticoliType ArticoliRec;

 if(PosizionePar==-1)

 cout<<"Articolo non trovato! ";

 else

 {

 ArticoliFile.open("articoli.dat", ios::in | ios::binary);

 if(!ArticoliFile) cout<<"Errore di apertura nel file! ";

 else

 {

 for(int i=0;i<PosizionePar-1;i++)

 ArticoliFile.read((char \*) &ArticoliRec, sizeof(ArticoliRec));

 }

 }

 ArticoliFile.close();

 //fase di sovrascrittura

 ArticoliFile.open("articoli.dat", ios::in | ios::out | ios::binary);

 if(!ArticoliFile) cout<<"Errore di apertura nel file! ";

 else

 {

 cout<<"Qual e' il nuovo prezzo dell'articolo inserito? ";

 cin>>ArticoliRec.PrezzoArt;

 ArticoliFile.write((char \*) &ArticoliRec, sizeof(ArticoliRec));

 }

 ArticoliFile.close();

}

Esercizio 14:

Creare un file di records con il seguente formato: codArt,descrArt,prezzoArt. Ricercare un articolo inserito da tastiera mediante il suo codice e restituire la sua posizione nel caso di ricerca con successo altrimenti -1. Dato il codice modificare il prezzo. Utilizzare "seekg".

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 23/10/2014

Numero es:14

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di effettuare la ricerca e la modifica in un file di records utilizzando "seekg";

Obiettivo del programma: Creare un file di records con il seguente formato: codArt,descrArt,prezzoArt. Ricercare un articolo inserito da tastiera mediante il suo codice e restituire la sua posizione nel caso di ricerca con successo altrimenti -1. Dato il codice modificare il prezzo.

Utilizzare "seekg".

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| ArticoliFile | I/O |  | File |  |
| ArticoliRec | I/O |  | ArticoliType | 3 BYTE |
| KcodArt | INPUT | Cod. da trovare | Carattere | 1 BYTE |
| NuovoPrezzo | INPUT |  | Carattere | 1 BYTE |
| risp | INPUT |  | Carattere | 1 BYTE |
| trovato | LAVORO |  | Intero | 4 BYTE |
| i | LAVORO |  | Intero | 4 BYTE |

\*/

#include<iostream>

#include<fstream>

#include<iomanip>

using namespace std;

struct ArticoliType

 {

 char CodArt[15];

 char DescArt[15];

 float PrezzoArt;

 };

char KCodArt[15];

float NuovoPrezzo;

//Prototipi

void CaricaFile();

void LeggiFile();

int RicercaLineare(char KCodArtPar[15]);

void ModificaPrezzo(char KCodArtPar[15], float NuovoPrezzoPar);

int main()

{

 CaricaFile();

 cout<<endl;

 LeggiFile();

 cout<<endl;

 cout<<"Inserire il codice dell'articolo da ricercare: ";

 cin>>KCodArt;

 cout<<"Qual e' il nuovo prezzo dell'articolo inserito? ";

 cin>>NuovoPrezzo;

 ModificaPrezzo(KCodArt,NuovoPrezzo);

 cout<<endl;

 cout<<"Dopo la modifica ";

 LeggiFile();

 cout<<endl;

 system("pause");

}

void CaricaFile()

{

 fstream ArticoliFile;

 ArticoliType ArticoliRec;

 char risp;

 ArticoliFile.open("articoli.dat", ios::out | ios::binary);

 do

 {

 cout<<"Inserire il codice dell'articolo: ";

 cin>>ArticoliRec.CodArt;

 cout<<"Inserire la descrizione dell'articolo: ";

 cin>>ArticoliRec.DescArt;

 cout<<"Inserire il prezzo dell'articolo: ";

 cin>>ArticoliRec.PrezzoArt;

 ArticoliFile.write((char \*) &ArticoliRec, sizeof(ArticoliRec));

 cout<<"Vuoi inserire un altro articolo? s/n ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 ArticoliFile.close();

}

void LeggiFile()

{

 fstream ArticoliFile;

 ArticoliType ArticoliRec;

 ArticoliFile.open("articoli.dat", ios::in | ios::binary);

 if(!ArticoliFile) cout<<"Errore di apertura del file! ";

 else

 {

 cout<<"Gli articoli presenti nel file sono: "<<endl;

 cout<<setw(16)<<"Codice"<<setw(16)<<"Descrizione"<<setw(10)<<"Prezzo"<<endl;

 ArticoliFile.read((char \*) &ArticoliRec, sizeof(ArticoliRec));

 while(!ArticoliFile.eof())

 {

 cout<<setw(16)<<ArticoliRec.CodArt<<setw(16)<<ArticoliRec.DescArt<<setw(10)<<ArticoliRec.PrezzoArt<<endl;

 ArticoliFile.read((char \*) &ArticoliRec, sizeof(ArticoliRec));

 }

 }

 ArticoliFile.close();

}

int RicercaLineare(char KCodArtPar[15])

{

 int trovato=-1,i=0;

 fstream ArticoliFile;

 ArticoliType ArticoliRec;

 ArticoliFile.open("articoli.dat", ios::in | ios::binary);

 if(!ArticoliFile) cout<<"Errore di apertura nel file! ";

 else

 {

 ArticoliFile.read((char \*) &ArticoliRec, sizeof(ArticoliRec));

 while(!ArticoliFile.eof() && trovato==-1)

 {

 i++;

 if(strcmp(ArticoliRec.CodArt,KCodArtPar)==0) trovato=i;

 ArticoliFile.read((char \*) &ArticoliRec, sizeof(ArticoliRec));

 }

 }

 ArticoliFile.close();

 return trovato;

}

void ModificaPrezzo(char KCodArtPar[15], float NuovoPrezzoPar)

{

 fstream ArticoliFile;

 ArticoliType ArticoliRec;

 int Posizione=RicercaLineare(KCodArtPar);

 if(Posizione==-1)

 cout<<"Articolo non trovato! ";

 else

 {

 ArticoliFile.open("articoli.dat", ios::in | ios::out | ios::binary);

 ArticoliFile.seekg((Posizione-1)\*sizeof(ArticoliRec)); //Ci posizioniamo sul record

 ArticoliFile.read((char \*) &ArticoliRec, sizeof(ArticoliRec));

 ArticoliRec.PrezzoArt=NuovoPrezzoPar;

 //fase di sovrascrittura

 ArticoliFile.seekg((Posizione-1)\*sizeof(ArticoliRec));

 ArticoliFile.write((char \*) &ArticoliRec, sizeof(ArticoliRec));

 }

 ArticoliFile.close();

}

Esercizio 15:

Cancellazione logica in un file con il seguente formato record: Codice,nome,età e tagcanc. La ricerca e la modifica devono essere effettuate mediante una funzione HASH che trasforma il codice (posizione logica nel file) in indirizzo fisico(posizione espressa in byte);

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 04/11/2014

Numero es:15

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di effettuare la cancellazione logica in un file;

Obiettivo del programma: Cancellazione logica in un file con il seguente formato record: Codice,nome,età e tagcanc. La ricerca e la modifica devono essere effettuate mediante una funzione HASH che trasforma il codice (posizione logica nel file) in indirizzo fisico(posizione espressa in byte);

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| PersonaFile  | I/O |  |  |  |
| PersonaRec | I/O |  | PersonaType | 8 BYTE |
| nomefisico | INPUT | Vettore | Carattere | 8 BYTE |
| risp | INPUT |  | Carattere | 1 BYTE |
| codicek | INPUT | Cod. da trovare | Long | 4 BYTE |

\*/

#include<iostream>

#include<fstream>

#include<iomanip>

using namespace std;

struct PersonaType

 {

 long codice;

 char nome;

 char eta;

 int tagcanc;

 };

char nomefisico[8];

long codicek;

//PROTOTIPI

void CaricaFile(char nomefisicopar[8]);

void StampaFile(char nomefisicopar[8]); //legge solo i record che hanno tagcanc a 0

long Hash(long codicekpar); /\* riceve in ingresso la posizione logica del record e restituisce la

 sua posizione in byte \*/

void CancellazioneLogica(char nomefisicopar[8],long codicekpar); //assegna al tagcanc valore 1

int main()

{

 cout<<"Inserire il nome del file: ";

 cin>>nomefisico;

 CaricaFile(nomefisico);

 StampaFile(nomefisico);

 cout<<"Inserisci il codice della persona da cancellare: ";

 cin>>codicek;

 CancellazioneLogica(nomefisico,codicek);

 cout<<endl;

 cout<<"Dopo la cancellazione logica: "<<endl;

 StampaFile(nomefisico);

 cout<<endl;

 system("pause");

}

void CaricaFile(char nomefisicopar[8])

{

 PersonaType PersonaRec;

 char risp;

 fstream PersonaFile;

 PersonaRec.tagcanc=0;

 PersonaFile.open(nomefisicopar, ios::out | ios::binary);

 do

 {

 cout<<"Inserire il codice della persona: ";

 cin>>PersonaRec.codice;

 cout<<"Inserire il nome della persona: ";

 cin>>PersonaRec.nome;

 cout<<"Inserire il eta della persona: ";

 cin>>PersonaRec.eta;

 PersonaFile.write((char \*) &PersonaRec, sizeof(PersonaRec));

 cout<<"Vuoi continuare? s/n ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 PersonaFile.close();

}

void StampaFile(char nomefisicopar[8])

{

 PersonaType PersonaRec;

 fstream PersonaFile;

 PersonaFile.open(nomefisicopar, ios::in | ios::binary);

 if(!PersonaFile) cout<<"Errore nell'apertura del file! ";

 else

 {

 cout<<setw(20)<<"Codice"<<setw(20)<<"Nome"<<setw(20)<<"Eta"<<setw(4)<<"Tag"<<endl;

 PersonaFile.read((char \*) &PersonaRec, sizeof(PersonaRec));

 while(!PersonaFile.eof())

 {

 if(PersonaRec.tagcanc==0) cout<<setw(20)<<PersonaRec.codice<<setw(20)<<PersonaRec.nome<<

 setw(20)<<PersonaRec.eta<<setw(4)<<PersonaRec.tagcanc<<endl;

 PersonaFile.read((char \*) &PersonaRec, sizeof(PersonaRec));

 }

 }

 PersonaFile.close();

}

long Hash(long codicekpar)

{

 PersonaType PersonaRec;

 return(codicekpar\*sizeof(PersonaRec));

}

void CancellazioneLogica(char nomefisicopar[8],long codicekpar)

{

 PersonaType PersonaRec;

 fstream PersonaFile;

 int pos=Hash(codicekpar);

 PersonaFile.open(nomefisicopar, ios::in | ios::out | ios::binary);

 if(!PersonaFile) cout<<"Errore nell'apertura del file! ";

 else

 {

 PersonaFile.seekg(pos);

 PersonaFile.read((char \*) &PersonaRec, sizeof(PersonaRec));

 PersonaRec.tagcanc=1;

 PersonaFile.seekg(pos);

 PersonaFile.write((char \*) &PersonaRec, sizeof(PersonaRec));

 }

 PersonaFile.close();

}

Esercizio 16:

Cancellazione logica in un file con il seguente formato record: Codice,nome,età e tagcanc. La ricerca e la modifica devono essere effettuate mediante una funzione HASH che trasforma il codice (posizione logica nel file) in indirizzo fisico(posizione espressa in byte)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 04/11/2014

Numero es:16

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di inizializzare un file ad accesso diretto;

Obiettivo del programma: Cancellazione logica in un file con il seguente formato record: Codice,nome,età e tagcanc. La ricerca e la modifica devono essere effettuate mediante una funzione HASH che trasforma il codice (posizione logica nel file) in indirizzo fisico(posizione espressa in byte);

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| DipendenteFile | I/O |  | File |  |
| DipendenteRec | I/O |  | DipendenteType |  |
| nomefisico | INPUT | Vettore | Carattere | 10 BYTE |
| Risp | INPUT |  | Carattere | 1 BYTE |
| codicek | INPUT | Cod. da trovare | Intero | 4 BYTE |
| nome | INPUT |  | Carattere | 1 BYTE |
| cognome | INPUT |  | Carattere | 1 BYTE |
| mansione | INPUT |  | Carattere | 1 BYTE |

\*/

#include<iostream>

#include<fstream>

#include<iomanip>

using namespace std;

struct DipendenteType

 {

 int codice;

 char nome;

 char cognome;

 char mansione;

 };

char nomefisico[10];

int codicek;

char nome,cognome,mansione,risp;

//PROTOTIPI

void InizializzaFile(char nomefisicopar[10]);

void StampaFile(char nomefisicopar[10]);

int Hash(int codicekpar); /\* riceve in ingresso la posizione logica del record e restituisce la

 sua posizione in byte \*/

void InserimentoFile(char nomefisicopar[10],int codicekpar,char nomepar,char cognomepar,char mansionepar);

int main()

{

 cout<<"Inserire il nome del file: ";

 cin>>nomefisico;

 InizializzaFile(nomefisico);

 cout<<endl;

 cout<<"Ecco il contenuto del file: "<<endl;

 StampaFile(nomefisico);

 do

 {

 cout<<"Inserire il codice del dipendente da modificare: ";

 cin>>codicek;

 cout<<"Inserire il nome del dipendente: ";

 cin>>nome;

 cout<<"Inserire il cognome del dipendente: ";

 cin>>cognome;

 cout<<"Inserire la mansione del dipendente: ";

 cin>>mansione;

 InserimentoFile(nomefisico,codicek,nome,cognome,mansione);

 cout<<endl;

 cout<<"Dopo la modifica il contenuto del file e': "<<endl;

 StampaFile(nomefisico);

 cout<<endl;

 cout<<"Vuoi continuare? ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 system("pause");

}

void InizializzaFile(char nomefisicopar[10])

{

 DipendenteType DipendenteRec;

 fstream DipendenteFile;

 DipendenteFile.open(nomefisicopar, ios::out | ios::binary);

 for(int i=0;i<10;i++)

 {

 DipendenteRec.codice=i;

 DipendenteRec.nome='-';

 DipendenteRec.cognome='-';

 DipendenteRec.mansione='-';

 DipendenteFile.write((char \*) &DipendenteRec, sizeof(DipendenteRec));

 }

 DipendenteFile.close();

}

void StampaFile(char nomefisicopar[10])

{

 DipendenteType DipendenteRec;

 fstream DipendenteFile;

 DipendenteFile.open(nomefisicopar, ios::in | ios::binary);

 if(!DipendenteFile) cout<<"Errore nell'apertura del file! ";

 else

 {

 cout<<setw(15)<<"Codice"<<setw(15)<<"Nome"<<setw(15)<<"Cognome"<<setw(15)<<"Mansione"<<endl;

 DipendenteFile.read((char \*) &DipendenteRec, sizeof(DipendenteRec));

 while(!DipendenteFile.eof())

 {

 cout<<setw(15)<<DipendenteRec.codice<<setw(15)<<DipendenteRec.nome<<setw(15)<<DipendenteRec.cognome<<setw(15)<<DipendenteRec.mansione<<endl;

 DipendenteFile.read((char \*) &DipendenteRec, sizeof(DipendenteRec));

 }

 }

 DipendenteFile.close();

}

int Hash(int codicekpar)

{

 DipendenteType DipendenteRec;

 return(codicekpar\*sizeof(DipendenteRec));

}

void InserimentoFile(char nomefisicopar[10],int codicekpar,char nomepar,char cognomepar,char mansionepar)

{

 DipendenteType DipendenteRec;

 fstream DipendenteFile;

 int pos=Hash(codicekpar);

 DipendenteFile.open(nomefisicopar, ios::in | ios::out | ios::binary);

 if(!DipendenteFile) cout<<"Errore nell'apertura! ";

 else

 {

 DipendenteFile.seekg(pos);

 DipendenteRec.codice=codicekpar;

 DipendenteRec.nome=nomepar;

 DipendenteRec.cognome=cognomepar;

 DipendenteRec.mansione=mansionepar;

 DipendenteFile.write((char \*) &DipendenteRec, sizeof(DipendenteRec));

 }

 DipendenteFile.close();

}

Esercizio 17:

Codificare l’ordinamento per selezione

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 10/11/2014

Numero es:17

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di applicare l’algoritmo risolutivo dell’ordinamento per selezione;

Obiettivo del programma: Codificare l’ordinamento per selezione

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| vet | I/O | VETTORE | INTERO | 40 BYTE |
| dimlog | INPUT |  | INTERO | 4 BYTE |
| dim | LAVORO |  | INTERO | 4 BYTE |
| i | LAVORO |  | INTERO | 4 BYTE |
| j | LAVORO |  | INTERO | 4 BYTE |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

int vet[10],dimlog;

//Prototipi

int OttieniDim();

void CaricaVet(int vetpar[10],int dimlogpar);

void StampaVet(int vetpar[10],int dimlogpar);

void OrdinamentoPerSelezione(int vetpar[10],int dimlogpar);

int main()

{

 cout<<"Quanti elementi vuoi inserire? ";

 dimlog=OttieniDim();

 CaricaVet(vet,dimlog);

 StampaVet(vet,dimlog);

 cout<<endl;

 cout<<"Dopo l'ordinamento: "<<endl;

 OrdinamentoPerSelezione(vet,dimlog);

 StampaVet(vet,dimlog);

 cout<<endl;

 system("pause");

}

int OttieniDim()

{

 int dim;

 do

 {

 cin>>dim;

 if(!(dim>0 && dim<=10)) cout<<"Hai sbagliato! Riprova! ";

 }

 while(!(dim>0 && dim<=10));

 return dim;

}

void CaricaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++)

 {

 cout<<"Inserire un numero: ";

 cin>>vetpar[i];

 }

}

void StampaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++) cout<<vetpar[i]<<" ";

}

void OrdinamentoPerSelezione(int vetpar[10],int dimlogpar)

{

 int i=0,j,k;

 do

 {

 k=i;

 j=i+1;

 do

 {

 if(vetpar[k]>vetpar[j]) k=j;

 j++;

 }

 while(j<dimlogpar);

 if(i!=k)

 {

 int app=vetpar[i];

 vetpar[i]=vetpar[k];

 vetpar[k]=app;

 }

 i++;

 }

 while(i<dimlogpar-1);

}

Esercizio 18:

Scrivere un programma che attraverso un menù consenta di:

1) Inserimento dei dati di un record;

2) Visualizzazione dei record del file (esclusi quelli cancellati logicamente);

3) Cancellazione logica;

4) Modifica dei dati di un record (escluso il codice);

5) Cancellazione fisica.

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 10/11/2014

Numero es:18

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare i file e di effettuare la cancellazione

fisica;

Obiettivo del programma: Scrivere un programma che attraverso un menù consenta di:

1) Inserimento dei dati di un record;

2) Visualizzazione dei record del file (esclusi quelli cancellati logicamente);

3) Cancellazione logica;

4) Modifica dei dati di un record (escluso il codice);

5) Cancellazione fisica.

