// See https://aka.ms/new-console-template for more information

Console.WriteLine("Hello, World!");

Person p = new();

p.GrownUp += P\_GrownUp;

p.GrownUp += (s => Console.WriteLine("Hey "+s));

p.GrownUp2 += P\_GrownUp21;

void P\_GrownUp21(object? sender, EventArgs e)

{

//Console.WriteLine(sender.ToString());

Console.WriteLine( (e as MyEventArgs)?.Message);

}

void P\_GrownUp(string Message)

{

Console.WriteLine("Hansi sagt mir :" + Message);

}

void P\_GrownUp2(string Message)

{

Console.WriteLine("ha? " + Message);

}

for (int i = 0; i < 20; i++)

{

p.CelebrateBirthday();

}

class Person

{

public delegate void WhatToTell(string Message);

public event WhatToTell GrownUp;

public event EventHandler GrownUp2;

public int Age { get; private set ; }

public void CelebrateBirthday()

{

Age++;

if (Age == 18)

OnGrownUp();

}

private void OnGrownUp()

{

// Sagen wir das Allen

if (GrownUp != null)

GrownUp("Endlich " + Age);

if (GrownUp2 != null)

GrownUp2(this, new MyEventArgs { Message= "finally " + Age });

}

}

public class MyEventArgs : EventArgs

{

public string Message { get; set; }

}

using System;

using System.Linq;

using System.Collections.Generic;

namespace PersonMitUndOhneLINQ

{

class Program

{

static void Main(string[] args)

{

MyFriends mf = new MyFriends();

mf.Add(new Person { Id = 1, Name = "Hugo", Geboren = new DateTime(2005, 1, 1), SVNr=1234 });

mf.Add(new Person { Id = 2, Name = "Susi", Geboren = new DateTime(2005, 2, 2),SVNr=2345 });

mf.Add(new Person { Id = 3, Name = "Felix", Geboren = new DateTime(2005, 3, 3),SVNr=3456 });

//Lösung ohne LINQ

List<Result> lr = mf.FilterByMonth(2);

foreach (var item in lr)

Console.WriteLine(item.Name + "/" + item.SVNr);

// Lösung mit LINQ

var erg = mf.Where((x) => x.Geboren.Month == 2)

.Select((x)=>new { x.Name, x.SVNr });//"result"

foreach (var item in erg)

Console.WriteLine(item.Name + "/" + item.SVNr);

}

}

class Person

{

public int Id { get; set; }

public string Name { get; set; }

public DateTime Geboren { get; set; }

public int SVNr { get; set; }

}

class Result {

public string Name { get; set; }

public int SVNr { get; set; }

}

class MyFriends : List<Person>

{

public List<Result> FilterByMonth(int m) {

List<Result> lr = new List<Result>();

foreach (var item in this)

{

if (item.Geboren.Month==m)

{

Result r = new Result();

r.Name = item.Name;

r.SVNr = item.SVNr;

lr.Add(r);

}

}

return lr;

}

}

}

using System;

using System.Collections;

using System.Linq;

namespace UebungLINQ

{

class Program

{

static void Main(string[] args)

{

//LINQ = SQL f+r Arrays

int[] data = { 1, 2, 3, 4, 5, 6, 7 };

string[] strings = { "1", "2", "3", "4", "5", "6" };

var even = //pretty print SQL syntax

from x in data

where (x % 2 == 0)

select x;

foreach (var item in even)

Console.Write(item);

Console.WriteLine();

var odd = //Methodensyntax

data.Where((x)=> x % 2 == 1);

var odd =

strings.Where(x => x.Contains("3"))

foreach (var item in odd)

Console.Write(item);

// --- Iterator (foreach bei input und output)

// delegate (lambda, anonymous method)

// -- object initializer

// -- extension methods

// -- implicitly typed local variable ("var")

// -- anonymous class

A a = new A();

a.Greet();

int xx = 23;

Console.WriteLine(xx.toHex());

var s = 3.5+"3";

var v = new { A = 2, B = "Felix" };

var w = new { A = 3, B = "Susi" };

v = w;

//w ist eine anonyme klasse

}

}

public sealed class A {

//erweitere die Klasse A - GEHT NED !!

}

public static class MyExtensions

{

public static void Greet(this A a)

{

Console.WriteLine("Hello from a");

}

public static string toHex(this int i) {

return i.ToString("x");

}

// MS schrieb eine Extension Methos "Where", welche

// IEnumerable um die Funktion eines als objekt übergebenen

// Lambda ausdrucks erweitert, welcher genau diese elemente

// als neues IEnumerable zurückgibt.

// SELECT,GROUPBY,JOIN,FROM,... wurschtwos siehe SQL

public static IEnumerable Where(this IEnumerable ie) {

/\* add code \*/

throw new NotImplementedException();

}

}

}