Tyson Duran

ECE 2310.03

Final Project

**Pool**

+temperature: Temperature

+location: Location

+Count: static int

+Pool()

+Pool(Temperature t, Location l)

+ToString: string

~Pool

**Location**

-x: int;

-y: int;

+Location(int a, int b)

+X

+Y

+FindDistance(Location sloc): double

+ToString: string

~Location

**Temperature**

-degree: int

-scale: string

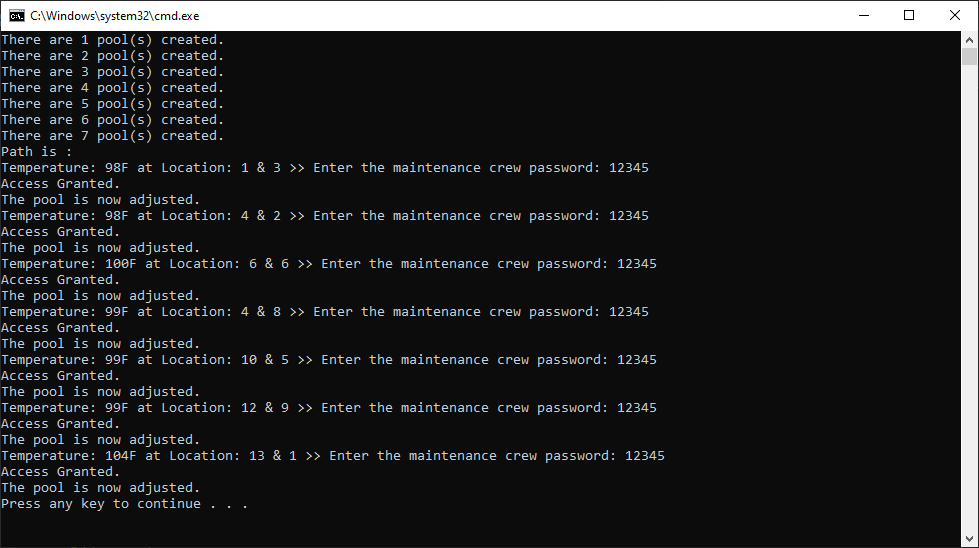
+Temperature()

+Temperature(int d)

+Degree

+ToString: string

~Temperature



namespace Final\_Project\_\_\_Duran\_\_Tyson

{

class Program

{

static void Main(string[] args)

{

ArrayList poolList = new ArrayList();

Random random = new Random();

// TEMPERATURES

Temperature[] temp = new Temperature[7];

for (int i = 0; i < 7; i++)

{

temp[i] = new Temperature(random.Next(98, 105));

}

// LOCATIONS

Location[] loc = new Location[7];

loc[0] = new Location(4, 8); // A

loc[1] = new Location(1, 3); // B

loc[2] = new Location(4, 2); // C

loc[3] = new Location(13, 1); // D

loc[4] = new Location(12, 9); // E

loc[5] = new Location(10, 5); // F

loc[6] = new Location(6, 6); // G

// POOLS

Pool[] p = new Pool[7];

for (int i = 0; i < 7; i++)

{

p[i] = new Pool(temp[i], loc[i]);

Console.WriteLine("There are " + Pool.Count + " pool(s) created.");

poolList.Add(p[i]);

}

Location location = new Location(0, 0);

Console.WriteLine("Path is : ");

Pool closest = null;

while (poolList.Count > 0)

{

double dist = double.MaxValue;

foreach (Pool i in poolList)

{

//double min = i.location.FindDistance(location);

if (i.location.FindDistance(location) < dist)

{

closest = i;

dist = i.location.FindDistance(location);

}

}

Console.Write(closest + " >> ");

closest.temperature.Degree = 100;

location = closest.location;

poolList.Remove(closest);

}

}

}

}

namespace Final\_Project\_\_\_Duran\_\_Tyson

{

class Pool

{

public Temperature temperature;

public Location location;

public static int Count = 0;

public Pool()

{

Count++;

}

public Pool(Temperature t, Location l)

{

temperature = t;

location = l;

Count++;

}

public override string ToString()

{

string output = "Temperature: " + temperature + " at " + location;

return output;

}

}

}

amespace Final\_Project\_\_\_Duran\_\_Tyson

{

class Temperature

{

int degree;

string scale;

public Temperature()

{

degree = 0;

scale = "F";

}

public Temperature(int d)

{

degree = d;

scale = "F";

}

public int Degree

{

get { return degree; }

set

{

int password = 12345;

Console.Write("Enter the maintenance crew password: ");

int input = Convert.ToInt32(Console.ReadLine());

if (input == password)

{

Console.WriteLine("Access Granted.\nThe pool is now adjusted.");

degree = value;

}

}

}

public override string ToString()

{

string output = degree + scale;

return output;

}

}

}

namespace Final\_Project\_\_\_Duran\_\_Tyson

{

class Location

{

int x;

int y;

public Location(int a, int b)

{

x = a;

y = b;

}

public int X

{

get { return x; }

set { x = value;}

}

public int Y

{

get { return y; }

set { y = value; }

}

public double FindDistance(Location sloc)

{

return Math.Sqrt((x - sloc.x) \* (x - sloc.x) + (y - sloc.y) \* (y - sloc.y));

}

public override string ToString()

{

string output = "Location: " + x + " & " + y;

return output;

}

}

}