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| AnimaleFile | I/O |  | File |  |
| AnimaleRec | I/O |  | AnimaleType | 11 BYTE |
| nomefisico | INPUT | VETTORE | CARATTERE | 15 BYTE |
| nome | INPUT | VETTORE | CARATTERE | 15 BYTE |
| razza | INPUT | VETTORE | CARATTERE | 15 BYTE |
| proprietario | INPUT | VETTORE | CARATTERE | 15 BYTE |
| risp | INPUT |  | CARATTERE | 1 BYTE |
| codicek | INPUT |  | INTERO | 4 BYTE |
| scelta | INPUT |  | INTERO | 4 BYTE |
| scelta2 | INPUT |  | INTERO | 4 BYTE |
| i | LAVORO |  | INTERO | 4 BYTE |
| pos | LAVORO |  | INTERO | 4 BYTE |

\*/

#include<iostream>

#include<fstream>

#include<iomanip>

#include<stdio.h>

#include<cstdlib>

#include<string.h>

using namespace std;

struct AnimaleType

 {

 int codice;

 char nome[15];

 char razza[15];

 char proprietario[15];

 int tagcanc;

 };

char nomefisico[15];

int codicek,scelta,scelta2;

char risp;

//Prototipi

void InizializzaFile(char nomefisicopar[15]);

void StampaFile(char nomefisicopar[15]);

void CancellazioneLogica(char nomefisicopar[15],int codicekpar);

void InserimentoFile(char nomefisicopar[15],int codicekpar,char nomepar[15],char razzapar[15],char proprietariopar[15]);

void CancellazioneFisica(char nomefisicopar[15]);

void ModificaFile(char nomefisicopar[15],int codicekpar,int scelta2par);

int Hash(int codicekpar);

void Ripristino(char nomefisicopar[15]);

int main()

{

 char nome[15],razza[15],proprietario[15];

 cout<<"Inserire il nome del file: ";

 cin>>nomefisico;

 InizializzaFile(nomefisico);

 cout<<endl;

 StampaFile(nomefisico);

 cout<<endl;

 do

 {

 cout<<"Cosa vuoi effettuare? "<<endl;

 cout<<"1)Inserimento dei dati di un record"<<endl;

 cout<<"2)Modifica dei dati di un record"<<endl;

 cout<<"3)Cancellazione logica di un record"<<endl;

 cout<<"4)Cancellazione fisica"<<endl;

 cout<<"5)Ripristino"<<endl;

 cout<<"6)Stampa"<<endl;

 cout<<"Scelta: ";

 cin>>scelta;

 switch(scelta)

 {

 case 1:

 cout<<"In quale codice vuoi inserire i dati? ";

 cin>>codicek;

 cout<<"Inserire il nome dell'animale: ";

 cin>>nome;

 cout<<"Inserire la razza dell'animale: ";

 cin>>razza;

 cout<<"Inserire il proprietario dell'animale: ";

 cin>>proprietario;

 InserimentoFile(nomefisico,codicek,nome,razza,proprietario);

 cout<<endl;

 break;

 case 2:

 cout<<"In quale codice vuoi modificare i dati? ";

 cin>>codicek;

 cout<<"Quale campo vuoi modificare? "<<endl;

 cout<<"1)Nome"<<endl;

 cout<<"2)Razza"<<endl;

 cout<<"3)Proprietario"<<endl;

 cin>>scelta2;

 ModificaFile(nomefisico,codicek,scelta2);

 cout<<endl;

 break;

 case 3:

 cout<<"Inserire il codice del record da cancellare logicamente: ";

 cin>>codicek;

 CancellazioneLogica(nomefisico,codicek);

 cout<<endl;

 break;

 case 4:

 CancellazioneFisica(nomefisico);

 cout<<"Cancellazione fisica effettuata correttamente! "<<endl;

 break;

 case 5:

 Ripristino(nomefisico);

 break;

 case 6:

 StampaFile(nomefisico);

 break;

 default:

 cout<<"Scelta sbagliata! ";

 }

 cout<<endl;

 cout<<"Vuoi continuare? s/n ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 system("pause");

}

void InizializzaFile(char nomefisicopar[15])

{

 AnimaleType AnimaleRec;

 fstream AnimaleFile;

 AnimaleFile.open(nomefisicopar, ios::out | ios::binary);

 for(int i=0;i<10;i++)

 {

 AnimaleRec.codice=i;

 strcpy(AnimaleRec.nome,"-");

 strcpy(AnimaleRec.razza,"-");

 strcpy(AnimaleRec.proprietario,"-");

 AnimaleRec.tagcanc=0;

 AnimaleFile.write((char \*) &AnimaleRec, sizeof(AnimaleRec));

 }

 AnimaleFile.close();

}

void InserimentoFile(char nomefisicopar[15],int codicekpar,char nomepar[15],char razzapar[15],char proprietariopar[15])

{

 AnimaleType AnimaleRec;

 fstream AnimaleFile;

 int pos=Hash(codicekpar);

 AnimaleFile.open(nomefisicopar, ios::in | ios::out | ios::binary);

 if(!AnimaleFile) cout<<"Errore nell'apertura! ";

 else

 {

 AnimaleFile.seekg(pos);

 AnimaleRec.codice=codicekpar;

 strcpy(AnimaleRec.nome,nomepar);

 strcpy(AnimaleRec.razza,razzapar);

 strcpy(AnimaleRec.proprietario,proprietariopar);

 AnimaleRec.tagcanc=0;

 AnimaleFile.write((char \*) &AnimaleRec,sizeof(AnimaleRec));

 AnimaleFile.close();

 }

}

void StampaFile(char nomefisicopar[15])

{

 AnimaleType AnimaleRec;

 fstream AnimaleFile;

 AnimaleFile.open(nomefisicopar, ios::in | ios::binary);

 if(!AnimaleFile) cout<<"Errore nell'apertura! ";

 else

 {

 cout<<setw(10)<<"Codice"<<setw(10)<<"Nome"<<setw(10)<<"Razza"<<

 setw(15)<<"Proprietario"<<setw(10)<<endl;

 AnimaleFile.read((char \*) &AnimaleRec,sizeof(AnimaleRec));

 while(!AnimaleFile.eof())

 {

 if(AnimaleRec.tagcanc==0)

 cout<<setw(10)<<AnimaleRec.codice<<setw(10)<<AnimaleRec.nome<<setw(10)<<AnimaleRec.razza<<

 setw(15)<<AnimaleRec.proprietario<<setw(5)<<AnimaleRec.tagcanc<<endl;

 AnimaleFile.read((char \*) &AnimaleRec,sizeof(AnimaleRec));

 }

 }

 AnimaleFile.close();

}

void ModificaFile(char nomefisicopar[15],int codicekpar,int scelta2par)

{

 AnimaleType AnimaleRec;

 fstream AnimaleFile;

 char nuovonome[15],nuovarazza[15],nuovoproprietario[15];

 int pos=Hash(codicekpar);

 AnimaleFile.open(nomefisicopar, ios::in | ios::out | ios::binary);

 if(!AnimaleFile) cout<<"Errore nell'apertura! ";

 else

 {

 if(scelta2par==1)

 {

 cout<<"Qual'e' il nuovo nome dell'animale? ";

 cin>>nuovonome;

 AnimaleFile.seekg(pos);

 AnimaleFile.read((char \*) &AnimaleRec,sizeof(AnimaleRec));

 AnimaleRec.codice=codicekpar;

 strcpy(AnimaleRec.nome,nuovonome);

 AnimaleRec.tagcanc=0;

 AnimaleFile.seekg(pos);

 AnimaleFile.write((char \*) &AnimaleRec,sizeof(AnimaleRec));

 }

 if(scelta2par==2)

 {

 cout<<"Qual'e' la nuova razza dell'animale? ";

 cin>>nuovarazza;

 AnimaleFile.seekg(pos);

 AnimaleFile.read((char \*) &AnimaleRec,sizeof(AnimaleRec));

 AnimaleRec.codice=codicekpar;

 strcpy(AnimaleRec.razza,nuovarazza);

 AnimaleRec.tagcanc=0;

 AnimaleFile.seekg(pos);

 AnimaleFile.write((char \*) &AnimaleRec,sizeof(AnimaleRec));

 }

 if(scelta2par==3)

 {

 cout<<"Qual'e' il nuovo proprietario dell'animale? ";

 cin>>nuovoproprietario;

 AnimaleFile.seekg(pos);

 AnimaleFile.read((char \*) &AnimaleRec,sizeof(AnimaleRec));

 AnimaleRec.codice=codicekpar;

 strcpy(AnimaleRec.proprietario,nuovoproprietario);

 AnimaleRec.tagcanc=0;

 AnimaleFile.seekg(pos);

 AnimaleFile.write((char \*) &AnimaleRec,sizeof(AnimaleRec));

 }

 AnimaleFile.close();

 }

}

void CancellazioneLogica(char nomefisicopar[15],int codicekpar)

{

 AnimaleType AnimaleRec;

 fstream AnimaleFile;

 int pos=Hash(codicekpar);

 AnimaleFile.open(nomefisicopar, ios::in | ios::out | ios::binary);

 if(!AnimaleFile) cout<<"Errore nell'apertura! ";

 else

 {

 AnimaleFile.seekg(pos);

 AnimaleFile.read((char \*) &AnimaleRec,sizeof(AnimaleRec));

 AnimaleRec.tagcanc=1;

 AnimaleFile.seekg(pos);

 AnimaleFile.write((char \*) &AnimaleRec,sizeof(AnimaleRec));

 AnimaleFile.close();

 }

}

void CancellazioneFisica(char nomefisicopar[15])

{

 AnimaleType AnimaleRec;

 fstream AnimaleFile,AnimaleAppoggio;

 AnimaleFile.open(nomefisicopar, ios::in | ios::binary);

 AnimaleAppoggio.open("appoggio.dat", ios::out | ios::binary);

 //copia di tutti i records tranne quelli cancellati logicamente in un altro file

 //e azzeramento di quelli cancellati logicamente

 AnimaleFile.read((char \*) &AnimaleRec,sizeof(AnimaleRec));

 while(!AnimaleFile.eof())

 {

 if(AnimaleRec.tagcanc==0)

 AnimaleAppoggio.write((char \*) &AnimaleRec,sizeof(AnimaleRec));

 if(AnimaleRec.tagcanc==1)

 {

 strcpy(AnimaleRec.nome,"-");

 strcpy(AnimaleRec.razza,"-");

 strcpy(AnimaleRec.proprietario,"-");

 AnimaleRec.tagcanc=0;

 AnimaleAppoggio.write((char \*) &AnimaleRec,sizeof(AnimaleRec));

 }

 AnimaleFile.read((char \*) &AnimaleRec,sizeof(AnimaleRec));

 }

 AnimaleFile.close();

 AnimaleAppoggio.close();

 remove(nomefisicopar); //rimozione file iniziale

 rename("appoggio.dat",nomefisicopar); //rinominazione del file

}

void Ripristino(char nomefisicopar[15])

{

 AnimaleType AnimaleRec,AnimaleRecApp;

 fstream AnimaleFile;

 int pos=0;

 AnimaleFile.open(nomefisicopar, ios::in | ios::out | ios::binary);

 if(!AnimaleFile) cout<<"Errore nell'apertura! ";

 else

 {

 AnimaleFile.read((char \*) &AnimaleRec, sizeof(AnimaleRec));

 while(!AnimaleFile.eof())

 {

 if(AnimaleRec.tagcanc==1)

 {

 AnimaleFile.seekg((pos)\*sizeof(AnimaleRecApp));

 AnimaleRecApp.codice=AnimaleRec.codice;

 strcpy(AnimaleRecApp.nome,AnimaleRec.nome);

 strcpy(AnimaleRecApp.razza,AnimaleRec.razza);

 strcpy(AnimaleRecApp.proprietario,AnimaleRec.proprietario);

 AnimaleRecApp.tagcanc=0;

 AnimaleFile.write((char \*) &AnimaleRecApp, sizeof(AnimaleRecApp));

 }

 pos++;

 AnimaleFile.read((char \*) &AnimaleRec, sizeof(AnimaleRec));

 }

 }

 AnimaleFile.close();

}

int Hash(int codicekpar)

{

 AnimaleType AnimaleRec;

 return(codicekpar\*sizeof(AnimaleRec));

}

Esercizio 19:

Caricamento e stampa di un file di testo

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 20/11/2014

Numero es:19

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare i file di testo;

Obiettivo del programma: Caricamento e stampa di un file di testo;

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| nomefisico | INPUT | VETTORE | CARATTERE | 20 BYTE |
| car | LAVORO |  | CARATTERE | 1 BYTE |
| risp | INPUT |  | CARATTERE | 1 BYTE |
| filetesto | I/O |  | FILE |  |

\*/

#include<iostream>

#include<fstream>

#include<string.h>

#include<cstdlib>

using namespace std;

char nomefisico[20];

//Prototipi

void CaricaFile(char nomefisicopar[20]);

void StampaFile(char nomefisicopar[20]);

int main()

{

 cout<<"Inserire il nome del file: ";

 cin>>nomefisico;

 CaricaFile(nomefisico);

 StampaFile(nomefisico);

 cout<<endl;

 system("pause");

}

void CaricaFile(char nomefisicopar[20])

{

 fstream filetesto;

 char car,risp;

 filetesto.open(nomefisicopar, ios::out);

 do

 {

 cout<<"Inserire un carattere: ";

 cin>>car;

 filetesto<<car;

 cout<<"Vuoi inserire un altro carattere? ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 filetesto.close();

}

void StampaFile(char nomefisicopar[20])

{

 fstream filetesto;

 char car;

 filetesto.open(nomefisicopar, ios::in);

 if(!filetesto) cout<<"Errore nell'apertura! ";

 else

 {

 cout<<"I caratteri sono: ";

 filetesto>>car;

 while(!filetesto.eof())

 {

 cout<<car<<" ";

 filetesto>>car;

 }

 }

 filetesto.close();

}

Esercizio 20:

Lettura di un file di testo creato con blocco note

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 20/11/2014

Numero es:20

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare i file di testo;

Obiettivo del programma: Lettura di un file di testo creato con blocco note;

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| nomefisico | INPUT | VETTORE | CARATTERE | 20 BYTE |
| parola | I/O | VETTORE | CARATTERE | 20 BYTE |
| filetesto | LAVORO |  | FILE |  |

\*/

#include<iostream>

#include<fstream>

#include<string.h>

#include<cstdlib>

using namespace std;

char nomefisico[20];

//Prototipi

void LeggiFile(char nomefisicopar[20]);

int main()

{

 cout<<"Inserire il nome del file: ";

 cin>>nomefisico;

 LeggiFile(nomefisico);

 cout<<endl;

 system("pause");

}

void LeggiFile(char nomefisicopar[20])

{

 fstream filetesto;

 char parola[20];

 filetesto.open(nomefisicopar, ios::in);

 if(!filetesto) cout<<"Errore nell'apertura! ";

 else

 {

 cout<<"I caratteri sono: "<<endl;

 filetesto>>parola;

 cout<<parola<<endl;

 while(!filetesto.eof())

 {

 filetesto>>parola;

 cout<<parola<<endl;

 }

 }

 filetesto.close();

}

Esercizio 21:

Leggere un file di testo creato con blocco note e copiarne il contenuto in un altro file di testo escludendo una determinata parola

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 20/11/2014

Numero es:21

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare i file di testo;

Obiettivo del programma: Leggere un file di testo creato con blocco note e copiarne il contenuto in un altro file di testo escludendo una determinata parola;

TABELLA DELLE VARIABILI

IDENTIFICATORE INPUT/OUTPUT/LAVORO DESCRIZIONE TIPO DIMENSIONE

nomefisicoO INPUT carattere 20 BYTE

nomefisicoM INPUT carattere 20 BYTE

parolaDaSost INPUT carattere 20 BYTE

parolaNuova INPUT carattere 20 BYTE

parola LAVORO carattere 20 BYTE

filetestoOriginale LAVORO file

filetestoModificato LAVORO file

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| nomefisicoO | INPUT | VETTORE | CARATTERE | 20 BYTE |
| nomefisicoM | INPUT | VETTORE | CARATTERE | 20 BYTE |
| parolaDaSost | INPUT | VETTORE | CARATTERE | 20 BYTE |
| parolaNuova | INPUT | VETTORE | CARATTERE | 20 BYTE |
| parola | LAVORO | VETTORE | CARATTERE | 20 BYTE |
| filetestoOriginale | LAVORO |  | FILE |  |
| filetestoModificato | LAVORO |  | FILE |  |

\*/

#include<iostream>

#include<fstream>

#include<string.h>

#include<cstdlib>

using namespace std;

char nomefisicoO[20],nomefisicoM[20];

char parolaDaSost[20],parolaNuova[20];

//Prototipi

void SostituisciParola(char nomefisicoOpar[20],char nomefisicoMpar[20],char parolaDaSostpar[20],char parolaNuovapar[20]);

int main()

{

 cout<<"Inserire il nome del file originale: ";

 cin>>nomefisicoO;

 cout<<"Inserire il nome del file modificato: ";

 cin>>nomefisicoM;

 cout<<"Inserire la parola che deve essere sostituita: ";

 cin>>parolaDaSost;

 cout<<"Inserire la nuova parola che sostituisce: ";

 cin>>parolaNuova;

 SostituisciParola(nomefisicoO,nomefisicoM,parolaDaSost,parolaNuova);

 system("pause");

}

void SostituisciParola(char nomefisicoOpar[20],char nomefisicoMpar[20],char parolaDaSostpar[20],char parolaNuovapar[20])

{

 fstream filetestoOriginale,filetestoModificato;

 char parola[20];

 filetestoOriginale.open(nomefisicoOpar,ios::in);

 filetestoModificato.open(nomefisicoMpar,ios::out);

 do

 {

 filetestoOriginale>>parola;

 if(strcmp(parolaDaSostpar,parola)==0)

 {

 strcpy(parola,parolaNuovapar);

 filetestoModificato<<parola<<" ";

 }

 else

 filetestoModificato<<parola<<" ";

 }

 while(!filetestoOriginale.eof());

 filetestoOriginale.close();

 filetestoModificato.close();

}

Esercizio 22:

Codificare l'ordinamento per inserzione

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 24/11/2014

Numero es:22

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di applicare l'algoritmo risolutivo dell'ordinamento per inserzione

Obiettivo del programma: Codificare l'ordinamento per inserzione

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| vet | I/O | VETTORE | INTERO | 40 BYTE |
| dimlog | INPUT |  | INTERO | 4 BYTE |
| dim | LAVORO |  | INTERO | 4 BYTE |
| i | LAVORO |  | INTERO | 4 BYTE |
| j | LAVORO |  | INTERO | 4 BYTE |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

int vet[10],dimlog;

//Prototipi

int OttieniDim();

void CaricaVet(int vetpar[10],int dimlogpar);

void StampaVet(int vetpar[10],int dimlogpar);

void OrdinamentoPerInserzione(int vetpar[10],int dimlogpar);

int main()

{

 cout<<"Quanti elementi vuoi inserire? ";

 dimlog=OttieniDim();

 CaricaVet(vet,dimlog);

 StampaVet(vet,dimlog);

 cout<<endl;

 cout<<"Dopo l'ordinamento: "<<endl;

 OrdinamentoPerInserzione(vet,dimlog);

 StampaVet(vet,dimlog);

 cout<<endl;

 system("pause");

}

int OttieniDim()

{

 int dim;

 do

 {

 cin>>dim;

 if(!(dim>0 && dim<=10)) cout<<"Hai sbagliato! Riprova! ";

 }

 while(!(dim>0 && dim<=10));

 return dim;

}

void CaricaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++)

 {

 cout<<"Inserire un numero: ";

 cin>>vetpar[i];

 }

}

void StampaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++) cout<<vetpar[i]<<" ";

}

void OrdinamentoPerInserzione(int vetpar[10],int dimlogpar)

{

 int j,k,i;

 j=1;

 do

 {

 i=j-1;

 k=vetpar[j];

 while(i>=0 && k<vetpar[i])

 {

 vetpar[i+1]=vetpar[i];

 i--;

 }

 vetpar[i+1]=k;

 j++;

 }

 while(j<dimlogpar);

}

Esercizio 23:

Codificare l'ordinamento per affioramento

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 29/11/2014

Numero es:23

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di applicare l'algoritmo risolutivo dell'ordinamento per affioramento

Obiettivo del programma: Codificare l'ordinamento per affioramento

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| vet | I/O | VETTORE | INTERO | 40 BYTE |
| dimlog | INPUT |  | INTERO | 4 BYTE |
| dim | LAVORO |  | INTERO | 4 BYTE |
| i | LAVORO |  | INTERO | 4 BYTE |
| j | LAVORO |  | INTERO | 4 BYTE |
| lim | LAVORO |  | INTERO | 4 BYTE |
| flag | LAVORO |  | INTERO | 4 BYTE |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

int vet[10],dimlog;

//Prototipi

int OttieniDim();

void CaricaVet(int vetpar[10],int dimlogpar);

void StampaVet(int vetpar[10],int dimlogpar);

void OrdinamentoPerAffioramento(int vetpar[10],int dimlogpar);

int main()

{

 cout<<"Quanti elementi vuoi inserire? ";

 dimlog=OttieniDim();

 CaricaVet(vet,dimlog);

 StampaVet(vet,dimlog);

 cout<<endl;

 cout<<"Dopo l'ordinamento: "<<endl;

 OrdinamentoPerAffioramento(vet,dimlog);

 StampaVet(vet,dimlog);

 cout<<endl;

 system("pause");

}

int OttieniDim()

{

 int dim;

 do

 {

 cin>>dim;

 if(!(dim>0 && dim<=10)) cout<<"Hai sbagliato! Riprova! ";

 }

 while(!(dim>0 && dim<=10));

 return dim;

}

void CaricaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++)

 {

 cout<<"Inserire un numero: ";

 cin>>vetpar[i];

 }

}

void StampaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++) cout<<vetpar[i]<<" ";

}

void OrdinamentoPerAffioramento(int vetpar[10],int dimlogpar)

{

 int lim,flag,j;

 do

 {

 lim=dimlogpar,flag=0,j=0;

 do

 {

 if(vetpar[j]>vetpar[j+1])

