

Template (Şablon)

Yazılım Tasarım Belgesi Software Design Document(SDD)

1.Giriş

1.1. Amaç ve kapsam

Projenin amacı, N kişinin katıldığı bir toplantı ve toplantıdaki herkesin tanıdığı(ünlü) kişiyi bulma üzerine yazılmıştır. Programa girdiler *TXT* dosyasından yapılıyor. Bu dosyaya, toplantıda kaç kişinin olduğu ve kimin kimi tanıdığı bilgileri yazılıyor. Program koşulduğu zaman, bu girdiler okunuyor ve ekrana ünlü kişiyi yazdırıyor. Ünlü yok ise, ünlü kişinin olmadığı bilgisi veriliyor.

1.2. Sözlük

str = Okunacak txt dosyasının adını tutan değişkendir.

persons = Dosyadaki kişileri içinde barındıran bir yığındır.

know = Birbirini tanıyanları tutan bir değişkendir.

i = Toplantıdaki kişilerin sayısını tutan değişken.

knows = knows?(a,b) sorgusunda kullanılan, tanıyıp tanımadığını sorguladığımız değişken.

a = tanıyan kişi.

b = tanınan kişi.

1.3. Referanslar

- <https://www.geeksforgeeks.org/the-celebrity-problem/>
- <https://stackoverflow.com/questions/29814436/optimal-solution-for-the-celebrity-algorithm>
- <https://www.geeksforgeeks.org/c-sharp-stack-class/>
- <https://www.youtube.com/watch?v=vTI7CuYCtds>

2.Tasarım Kararları

Yazılım üç etaptan oluşuyor. İlk etapta; Hangi dosyadan veri okunacak ise o dosyanın adı isteniyor, dosya adı girildikten sonra verileri okuyor.

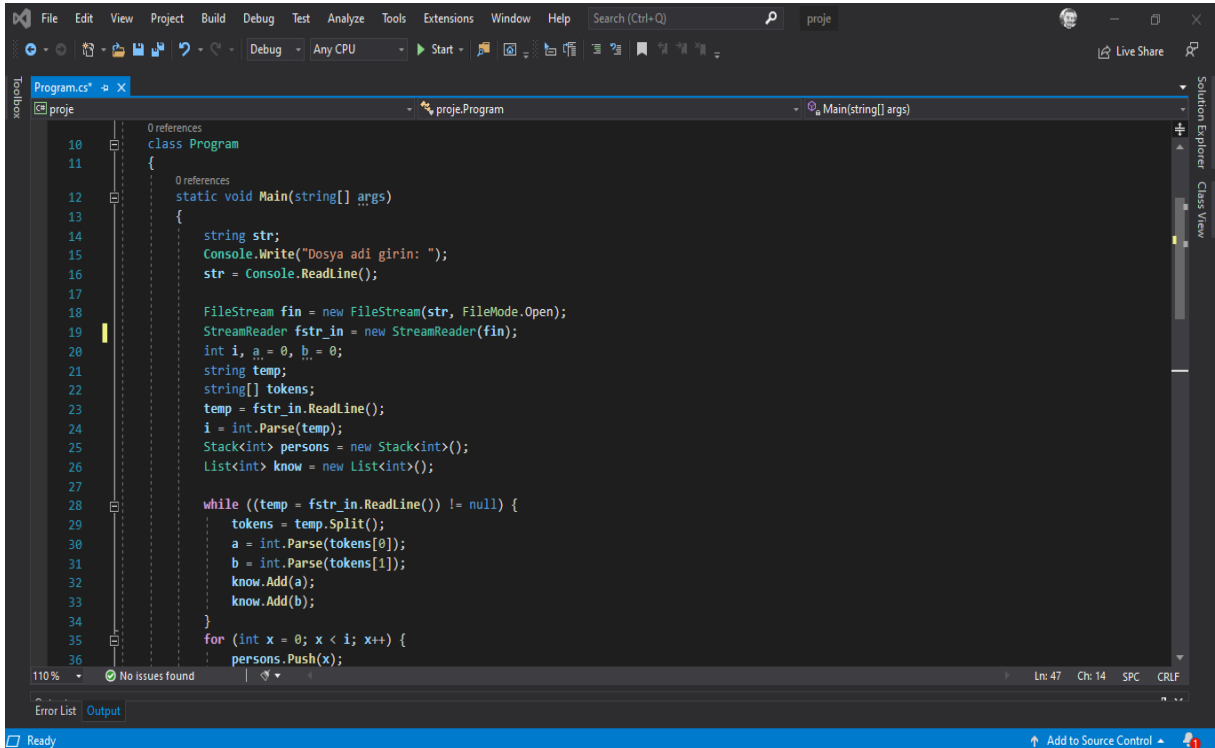
İkinci etapta; Bu dosyadaki bütün kişileri “persons” isimli değişkene atıyor. Ardından bu kişiler “knows” metodunda işleniyor, örneğin; knows?(a,b). A, b’yi tanıyor mu? Sorgusu dönüyor. Sorgu sonucunda belirlediğimiz kişi çöpe(pop) atılıyor, kalan kişi tekrar “persons” yığına itiliyor(push). Bu işlem yığındaki herkes için tekrarlanıyor.

