class Person

{

//feild:

string firstName;

string lastName;

string id;

//getter and setter;

public string GetFirstName()

{

return firstName;

}

public string GetLastName()

{

return lastName;

}

public string GetId()

{

return id;

}

public bool SetFirstName(string firstName)

{

foreach (char tav in firstName)

{

if (!(tav >= 'a' && tav <= 'z') && !(tav >= 'A' && tav <= 'Z'))// if (not small letters or not capital letters)

{

Console.WriteLine("The first name contains characters that are letters");

return false;

}

}

this.firstName = firstName;

return true;

}

public bool SetLastName(string lastName)

{

foreach (char tav in lastName)

{

if (!(tav >= 'a' && tav <= 'z') && !(tav >= 'A' && tav <= 'Z'))// if (not small letters or not capital letters)

{

Console.WriteLine("The last name contains characters that are letters");

return false;

}

}

this.lastName = lastName;

return true;

}

public bool SetId(string id)

{

foreach (char val in id)

{

if(!(val>='0' && val <= '9'))

{

Console.WriteLine("The string contains non-numeric characters");

return false;

}

}

this.id = id;

return true;

}

//contructors:

//Default cto'r

public Person()

{

id = firstName = lastName = "";

}

//Parameter cto'r:

public Person(string id,string firstName,string lastName)

{

while (!SetId(id))

{

Console.Write("entar id: ");

id = Console.ReadLine();

}

while (!SetFirstName(firstName))

{

Console.Write("entar first name: ");

firstName = Console.ReadLine();

}

while (!SetLastName(lastName))

{

Console.Write("entar last name: ");

lastName = Console.ReadLine();

}

}

//copy cto'r:

public Person(Person other):this(other.id,other.firstName,other.lastName)

{

//option 2:

//id = other.id;

//firstName = other.firstName;

//lastName = other.lastName;

}

public void Print()

{

Console.WriteLine("Full Name: {0} {1}, Id: {2}", firstName, lastName,id);

}

}

class Program

{

static void Main(string[] args)

{

Person[] arr=null;

int choice = 0;

while (choice!=3)

{

Console.WriteLine("\n\nSelect:\n1.Printing a list of people\n2.Adding personal details\n3.Exit");

choice = int.Parse(Console.ReadLine());

switch (choice)

{

case 1:

PrintList(arr);

break;

case 2:

AddPerson(ref arr);

break;

case 3:

Console.WriteLine("by by!");

break;

}

}

}

static void PrintList(Person [] arr)

{

if(arr==null || arr.Length == 0)

{

Console.WriteLine("list is empty!");

return;

}

foreach (Person p in arr)

{

p.Print();

}

}

static void AddPerson(ref Person[] arr)

{

Person[] newarr;

if (arr == null || arr.Length == 0)

{

newarr = new Person[1];

}

else

{

newarr = new Person[arr.Length + 1];

Array.Copy(arr, newarr, arr.Length);

}

Console.WriteLine("enter id, firstName, lastName: ");

newarr[newarr.Length - 1] = new Person(Console.ReadLine(), Console.ReadLine(), Console.ReadLine());

arr = newarr;

}

}