 {

 int app=vetpar[j];

 vetpar[j]=vetpar[j+1];

 vetpar[j+1]=app;

 flag=j;

 }

 j++;

 }

 while(!(j>lim-2));

 if(flag!=0) lim=flag;

 }

 while(flag!=0);

}

Esercizio 24:

Codificare l'algoritmo di partizione

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 28/11/2014

Numero es:24

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di applicare l'algoritmo di partizione

Obiettivo del programma: Codificare l'algoritmo di partizione

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| vet | I/O | VETTORE | INTERO | 40 BYTE |
| dimlog | INPUT |  | INTERO | 4 BYTE |
| dim | LAVORO |  | INTERO | 4 BYTE |
| i | LAVORO |  | INTERO | 4 BYTE |
| j | LAVORO |  | INTERO | 4 BYTE |
| app | LAVORO |  | INTERO | 4 BYTE |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

int vet[10],dimlog;

//Prototipi

int OttieniDim();

void CaricaVet(int vetpar[10],int dimlogpar);

void StampaVet(int vetpar[10],int dimlogpar);

void AlgoritmoDiPartizione(int vetpar[10],int dimlogpar);

int main()

{

 cout<<"Quanti elementi vuoi inserire? ";

 dimlog=OttieniDim();

 CaricaVet(vet,dimlog);

 cout<<"Ecco il vettore: "<<endl;

 StampaVet(vet,dimlog);

 cout<<endl;

 cout<<"Ecco il vettore partizionato rispetto al perno (primo elemento): "<<endl;

 AlgoritmoDiPartizione(vet,dimlog);

 StampaVet(vet,dimlog);

 cout<<endl;

 system("pause");

}

int OttieniDim()

{

 int dim;

 do

 {

 cin>>dim;

 if(!(dim>0 && dim<=10)) cout<<"Hai sbagliato! Riprova! ";

 }

 while(!(dim>0 && dim<=10));

 return dim;

}

void CaricaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++)

 {

 cout<<"Inserire un numero: ";

 cin>>vetpar[i];

 }

}

void StampaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++) cout<<vetpar[i]<<" ";

}

void AlgoritmoDiPartizione(int vetpar[10],int dimlogpar)

{

 int app;

 int i=0;

 int j=dimlogpar;

 while(i<j)

 {

 do

 {

 j--;

 }

 while(vetpar[i]<vetpar[j] && i<j);

 if(i<j)

 {

 app=vetpar[i];

 vetpar[i]=vetpar[j];

 vetpar[j]=app;

 do

 {

 i++;

 }

 while(vetpar[i]<vetpar[j] && i<j);

 }

 if(i<j)

 {

 app=vetpar[i];

 vetpar[i]=vetpar[j];

 vetpar[j]=app;

 }

 }

}

Esercizio 25:

Codificare l'algoritmo di quick-sort

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 05/01/2015

Numero es:25

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di applicare l'algoritmo di quick sort

Obiettivo del programma: Codificare l'algoritmo di quick sort

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| vet | I/O | VETTORE | INTERO | 40 BYTE |
| dimlog | INPUT |  | INTERO | 4 BYTE |
| dim | LAVORO |  | INTERO | 4 BYTE |
| i | LAVORO |  | INTERO | 4 BYTE |
| j | LAVORO |  | INTERO | 4 BYTE |
| app | LAVORO |  | INTERO | 4 BYTE |
| inf | INPUT |  | INTERO | 4 BYTE |
| sup | INPUT |  | INTERO | 4 BYTE |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

int vet[10],dimlog,inf,sup;

//Prototipi

int OttieniDim();

void CaricaVet(int vetpar[10],int dimlogpar);

void StampaVet(int vetpar[10],int dimlogpar);

int AlgoritmoDiPartizione(int vetpar[10],int infpar,int suppar);

void QuickSort(int vetpar[10],int infpar,int suppar);

int main()

{

 cout<<"Quanti elementi vuoi inserire? ";

 dimlog=OttieniDim();

 CaricaVet(vet,dimlog);

 cout<<"Ecco il vettore: "<<endl;

 StampaVet(vet,dimlog);

 cout<<endl;

 cout<<"Inserisci inf: ";

 cin>>inf;

 cout<<"Inserisci sup: ";

 cin>>sup;

 QuickSort(vet,inf,sup);

 cout<<"Ecco il vettore dopo l'ordinamento quick-sort: "<<endl;

 StampaVet(vet,dimlog);

 cout<<endl;

 system("pause");

}

int OttieniDim()

{

 int dim;

 do

 {

 cin>>dim;

 if(!(dim>0 && dim<=10)) cout<<"Hai sbagliato! Riprova! ";

 }

 while(!(dim>0 && dim<=10));

 return dim;

}

void CaricaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++)

 {

 cout<<"Inserire un numero: ";

 cin>>vetpar[i];

 }

}

void StampaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++) cout<<vetpar[i]<<" ";

}

int AlgoritmoDiPartizione(int vetpar[10],int infpar,int suppar)

{

 int pivot=vetpar[infpar];

 while(infpar<suppar)

 {

 while(vetpar[infpar]<pivot)

 infpar++;

 while(vetpar[suppar]>pivot)

 suppar--;

 if(vetpar[infpar]==vetpar[suppar])

 infpar++;

 else if(infpar<suppar)

 {

 int app = vetpar[infpar];

 vetpar[infpar]=vetpar[suppar];

 vetpar[suppar]=app;

 }

 }

 return suppar;

}

void QuickSort(int vetpar[10], int infpar, int suppar)

{

 if(infpar<suppar)

 {

 int j=AlgoritmoDiPartizione(vetpar,infpar,suppar);

 QuickSort(vetpar,infpar,j-1);

 QuickSort(vetpar,j+1,suppar);

 }

}

Esercizio 26:

Codificare l'algoritmo di fusione

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 08/01/2015

Numero es:26

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di applicare l'algoritmo di fusione

Obiettivo del programma: Codificare l'algoritmo di fusione

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| vet1 | INPUT | VETTORE | INTERO | 40 BYTE |
| vet2 | INPUT | VETTORE | INTERO | 40 BYTE |
| vet3 | OUTPUT | VETTORE | INTERO |  |
| dimlog1 | INPUT |  | INTERO | 4 BYTE |
| dimlog2 | INPUT |  | INTERO | 4 BYTE |
| dim | LAVORO |  | INTERO | 4 BYTE |
| i | LAVORO |  | INTERO | 4 BYTE |
| j | LAVORO |  | INTERO | 4 BYTE |
| k | LAVORO |  | INTERO | 4 BYTE |
| z | LAVORO |  | INTERO | 4 BYTE |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

int vet1[10],vet2[10],dimlog1,dimlog2;

//Prototipi

int OttieniDim();

void CaricaVet(int vetpar[10],int dimlogpar);

void AlgoritmoDiFusione(int vet1par[10],int vet2par[10],int dimlog1par,int dimlog2par);

int main()

{

 cout<<"Quanti elementi vuoi inserire nel primo vettore? ";

 dimlog1=OttieniDim();

 CaricaVet(vet1,dimlog1);

 cout<<"Quanti elementi vuoi inserire nel secondo vettore? ";

 dimlog2=OttieniDim();

 CaricaVet(vet2,dimlog2);

 cout<<"Ecco il vettore creato dopo l'algoritmo di fusione: "<<endl;

 AlgoritmoDiFusione(vet1,vet2,dimlog1,dimlog2);

 cout<<endl;

 system("pause");

}

int OttieniDim()

{

 int dim;

 do

 {

 cin>>dim;

 if(!(dim>0 && dim<=10)) cout<<"Hai sbagliato! Riprova! ";

 }

 while(!(dim>0 && dim<=10));

 return dim;

}

void CaricaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++)

 {

 cout<<"Inserire un numero: ";

 cin>>vetpar[i];

 }

}

void AlgoritmoDiFusione(int vet1par[10],int vet2par[10],int dimlog1par,int dimlog2par)

{

 int vet3[dimlog1par+dimlog2par];

 int i=0,j=0,k=0;

 do

 {

 if(vet1par[i]<=vet2par[j])

 {

 vet3[k]=vet1par[i];

 k++;

 i++;

 }

 else

 {

 vet3[k]=vet2par[j];

 k++;

 j++;

 }

 }

 while(i<dimlog1par && j<dimlog2par);

 if(i<=dimlog1par)

 {

 while(i<dimlog1par)

 {

 vet3[k]=vet1par[i];

 k++;

 i++;

 }

 }

 if(j<=dimlog2par)

 {

 while(j<dimlog2par)

 {

 vet3[k]=vet2par[j];

 k++;

 j++;

 }

 }

 //Stampa

 for(int z=0;z<dimlog1par+dimlog2par;z++) cout<<vet3[z]<<" ";

}

Esercizio 27:

Codificare l'algoritmo di sort-merge

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 09/01/2015

Numero es:27

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di applicare l'algoritmo di sort merge

Obiettivo del programma: Codificare l'algoritmo di sort merge

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| vet | I/O | VETTORE | INTERO | 40 BYTE |
| dimlog | INPUT |  | INTERO | 4 BYTE |
| dim | LAVORO |  | INTERO | 4 BYTE |
| i | LAVORO |  | INTERO | 4 BYTE |
| j | LAVORO |  | INTERO | 4 BYTE |
| k | LAVORO |  | INTERO | 4 BYTE |
| inf | INPUT |  | INTERO | 4 BYTE |
| sup | INPUT |  | INTERO | 4 BYTE |
| centro | LAVORO |  | INTERO | 4 BYTE |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

int vet[10],dimlog;

//Prototipi

int OttieniDim();

void CaricaVet(int vetpar[10],int dimlogpar);

void StampaVet(int vetpar[10],int dimlogpar);

void SortMerge(int vetpar[10],int infpar,int suppar);

void AlgoritmoDiFusione(int vetpar[10],int infpar,int centropar,int suppar);

main()

{

 cout<<"Quanti elementi vuoi inserire? ";

 dimlog=OttieniDim();

 CaricaVet(vet,dimlog);

 StampaVet(vet,dimlog);

 cout<<endl;

 cout<<"Dopo l'ordinamento: "<<endl;

 SortMerge(vet,0,dimlog-1);

 StampaVet(vet,dimlog);

 cout<<endl;

 system("pause");

}

int OttieniDim()

{

 int dim;

 do

 {

 cin>>dim;

 if(!(dim>0 && dim<=10)) cout<<"Hai sbagliato! Riprova! ";

 }

 while(!(dim>0 && dim<=10));

 return dim;

}

void CaricaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++)

 {

 cout<<"Inserire un numero: ";

 cin>>vetpar[i];

 }

}

void StampaVet(int vetpar[10],int dimlogpar)

{

 for(int i=0;i<dimlogpar;i++) cout<<vetpar[i]<<" ";

}

void SortMerge(int vetpar[10],int infpar,int suppar)

{

 if(infpar<suppar)

 {

 int centro = (infpar+suppar)/2;

 SortMerge(vetpar,infpar,centro);

 SortMerge(vetpar,centro+1,suppar);

 AlgoritmoDiFusione(vetpar,infpar,centro,suppar);

 }

}

void AlgoritmoDiFusione(int vetpar[10],int infpar,int centropar,int suppar)

{

 int i,j,k;

 int vet2[8];

 i=infpar;

 j=centropar+1;

 k=0;

 while((i<=centropar)&&(j<=suppar))

 {

 if (vetpar[i]<=vetpar[j])

 {

 vet2[k]=vetpar[i];

 i++;

 }

 else

 {

 vet2[k]=vetpar[j];

 j++;

 }

 k++;

 }

 while (i<=centropar)

 {

 vet2[k]=vetpar[i];

 i++;

 k++;

 }

 while (j<=suppar)

 {

 vet2[k]=vetpar[j];

 j++;

 k++;

 }

 for (k=infpar;k<=suppar;k++)

 {

 vetpar[k]=vet2[k-infpar];

 }

}

Esercizio 28:

Pila di autovetture

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 12/01/2015

Numero es:28

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di usare le liste L.I.F.O. (Pila)

Obiettivo del programma: Pila di autovetture (Lista L.I.F.O.)

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| \*p | I/O |  | STRUCT VETTURA |  |
| \*testa | LAVORO |  | STRUCT VETTURA |  |
| scelta | LAVORO |  | INTERO | 4 BYTE |

\*/

#include<iostream>

#include<cstdlib>

#include<string>

using namespace std;

struct vettura

 {

 string targa;

 int cil;

 struct vettura \*next;

 };

int main()

{

 vettura \*testa;

 int scelta;

 vettura \*p;

 testa=NULL;

 do

 {

 cout<<"1. Inscerisci auto (push)"<<endl;

 cout<<"2. Elimina auto (pop)"<<endl;

 cout<<"3. Stampa elenco auto presenti nel garage"<<endl;

 cout<<"0. Esci"<<endl;

 cout<<"Fai una scelta: ";

 cin>>scelta;

 switch(scelta)

 {

 case 1:

 p=new(vettura);

 cout<<"Inserire il numero di targa: ";

 cin>>p->targa;

 cout<<"Inserire la cilindrata: ";

 cin>>p->cil;

 p->next=testa;

 testa=p;

 cout<<endl;

 break;

 case 2:

 if(testa==NULL)

 cout<<"Non ci sono autovetture disponibili! Attendere!"<<endl;

 else

 {

 p=testa;

 testa=p->next;

 cout<<"La vettura assegnata ha targa "<<p->targa<<" e cilindrata"<<p->cil<<endl;

 delete p;

 }

 cout<<endl;

 break;

 case 3:

 if(testa==NULL)

 cout<<"Non ci sono autovetture disponibili! Attendere!"<<endl;

 else

 {

 cout<<"Elenco autovetture disponibili: "<<endl;

 cout<<"TARGA\tCILINDRATA "<<endl;

 for(p=testa;p!=NULL;p=p->next)

 cout<<p->targa<<"\t"<<p->cil<<endl;

 }

 cout<<endl;

 break;

 }

 }

 while(scelta != 0);

 system("pause");

}

Esercizio 29:

Pila di caratteri

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 13/01/2015

Numero es:29

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di usare le liste L.I.F.O. (Pila)

Obiettivo del programma: Pila di caratteri (Lista L.I.F.O.)

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| \*p | I/O |  | STRUCT NODO |  |
| \*testa | LAVORO |  | STRUCT NODO |  |
| scelta | LAVORO |  | INTERO | 4 BYTE |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

struct nodo

 {

 char carattere;

 struct nodo \*next;

 };

int main()

{

 nodo \*testa;

 nodo \*p;

 testa=NULL;

 int scelta;

 do

 {

 cout<<"1)Inserimento carattere"<<endl;

 cout<<"2)Stampa caratteri"<<endl;

 cout<<"Fai una scelta: (0 per terminare) ";

 cin>>scelta;

 switch(scelta)

 {

 case 1:

 p=new(nodo);

 cout<<"Inserire il carattere: ";

 cin>>p->carattere;

 p->next=testa;

 testa=p;

 cout<<endl;

 break;

 case 2:

 if(testa==NULL)

 cout<<"Non ci sono autovetture disponibili! Attendere!"<<endl;

 else

 {

 cout<<"I caratteri sono: "<<endl;

 for(p=testa;p!=NULL;p=p->next)

 cout<<p->carattere<<" ";

 }

 cout<<endl;

 break;

 }

 }

 while(scelta != 0);

 system("pause");

}

Esercizio 30:

Coda di pazienti

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 15/01/2015

Numero es:30

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di usare le liste F.I.F.O. (Coda)

Obiettivo del programma: Coda di pazienti (Lista F.I.F.O.)

|  |
| --- |
| TABELLA DELLE VARIABILI |
| IDENTIF. | I/O/LAVORO | DESC. | TIPO | DIMENSIONE |
| \*p | I/O |  | STRUCT PAZIENTE |  |
| \*testa | LAVORO |  | STRUCT PAZIENTE |  |
| \*coda | LAVORO |  | STRUCT PAZIENTE |  |
| scelta | LAVORO |  | INTERO | 4 BYTE |
| num | LAVORO |  | INTERO | 4 BYTE |

\*/

#include <iostream>

#include <string>

using namespace std;

struct paziente

 {

 string nome;

 int num;

 struct paziente \*next;

 };

int main()

{

 paziente \*testa,\*coda;

 int scelta;

 paziente \*p;

 int num;

 testa=NULL;

 coda=NULL;

 num=1;

 do

 {

 cout<<"1. Inserisci paziente"<<endl;

 cout<<"2. Estrai paziente"<<endl;

 cout<<"3. Stampa elenco pazienti in attesa"<<endl;

 cout<<"0. Esci"<<endl;

 cout<<"Fai una scelta: "<<endl;

 cin>>scelta;

 switch(scelta)

 {

 case 1:

 p=new(paziente);

 cout<<"Inserire nome paziente: ";

 cin>>p->nome;

 p->num=num++;

 cout<<"Il paziente "<<p->nome<<" ha il numero "<<p->num<<endl;

 p->next=NULL;

 if(testa==NULL)

 testa=coda=p;

 else

 {

 coda->next=p;

 coda=p;

 }

 break;

 case 2:

 if(testa==NULL)

 cout<<"Non ci sono pazienti in coda"<<endl;

 else

 {

 p=testa;

 testa=p->next;

 cout<<"Il prossimo da visitare e' "<<p->nome<<" e ha il numero "<<p->num<<endl;

 delete p;

 }

 break;

 case 3:

 if (testa==NULL)

 cout<<"Non ci sono pazienti in coda"<<endl;

 else

 {

 cout<<"Elenco pazienti in attesa: "<<endl;

 cout<<"NOME\tNUMERO"<<endl;

 for (p=testa;p!=NULL;p=p->next)

 cout<<p->nome<<"\t"<<p->num<<endl;

 }

 break;

 }

 }

 while(scelta!=0);

 system("pause");

}

Esercizio 31:

La classe “Motocicletta” (1)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 27/01/2015

Numero es:31

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di usare le classi in c++

Obiettivo del programma: Realizzare la classe "Motocicletta" (1)

|  |
| --- |
| MotociclettaClasse |
| - char marca;- int StatoMotore; |
| MotociclettaClasse();+ AvviaMotore();+ MostraStato(); |

\*/

#include<iostream>

using namespace std;

class MotociclettaClasse

{

 private:

 char marca;

 int StatoMotore;

 public:

 //costruttore stesso nome della classe e richiamato automaticamente

 //MotociclettaClasse::MotociclettaClasse oppure

 MotociclettaClasse()

 {

 marca='c';

 StatoMotore=0;

 };

 void AvviaMotore();

 void MostraStato();

};

void MotociclettaClasse::AvviaMotore()

{

 if (StatoMotore==1) cout<<"Il motore e' gia' acceso"<<endl;

 else

 {

 StatoMotore=1;

 cout<<"Il motore e' ora acceso"<<endl;

 }

}

void MotociclettaClasse::MostraStato()

{

 cout<<"Questa motocicletta e' una "<<marca<<" e ";

 if (StatoMotore==1) cout<<"il motore e' gia' acceso"<<endl;

 else cout<<"il motore e' spento"<<endl;

}

int main()

{

 MotociclettaClasse Motocicletta;

 Motocicletta.MostraStato();

 Motocicletta.AvviaMotore();

 Motocicletta.MostraStato();

 system("pause");

}

Esercizio 32:

La classe “Telecomando”

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 27/01/2015

Numero es:32

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di usare classi in c++

Obiettivo del programma: Realizzare la classe "Telecomando"

|  |
| --- |
| TelecomandoClasse |
| - int NumCanale; |
| TelecomandoClasse();TelecomandoClasse(int NumCanalePar);+ SetCanale(int CanaleSceltoPar);+ GetCanale(); |

\*/

#include<iostream>

using namespace std;

class TelecomandoClasse

{

 private:

 int NumCanale;

 public:

 TelecomandoClasse() { NumCanale=1; };

 TelecomandoClasse(int NumCanalePar) { NumCanale=NumCanalePar; };

 void SetCanale(int CanaleSceltoPar);

 int GetCanale();

};

int TelecomandoClasse::GetCanale()

{

 return NumCanale;

}

void TelecomandoClasse::SetCanale(int CanaleSceltoPar)

{

 NumCanale=CanaleSceltoPar;

}

int main()

{

 int CanaleScelto;

 TelecomandoClasse Telecomando(7);

 cout<<"Il canale impostato sul telecomando e': "<<Telecomando.GetCanale()<<endl;

 cout<<"Quale canale vuoi impostare? ";

 cin>>CanaleScelto;