Üçüncü etapta; son kalan kişi, başka bir kişiyi tanıyor mu kontrolü sağlanıyor.

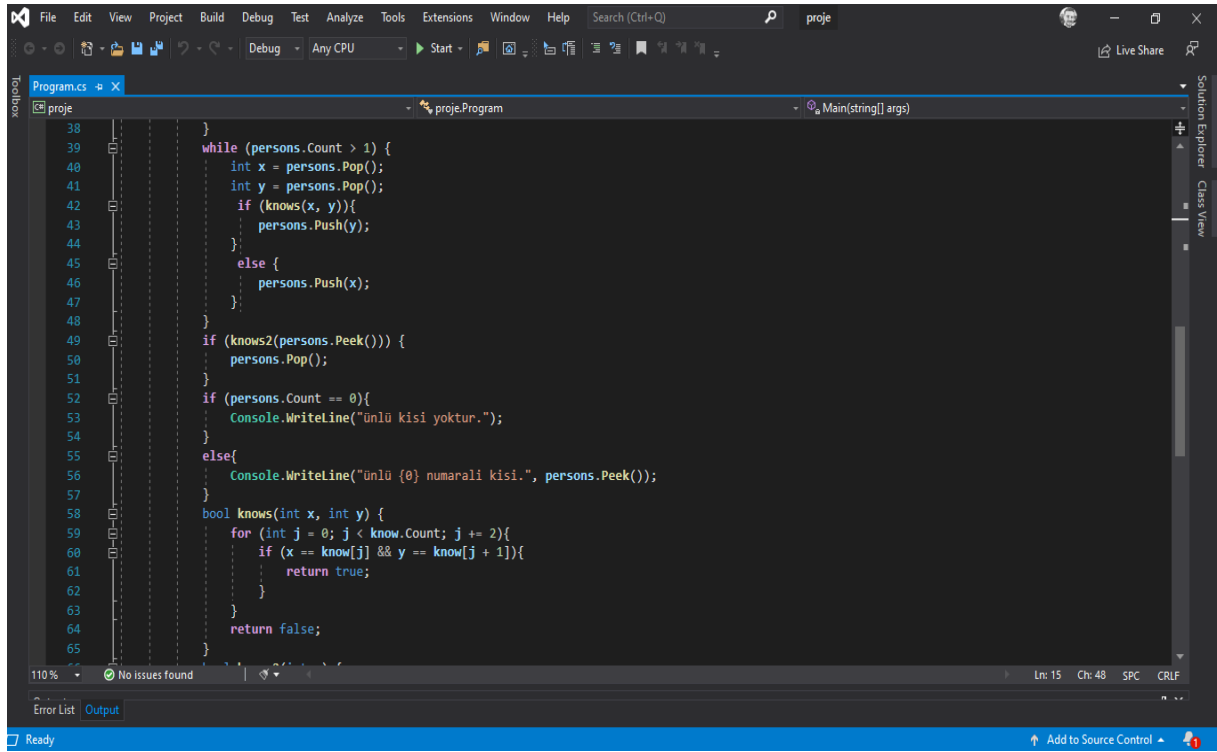
3.Tasarım Detayları

3.1. Yazılım Bileşenleri

Main fonksiyonu:

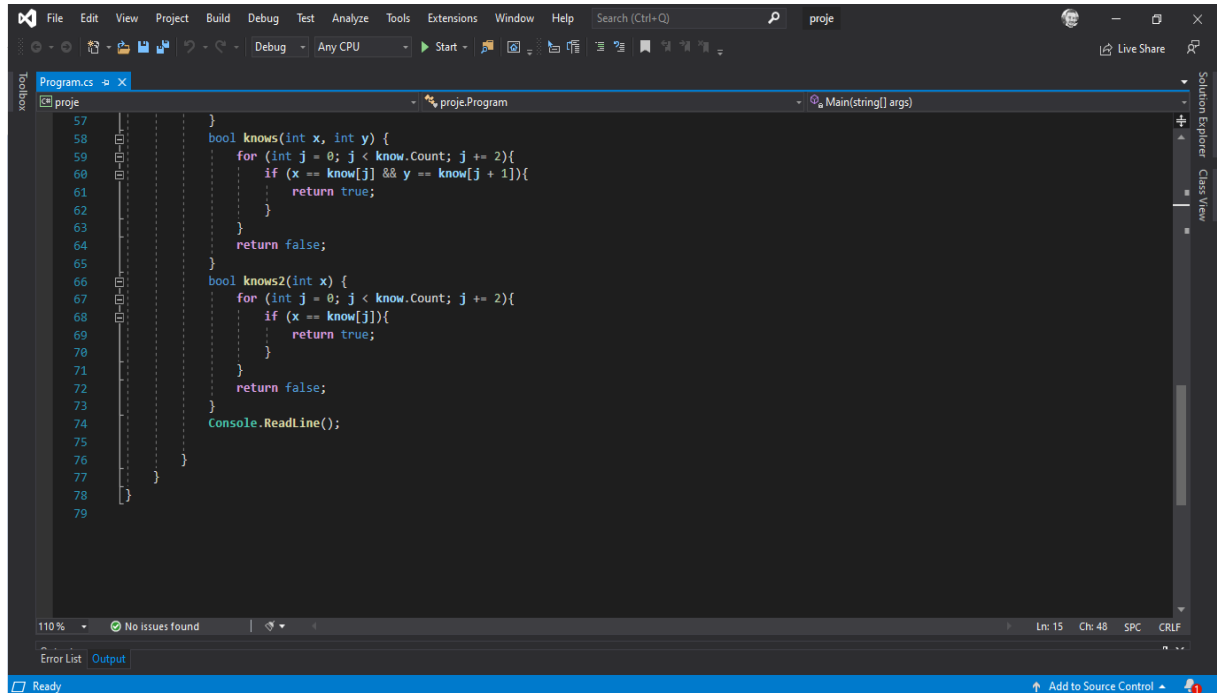


```
10 class Program
11 {
12     static void Main(string[] args)
13     {
14         string str;
15         Console.WriteLine("Dosya adı girin: ");
16         str = Console.ReadLine();
17
18         FileStream fin = new FileStream(str, FileMode.Open);
19         StreamReader fstr_in = new StreamReader(fin);
20         int i, a = 0, b = 0;
21         string temp;
22         string[] tokens;
23         temp = fstr_in.ReadLine();
24         i = int.Parse(temp);
25         Stack<int> persons = new Stack<int>();
26         List<int> know = new List<int>();
27
28         while ((temp = fstr_in.ReadLine()) != null) {
29             tokens = temp.Split();
30             a = int.Parse(tokens[0]);
31             b = int.Parse(tokens[1]);
32             know.Add(a);
33             know.Add(b);
34         }
35         for (int x = 0; x < i; x++) {
36             persons.Push(x);
37         }
38     }
39 }
```



```
38
39 while (persons.Count > 1) {
40     int x = persons.Pop();
41     int y = persons.Pop();
42     if (knows(x, y)){
43         persons.Push(y);
44     }
45     else {
46         persons.Push(x);
47     }
48 }
49 if (knows2(persons.Peek())) {
50     persons.Pop();
51 }
52 if (persons.Count == 0){
53     Console.WriteLine(" nl  kisi yoktur.");
54 }
55 else{
56     Console.WriteLine(" nl  {0} numaralı kisi.", persons.Peek());
57 }
58 bool knows(int x, int y) {
59     for (int j = 0; j < know.Count; j += 2){
60         if (x == know[j] && y == know[j + 1]){
61             return true;
62         }
63     }
64     return false;
65 }
```

knows ve knows2 metodları:



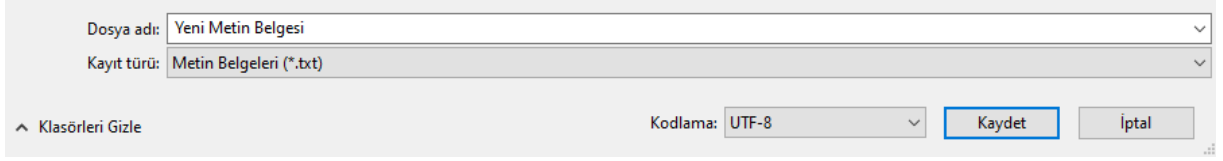
```
57
58 bool knows(int x, int y) {
59     for (int j = 0; j < know.Count; j += 2){
60         if (x == know[j] && y == know[j + 1]){
61             return true;
62         }
63     }
64     return false;
65 }
66 bool knows2(int x) {
67     for (int j = 0; j < know.Count; j += 2){
68         if (x == know[j]){
69             return true;
70         }
71     }
72     return false;
73 }
74 Console.ReadLine();
75
76
77
78
79 }
```

