**שיעורי בית יסודות מערך עצמים – אופיר הופמן י3**

**תרגיל 1**

class Car

{

private string num;

private string model;

private int year;

public Car(string num, string model, int year)

{

this.num = num;

this.model = model;

this.year = year;

}

public string GetModel()

{

return this.model;

}

public int GetYear()

{

return this.year;

}

public bool Older(Car OtherCar)

{

return this.year < OtherCar.year;

}

public bool Equals(Car OtherCar)

{

return this.year == OtherCar.year;

}

public override string ToString()

{

string s = "Num: " + this.num + ", Model: " + this.model + ", Year " + this.year;

return s;

}

}

class UsedCars

{

private const int MaxCars = 1000;

private Car[] usedCars;

private int empty;

public UsedCars ()

{

this.usedCars = new Car[MaxCars];

}

public void AddCar(Car c)

{

if (empty < 1000)

{

usedCars[empty] = c;

empty++;

}

}

public void AddCar(string num, string model, int year)

{

Car car = new Car(num, model, year);

AddCar(car);

}

public void PrintAllCarsProducedBefor(int year)

{

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

{

if (usedCars[i].GetYear() < year)

{

Console.Write(usedCars[i]);

Console.WriteLine();

}

}

}

public int HowMuch(string model)

{

int cnt = 0;

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

{

if (usedCars[i].GetModel() == model)

cnt++;

}

return cnt;

}

}

static void Main(string[] args)

{

UsedCars uc = new UsedCars();

uc.AddCar("90-123-56", "toyota", 2006);

uc.AddCar("23-003-14", "ford", 2007);

uc.AddCar("83-101-64", "mazda", 2008);

Car c1 = new Car("13-501-36", "toyota", 2000);

Car c2 = new Car("19-502-68", "mazda", 2009);

uc.AddCar(c1);

uc.PrintAllCarsProducedBefor(2008);

Console.WriteLine(uc.HowMuch("toyota"));

}

**תרגיל 2**

class Playlist

{

private string name;

private int length;

private Song[] songs;

public Playlist(string name)

{

this.name = name;

this.songs = new Song[100];

this.length = 0;

}

public bool AddSong(Song song)

{

if (length < 100)

{

songs[length] = song;

length++;

return true;

}

return false;

}

public bool AddSong(string name, string Singer, int length)

{

Song song = new Song(name, Singer, length);

return AddSong(song);

}

public int OverAll()

{

int sum = 0;

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

{

sum += songs[i].GetLength();

}

return sum;

}

public string Longest()

{

Song longestSong = songs[0];

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

{

if (songs[i].GetLength() > longestSong.GetLength())

longestSong = songs[i];

}

return longestSong.GetName();

}

}

**תרגיל 3**

class CellPhone

{

private string num;

private string name;

private bool isOnline;

private CellPhone[] incomingCalls;

private CellPhone[] outgoingCalls;

private int numOfIncoming;

private int numOfOutgoing;

public CellPhone(string num, string name, bool isOnline)

{

this.num = num;

this.name = name;

this.isOnline = isOnline;

incomingCalls = new CellPhone[100];

outgoingCalls = new CellPhone[100];

numOfIncoming = 0;

numOfOutgoing = 0;

}

public void SetNum(string num)

{

this.num = num;

}

public void SetName(string name)

{

this.name = name;

}

public void set\_isOnline(bool isOnline)

{

this.isOnline = isOnline;

}

public string GetNum()

{

return this.num;

}

public string GetName()

{

return this.name;

}

public bool Get\_isOnline()

{

return this.isOnline;

}

public CellPhone[] GetIncomingCalls()

{

return this.incomingCalls;

}

public CellPhone[] GetOutgoingCalls()

{

return this.outgoingCalls;

}

public int GetNumOfIncoming()

{

return this.numOfIncoming;

}

public int GetNumOfOutgoing()

{

return this.numOfOutgoing;

}

public override string ToString()

{

string s = "Name: " + this.name + ", number: " + this.num + ", Is Online: " + this.isOnline + ", Incoming calls: " + numOfIncoming + ", Outgoing calls: " + numOfOutgoing;

return s;

}

public void MakeCall(CellPhone ToCell)

{

outgoingCalls[numOfOutgoing] = ToCell;

numOfOutgoing++;

}

public void GetCall(CellPhone fromCell)

{

incomingCalls[numOfIncoming] = fromCell;

numOfIncoming++;

}

}

internal class Program

{

public static void Call(CellPhone phone1, CellPhone phone2)

{

if (phone1 != phone2)

{

phone1.MakeCall(phone2);

phone2.GetCall(phone1);

}

}

static void Main(string[] args)

{

CellPhone[] cellphones = new CellPhone[10];

CellPhone cell1 = new CellPhone("0547783489", "avi", true);

cellphones[0] = cell1;

CellPhone cell2 = new CellPhone("0573385987", "Yossi", true);

cellphones[1] = cell2;

CellPhone cell3 = new CellPhone("0526670728", "Moshe", true);

cellphones[2] = cell3;

CellPhone cell4 = new CellPhone("0558908395", "Alon", false);

cellphones[3] = cell4;

CellPhone cell5 = new CellPhone("0537894783", "Yaron", true);

cellphones[4] = cell5;

CellPhone cell6 = new CellPhone("0532787789", "Refael", true);

cellphones[5] = cell6;

CellPhone cell7 = new CellPhone("0527784950", "Moti", true);

cellphones[6] = cell7;

CellPhone cell8 = new CellPhone("0557783957", "Amnon", false);

cellphones[7] = cell8;

CellPhone cell9 = new CellPhone("0527780909", "Adam", true);

cellphones[8] = cell9;

CellPhone cell10 = new CellPhone("0547789457", "Yotam", true);

cellphones[9] = cell10;

for (int i = 0; i < cellphones.Length; i++)

{

Console.WriteLine(cellphones[i]);

}

Random rnd = new Random();

for (int i = 0; i < 100; i++)

{

CellPhone phone1 = cellphones[rnd.Next(10)];

CellPhone phone2 = cellphones[rnd.Next(10)];

if (phone1 == phone2)

i++;

else

{

Call(phone1, phone2);

}

}

CellPhone maxIncoming = cellphones[0];

CellPhone minOutgoing = cellphones[0];

for (int i = 0; i < cellphones.Length; i++)

{

if (cellphones[i].GetNumOfIncoming() > maxIncoming.GetNumOfIncoming())

maxIncoming = cellphones[i];

else if (cellphones[i].GetNumOfOutgoing() < minOutgoing.GetNumOfOutgoing())

minOutgoing = cellphones[i];

}

Console.WriteLine("The most Incoming calls: " + maxIncoming.GetName());

Console.WriteLine("Calls list:");

for (int i = 0; i < maxIncoming.GetNumOfIncoming(); i++)

{

Console.WriteLine(maxIncoming.GetIncomingCalls()[i].GetName());

}

Console.WriteLine("The list Outgoing calls: " + minOutgoing.GetName());

Console.WriteLine("Calls list:");

for (int i = 0; i < minOutgoing.GetNumOfOutgoing(); i++)

{

Console.WriteLine(minOutgoing.GetOutgoingCalls()[i].GetName());

}

}

}