 Telecomando.SetCanale(CanaleScelto);

 cout<<"Il canale impostato sul telecomando e': "<<Telecomando.GetCanale()<<endl;

 system("pause");

}

Esercizio 33:

La classe “Motocicletta” (2)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 30/01/2015

Numero es:33

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di usare le classi in c++

Obiettivo del programma: Realizzare la classe "Motocicletta" (2)

Rappresentazione UML della classe:

|  |
| --- |
| MotociclettaClasse |
| - char marca[10];- char colore[10];- int cilindrata;- int StatoMotore;- int serbatoio; |
| MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar);MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar, int serbatoiopar);+ void AvviaMotore();+ void SetSerbatoio(int Bpar);+ int GetSerbatoio();+ void MostraStato(); |

\*/

#include<iostream>

using namespace std;

class MotociclettaClasse

{

 private:

 char marca[10];

 char colore[10];

 int cilindrata;

 int StatoMotore;

 int serbatoio;

 public:

 MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar);

 MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar, int serbatoiopar);

 void AvviaMotore();

 void SetSerbatoio(int Bpar);

 int GetSerbatoio();

 void MostraStato();

};

MotociclettaClasse::MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar)

{

 strcpy(marca,marcapar);

 strcpy(colore,colorepar);

 cilindrata=cilindratapar;

 StatoMotore=0;

 serbatoio=0;

}

MotociclettaClasse::MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar, int serbatoiopar)

{

 strcpy(marca,marcapar);

 strcpy(colore,colorepar);

 cilindrata=cilindratapar;

 StatoMotore=0;

 serbatoio=0;

 if(serbatoiopar<=25) serbatoio=serbatoiopar;

 else cout<<"Troppa benzina!"<<endl;

}

void MotociclettaClasse::AvviaMotore()

{

 if (StatoMotore==1) cout<<"Il motore e' gia' acceso"<<endl;

 else

 {

 StatoMotore=1;

 cout<<"Il motore e' ora acceso"<<endl;

 }

}

void MotociclettaClasse::SetSerbatoio(int Bpar)

{

 if((serbatoio+Bpar)<25)

 serbatoio=serbatoio+Bpar;

 else cout<<"Il carburante eccede la capacita' massima del serbatoio (25 litri)"<<endl;

}

int MotociclettaClasse::GetSerbatoio()

{

 return serbatoio;

}

void MotociclettaClasse::MostraStato()

{

 cout<<"Questa motocicletta e' una "<<marca<<" "<<cilindrata<<" "<<colore<<" e ";

 cout<<"nel serbatoio ci sono "<<serbatoio<<" litri di "<<endl<<"carburante. ";

 if (StatoMotore==1) cout<<"Il motore e' acceso"<<endl;

 else cout<<"Il motore e' spento."<<endl;

}

int main()

{

 int scelta,B;

 MotociclettaClasse Motocicletta("BMW","Nera",2000);

 do

 {

 cout<<endl;

 cout<<"1)Visualizzare lo stato della motocicletta"<<endl;

 cout<<"2)Visualizzare lo stato del serbatoio"<<endl;

 cout<<"3)Inserire benzina nel serbatoio"<<endl;

 cout<<"4)Mettere in moto"<<endl;

 cout<<"Cosa vuoi effettuare? (0 per terminare) ";

 cin>>scelta;

 switch (scelta)

 {

 case 1:

 Motocicletta.MostraStato();

 break;

 case 2:

 cout<<"Nel serbatoio ci sono "<<Motocicletta.GetSerbatoio()<<" litri di carburante "<<endl;

 break;

 case 3:

 cout<<"Quanti litri vuoi inserire? ";

 cin>>B;

 Motocicletta.SetSerbatoio(B);

 break;

 case 4:

 Motocicletta.AvviaMotore();

 break;

 }

 }

 while(scelta!=0);

 system("pause");

}

Esercizio 34:

La classe “Motocicletta” (3)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 03/02/2015

Numero es:34

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di definire le classi ed utilizzare gli oggetti in c++

Obiettivo del programma: Realizzare la classe "Motocicletta" (3);

Rappresentazione UML della classe:

|  |
| --- |
| MotociclettaClasse |
| - char marca[10];- char colore[10];- int cilindrata;- int serbatoio;- int velocita;- int StatoMotore; |
| MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar);MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar, int serbatoiopar);+ void AvviaMotore();+ void SetSerbatoio(int Bpar);+ int GetSerbatoio();+ void Accelerare(int accelerazionepar);+ void Decelerare(int decelerazionepar);+ void MostraStato(); |

\*/

#include<iostream>

using namespace std;

class MotociclettaClasse

{

 private:

 char marca[10];

 char colore[10];

 int cilindrata;

 int serbatoio;

 int velocita;

 int StatoMotore;

 public:

 MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar);

 MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar, int serbatoiopar);

 void AvviaMotore();

 void SetSerbatoio(int Bpar);

 int GetSerbatoio();

 void Accelerare(int accelerazionepar);

 void Decelerare(int decelerazionepar);

 void MostraStato();

};

MotociclettaClasse::MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar)

{

 strcpy(marca,marcapar);

 strcpy(colore,colorepar);

 cilindrata=cilindratapar;

 StatoMotore=0;

 serbatoio=0;

 velocita=0;

}

MotociclettaClasse::MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar, int serbatoiopar)

{

 strcpy(marca,marcapar);

 strcpy(colore,colorepar);

 cilindrata=cilindratapar;

 StatoMotore=0;

 velocita=0;

 serbatoio=0;

 if(serbatoiopar<=25) serbatoio=serbatoiopar;

 else cout<<"Troppa benzina!"<<endl;

}

void MotociclettaClasse::AvviaMotore()

{

 if (StatoMotore==1) cout<<"Il motore e' gia' acceso"<<endl;

 else

 {

 StatoMotore=1;

 cout<<"Il motore e' ora acceso"<<endl;

 }

}

void MotociclettaClasse::SetSerbatoio(int Bpar)

{

 if((serbatoio+Bpar)<=25) serbatoio=serbatoio+Bpar;

 else cout<<"Il carburante eccede la capacita' massima del serbatoio (25 litri)"<<endl;

}

int MotociclettaClasse::GetSerbatoio()

{

 return serbatoio;

}

void MotociclettaClasse::MostraStato()

{

 cout<<"Questa motocicletta e' una "<<marca<<" "<<cilindrata<<" "<<colore<<" e ";

 cout<<"nel serbatoio ci sono "<<serbatoio<<" litri di "<<"carburante. ";

 if (StatoMotore==1) cout<<"La velocita' e' di "<<velocita<<" km/h"<<endl;

 else cout<<"Il motore e' spento."<<endl;

}

void MotociclettaClasse::Accelerare(int accelerazionepar)

{

 if(StatoMotore==1 && serbatoio!=0)

 {

 if(velocita+accelerazionepar<=200)

 {

 velocita=velocita+accelerazionepar;

 serbatoio--;

 }

 else cout<<"Impossibile accelerare! La velocità massima e' 200 km/h"<<endl;

 }

 else cout<<"Impossibile accelerare! Il motore è spento."<<endl;

}

void MotociclettaClasse::Decelerare(int decelerazionepar)

{

 if(StatoMotore==1) velocita=velocita-decelerazionepar;

 else cout<<"Impossibile decelerare! Il motore è spento."<<endl;

}

int main()

{

 int scelta,B,accelerazione,decelerazione;

 MotociclettaClasse Motocicletta("Yamaha","rossa",1000);

 cout<<"\t\t\t MOTOCICLETTA"<<endl;

 do

 {

 cout<<endl;

 cout<<"1)Mettere in moto"<<endl;

 cout<<"2)Visualizzare lo stato della motocicletta"<<endl;

 cout<<"3)Visualizzare lo stato del serbatoio"<<endl;

 cout<<"4)Inserire benzina nel serbatoio"<<endl;

 cout<<"5)Accelerare"<<endl;

 cout<<"6)Decelerare"<<endl;

 cout<<"Cosa vuoi effettuare? (0 per terminare) ";

 cin>>scelta;

 switch (scelta)

 {

 case 1:

 Motocicletta.AvviaMotore();

 break;

 case 2:

 Motocicletta.MostraStato();

 break;

 case 3:

 cout<<"Nel serbatoio ci sono "<<Motocicletta.GetSerbatoio()<<" litri di benzina "<<endl;

 break;

 case 4:

 cout<<"Quanti litri vuoi inserire? ";

 cin>>B;

 Motocicletta.SetSerbatoio(B);

 break;

 case 5:

 cout<<"Di quanto vuoi accelererare? (km/h) ";

 cin>>accelerazione;

 Motocicletta.Accelerare(accelerazione);

 break;

 case 6:

 cout<<"Di quanto vuoi decelerare? (km/h) ";

 cin>>decelerazione;

 Motocicletta.Decelerare(decelerazione);

 break;

 }

 }

 while(scelta!=0);

 system("pause");

}

Esercizio 35:

La classe “Motocicletta” (4)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 05/02/2015

Numero es:35

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di definire le classi ed utilizzare gli oggetti in c++

Obiettivo del programma: Realizzare la classe "Motocicletta" (4)

Rappresentazione UML della classe:

|  |
| --- |
| MotociclettaClasse |
| - char marca[10];- char colore[10];- int cilindrata;- int serbatoio;- int velocita;- int StatoMotore;- float OlioMotore;- float UsuraFreni; |
| MotociclettaClasse()MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar);MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar, int serbatoiopar);+ void AvviaMotore();+ void SetSerbatoio(int Bpar);+ int GetSerbatoio();+ void Accelerare(int accelerazionepar);+ void Decelerare(int decelerazionepar);+ void MostraStato();+ void SetOlioMotore(int Opar);+ float GetOlioMotore();- TuttoOK(); |

\*/

#include<iostream>

using namespace std;

class MotociclettaClasse

{

 private:

 //Attributi privati

 char marca[10];

 char colore[10];

 int cilindrata;

 float serbatoio;

 int velocita;

 float OlioMotore;

 float UsuraFreni;

 int StatoMotore;

 //Metodi privati

 int TuttoOK();

 public:

 MotociclettaClasse()

 {

 strcpy(marca,""); strcpy(colore,""); cilindrata=0; StatoMotore=0; serbatoio=0; velocita=0;

 OlioMotore=5; UsuraFreni=5;

 }

 MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar);

 MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar, int serbatoiopar);

 void AvviaMotore();

 void SetSerbatoio(int Bpar);

 void SetOlioMotore(int Opar);

 float GetOlioMotore();

 float GetSerbatoio();

 void Accelerare(int accelerazionepar);

 void Decelerare(int decelerazionepar);

 void MostraStato();

};

MotociclettaClasse::MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar)

{

 strcpy(marca,marcapar);

 strcpy(colore,colorepar);

 cilindrata=cilindratapar;

 StatoMotore=0;

 serbatoio=0;

 velocita=0;

 OlioMotore=5;

 UsuraFreni=5;

}

MotociclettaClasse::MotociclettaClasse(char marcapar[10],char colorepar[10], int cilindratapar, int serbatoiopar)

{

 strcpy(marca,marcapar);

 strcpy(colore,colorepar);

 cilindrata=cilindratapar;

 StatoMotore=0;

 velocita=0;

 OlioMotore=5;

 UsuraFreni=5;

 serbatoio=0;

 if(serbatoiopar<25) serbatoio=serbatoiopar;

 else

 {

 serbatoio=25;

 cout<<"Il carburante eccede la capacita' massima del serbatoio (25 litri)! "<<endl;

 cout<<"Sono avanzati "<<serbatoiopar-25<<" litri di benzina."<<endl;

 }

}

void MotociclettaClasse::AvviaMotore()

{

 if (StatoMotore==1) cout<<"Il motore e' gia' acceso"<<endl;

 else

 {

 if(TuttoOK()==1)

 {

 StatoMotore=1;

 cout<<"Il motore e' ora acceso"<<endl;

 }

 else cout<<"Olio motore insufficiente o anomalia ai freni."<<endl;

 }

}

void MotociclettaClasse::SetSerbatoio(int Bpar)

{

 if((serbatoio+Bpar)<25) serbatoio=serbatoio+Bpar;

 else

 {

 serbatoio=25;

 cout<<"Il carburante eccede la capacita' massima del serbatoio (25 litri)! "<<endl;

 cout<<"Sono avanzati "<<Bpar-25<<" litri di benzina."<<endl;

 }

}

float MotociclettaClasse::GetSerbatoio()

{

 return serbatoio;

}

float MotociclettaClasse::GetOlioMotore()

{

 return OlioMotore;

}

void MotociclettaClasse::MostraStato()

{

 cout<<"Questa motocicletta e' una "<<marca<<" "<<cilindrata<<" "<<colore<<" e ";

 cout<<"nel serbatoio ci sono "<<serbatoio<<" litri di "<<"carburante. Ci sono "<<OlioMotore;

 cout<<" litri di olio motore e i freni sono al "<<UsuraFreni<<" % ."<<endl;

 if (StatoMotore==1) cout<<"La velocita' e' di "<<velocita<<" km/h"<<endl;

 else cout<<"Il motore e' spento."<<endl;

}

int MotociclettaClasse::TuttoOK()

{

 int flag=0;

 if(OlioMotore>1 && UsuraFreni>1) flag=1;

 return flag;

}

void MotociclettaClasse::Accelerare(int accelerazionepar)

{

 if(StatoMotore==1 && serbatoio!=0)

 {

 if(velocita+accelerazionepar<200)

 {

 velocita=velocita+accelerazionepar;

 serbatoio=serbatoio-0.1;

 OlioMotore=OlioMotore-0.1;

 }

 else

 {

 velocita=200;

 serbatoio=serbatoio-0.2;

 OlioMotore=OlioMotore-0.2;

 cout<<"Si e' raggiunta la velocita' massima (200 km/h)"<<endl;

 }

 }

 else cout<<"Impossibile accelerare! Il motore e' spento oppure non c'e' carburante sufficiente."<<endl;

}

void MotociclettaClasse::Decelerare(int decelerazionepar)

{

 Accelerare(-decelerazionepar);

 UsuraFreni=UsuraFreni-0.1;

}

void MotociclettaClasse::SetOlioMotore(int Opar)

{

 if((OlioMotore+Opar)<25) OlioMotore=OlioMotore+Opar;

 else

 {

 OlioMotore=5;

 cout<<"L'olio eccede la capacita' massima del serbatoio (5 litri)! "<<endl;

 cout<<"Sono avanzati "<<Opar-5<<" litri di olio motore."<<endl;

 }

}

int main()

{

 int scelta,B,O,accelerazione,decelerazione;

 MotociclettaClasse Motocicletta("Yamaha","rossa",1000);

 do

 {

 cout<<endl;

 cout<<"1)Mettere in moto"<<endl;

 cout<<"2)Visualizzare lo stato della motocicletta"<<endl;

 cout<<"3)Visualizzare lo stato del serbatoio"<<endl;

 cout<<"4)Inserire benzina nel serbatoio"<<endl;

 cout<<"5)Accelerare"<<endl;

 cout<<"6)Decelerare"<<endl;

 cout<<"7)Inserire olio nel motore"<<endl;

 cout<<"8)Visualizzare lo stato dell'olio motore"<<endl;

 cout<<"Cosa vuoi effettuare? (0 per terminare) ";

 cin>>scelta;

 switch (scelta)

 {

 case 1:

 Motocicletta.AvviaMotore();

 break;

 case 2:

 Motocicletta.MostraStato();

 break;

 case 3:

 cout<<"Nel serbatoio ci sono "<<Motocicletta.GetSerbatoio()<<" litri di benzina "<<endl;

 break;

 case 4:

 cout<<"Quanti litri vuoi inserire? (max 25) ";

 cin>>B;

 Motocicletta.SetSerbatoio(B);

 break;

 case 5:

 cout<<"Di quanto vuoi accelererare? (km/h) ";

 cin>>accelerazione;

 Motocicletta.Accelerare(accelerazione);

 break;

 case 6:

 cout<<"Di quanto vuoi decelerare? (km/h) ";

 cin>>decelerazione;

 Motocicletta.Decelerare(decelerazione);

 break;

 case 7:

 cout<<"Quanti litri vuoi inserire? (max 5) ";

 cin>>O;

 Motocicletta.SetOlioMotore(O);

 break;

 case 8:

 cout<<"Nel motore ci sono "<<Motocicletta.GetOlioMotore()<<" litri di olio"<<endl;

 break;

 }

 }

 while(scelta!=0);

 system("pause");

}

Esercizio 36:

La classe “Calcolatrice”

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 05/02/2015

Numero es:35

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di definire le classi ed utilizzare gli oggetti in c++

Obiettivo del programma: Realizzare la classe "Calcolatrice"

Rappresentazione UML della classe:

|  |
| --- |
| CalcolatriceClasse |
|  |
| $ + Addiziona(float op1,float op2);$ + Sottrai(float op1,float op2);$ + Moltiplica(float op1,float op2);$ + Dividi(float op1,float op2); |

\*/

#include<iostream>

using namespace std;

class CalcolatriceClasse

{

 public:

 static float Addiziona(float op1,float op2)

 {

 return(op1+op2);

 }

 static float Sottrai(float op1,float op2)

 {

 return(op1-op2);

 }

 static float Moltiplica(float op1,float op2)

 {

 return(op1\*op2);

 }

 static float Dividi(float op1,float op2)

 {

 return(op1/op2);

 }

};

int main()

{

 int scelta;

 float op1,op2;

 do

 {

 cout<<endl;

 cout<<"Inserisci il primo operando: ";

 cin>>op1;

 cout<<"Inserisci il secondo operando: ";

 cin>>op2;

 cout<<endl;

 cout<<"1)Digita 1 per effettuare l'addizione"<<endl;

 cout<<"2)Digita 2 per effettuare la sottrazione"<<endl;

 cout<<"3)Digita 3 per effettuare la moltiplicazione"<<endl;

 cout<<"4)Digita 4 per effettuare la divisione"<<endl;

 cout<<"Cosa vuoi effettuare? (0 per terminare) ";

 cin>>scelta;

 switch(scelta)

 {

 case 1:

 cout<<"La somma tra "<<op1<<" e "<<op2<<" e': "<<CalcolatriceClasse::Addiziona(op1,op2)<<endl;

 break;

 case 2:

 cout<<"La differenza tra "<<op1<<" e "<<op2<<" e': "<<CalcolatriceClasse::Sottrai(op1,op2)<<endl;

 break;

 case 3:

 cout<<"Il prodotto tra "<<op1<<" e "<<op2<<" e': "<<CalcolatriceClasse::Moltiplica(op1,op2)<<endl;

 break;

 case 4:

 cout<<"La divisione tra "<<op1<<" e "<<op2<<" e': "<<CalcolatriceClasse::Dividi(op1,op2)<<endl;

 break;

 default:

 cout<<"Il tasto premuto non consente di effettuare alcuna operazione. "<<endl;

 }

 }

 while(scelta!=0);

 system("pause");

}

Esercizio 37 (Pag.74 n.1)

* I metodi privati di una classe non fanno parte della sua interfaccia con l’esterno. Vero.
* I metodi pubblici e privati sono incapsulati nell’oggetto. Vero.
* Non è possibile vedere all’esterno l’implementazione di un metodo pubblico. Vero.
* Non è possibile vedere all’esterno l’implementazione di un metodo privato. Vero.
* Information hiding significa nascondere l’implementazione dei metodi. Vero.
* Una modifica a una varibile di classe si ripercuote su tutte le istanze di quella classe.Vero.
* Un oggetto può modificare una variabile di classe. Falso.
* E’ possibile richiamare un metodo di classe indipendentemente dall’esistenza di un’istanza di classe. Vero.
* E’ possibile utilizzare oggetti come parametri. Vero.
* E’ possibile utilizzare oggetti come valori di ritorno. Vero.

Esercizio 38 (Pag.74 n.2)

* In che senso la OOP è una evoluzione della programmazione imperativa?

Nel senso che aggiunge aggiunge ai concetti già analizzati della programmazione imperativa tre caratteristiche fondamentali: incapsulamento, ereditarietà e polimorfismo.

Esercizio 39 (Pag.74 n.3)

* Qual è un grosso inconveniente nell’utilizzo dei moduli?

Il grosso inconveniente nell’utilizzo dei moduli e’ che non hanno dati propri.

Esercizio 40 (Pag.74 n.4)

* Che cosa significa incapsulamento?

L’incapsulamento significa racchiudere all’interno di una classe tutte le sue caratteristiche e i suoi comportamenti.