3.3. Veri Modeli

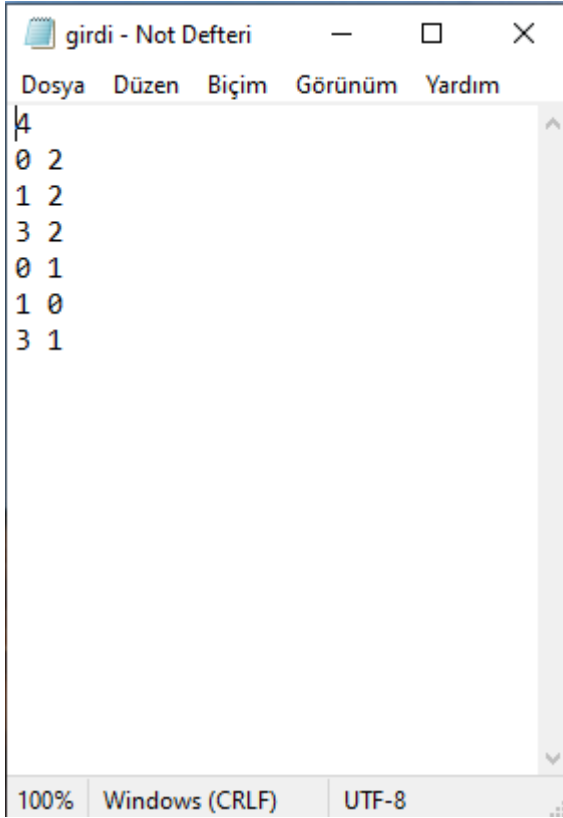
Veriler metin belgesinden(txt) okunuyor. İlk satında grupta kaç kişinin olduđu bilgisi veriliyor. Kalan satırlarda, aralarında bir boşluk olacak şekilde yan

yana yazılan kişiler veriliyor. Sol tarafta kalan kişi tanıyan, sağ tarafta kalan ise tanınan kişi oluyor. Girdiler tam sayı olacak şekilde girilmesi gerekiyor.

Veri kayıt türü:

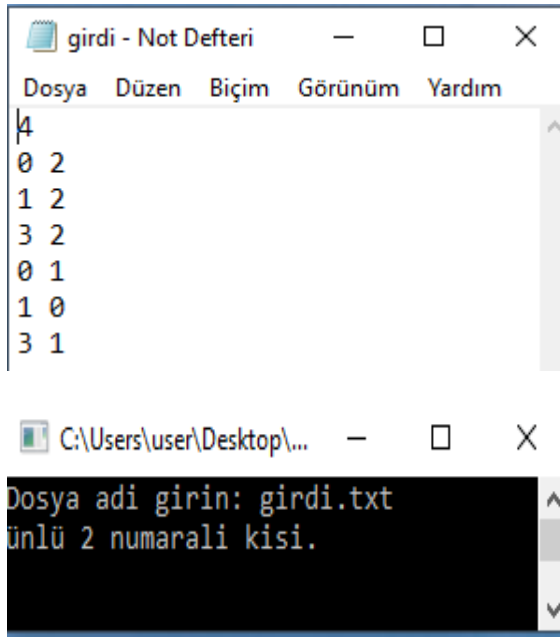


Veri modeli örnek:

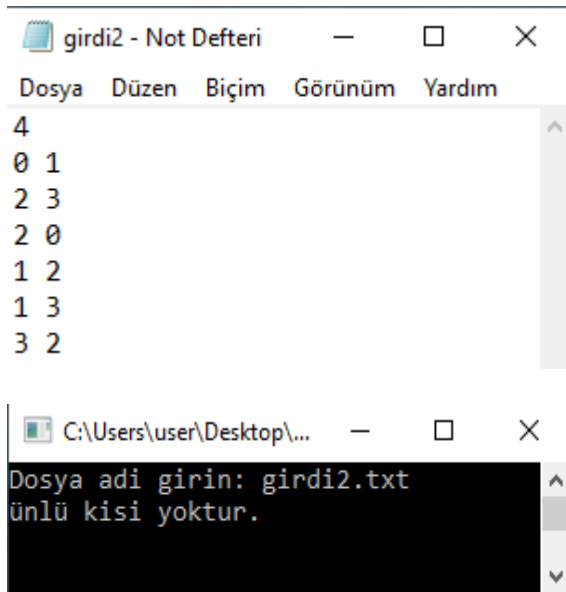


4. Girdi/Çıktı Örnekleri

1.Girdi/Çıktı:



2.Girdi/Çıktı:



5.Sonuç

Yazılım anlatıldığı gibi sorunsuz çalışmaktadır. Görsellerdeki girdiler, programın “debug” klasörüne eklenmiştir. İstenildiği anda çalıştırılıp çıktı alınabilir ve bu girdiler değiştirilebilir.