Esercizio 41 (Pag.74 n.5)

* Che cosa indicano le parole chiave PUBBLICO e PRIVATO?

Le parole chiave PUBBLICO e PRIVATO sono specificatore di accesso. Pubblico rende visibile all’esterno le segnature dei metodi e gli attributi; privato nasconde all’esterno segnature e metodi.

Esercizio 42 (Pag.74 n.6)

* In cosa consiste l’information hiding?

L’information hiding consiste nel nascondere l’implementazione dei metodi e rendere noto all’esterno solo la loro segnatura.

Esercizio 43 (Pag.74 n.7)

* Quale problema della programmazione “tradizionale” risolve l’incapsulamento?

L’incapsulamento rende il codice molto più ordinato e dunque più sicuro, dato che tutti i dati e i comportamenti di un oggetto si trovano in un’unità sintattica semantica.

Esercizio 44 (Pag.74 n.8)

* Come si esprime in UML l’information hiding?

Per indicare metodi/attributi pubblici si utlizza il simbolo “+”; per indicare metodi/attributi privati si utilizza il simbolo “-“.

Esercizio 45 (Pag.74 n.9)

* Che cos’è la relazione di aggregazione stretta? E la relazione di aggregazione lasca?

Un’aggregazione lasca indica l’indipendenza del ciclo di vita dell’oggetto contenuto dall’oggetto contenitore; un’aggregazione stretta indica che l’oggetto contenuto non ha una vita propria ma esiste unicamente perché è stato creato l’oggetto contenitore.

Esercizio 46 (Pag.74 n.10)

* Come si rappresenta in UML la relazione di aggregazione stretta? E la relazione di aggregazione lasca?

L’aggregazione stretta si rappresenta con un rombo pieno; l’aggregazione lasca si rappresenta con un rombo vuoto.

Esercizio 47 (Pag.74 n.11)

* Che differenza c’è tra variabile istanza e variabile di classe?

La variabile di classe non è collegata ad una particolare istanza della classe, cosicchè è possibile accedere alla variabile anche se non è stato mai creato un oggetto di quella classe.

Esercizio 48 (Pag.74 n.12)

* Che differenza c’è tra metodo istanza e metodo di classe?

Il metodo di classe non è collegato ad una particolare istanza della classe, cosicchè è possibile accedere al metodo anche se non è stato mai creato un oggetto di quella classe.

Esercizio 49 (Pag.75 n.1)

* Descrivi l’interfaccia con l’esterno della classe denominata Autoradio.

class AutoradioClasse

{

 private:

 int StatoAutoradio;

 float frequenza;

 int volume;

 public:

 void Accensione();

 void ModificaFrequenza(float frequenzapar);

 void ModificaVolume(int volumepar);

};

Esercizio 50:

La classe “Triangolo”

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 19/02/2015

Numero es:50

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di definire le classi ed utilizzare gli oggetti in c++

Obiettivo del programma: Codifica c++ classe "Triangolo" pag.76 n.24

Rappresentazione UML della classi:

|  |
| --- |
| PuntoClasse |
| - float x;- float y; |
| PuntoClasse() { x=0; y=0; }; //costruttore che crea un punto nell'originePuntoClasse(float xpar, float ypar) { x=xpar; y=ypar; }; //costruttore+ float Distanza(PuntoClasse P); |

|  |
| --- |
| TriangoloClasse |
| - PuntoClasse P1;- PuntoClasse P2;- PuntoClasse P3; |
| TriangoloClasse(PuntoClasse Punto1par,PuntoClasse Punto2par,PuntoClasse Punto3par);+ float perimetro(); |

\*/

#include<iostream>

#include<math.h>

using namespace std;

class PuntoClasse

{

 private:

 float x;

 float y;

 public:

 PuntoClasse() { x=0; y=0; }; //costruttore che crea un punto nell'origine

 PuntoClasse(float xpar, float ypar) { x=xpar; y=ypar; }; //costruttore

 float Distanza(PuntoClasse P);

};

float PuntoClasse::Distanza(PuntoClasse P)

{

 float ris;

 ris=sqrt(pow((this->x - P.x),2)+pow((this->y - P.y),2));

 return ris;

}

class TriangoloClasse

{

 private:

 PuntoClasse P1;

 PuntoClasse P2;

 PuntoClasse P3;

 public:

 TriangoloClasse(PuntoClasse Punto1par,PuntoClasse Punto2par,PuntoClasse Punto3par);

 float perimetro();

};

TriangoloClasse::TriangoloClasse(PuntoClasse Punto1par,PuntoClasse Punto2par,PuntoClasse Punto3par)

{

 P1=Punto1par;

 P2=Punto2par;

 P3=Punto3par;

}

float TriangoloClasse::perimetro()

{

 float p;

 p=P1.Distanza(P2)+P2.Distanza(P3)+P3.Distanza(P1);

 return p;

}

int main()

{

 PuntoClasse Punto1(0,0);

 PuntoClasse Punto2(2,0);

 PuntoClasse Punto3(0,2);

 TriangoloClasse Triangolo(Punto1,Punto2,Punto3);

 cout<<"Il perimetro del triangolo e': "<<Triangolo.perimetro()<<endl; //4+rad8=6,83

 system("pause");

}

Esercizio 51:

Pix1

#include <graphics.h> // declarations for graphics library

#include <conio.h> // for getch() function

enum Boolean {false, true};

class Locazione {

protected: // allows derived class to access private data

 int X;

 int Y;

public: // these functions can be accessed from outside

 Locazione(int InizX, int InizY);

 int daiX();

 int daiY();

};

class Punto : public Locazione { // derived from class Location

 protected:

 Boolean Visibile; // classes derived from Point will need access

public:

 Punto(int InizX, int InizY); // constructor

 void Mostra();

 void Nascondi();

 Boolean Visibile\_si();

 void Muovi\_a(int NuovoX, int NuovoY);

};

// member functions for the Location class

Locazione::Locazione(int InizX, int InizY) {

 X = InizX;

 Y = InizY;

};

int Locazione::daiX(void) {

 return X;

};

int Locazione::daiY(void) {

 return Y;

};

// member functions for the Point class: These assume

// the main program has initialized the graphics system

Punto::Punto(int InizX, int InizY) : Locazione(InizX,InizY) {

 Visibile = false; // make invisible by default

};

void Punto::Mostra(void) {

 Visibile = true;

 putpixel(X, Y, getcolor()); // uses default color

};

void Punto::Nascondi(void) {

 Visibile = false;

 putpixel(X, Y, getbkcolor()); // uses background color to erase

};

Boolean Punto::Visibile\_si(void) {

 return Visibile;

};

void Punto::Muovi\_a(int NuovoX, int NuovoY) {

 Nascondi(); // make current point invisible

 X = NuovoX; // change X and Y coordinates to new location

 Y = NuovoY;

 Mostra(); // show point at new location

};

int main()

{

 // initialize the graphics system

 int pilotagrafico = DETECT, modografico;

 initgraph(&pilotagrafico, &modografico, "..\\bgi");

 // move a point across the screen

 Punto UnPunto(100, 50); // Initial X, Y at 100, 50

 UnPunto.Mostra(); // Unpunto turns itself on

 getch(); // Wait for keypress

 UnPunto.Muovi\_a(300, 150); // Unpunto moves to 300,150

 getch(); // Wait for keypress

 UnPunto.Nascondi(); // Unpunto turns itself off

 getch(); // Wait for keypress

 closegraph(); // Restore original screen

 return 0;

}

Esercizio 52:

Pix2

#include <graphics.h> // declarations for graphics library

#include <conio.h> // for getch() function

enum Boolean {false, true};

class Locazione {

protected: // allows derived class to access private data

 int X;

 int Y;

public: // these functions can be accessed from outside

 Locazione(int InizX, int InizY);

 int GetX();

 int GetY();

};

class Punto : public Locazione { // derived from class Location

 protected:

 Boolean Visibile; // classes derived from Point will need access

public:

 Punto(int InizX, int InizY); // constructor

 void Mostra();

 void Nascondi();

 Boolean Visibile\_si();

 void Muovi\_a(int NuovoX, int NuovoY);

};

// member functions for the Location class

Locazione::Locazione(int InizX, int InizY) {

 X = InizX;

 Y = InizY;

};

int Locazione::GetX(void) {

 return X;

};

int Locazione::GetY(void) {

 return Y;

};

// member functions for the Point class: These assume

// the main program has initialized the graphics system

Punto::Punto(int InizX, int InizY) : Locazione(InizX,InizY) {

 Visibile = false; // make invisible by default

};

void Punto::Mostra(void) {

 Visibile = true;

 putpixel(X, Y, getcolor()); // uses default color

};

void Punto::Nascondi(void) {

 Visibile = false;

 putpixel(X, Y, getbkcolor()); // uses background color to erase

};

Boolean Punto::Visibile\_si(void) {

 return Visibile;

};

void Punto::Muovi\_a(int NuovoX, int NuovoY) {

 Nascondi(); // make current point invisible

 X = NuovoX; // change X and Y coordinates to new location

 Y = NuovoY;

 Mostra(); // show point at new location

};

// CIRCLE A Circle class derived from Point

class Circonferenza : Punto { // derived privately from class Point

 // and ultimately from class Location

 int Raggio; // private by default

public:

 Circonferenza(int InizX, int InizY, int InitRaggio);

 void Mostra(void);

 void Nascondi(void);

 void Espandi(int Espandi\_di);

 void Muovi\_a(int NuovoX, int NuovoY);

 void Contrai(int Contrai\_di);

};

Circonferenza::Circonferenza(int InizX, int InizY, int InizRaggio)

 : Punto(InizX,InizY)

{

 Raggio = InizRaggio;

};

void Circonferenza::Mostra(void)

{

 Visibile = true;

 circle(X, Y, Raggio); // draw the circle

}

void Circonferenza::Nascondi(void)

{

 unsigned int TemporaneoColore; // to save current color

 TemporaneoColore = getcolor(); // set to current color

 setcolor(getbkcolor()); // set drawing color to background

 Visibile = false;

 circle(X, Y, Raggio); // draw in background color to erase

 setcolor(TemporaneoColore); // set color back to current color

};

void Circonferenza::Espandi(int Espandi\_di)

{

 Nascondi(); // erase old circle

 Raggio += Espandi\_di; // expand radius

 if (Raggio < 0) // avoid negative radius

 Raggio = 0;

 Mostra(); // draw new circle

};

void Circonferenza::Contrai(int Contrai\_di)

{

 Espandi(-Contrai\_di); // redraws with (Radius - Contrai\_di)

};

void Circonferenza::Muovi\_a(int NuovoX, int NuovoY)

{

 Nascondi(); // erase old circle

 X = NuovoX; // set new location

 Y = NuovoY;

 Mostra(); // draw in new location

};

main() // test the functions

{

 // initialize the graphics system

 int pilotagrafico = DETECT, modografico;

 initgraph(&pilotagrafico, &modografico, "..\\bgi");

 Circonferenza MiaCirconferenza(100, 200, 50); // declare a circle object

 MiaCirconferenza.Mostra(); // show it

 getch(); // wait for keypress

 MiaCirconferenza.Muovi\_a(200, 250); // move the circle (tests hide

 // and show also)

 getch();

 MiaCirconferenza.Espandi(50); // make it bigger

 getch();

 MiaCirconferenza.Contrai(75); // make it smaller

 getch();

 closegraph();

 return 0;

}

Esercizio 53:

Pix3

#include <conio.h> // for getch() function

#include <graphics.h>

enum Boolean {false, true};

class Location {

protected: // allows derived class to access private data

 int X;

 int Y;

public: // these functions can be accessed from outside

 Location(int InitX, int InitY);

 int GetX();

 int GetY();

};

class Point : public Location { // derived from class Location

 protected:

 Boolean Visible; // classes derived from Point will need access

public:

 Point(int InitX, int InitY); // constructor

 void Show();

 void Hide();

 Boolean IsVisible();

 void MoveTo(int NewX, int NewY);

};

// member functions for the Location class

Location::Location(int InitX, int InitY) {

 X = InitX;

 Y = InitY;

};

int Location::GetX(void) {

 return X;

};

int Location::GetY(void) {

 return Y;

};

// member functions for the Point class: These assume

// the main program has initialized the graphics system

Point::Point(int InitX, int InitY) : Location(InitX,InitY) {

 Visible = false; // make invisible by default

};

void Point::Show(void) {

 Visible = true;

 putpixel(X, Y, getcolor()); // uses default color

};

void Point::Hide(void) {

 Visible = false;

 putpixel(X, Y, getbkcolor()); // uses background color to erase

};

Boolean Point::IsVisible(void) {

 return Visible;

};

void Point::MoveTo(int NewX, int NewY) {

 Hide(); // make current point invisible

 X = NewX; // change X and Y coordinates to new location

 Y = NewY;

 Show(); // show point at new location

};

// A Circle class derived from Point

class Circle : public Point { // derived from class Point

 // and ultimately from class Location

 int Radius; // private by default

public:

 Circle(int InitX, int InitY, int InitRadius);

 void Show(void);

 void Hide(void);

 void Expand(int ExpandBy);

 void Contract(int ContractBy);

 //void MoveTo(int NewX, int NewY);

};

// Circle constructor calls base Point constructor first

Circle::Circle(int InitX, int InitY, int InitRadius) : Point(InitX,InitY)

{

 Radius = InitRadius;

};

void Circle::Show()

{

 Visible = true;

 circle(X, Y, Radius); // draw the circle using BGI function

}

void Circle::Hide()

{

 if (!Visible) return; // no need to hide

 unsigned int TempColor; // to save current color

 TempColor = getcolor(); // set to current color

 setcolor(getbkcolor()); // set drawing color to background

 Visible = false;

 circle(X, Y, Radius); // draw in background color to erase

 setcolor(TempColor); // set color back to current color

};

void Circle::Expand(int ExpandBy)

{

 Boolean vis = Visible; // is current circle visible?

 if (vis) Hide(); // if so, hide it

 Radius += ExpandBy; // expand radius

 if (Radius < 0) // avoid negative radius

 Radius = 0;

 if (vis) Show(); // draw new circle if previously visible

};

inline void Circle::Contract(int ContractBy)

{

 Expand(-ContractBy); // redraws with (Radius - ContractBy)

};

/\*

void Circle::MoveTo(int NewX, int NewY) {

 Hide(); // make current point invisible

 X = NewX; // change X and Y coordinates to new location

 Y = NewY;

 Show(); // show point at new location

};

 \*/

main() // test the functions

{

 // initialize the graphics system

 int graphdriver = DETECT, graphmode;

 initgraph(&graphdriver, &graphmode, "..\\bgi");

 Circle MyCircle(100, 200, 50); // declare a circle object

 MyCircle.Show(); // show it

 getch(); // wait for keypress

 MyCircle.MoveTo(200, 250); // move the circle (tests hide

 // and show also)

 getch();

 MyCircle.Expand(50); // make it bigger

 getch();

 MyCircle.Contract(75); // make it smaller

 getch();

 closegraph();

 return 0;

}

Esercizio 54:

Pix4

#include <conio.h> // for getch() function

#include <graphics.h>

enum Boolean {false, true};

class Location {

protected: // allows derived class to access private data

 int X;

 int Y;

public: // these functions can be accessed from outside

 Location(int InitX, int InitY);

 int GetX();

 int GetY();

};

class Point : public Location { // derived from class Location

 protected:

 Boolean Visible; // classes derived from Point will need access

public:

 Point(int InitX, int InitY); // constructor

 virtual void Show();

 virtual void Hide();

 Boolean IsVisible();

 void MoveTo(int NewX, int NewY);

};

// member functions for the Location class

Location::Location(int InitX, int InitY) {

 X = InitX;

 Y = InitY;

};

int Location::GetX(void) {

 return X;

};

int Location::GetY(void) {

 return Y;

};

// member functions for the Point class: These assume

// the main program has initialized the graphics system

Point::Point(int InitX, int InitY) : Location(InitX,InitY) {

 Visible = false; // make invisible by default

};

void Point::Show(void) {

 Visible = true;

 putpixel(X, Y, getcolor()); // uses default color

};

void Point::Hide(void) {

 Visible = false;

 putpixel(X, Y, getbkcolor()); // uses background color to erase

};

Boolean Point::IsVisible(void) {

 return Visible;

};

void Point::MoveTo(int NewX, int NewY) {

 Hide(); // make current point invisible

 X = NewX; // change X and Y coordinates to new location

 Y = NewY;

 Show(); // show point at new location

};

// A Circle class derived from Point

class Circle : public Point { // derived from class Point

 // and ultimately from class Location

 int Radius; // private by default

public:

 Circle(int InitX, int InitY, int InitRadius);

 void Show(void);

 void Hide(void);

 void Expand(int ExpandBy);

 void Contract(int ContractBy);

 //void MoveTo(int NewX, int NewY);

};

// Circle constructor calls base Point constructor first

Circle::Circle(int InitX, int InitY, int InitRadius) : Point(InitX,InitY)

{

 Radius = InitRadius;

};

void Circle::Show()

{

 Visible = true;

 circle(X, Y, Radius); // draw the circle using BGI function

}

void Circle::Hide()

{

 if (!Visible) return; // no need to hide

 unsigned int TempColor; // to save current color

 TempColor = getcolor(); // set to current color

 setcolor(getbkcolor()); // set drawing color to background

 Visible = false;

 circle(X, Y, Radius); // draw in background color to erase

 setcolor(TempColor); // set color back to current color

};

void Circle::Expand(int ExpandBy)

{

 Boolean vis = Visible; // is current circle visible?

 if (vis) Hide(); // if so, hide it

 Radius += ExpandBy; // expand radius

 if (Radius < 0) // avoid negative radius

 Radius = 0;

 if (vis) Show(); // draw new circle if previously visible

};

inline void Circle::Contract(int ContractBy)

{

 Expand(-ContractBy); // redraws with (Radius - ContractBy)

};

/\*

void Circle::MoveTo(int NewX, int NewY) {

 Hide(); // make current point invisible

 X = NewX; // change X and Y coordinates to new location

 Y = NewY;

 Show(); // show point at new location

};

 \*/

main() // test the functions

{

 // initialize the graphics system

 int graphdriver = DETECT, graphmode;

 initgraph(&graphdriver, &graphmode, "..\\bgi");

 Circle MyCircle(100, 200, 50); // declare a circle object

 MyCircle.Show(); // show it

 getch(); // wait for keypress

 MyCircle.MoveTo(200, 250); // move the circle (tests hide

 // and show also)

 getch();

 MyCircle.Expand(50); // make it bigger

 getch();

 MyCircle.Contract(75); // make it smaller

 getch();

 closegraph();

 return 0;

}

Esercizio 55:

La classe “Cilindro" derivata dalla classe “Cerchio”

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 23/02/2015

Numero es:55

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di definire le classi ed utilizzare gli oggetti in c++

Obiettivo del programma: Codifica c++ La classe “Cilindro" derivata dalla classe “Cerchio”;

Rappresentazione UML della classi:

|  |
| --- |
| CerchioClasse |
| - float raggio; |
| CerchioClasse(float raggio) + void SetRaggio(float raggio) + void GetRaggio(float raggio) + float Area()+ float Circonferenza(); |

|  |
| --- |
| CilindroClasse |
| - float altezza; |
| CilindroClasse(float raggio,float altezza);+ void SetAltezza(float altezza);+ float Volume();+ float Area(); |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

class CerchioClasse

{

 private:

 float raggio;

 public:

 CerchioClasse(float raggio) { this->raggio=raggio; };

 void SetRaggio(float raggio) { this->raggio=raggio; };

 float Area() { return(raggio\*raggio\*3.14); };

 float Circonferenza() { return(raggio\*2\*3.14); };

};

class CilindroClasse : public CerchioClasse //la classe cilindro deriva dalla classe cerchio

{

 private:

 float altezza;

 public:

 CilindroClasse(float raggio,float altezza) : CerchioClasse(raggio)

 { this->altezza=altezza; };

 void SetAltezza(float altezza) { this->altezza=altezza; };

 float Volume() { return (CerchioClasse::Area()\*altezza); }; //funzione Area() della classe Cerchio

 float Area();

};

int main()

{

 CilindroClasse Cilindro(4,10);

 cout<<"L'area del cilindro e': "<<Cilindro.Area()<<endl; //351,68

 cout<<"Il volume del cilindro e': "<<Cilindro.Volume()<<endl; //502,4

 system("pause");

}

float CilindroClasse::Area()

{

 float AreaBase,AreaLaterale;

 AreaBase=CerchioClasse::Area()\*2; //funzione Area() della classe Cerchio

 AreaLaterale=CerchioClasse::Circonferenza()\*altezza; //funzione Circonferenza() della classe Cerchio

 return (AreaBase+AreaLaterale);

};

Esercizio 56:

La classe “Penna colorata" derivata dalla classe “Penna”

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 02/03/2015

Numero es:56

Versione:1.1

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di definire le classi ed utilizzare gli oggetti in c++

Obiettivo del programma: Codifica c++ "La classe penna colorata derivata dalla classe penna"

Rappresentazione UML della classi:

|  |
| --- |
| PennaClasse |
| - int LivelloInchiostroNero; |
| PennaClasse();PennaClasse(int LivelloInchiostroNero);+ int GetLivelloInchiostroNero(){ return LivelloInchiostroNero; }; + void Scrivi(int CaratteriDaScrivere); |

|  |
| --- |
| PennaColorataClasse |
| - int LivelloInchiostroColorato; |
| PennaColorataClasse(int LivelloInchiostroNero,int LivelloInchiostroColorato);+ void GetLivelloInchiostroColorato() + void Scrivi(int ColorePenna,int CaratteriDaScrivere); |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

class PennaClasse

{

 private:

 int LivelloInchiostroNero;

 public:

 PennaClasse() { LivelloInchiostroNero = 2000; };

 //numero di caratteri che si possono scrivere

 PennaClasse(int LivelloInchiostroNero) { this->LivelloInchiostroNero = LivelloInchiostroNero; };

 int GetLivelloInchiostroNero(){ return LivelloInchiostroNero; };

 void Scrivi(int CaratteriDaScrivere);

};

class PennaColorataClasse : public PennaClasse

{

 private:

 int LivelloInchiostroColorato;

 public:

 PennaColorataClasse(int LivelloInchiostroNero,int LivelloInchiostroColorato) : PennaClasse(LivelloInchiostroNero)

 { this->LivelloInchiostroColorato = LivelloInchiostroColorato; };

 void GetLivelloInchiostroColorato()

 {

 cout<<"Il livello dell'inchiostro nero e': "<<PennaClasse::GetLivelloInchiostroNero()<<endl;

 cout<<"Il livello dell'inchiostro colorato e': "<<LivelloInchiostroColorato<<endl;

 };

 void Scrivi(int ColorePenna,int CaratteriDaScrivere);

};

int main()

{

 int CaratteriDaScrivere,ColorePenna;

 PennaClasse Penna;

 PennaColorataClasse PennaColorata(1000,500);

 PennaColorata.GetLivelloInchiostroColorato();

 cout<<"Vuoi usare la penna nera (0) o colorata(1)? ";

 cin>>ColorePenna;

 cout<<"Quanti caratteri vuoi scrivere? ";

 cin>>CaratteriDaScrivere;

 PennaColorata.Scrivi(ColorePenna,CaratteriDaScrivere);

 PennaColorata.GetLivelloInchiostroColorato();

 system("pause");

}

void PennaClasse::Scrivi(int CaratteriDaScrivere)

{

 if(CaratteriDaScrivere<=LivelloInchiostroNero)

 LivelloInchiostroNero=LivelloInchiostroNero-CaratteriDaScrivere;

 else

 cout<<"Impossibile scrivere i caratteri. Livello inchiostro insufficiente.";

}

void PennaColorataClasse::Scrivi(int ColorePenna,int CaratteriDaScrivere)

{

 if(ColorePenna==0)

 PennaClasse::Scrivi(CaratteriDaScrivere);

 else

 LivelloInchiostroColorato=LivelloInchiostroColorato-CaratteriDaScrivere;

}

Esercizio 57:

La classe “Motocicletta" derivata dalla classe “Motore”

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 03/03/2015

Numero es:57

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di definire le classi ed utilizzare gli oggetti in c++

Obiettivo del programma: Codifica c++ "La classe motocicletta derivata dalla classe motore"

Rappresentazione UML della classi:

|  |
| --- |
| MotoreClasse |
| - int NumGiri; |
| MotoreClasse() { NumGiri=0; };MotoreClasse(int NumGiri);+ void AvviaMotore();+ int GetNumGiri() ;+ void Accelerare(int accelerazionepar); |

|  |
| --- |
| MotociclettaClasse |
| - char marca;- char targa;- char colore; |
| MotociclettaClasse();+ void MostraStato(); |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

class MotoreClasse

{

 private:

 int NumGiri;

 public:

 MotoreClasse() { NumGiri=0; };

 MotoreClasse(int NumGiri) { this->NumGiri=NumGiri; };

 void AvviaMotore();

 int GetNumGiri() { return NumGiri; };

 void Accelerare(int accelerazionepar) { NumGiri=NumGiri+accelerazionepar; };

};

class MotociclettaClasse : public MotoreClasse

{

 private:

 char marca;

 char targa;

 char colore;

 public:

 MotociclettaClasse() : MotoreClasse()

 {

 marca='a';

 targa='b';

 colore='c';

 };

 void MostraStato();

};

int main()

{

 int scelta,accelerazione;

 char risp;

 MotociclettaClasse Motocicletta;

 do

 {

 cout<<"Cosa vuoi effettuare? "<<endl;

 cout<<"1) Mettere in moto "<<endl;

 cout<<"2) Accelerare "<<endl;

 cout<<"3) Decelerare "<<endl;

 cout<<"4) Visualizzare lo stato della moto "<<endl;

 cin>>scelta;

 switch(scelta)

 {

 case 1:

 Motocicletta.AvviaMotore();

 break;

 case 2:

 cout<<"Di quanto vuoi accelerare? ";

 cin>>accelerazione;

 Motocicletta.Accelerare(accelerazione);

 break;

 case 3:

 cout<<"Di quanto vuoi decelerare? ";

 cin>>accelerazione;

 Motocicletta.Accelerare(-accelerazione);

 break;

 case 4:

 Motocicletta.MostraStato();

 break;

 }

 cout<<"Vuoi continuare? s/n ";

 cin>>risp;

 }

 while(risp=='s' || risp=='S');

 system("pause");

}

void MotoreClasse::AvviaMotore()

{

 if(NumGiri!=0)

 cout<<"Il motore e' gia' acceso "<<endl;

 else

 {

 NumGiri=100;

 cout<<"Il motore e' ora acceso "<<endl;

 }

}

void MotociclettaClasse::MostraStato()

{

 cout<<"La moto e' una "<<marca<<" di colore "<<colore<<" con targa "<<targa<<endl;

 cout<<"Il numero di giri del motore e': "<<GetNumGiri()<<endl;

}

Esercizio 58: Pag.88 n.1 (Libro Cloud)

Esercizio 59: Pag.88 n.2 (Libro Cloud)

Esercizio 60:

La classe “Scaldabagno" derivata dalla classe “Cilindro”

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 09/03/2015

Numero es:60

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di derivare classi in c++ (ereditarietà)

Obiettivo del programma: Codifica c++ "La classe scaldabagno a forma cilindrtica" (pag.108 n.26)

Rappresentazione UML della classi:

|  |
| --- |
| CilindroClasse |
| - float raggio;- float altezza; |
| CilindroClasse(float raggio,float altezza);+ float Volume() ;+ float Area(); |

|  |
| --- |
| ScaldabagnoClasse |
| - int LivelloAcqua;- int Temperatura; |
| ScaldabagnoClasse(float raggio,float altezza,int LivelloAcqua,int Temperatura);+ void SetLivelloAcqua(int LivelloAcqua);+ int GetLivelloAcqua();+ void SetTemperatura(int Temperatura);+ int GetTemperatura(); |

\*/

#include<iostream>

#include<cstdlib>

using namespace std;

class CilindroClasse

{

 private:

 float raggio;

 float altezza;

 public:

 CilindroClasse(float raggio,float altezza)

 { this->altezza=altezza; this->raggio=raggio; };

 float Volume() { return (raggio\*raggio\*3.14\*altezza); }; //AreaCerchio\*altezza

 float Area();

};

class ScaldabagnoClasse : public CilindroClasse

{

 private:

 int LivelloAcqua;

 int Temperatura;

 public:

 ScaldabagnoClasse(float raggio,float altezza,int LivelloAcqua,int Temperatura) :

 CilindroClasse(raggio,altezza)

 { this->LivelloAcqua=LivelloAcqua; this->Temperatura=Temperatura; };

 void SetLivelloAcqua(int LivelloAcqua) { this->LivelloAcqua=LivelloAcqua; };

 int GetLivelloAcqua() { return LivelloAcqua; };

 void SetTemperatura(int Temperatura) { this->Temperatura=Temperatura; };

 int GetTemperatura() { return Temperatura; };

};

int main()

{

 ScaldabagnoClasse Scaldabagno(4,10,5,38);

 cout<<"Il volume dello scaldabagno a forma cilindrica e': "<<Scaldabagno.Volume()<<endl;

 cout<<"Il livello dell'acqua e': "<<Scaldabagno.GetLivelloAcqua()<<endl;

 cout<<"La temperatura dell'acqua e' "<<Scaldabagno.GetTemperatura()<<" gradi."<<endl;

 system("pause");

}

float CilindroClasse::Area()

{

 float AreaBase,AreaLaterale;

 AreaBase=raggio\*raggio\*3.14\*2; //AreaCerchio\*2

 AreaLaterale=raggio\*2\*3.14\*altezza; //CirconferenzaCerchio\*altezza

 return (AreaBase+AreaLaterale);

};

Esercizio 61:

Grafica in c++: animazione linea

/\*

ITIS-LS F.Giordani CaseAnno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 09/03/2015

Numero es:61

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica in c++

Obiettivo del programma: Animazione linea

\*/

#include<graphics.h> // declarations for graphics library

#include<conio.h> // for getch() function

#include<dos.h>

int main()

{

 // initialize the graphics system

 int pilotagrafico = DETECT, modografico;

 initgraph(&pilotagrafico, &modografico, "..\\bgi");

 int x1=100,x2=200,y1=100,y2=100;

 line (x1,y1,x2,y2);

 for(int i=1;i<50;i++)

 {

 setcolor(getbkcolor());

 line (x1,y1,x2,y2);

 y1=y1+5;

 y2=y2+5;

 setcolor(15);

 line (x1,y1,x2,y2);

 delay(100);

 }

 getch(); // Wait for keypress

 closegraph(); // Restore original screen

 return 0;

}

Esercizio 62:

“Grafica1” in c++

#include<graphics.h>

int main()

{

 initwindow(700,700,"MY First Program");

 circle(200,200,150);

 getch();

 return 0;

}

Esercizio 63:

“Grafica2” in c++

#include<graphics.h>

int main( )

 {

 initwindow(100,100); //apre una finestra grafica 100x100

 moveto(10,10); // inizio di creazione del primo punto

 getch();

 lineto(10,50); // crea delle linee secondo coordinate cartesiane

 getch();

 lineto(50,50);

 getch();

 lineto(50,10);

 getch();

 lineto(10,10);

 getch();

 lineto(50,50);

 getch();

 lineto(10,50);

 getch();

 lineto(50,10);

 getch();

 return 0;

}

Esercizio 64:

Grafica in c++: segmento con putpixel

#include<graphics.h>

void segmento (int x1, int y1, int x2, int y2, int col);

int main( )

 {

 initwindow(500,500); //apre una finestra grafica 100x100

 segmento(100,100,200,100,12);

 getch();

 return 0;

}

void segmento (int x1, int y1, int x2, int y2, int col)

{

double x,y;

int i;

for (x=x1,y=y1,i=0;i<1000;i++)

 {

 putpixel((int) x, (int) y, col);

 x = x + (x2-x1)/1000.0;

 y = y + (y2-y1)/1000.0;

 }

}

Esercizio 65:

Graficain c++: rettangolo con putpixel

#include<graphics.h>

void segmento (int x1, int y1, int x2, int y2, int col);

void rettangolo(int xsup, int ysup, int xinf, int yinf, int col);

int main( )

 {

 initwindow(500,500); //apre una finestra grafica 100x100

 rettangolo(100,100,300,300,10);

 getch();

 return 0;

}

void segmento (int x1, int y1, int x2, int y2, int col)

{

double x,y;

int i;

for (x=x1,y=y1,i=0;i<1000;i++)

 {

 putpixel((int) x, (int) y, col);

 x = x + (x2-x1)/1000.0;

 y = y + (y2-y1)/1000.0;

 }

}

void rettangolo(int xsup, int ysup, int xinf, int yinf, int col)

{

segmento(xsup,ysup,xinf,ysup,col);

segmento(xsup,ysup,xsup,yinf,col);

segmento(xsup,yinf,xinf,yinf,col);

segmento(xinf,ysup,xinf,yinf,col);

}

Esercizio 66:

Graficain c++: cerchio con putpixel

#include<graphics.h>

#include <math.h>

void cerchio(int x, int y, int r, int color);

int main( )

 {

 initwindow(500,500); //apre una finestra grafica 100x100

 cerchio(250,250,100,14);

 getch();

 return 0;

}

void cerchio(int x, int y, int r, int color)

{

static const double PI = 3.1415926535;

double i, angle, x1, y1;

for(i = 0; i < 360; i += 0.1)

 {

 angle = i;

 x1 = r \* cos(angle \* PI / 180);

 y1 = r \* sin(angle \* PI / 180);

 putpixel(x + x1, y + y1, color);

 }

}

Esercizio 67:

Classe scooter derivata dalla classe VeicoloAMotore

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 12/03/2015

Numero es:67

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di derivare classi in c++ (ereditarietà)

Obiettivo del programma: Classe scooter derivata dalla classe VeicoloAMotore

Rappresentazione UML della classi:

|  |
| --- |
| VeicoloAMotoreClasse |
|  |
| + int StatoMotore;+ VeicoloAMotoreClasse() + void AvviaMotore()+ void MostraStato(); |

|  |
| --- |
| ScooterClasse |
|  |
| + void MostraStato(); |

\*/

#include <iostream>

using namespace std;

class VeicoloAMotoreClasse

{

 public:

 int StatoMotore;

 VeicoloAMotoreClasse() { StatoMotore=0; };

 void AvviaMotore() { StatoMotore=1; };

 void MostraStato();

};

class ScooterClasse : public VeicoloAMotoreClasse

{

 public:

 void MostraStato();

};

int main()

{

 int x=0;

 if(x==0)

 ScooterClasse S;

 else

 VeicoloAMotoreClasse S;

 S.MostraStato();

 system("pause");

}

void VeicoloAMotoreClasse::MostraStato()

{

 cout<<"E' stato richiamato MostraStato di VeicoloAMotoreClasse"<<endl;

 if(StatoMotore==0)

 cout<<"Il motore e' spento"<<endl;

 else

 cout<<"Il motore e' acceso"<<endl;

}

void ScooterClasse::MostraStato()

{

 cout<<"E' stato richiamato MostraStato di ScooterClasse"<<endl;

 if(StatoMotore==0)

 cout<<"Il motore e' spento"<<endl;

 else

 cout<<"Il motore e' acceso"<<endl;

}

Esercizio 68:

Primo esercizio Java: l’output

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 13/03/2015

Numero es:68

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di visualizzare l’output in java;

Obiettivo del programma: Visualizzare “Quanti anni hai?”;

\*/

class ProgDomanda

{

 public static void main(String args[])

 {

 System.out.println(" Quanti anni hai? ");

 }

}

Esercizio 69:

Classe VeicoliMotore in java

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 13/03/2015

Numero es:69

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di derivare classi in java (ereditarietà)

Obiettivo del programma: Classe VeicoliMotore

Rappresentazione UML della classi:

|  |
| --- |
| VeicoloAMotoreClasse |
| - int numGiri; |
| + int StatoMotore;+ VeicoliMotoreClasse(int numGiri) + void accelera(int numGiri)+ int getNumGiri() |

|  |
| --- |
| MotoClasse |
| - String nome; |
| + MotoClasse(String nome);+ String getNome(); |

class VeicoliMotoreClasse

{

 private int numGiri;

 public VeicoliMotoreClasse(int numGiri)

 {

 this.numGiri=numGiri;

 }

 public void accelera(int numGiri)

 {

 this.numGiri=this.numGiri+numGiri;

 }

 public int getNumGiri()

 {

 return numGiri;

 }

 public class MotoClasse

 {

 private String nome;

 public MotoClasse(String nome)

 {

 this.nome=nome;

 }

 public String getNome()

 {

 return nome;

 }

 }

}

class ProgVeicoliMotore

{

 public static void main(String argv[])

 {

 VeicoliMotoreClasse VeicoliMotore =new VeicoliMotoreClasse(10);

 VeicoliMotoreClasse.MotoClasse Moto =VeicoliMotore.new MotoClasse("Guzzi");

 System.out.println("Nome: " + Moto.getNome());

 System.out.println("Numero di giri del motore: " + VeicoliMotore.getNumGiri());

 }

}

Esercizio 70:

Classe Cerchio in java

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 16/03/2015

Numero es:70

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di derivare classi in java (ereditarietà)

Obiettivo del programma: Classe Cerchio in java

Rappresentazione UML della classi:

|  |
| --- |
| CerchioClasse |
| - float raggio; |
| CerchioClasse(float raggio) + void SetRaggio(float raggio) + void GetRaggio(float raggio) + float Area()+ float Circonferenza(); |

class CerchioClasse

{

 private double raggio;

 public CerchioClasse()

 {

 raggio=0;

 }

 public CerchioClasse(double raggio)

 {

 this.raggio=raggio;

 }

 public double Area()

 {

 return(raggio\*raggio\*3.14);

 }

 public double Circonferenza()

 {

 return(2\*3.14\*raggio);

 }

 public void SetRaggio(double raggio)

 {

 this.raggio=raggio;

 }

 public double GetRaggio()

 {

 return(raggio);

 }

}

class ProgCerchio

{

 public static void main(String argv[])

 {

 CerchioClasse Cerchio = new CerchioClasse(4);

 System.out.println("Il raggio e': "+Cerchio.GetRaggio());

 System.out.println("L'area e': "+Cerchio.Area());

 System.out.println("L'area e': "+Cerchio.Circonferenza());

 }

}

Esercizio 71:

Calcolatrice in VisualBasic

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 23/03/2015

Numero es:71

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di derivare classi in java (ereditarietà)

Obiettivo del programma: Calcolatrice in VisualBasic

\*/

Private Sub cmdSomma\_Click()

a = Val(txtAddendo1.Text)

b = Val(txtAddendo2.Text)

c = Str$(a + b)

txtRisultato.Text = c

End Sub

Esercizio 72:

Parametri del main in java

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 23/03/2015

Numero es:72

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Parametri del main in java

\*/

class ProgSemplice

{

 public static void main(String argv[])

 {

 System.out.print("Ciao mondo, sono il primo programma in Java ");

 System.out.println("di "+ argv[0] +" "+ argv[1]);

 }

}

Esercizio 73:

Leggi stringa in input in java

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 23/03/2015

Numero es:73

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Leggi stringa in input in java

\*/

class ProgLeggiStringa

{

 public static void main(String argv[])

 {

 String str;

 InputStreamReader mioInput = new InputStreamReader(System.in);

 BufferedReader miaTastiera = new BufferedReader(mioInput);

 str=miaTastiera.readLine();

 System.out.println("questa è la stringa letta: "+str);

 }

}

Esercizio 74:

Controlli VisualBasic

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 30/03/2015

Numero es:74

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Controlli VisualBasic

\*/

Private Sub chkCorsivo\_Click()

lblMessaggio.Font.Italic = chkCorsivo.Value

End Sub

Private Sub chkGrassetto\_Click()

lblMessaggio.Font.Bold = chkGrassetto.Value

End Sub

Private Sub chkSottolineato\_Click()

lblMessaggio.Font.Underline = chkSottolineato.Value

End Sub

Private Sub cmdAzzera\_Click()

lblMessaggio.Caption = ""

txtNome.Text = ""

imgGiallo.Visible = False

imgVerde.Visible = False

imgRosso.Visible = False

imgSpento.Visible = False

optRosso.Value = False

optGiallo.Value = False

optVerde.Value = False

optSpento.Value = False

chkGrassetto.Value = Unchecked

chkCorsivo.Value = Unchecked

chkSottolineato.Value = Unchecked

txtNome.SetFocus

End Sub

Private Sub cmdEsci\_Click()

End

End Sub

Private Sub cmdStampa\_Click()

PrintForm

End Sub

Private Sub optGiallo\_Click()

imgGiallo.Visible = True

imgVerde.Visible = False

imgRosso.Visible = False

imgSpento.Visible = False

lblMessaggio.Caption = "Questo è il semaforo giallo, " & txtNome.Text

End Sub

Private Sub optRosso\_Click()

imgGiallo.Visible = False

imgVerde.Visible = False

imgRosso.Visible = True

imgSpento.Visible = False

lblMessaggio.Caption = "Questo è il semaforo rosso, " & txtNome.Text

End Sub

Private Sub optSpento\_Click()

imgGiallo.Visible = False

imgVerde.Visible = False

imgRosso.Visible = False

imgSpento.Visible = True

lblMessaggio.Caption = "Questo è il semaforo spento, " & txtNome.Text

End Sub

Private Sub optVerde\_Click()

imgGiallo.Visible = False

imgVerde.Visible = True

imgRosso.Visible = False

imgSpento.Visible = False

lblMessaggio.Caption = "Questo è il semaforo verde, " & txtNome.Text

End Sub

Esercizio 75:

Timer in VisualBasic

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 30/03/2015

Numero es:75

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Timer in VisualBasic

\*/

Option Explicit

Dim x, y As Integer

Private Sub Form\_Load()

Timer1.Interval = 1000

x = 100

y = 100

End Sub

Private Sub Timer1\_Timer()

Image1.Move Image1.Left + x, Image1.Top + y

End Sub

Esercizio 76:

Line in VisualBasic

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 30/03/2015

Numero es:76

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Line in VisualBasic

\*/

Private Sub Form\_Load()

 Line1.X1 = 0

 Line1.X2 = Form1.Width

 Line1.Y1 = Form1.Height \ 2

 Line1.Y2 = Form1.Height \ 2

End Sub

Esercizio 77:

Hscroll in VisualBasic

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 30/03/2015

Numero es:77

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: HScroll in VisualBasic

\*/

Private Sub Form\_Load()

HScroll1.Min = 0

HScroll1.Max = 1000

HScroll1.LargeChange = 100

 'largechange regola lo spostamento da effettuare

 'se l'utente fa click tra freccia e cursore

HScroll1.SmallChange = 1

End Sub

Private Sub HScroll1\_Change()

Text1.Text = HScroll1.Value

End Sub

Esercizio 78:

Primo esercizio in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:78

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Primo esercizio in Javascript

-->

<html>

 <head>

 <title>Ciao a Tutti (title) </title>

 <SCRIPT language=Javascript>

 document.write("<h1>Ciao a Tutti! </h1>");

 </SCRIPT>

 </head>

<body></body>

</html>

Esercizio 79:

Somma in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:79

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Somma in Javascript

-->

<html>

 <head>

 <title>Somma di due numeri</title>

 </head>

<body>

<form name="somma">

<input name="operando1" >+<input name="operando2" >

=<input name="risultato" value="0">

<input type="button" value="Fai la somma!" onClick=

"somma.risultato.value=Number(somma.operando1.value)+ Number(somma.operando2.value)">

</form>

</body>

</html>

Esercizio 80:

Media in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:80

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Media in Javascript

-->

<html>

 <head>

 <title>Calcola la media di una serie di numeri</title>

 <script language=javascript>

 function calcolamedia(lista)

 {

 dati = lista.split(" ")

 if (dati.length > 0)

 {

 var somma = 0;

 for (var i = 0; i < dati.length; i++)

 {

 somma = somma + Number(dati[i])

 }

 return (somma / dati.length);

 }

 }

 </script>

</head>

<body>

<form name="media">

Lista di valori

<input name="valori">

<br>Risultato

<input name="risultato">

<input type=button value="Calcola la media!"

onClick="media.risultato.value=calcolamedia(media.valori.value)">

</form>

</body>

</html>

Esercizio 81:

onMouseOver in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:81

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: onMouseOver in Javascript

-->

<html>

<head>

<script language="JavaScript">

 function ciao()

 { alert("Ciao!"); }

</script>

</head>

<body>

<a href="" onMouseOver="ciao()">link</a>

</body>

</html>

Esercizio 82:

Data in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:82

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Data in Javascript

-->

<html>

<head>

<script language="JavaScript">

 oggi = new Date()

 document.write("L'ora attuale è: ",oggi.getHours(),":",oggi.getMinutes())

 document.write("La data di oggi è: ", oggi.getDate(),"/",oggi.getMonth()+1,"/",oggi.getYear());

</script>

</head>

<body>

</body>

</html>

Esercizio 83:

ArrayMix in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:83

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma:ArrayMix in Javascript

-->

<html>

<head>

<script language="JavaScript">

 ArrayMix= new Array(17,"Ehi",103);

 document.write("Element No. 1: "+ArrayMix[0]+"<br>");

 document.write("Element No. 2: "+ArrayMix[1]+"<br>");

 document.write("Element No. 3: "+ArrayMix[2]+"<br>");

</script>

</head>

<body>

</body>

</html>

Esercizio 84:

Vettori paralleli in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:84

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Vettori paralleli in Javascript

-->

<html>

<table border="1">

<tr>

 <td><B>Voti</B></td>

 <td><B>Alunni</B></td>

</tr>

 <script language="JavaScript">

 alunni =new Array("Aldo","Giovanni","Giacomo","Mario","Gianni","Monica");

 voti= new Array(3,8,5,7,4,4); // a ogni nome corrisponde un voto

 i=0;

 while(i<alunni.length)

 {

 document.write("<tr>");

 document.write("<td>"+alunni[i]+"</td>");

 document.write("<td>"+voti[i]+"</td>");

 document.write("</tr>");

 i++;

 }

</script>

</table>

</html>

Esercizio 85:

Vettori associativi in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:85

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Vettori associativi in Javascript

-->

<html>

 <script language="JavaScript">

 var o = new Object();

 o["one"] = 1;

 o["two"] = 2;

 o["three"] = 3;

for(var i in o)

{

 document.write(i + "=" + o[i] + '<br>');

}

</script>

</html>

Esercizio 86:

Calcolatrice in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:86

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Calcolatrice in Javascript

-->

<!doctype html>

<html lang="it">

<head>

<script language="javascript" src="calcola.js">

</script>

</head>

<title>Calcolatrice</title>

<body>

<table cellspacing="0" cellpadding="0">

<tr>

<td height="40" colspan="4" align="right">

<input type="text" id="display" style="width:80px; height:20px;">

</td>

</tr>

<tr>

<td><input type="button" value="7" style="width:20px; height:20px;" onclick="document.getElementById('display').value+=7"> </td>

<td><input type="button" value="8" style="width:20px; height:20px;" onclick="document.getElementById('display').value+=8"> </td>

<td><input type="button" value="9" style="width:20px; height:20px;" onclick="document.getElementById('display').value+=9"> </td>

<td><input type="button" value="/" style="width:20px; height:20px;" onclick="diviso()"> </td>

</tr>

<tr>

<td><input type="button" value="4" style="width:20px; height:20px;" onclick="document.getElementById('display').value+=4"> </td>

<td><input type="button" value="5" style="width:20px; height:20px;" onclick="document.getElementById('display').value+=5"> </td>

<td><input type="button" value="6" style="width:20px; height:20px;" onclick="document.getElementById('display').value+=6"> </td>

<td><input type="button" value="x" style="width:20px; height:20px;" onclick="per()"> </td>

</tr>

<tr>

<td><input type="button" value="1" style="width:20px; height:20px;" onclick="document.getElementById('display').value+=1"> </td>

<td><input type="button" value="2" style="width:20px; height:20px;" onclick="document.getElementById('display').value+=2"> </td>

<td><input type="button" value="3" style="width:20px; height:20px;" onclick="document.getElementById('display').value+=3"> </td>

<td><input type="button" value="-" style="width:20px; height:20px;" onclick="meno()"> </td>

</tr>

<tr>

<td><input type="button" value="0" style="width:20px; height:20px;" onclick="document.getElementById('display').value+=0"> </td>

<td><input type="button" value="C" style="width:20px; height:20px;" onclick="document.getElementById('display').value=' ';val1=0;"> </td>

<td><input type="button" value="=" style="width:20px; height:20px;" onclick="uguale()"> </td>

<td><input type="button" value="+" style="width:20px; height:20px;" onclick="piu()"> </td>

</tr>

</table>

</body>

</html>

<!-- File Javascript -->

function piu(){

val1=eval(document.getElementById("display").value);

op="piu";

document.getElementById("display").value=' ';

}

function meno(){

val1=eval(document.getElementById("display").value);

op="meno";

document.getElementById("display").value=' ';

}

function per(){

val1=eval(document.getElementById("display").value);

op="per";

document.getElementById("display").value=' ';

}

function diviso(){

val1=eval(document.getElementById("display").value);

op="diviso";

document.getElementById("display").value=' ';

}

function uguale(){

val2=eval(document.getElementById("display").value);

switch(op){

case "diviso": risultato=val1/val2;

break;

case "per": risultato=val1\*val2;

break;

case "piu": risultato=val1+val2;

break;

case "meno": risultato=val1-val2;

break;

default: risultato="errore";

}

document.getElementById("display").value= risultato;

}

Esercizio 87:

Quiz in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:87

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Quiz in Javascript

-->

<!doctype html>

<html lang="it">

<head>

<title> Quiz </title>

<script type="text/javascript">

//Questa funzione avvisa che la risposta è errata

function Errore(){

 alert("Risposta sbagliata, riprova! ");

}

//Questa funzione avvisa che la risposta è corretta

function Esatto(){

 alert("Risposta esatta, bene! ");

}

</script>

</head>

<body>

<form>

 <h2> Come si calcola l'area del triangolo ? </h2>

 <p>

 <input type="radio" value="1" name="q1" onclick="Errore()"> base x altezza <br />

 <input type="radio" value="2" name="q1" onclick="Esatto()"> base x altezza:2 <br />

 <input type="radio" value="3" name="q1" onclick="Errore()"> base + altezza <br />

 <input type="radio" value="4" name="q1" onclick="Errore()"> base + altezza x 2

 </p>

</form>

</body>

</html>

Esercizio 88:

Classi in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:88

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L’alunno è in grado di definire ed utilizzare le classi in javascript;

Obiettivo del programma: Classi in Javascript

-->

<html>

 <script language="JavaScript">

 function Persona(nome, cognome)

 {

 this.nome = nome;

 this.cognome = cognome;

 }

 Persona.prototype.getNomeCognome = function()

 {

 return this.nome + ' ' + this.cognome;

 };

 var personaObj = new Persona("A", "B");

 document.write(personaObj.getNomeCognome());

 </script>

</html>

Esercizio 89:

Classe derivata in Javascript

<!--

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 06/04/2015

Numero es:88

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L’alunno è in grado di utilizzare l’ereditarietà in javascript;

Obiettivo del programma: Classe derivata in Javascript

-->

<html>

 <script language="JavaScript">

// definizione della classe Persona

function Persona(sesso)

 {

 this.sesso = sesso;

 }

Persona.prototype.getSesso = function(){

 return(this.sesso);

};

Persona.prototype.cammina = function(){

 document.write('Sto camminando!');

};

Persona.prototype.saluta = function(){

 document.write('ciao');

};

// definizione della classe Studente

function Studente() {

 // Chiama il costruttore della classe Persona

 Persona.call(this);

}

// eredita Persona

Studente.prototype = new Persona();

// corregge il puntatore del costruttore perchè punta a Persona

Studente.prototype.constructor = Studente;

// riscrive il metodo saluta

Studente.prototype.saluta = function(){

 alert('ciao, sono uno studente');

}

// aggiunge il metodo arrivederci

Studente.prototype.arrivederci = function(){

 alert('arrivederci');

}

var personaObj = new Persona("maschio");

document.write(personaObj.getSesso());

document.write('<br>');

personaObj.cammina();

var studenteObj = new Studente("maschio");

studenteObj.saluta();

studenteObj.cammina();

studenteObj.arrivederci();

// test ereditarietà

alert(studenteObj instanceof Persona); // true

alert(studenteObj instanceof Studente); // true

</script>

</html>

Esercizio 90:

Classe Cilindro in java

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 10/04/2015

Numero es:90

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di derivare classi in java (ereditarietà)

Obiettivo del programma: Classe Cilindro in java

Rappresentazione UML della classi:

|  |
| --- |
| CerchioClasse |
| - float raggio; |
| CerchioClasse(float raggio) + void SetRaggio(float raggio) + void GetRaggio(float raggio) + float Area()+ float Circonferenza(); |

|  |
| --- |
| CilindroClasse |
| - float altezza; |
| CilindroClasse(float raggio,float altezza);+ void SetAltezza(float altezza);+ float Volume();+ float Area(); |

\*/

class Cerchio

{

 private double raggio;

 public Cerchio(double raggio)

 {

 this.raggio=raggio;

 }

 public void setRaggio(double raggio)

 {

 this.raggio=raggio;

 }

 public double area()

 {

 return (raggio\*raggio\*3.14);

 }

 public double circonferenza()

 {

 return (raggio\*2\*3.14);

 }

}

class Cilindro extends Cerchio

{

 private double altezza;

 public Cilindro(double raggio, double altezza)

 {

 super(raggio);

 this.altezza=altezza;

 }

 public void setAltezza(double altezza)

 {

 this.altezza=altezza;

 }

 public double volume()

 {

 return (super.area()\*altezza);

 }

 public double area()

 {

 double AreaBase,AreaLaterale;

 AreaBase=super.area()\*2;

 AreaLaterale=circonferenza()\*altezza;

 return (AreaBase+AreaLaterale);

 }

}

class ProgCilindro

{

 public static void main(String args[])

 {

 Cilindro cilindroObj=new Cilindro(4,10);

 System.out.println("L'area del cilindro e': "+cilindroObj.area());

 System.out.println("Il volume del cilindro e': "+cilindroObj.volume());

 }

}

Esercizio 91:

Calcolare la media (while) di una matrice in java

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 13/04/2015

Numero es:91

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Calcolare la media (while) di una matrice in java

\*/

//temperature.length restituisce il numero di righe

//temperature[0].length restituisce il numero di colonne

import java.io.\*;

class ProgCalcolaMediaw

{

 public static void main(String argv[])

 {

 double[][]temperature = new double [4][3];

 InputStreamReader mioInput = new InputStreamReader(System.in);

 BufferedReader miaTastiera = new BufferedReader(mioInput);

 int i=0,j=0;

 double somma=0,media=0;

 try

 {

 while(i<temperature.length)

 {

 j=0;

 while(j<temperature[0].length)

 {

 System.out.print("Elemento di posizione ["+ i+","+j+"]: ");

 temperature[i][j]=Double.parseDouble(miaTastiera.readLine());

 j++;

 }

 i++;

 } //fine caricamento

 i=0;

 while(i<temperature.length)

 {

 j=0;

 while(j<temperature[0].length)

 {

 somma=somma+temperature[i][j];

 j++;

 }

 i++;

 }

 media=somma/(temperature.length\*temperature[0].length);

 System.out.print("\n La media e': "+media);

 }

 catch(Exception e)

 {

 System.out.println("Si e' verificata un'eccezione");

 }

 }

}

Esercizio 92:

Conta vocali di una stringa in java

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 13/04/2015

Numero es:92

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Conta vocali di una stringa in java

\*/

//s.substring(i,j) seleziona una sottostringa formata dai caratteri da i a j

//vocali.indexOf(char c) restituisce la posizione del carattere c;

//se il carattere c e' presente piu' volte restituisce la prima posizione

import java.io.\*;

class ProgContaVocali

{

 public static void main(String argv[])

 {

 String s="ciao buongiorno a tutti";

 String vocali="aAeEiIoOuU";

 int NumVocali=0;

 for(int i=0;i<s.length();i++)

 {

 if(vocali.indexOf(s.substring(i,i+1)) != -1)

 NumVocali++;

 }

 System.out.println("Il numero di vocali e': "+NumVocali);

 }

Esercizio 93:

Calcolare la media (for) di una matrice in java

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 14/04/2015

Numero es:93

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico:

Obiettivo del programma: Calcolare la media (while) di una matrice in java

\*/

//temperature.length restituisce il numero di righe

//temperature[0].length restituisce il numero di colonne

import java.io.\*;

class ProgCalcolaMediaw

{

 public static void main(String argv[])

 {

 double[][]temperature = new double [4][3];

 InputStreamReader mioInput = new InputStreamReader(System.in);

 BufferedReader miaTastiera = new BufferedReader(mioInput);

 int i=0,j=0;

 double somma=0,media=0;

 try

 {

 while(i<temperature.length)

 {

 j=0;

 while(j<temperature[0].length)

 {

 System.out.print("Elemento di posizione ["+ i+","+j+"]: ");

 temperature[i][j]=Double.parseDouble(miaTastiera.readLine());

 j++;

 }

 i++;

 } //fine caricamento

 i=0;

 while(i<temperature.length)

 {

 j=0;

 while(j<temperature[0].length)

 {

 somma=somma+temperature[i][j];

 j++;

 }

 i++;

 }

 media=somma/(temperature.length\*temperature[0].length);

 System.out.print("\n La media e': "+media);

 }

 catch(Exception e)

 {

 System.out.println("Si e' verificata un'eccezione");

 }

 }

}

Esercizio 94:

Classe vettore in java

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 16/04/2015

Numero es:94

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare le classi in java

Obiettivo del programma: Classe Vettore in java

Rappresentazione UML della classi:

|  |
| --- |
| VettoreClasse |
| - int vet[];- int dimlog; |
| VettoreClasse() ;VettoreClasse(int dimlog);+ void CaricaVet() + void StampaVet() + double Media() |

\*/

import java.io.\*;

class VettoreClasse

{

 private int vet[];

 private int dimlog;

 public VettoreClasse()

 {

 vet‑ = new int(dimlog);

 dimlog=0;

 vet‑[] = { 0,0,0,0,0,0,0,0,0,0 }; //inizializzazione vettore

 }

 public VettoreClasse(int dimlog)

 {

 vet‑ = new int(dimlog);

 this.dimlog=dimlog;

 vet‑[] = { 0,0,0,0,0,0,0,0,0,0 }; //inizializzazione vettore

 }

 public void CaricaVet()

 {

 InputStreamReader mioInput = new InputStreamReader(System.in);

 BufferedReader miaTastiera = new BufferedReader(mioInput);

 try

 {

 for(int i=0;i<dimlog;i++)

 {

 System.out.print("Inserire un numero: ");

 vet[i] = Integer.parseInt(miaTastiera.readLine());

 }

 }

 catch(Exception e)

 {

 System.out.println("Si e' verificata un'eccezione");

 }

 }

 public void StampaVet()

 {

 try

 {

 for(int i=0;i<dimlog;i++)

 System.out.print(vet[i]+" ");

 }

 catch(Exception e)

 {

 System.out.println("Si e' verificata un'eccezione");

 }

 }

 public double Media()

 {

 int somma=0,cont=0;

 double media=0;

 try

 {

 for(int i=0;i<dimlog;i++)

 {

 somma=somma+vet[i];

 cont++;

 }

 media=somma/cont;

 }

 catch(Exception e)

 {

 System.out.println("Si e' verificata un'eccezione");

 }

 return media;

 }

}

class ProgVettore

{

 public static void main(String argv[])

 {

 int dim=0; //senza quest'inizializzazione dà errore

 InputStreamReader mioInput = new InputStreamReader(System.in);

 BufferedReader miaTastiera = new BufferedReader(mioInput);

 System.out.print("Quanti elementi vuoi inserire? ");

 try

 {

 dim = Integer.parseInt(miaTastiera.readLine());

 }

 catch (Exception e)

 {

 System.out.println("Si e' verificata un'eccezione");

 }

 VettoreClasse Vettore = new VettoreClasse(dim);

 Vettore.CaricaVet();

 System.out.print("Ecco il contenuto del vettore: ");

 Vettore.StampaVet();

 System.out.println("");

 System.out.print("La media e': "+Vettore.Media());

 }

}

Esercizio 95:

La grafica in Java: Finestra e label (AWT)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 20/04/2015

Numero es:95

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica in java

Obiettivo del programma: La grafica in Java: Finestra e label (AWT)

\*/

import javax.swing.\*;

import java.awt.\*;

public class ProgLabel

{

 public static void main(String argv[])

 {

 Label etichetta = new Label("Ciao mondo!");

 Frame f = new Frame("Primo esempio");

 f.add(etichetta);

 f.setVisible(true);

 f.setSize(600,200);

 f.setLocation(100,200);

 }

}

Esercizio 96:

La grafica in Java: Jlabel e JButton (Swing)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 20/04/2015

Numero es:96

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica in java

Obiettivo del programma: La grafica in Java: Jlabel e JButton (Swing)

\*/

import javax.swing.\*;

import java.awt.\*;

class ProgJButton

{

 public static void main (String args[])

 {

 JFrame FrameObj=new JFrame();

 JPanel PanelObj=new JPanel();

 JLabel LabelObj=new JLabel("Etichetta");

 JButton ButtonObj=new JButton("Bottone");

 Container ContainerObj=FrameObj.getContentPane();

 ContainerObj.add(PanelObj);

 PanelObj.add(LabelObj);

 PanelObj.add(ButtonObj);

 FrameObj.setSize(300,200);

 FrameObj.setVisible(true);

 }

}

Esercizio 97:

La grafica in Java: I tre costruttori della label (Swing)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 20/04/2015

Numero es:97

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica in java

Obiettivo del programma: La grafica in Java: I tre costruttori della label (Swing)

\*/

import javax.swing.\*;

import java.awt.\*;

class ProgCostruttoriLabel

{

 public static void main(String args[])

 {

 JFrame FrameObj = new JFrame();

 JPanel PanelObj = new JPanel();

 JLabel LabelObj = new JLabel();

 JLabel LabelObj2 = new JLabel("Etichetta");

 JLabel LabelObj3 = new JLabel("Etichetta2",10);

 JButton ButtonObj = new JButton("OK");

 PanelObj.add(LabelObj);

 PanelObj.add(LabelObj2);

 PanelObj.add(LabelObj3);

 PanelObj.add(ButtonObj);

 Container ContainerObj = FrameObj.getContentPane();

 ContainerObj.add(PanelObj);

 FrameObj.setSize(640,480);

 FrameObj.setVisible(true);

 }

}

Esercizio 98:

La grafica in Java: SetEnabled (Swing)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 27/04/2015

Numero es:98

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica in java

Obiettivo del programma: La grafica in Java: SetEnabled (Swing)

\*/

import javax.swing.\*;

import java.awt.\*;

class ProgSetEnabled

{

 public static void main(String args[])

 {

 JFrame FrameObj = new JFrame();

 JPanel PanelObj = new JPanel();

 JLabel LabelObj = new JLabel("Test bottone: ");

 JButton ButtonObj = new JButton("OK");

 PanelObj.add(LabelObj);

 PanelObj.add(ButtonObj);

 ButtonObj.setEnabled(false);

 Container ContainerObj = FrameObj.getContentPane();

 ContainerObj.add(PanelObj);

 FrameObj.setSize(300,200);

 FrameObj.setVisible(true);

 }

}

Esercizio 99:

La grafica in Java: Casella e area di testo (Swing)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 27/04/2015

Numero es:99

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica in java

Obiettivo del programma: La grafica in Java: Casella e area di testo (Swing)

\*/

import javax.swing.\*;

import java.awt.\*;

class ProgCasellaAreaTesto

{

 public static void main (String args[])

 {

 JFrame FrameObj=new JFrame();

 JPanel PanelObj=new JPanel();

 JTextField TextObj0=new JTextField();

 JTextField TextObj=new JTextField("Casella di testo");

 JTextField TextObj2=new JTextField("Casella di testo 2",15);

 TextObj.setEditable(true);

 TextObj2.setEditable(false);

 TextObj2.setText("Casella di testo 2 modificata");

 PanelObj.add(TextObj0);

 PanelObj.add(TextObj);

 PanelObj.add(TextObj2);

 JTextArea TextAreaObj=new JTextArea("Area di testo");

 JTextArea TextAreaObj2=new JTextArea("Area di testo2",2,10);

 PanelObj.add(TextAreaObj);

 PanelObj.add(TextAreaObj2);

 Container ContainerObj=FrameObj.getContentPane();

 ContainerObj.add(PanelObj);

 FrameObj.setSize(700,400);

 FrameObj.setVisible(true);

 }

}

Esercizio 100:

La grafica in Java: Componenti (Swing)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 04/05/2015

Numero es:100

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica in java

Obiettivo del programma: La grafica in Java: Componenti (Swing)

\*/

import java.awt.\*;

import javax.swing.\*;

class ProgComponenti

{

 public static void main(String[] args)

 {

 JFrame frmF = new JFrame("Ecco i prncipali componenti della GUI");

 JPanel pnlP = new JPanel();

 JLabel lblEtichetta = new JLabel("Questa è un'etichetta");

 JButton btnPulsante = new JButton("Questo è un pulsante");

 JTextField txtCasellaTesto = new JTextField("inserire qui il testo", 30);

 JTextArea txtAreaTesto = new JTextArea("inserire in questa area il testo", 4,30);

 JCheckBox chkCheck = new JCheckBox("scelta già spuntata",true);

 ButtonGroup radioGroup = new ButtonGroup();

 JRadioButton rad1 = new JRadioButton("1 scelta spuntata",true);

 JRadioButton rad2 = new JRadioButton("2 scelta non spuntata",false);

 radioGroup.add(rad1);

 radioGroup.add(rad2);

 Choice cboC = new Choice();

 cboC.addItem("prima scelta");

 cboC.addItem("seconda scelta");

 pnlP.add(lblEtichetta);

 pnlP.add(btnPulsante);

 pnlP.add(txtCasellaTesto);

 pnlP.add(txtAreaTesto);

 pnlP.add(chkCheck);

 pnlP.add(chkC1);

 pnlP.add(chkC2);

 pnlP.add(cboC);

 Container cntC = frmF.getContentPane();

 cntC.add(pnlP);

 frmF.setSize(600,500);

 frmF.setLocation(200,200);

 frmF.setVisible(true);

 }

}

Esercizio 101:

Pulsante con evento in java (Swing)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 04/05/2015

Numero es:101

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica e gestire gli eventi in java

Obiettivo del programma: Pulsante con evento in java (Swing)

\*/

import java.awt.\*;

import javax.swing.\*;

class ProgComponenti

{

 public static void main(String[] args)

 {

 JFrame frmF = new JFrame("Ecco i prncipali componenti della GUI");

 JPanel pnlP = new JPanel();

 JLabel lblEtichetta = new JLabel("Questa è un'etichetta");

 JButton btnPulsante = new JButton("Questo è un pulsante");

 JTextField txtCasellaTesto = new JTextField("inserire qui il testo", 30);

 JTextArea txtAreaTesto = new JTextArea("inserire in questa area il testo", 4,30);

 JCheckBox chkCheck = new JCheckBox("scelta già spuntata",true);

 ButtonGroup radioGroup = new ButtonGroup();

 JRadioButton chkC1 = new JRadioButton("1 scelta spuntata",true);

 JRadioButton chkC2 = new JRadioButton("2 scelta non spuntata",false);

 radioGroup.add(chkC1);

 radioGroup.add(chkC2);

 Choice cboC = new Choice();

 cboC.addItem("prima scelta");

 cboC.addItem("seconda scelta");

 pnlP.add(lblEtichetta);

 pnlP.add(btnPulsante);

 pnlP.add(txtCasellaTesto);

 pnlP.add(txtAreaTesto);

 pnlP.add(chkCheck);

 pnlP.add(chkC1);

 pnlP.add(chkC2);

 pnlP.add(cboC);

 Container cntC = frmF.getContentPane();

 cntC.add(pnlP);

 frmF.setSize(600,500);

 frmF.setLocation(200,200);

 frmF.setVisible(true);

 }

}

Esercizio 102:

Pulsanti con evento in java (Swing)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 11/05/2015

Numero es:102

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica e gestire gli eventi in java

Obiettivo del programma: Pulsanti con evento in java (Swing)

\*/

import java.awt.Graphics;

import java.awt.Color;

import java.awt.\*;

import java.awt.event.\*;

class PulsantiConEvento

{

 TextField txtT;

 public PulsantiConEvento()

 {

 Frame frmF = new Frame("Pulsanti con Evento");

 txtT = new TextField(30);

 Button btnB1 = new Button("Premi qui");

 Button btnB2 = new Button("Chiudi applicazione");

 frmF.setLayout(new FlowLayout());

 btnB1.addActionListener(new ascoltaPulsante1());

 btnB2.addActionListener(new ascoltaPulsante2());

 frmF.add(txtT);

 frmF.add(btnB1);

 frmF.add(btnB2);

 frmF.pack();

 frmF.setVisible(true);

 }

 private class ascoltaPulsante1 implements ActionListener

 {

 public void actionPerformed(ActionEvent e)

 {

 txtT.setText("Ciao anche a te");

 }

 }

 private class ascoltaPulsante2 implements ActionListener

 {

 public void actionPerformed(ActionEvent e)

 {

 System.exit(0);

 }

 }

}

public class ProgPulsantiConEvento

{

 public static void main(String args[])

 {

 PulsantiConEvento PulsantiConEventoObj = new PulsantiConEvento();

 }

}

Esercizio 103:

Pulsanti con evento in java (stesso ascoltatore) (Swing)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 11/05/2015

Numero es:103

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica e gestire gli eventi in java

Obiettivo del programma: Pulsanti con evento in java (stesso ascoltatore) (Swing)

\*/

import java.awt.Graphics;

import java.awt.Color;

import java.awt.\*;

import java.awt.event.\*;

class PulsantiConEvento

{

 TextField txtT;

 public PulsantiConEvento()

 {

 Frame frmF = new Frame("Pulsanti con Evento");

 txtT = new TextField(30);

 Button btnB1 = new Button("Premi qui");

 Button btnB2 = new Button("Chiudi applicazione");

 frmF.setLayout(new FlowLayout());

 btnB1.addActionListener(new ascoltaPulsante1());

 btnB2.addActionListener(new ascoltaPulsante2());

 frmF.add(txtT);

 frmF.add(btnB1);

 frmF.add(btnB2);

 frmF.pack();

 frmF.setVisible(true);

 }

 private class ascoltaPulsante1 implements ActionListener

 {

 public void actionPerformed(ActionEvent e)

 {

 txtT.setText("Ciao anche a te");

 }

 }

 private class ascoltaPulsante2 implements ActionListener

 {

 public void actionPerformed(ActionEvent e)

 {

 System.exit(0);

 }

 }

}

public class ProgPulsantiConEvento

{

 public static void main(String args[])

 {

 PulsantiConEvento PulsantiConEventoObj = new PulsantiConEvento();

 }

}

Esercizio 104:

Pallino (eventi del mouse in java) (Swing)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 11/05/2015

Numero es:104

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica e gestire gli eventi in java

Obiettivo del programma: Pallino (eventi del mouse in java) (Swing)

\*/

import javax.swing.\*;

import java.awt.Graphics;

import java.awt.Color;

import java.awt.event.\*;

import java.awt.\*;

class Pallino extends JFrame

{

 int x=0;

 int y=0;

 MiaClasseAscoltoMouse a1;

 public Pallino()

 {

 a1 = new MiaClasseAscoltoMouse();

 addMouseListener(a1);

 }

 private class MiaClasseAscoltoMouse implements MouseListener

 {

 public void mousePressed(MouseEvent e)

 {

 x = e.getX();

 y = e.getY();

 repaint();

 }

 public void mouseReleased(MouseEvent e) { }

 public void mouseEntered(MouseEvent e) { }

 public void mouseExited(MouseEvent e) { }

 public void mouseClicked(MouseEvent e) { }

 }

 public void paint(Graphics g)

 {

 super.paint(g);

 g.setColor(Color.red);

 g.fillOval(x,y,20,20);

 }

}

public class ProgPallino

{

 public static void main(String args[])

 {

 JFrame frmF = new Pallino();

 frmF.setSize(300,200);

 frmF.setLocation(50,50);

 frmF.setVisible(true);

 }

}

Esercizio 105:

Primo esercizio con NetBeans

Esercizio 106:

Il costruttore di JLabel(String,int) (Swing)

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 15/05/2015

Numero es:106

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica in java

Obiettivo del programma: Il costruttore di JLabel(String,int) (Swing)

\*/

import javax.swing.\*;

import java.awt.\*;

class ProgLabel

{

 public static void main(String args[])

 {

 JFrame FrameObj = new JFrame();

 JPanel PanelObj = new JPanel();

 JLabel LabelObj = new JLabel("Etichetta",10);

 PanelObj.add(LabelObj);

 Container ContainerObj = FrameObj.getContentPane();

 ContainerObj.add(PanelObj);

 FrameObj.setSize(640,480);

 FrameObj.setVisible(true);

 }

}

Esercizio 107:

Le applet in java

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 15/05/2015

Numero es:107

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare le applet di java

Obiettivo del programma: Le applet in java

\*/

import java.applet.\*;

import javax.swing.\*;

import java.awt.\*;

public class CiaoMondo extends JApplet

{

 public void init()

 {

 Container sfondo=getContentPane();

 sfondo.setLayout(new FlowLayout());

 JLabel miaLabel=new JLabel("Ciao, sono la vostra prima applet!");

 sfondo.add(miaLabel);

 }

}

Esercizio 108:

Realizzare una form in cui sia possibile convertire una temperatura da gradi Celsius a gradi Fahrenheit.

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 15/05/2015

Numero es:108

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica e gestire gli eventi in java

Obiettivo del programma: Realizzare una form in cui sia possibile convertire una temperatura da gradi Celsius a gradi Fahrenheit.

\*/

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class ConvertiFrame extends JFrame implements ActionListener

{

 private JPanel p1 = new JPanel();

 private JPanel p2 = new JPanel();

 private JTextField txtCentigradi = new JTextField(15);

 private JTextField txtFahrenheit = new JTextField(15);

 private JButton btnConverti = new JButton("Converti");

 public ConvertiFrame()

 {

 super("Convertitore Centigradi->Fahrenheit");

 addWindowListener(new GestoreFinestra());

 //Inserisce le componenti nei pannelli

 p1.add(new JLabel("Gradi Centigradi: "));

 p1.add(txtCentigradi);

 p2.add(new JLabel(": Fahrenheit"));

 p2.add(txtFahrenheit);

 //Inserisce le componenti nella finestra disponendole con una griglia

 setLayout(new GridLayout(3,1,5,10));

 add(p1);

 add(btnConverti);

 add(p2);

 btnConverti.addActionListener(this);

 }

 public void actionPerformed(ActionEvent e)

 {

 String pulsante = e.getActionCommand();

 double cent,fahr;

 if(pulsante.equals("Converti"))

 {

 try

 {

 String numeroLetto = txtCentigradi.getText();

 cent = Double.valueOf(numeroLetto).doubleValue();

 fahr = 32+(cent/100)\*180;

 txtFahrenheit.setText(""+fahr);

 }

 catch (Exception exc)

 {

 txtFahrenheit.setText("");

 }

 }

 }

 class GestoreFinestra implements WindowListener

 {

 public void windowIconified(WindowEvent e) { }

 public void windowDeiconified(WindowEvent e) { }

 public void windowActivated(WindowEvent e) { }

 public void windowDeactivated(WindowEvent e) { }

 public void windowOpened(WindowEvent e) { }

 public void windowClosed(WindowEvent e) { }

 public void windowClosing(WindowEvent e)

 {

 System.out.println("Programma terminato.");

 System.exit(0);

 }

 }

}

class ProgConvertiFrame

{

 public static void main(String args[])

 {

 ConvertiFrame f = new ConvertiFrame();

 f.pack();

 f.setVisible(true);

 }

}

Esercizio 109:

Gestire due pannelli in java: “Cambia frame mediante una ComboBox”

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 15/05/2015

Numero es:109

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica e gestire gli eventi in java

Obiettivo del programma: Gestire due pannelli in java: “Cambia frame mediante una ComboBox”

\*/

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class CambiaFrame extends Frame implements ActionListener

{

 private Color sfondo = Color.white;

 private JPanel p = new JPanel();

 private JLabel lblColori = new JLabel("Colori di sfondo: ");

 private JPanel panArea = new JPanel();

 private Choice cboColori = new Choice();

 private JButton btnCambia = new JButton("Cambia");

 public CambiaFrame()

 {

 super("Cambia sfondo!");

 setSize(400,250);

 addWindowListener(new GestoreFinestra());

 inizializzaCombo();

 panArea.setBackground(sfondo);

 //inserisce le componenti nel pannello in alto

 p.add(lblColori);

 p.add(cboColori);

 p.add(btnCambia);

 //inserisce le componenti nella finestra

 add(p, "North");

 add(panArea, "Center");

 btnCambia.addActionListener(this);

 }

 private void inizializzaCombo()

 {

 //aggiunge le voci alla combo box

 cboColori.addItem("bianco");

 cboColori.addItem("rosso");

 cboColori.addItem("arancione");

 cboColori.addItem("giallo");

 cboColori.addItem("verde");

 cboColori.addItem("blu");

 cboColori.addItem("nero");

 }

 public void actionPerformed (ActionEvent e)

 {

 String pulsante = e.getActionCommand();

 if(pulsante.equals("Cambia"))

 {

 switch(cboColori.getSelectedIndex())

 {

 case 0: sfondo = Color.white; break;

 case 1: sfondo = Color.red; break;

 case 2: sfondo = Color.orange; break;

 case 3: sfondo = Color.yellow; break;

 case 4: sfondo = Color.green; break;

 case 5: sfondo = Color.blue; break;

 case 6: sfondo = Color.black; break;

 }

 panArea.setBackground(sfondo);

 }

 }

 class GestoreFinestra implements WindowListener

 {

 public void windowIconified(WindowEvent e) { }

 public void windowDeiconified(WindowEvent e) { }

 public void windowActivated(WindowEvent e) { }

 public void windowDeactivated(WindowEvent e) { }

 public void windowOpened(WindowEvent e) { }

 public void windowClosed(WindowEvent e) { }

 public void windowClosing(WindowEvent e)

 {

 System.out.println("Programma terminato.");

 System.exit(0);

 }

 }

}

class ProgCambiaFrame

{

 public static void main(String args[])

 {

 CambiaFrame F = new CambiaFrame();

 F.setVisible(true);

 }

}

Esercizio 110:

Le finestre di dialogo in java

/\*

ITIS-LS F.Giordani Caserta

Anno scolastico 2014/2015

Classe 4^ sez.D spec. Informatica

Data: 15/05/2015

Numero es:109

Versione:1.0

Programmatore/i: Sebastiano Fusco

Sistema Operativo:Windows XP

Compilatore/Interprete:Dev-C++ 4.9.9.2

Obiettivo didattico: L'alunno e' in grado di utilizzare la grafica e gestire gli eventi in java

Obiettivo del programma: Le finestre di dialogo in java

\*/

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

class CambiaFrame extends Frame implements ActionListener

{

 private Color sfondo = Color.white;

 private JPanel p = new JPanel();

 private JPanel panArea = new JPanel();

 private JComboBox cbColori = new JComboBox();

 private JButton btnCambia = new JButton("Cambia");

 public CambiaFrame()

 {

 super("Cambia sfondo!");

 setSize(400,250);

 addWindowListener(new GestoreFinestra());

 inizializzaCombo();

 panArea.setBackground(sfondo);

 //inserisce le componenti nel pannello in alto

 p.add(new Label("Colori di sfondo: "));

 p.add(cbColori);

 p.add(btnCambia);

 //inserisce le componenti nella finestra

 add(p, "North");

 add(panArea, "Center");

 btnCambia.addActionListener(this);

 }

 private void inizializzaCombo()

 {

 //aggiunge le voci alla combo box

 cbColori.addItem("bianco");

 cbColori.addItem("rosso");

 cbColori.addItem("arancione");

 cbColori.addItem("giallo");

 cbColori.addItem("verde");

 cbColori.addItem("blu");

 cbColori.addItem("nero");

 }

 public void actionPerformed (ActionEvent e)

 {

 String pulsante = e.getActionCommand();

 if(pulsante.equals("Cambia"))

 {

 switch(cbColori.getSelectedIndex())

 {

 case 0: sfondo = Color.white; break;

 case 1: sfondo = Color.red; break;

 case 2: sfondo = Color.orange; break;

 case 3: sfondo = Color.yellow; break;

 case 4: sfondo = Color.green; break;

 case 5: sfondo = Color.blue; break;

 case 6: sfondo = Color.black; break;

 }

 panArea.setBackground(sfondo);

 }

 }

 class GestoreFinestra implements WindowListener

 {

 public void windowIconified(WindowEvent e) { }

 public void windowDeiconified(WindowEvent e) { }

 public void windowActivated(WindowEvent e) { }

 public void windowDeactivated(WindowEvent e) { }

 public void windowOpened(WindowEvent e) { }

 public void windowClosed(WindowEvent e) { }

 public void windowClosing(WindowEvent e)

 {

 System.out.println("Programma terminato.");

 System.exit(0);

 }

 }

}

class ProgCambiaFrame

{

 public static void main(String args[])

 {

 CambiaFrame F = new CambiaFrame();

 F.setVisible(true);

 }

}