

## CONSOLE EKRANI

### CONSOLE EKRANI

✓ Proje 2

### KAZANIMLAR

✓ Genel Tekrar

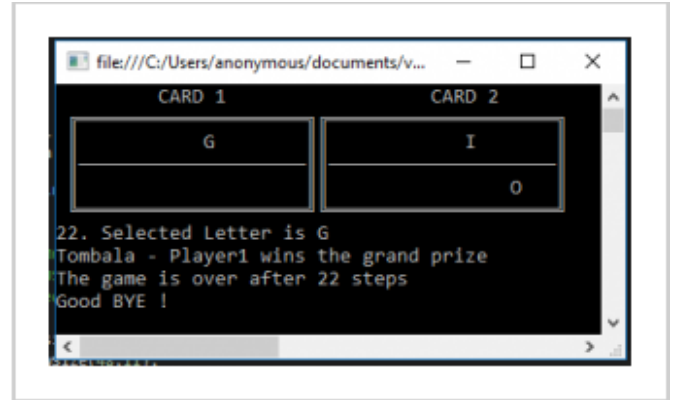
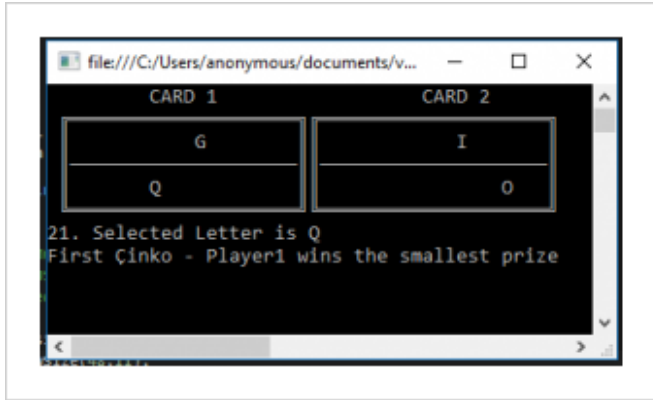


**Proje:** Console Ekranında Harfler ile Tombala Oyunu Uygulamasını yapacağız.

Oyunda;iki oyuncu bulunuyor. Her oyuncunun 2 satır ve 4 sütundan oluşan kartı var.İngilizce Alfabesine göre oyuncuların kartlarına benzersiz olarak rasgele bu harflerden toplam 16 tane har rasgele dağıtılır.Kartlar oluşturulduktan sonra, elimizde A-Z harfler ve '#' jokeri bulunan torbadan harf çekmeye başlarız.Gelen harf hangi kartta var ise o alanlar silinir. Joker geldiğinde her iki oyuncunun kartlarında bulunan alfabeye göre en büyük harf silinir. Hangi oyuncu kartının her hangi bir satırını tamamen bitirdiğinde Çinko yapmış olur. İki satırı da bittiğinde oyun tamamlanır ve Tombala olmuş olur.

## Algoritması

Kullanılacak verilerin değişkenleri ilk olarak tanımlanır. Harfler ve Çanta olarak iki değişken oluşturulur. Harfler ve Çanta'dan harfler dağıtılırken her harfin benzersiz olması için dağıtılırken iki adet kontrol için kontrol dizisi oluşturuldu ve değerleri resetlendi. Ardından oyuncuların kartları oluşturuldu. Harflerin kontrol edilerek yerleştirildi. Oyuncu, herhangi bir satırı bitirdiğinde çinko yapmış olacak ve iki satırı bitirdiğinde ise tombala yapmış olacaktır.



## Kod Blokları

```
using System;
namespace Hw4
{
    class Program
    {
        static bool[] Control = new bool[4];
        static bool[] Written = new bool[5];
        static void Main(string[] args)
        {
            /*
            * Published on Muslu.Net at 26.11.16
            * by M.Tayyip MUSLU
            * To be Countined C# Console Application Examples
            */
            Console.CursorVisible = false;
            Console.SetWindowSize(48, 11);
            char[] Letters = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N',
            'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z' }; // Letters to use random cards
            bool[] LettersC = new bool[Letters.Length]; // letters location control
            char[,] Cards = new char[4, 4];
            char[] Bag = { 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O',
            'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', '#' }; // random get a letter
```

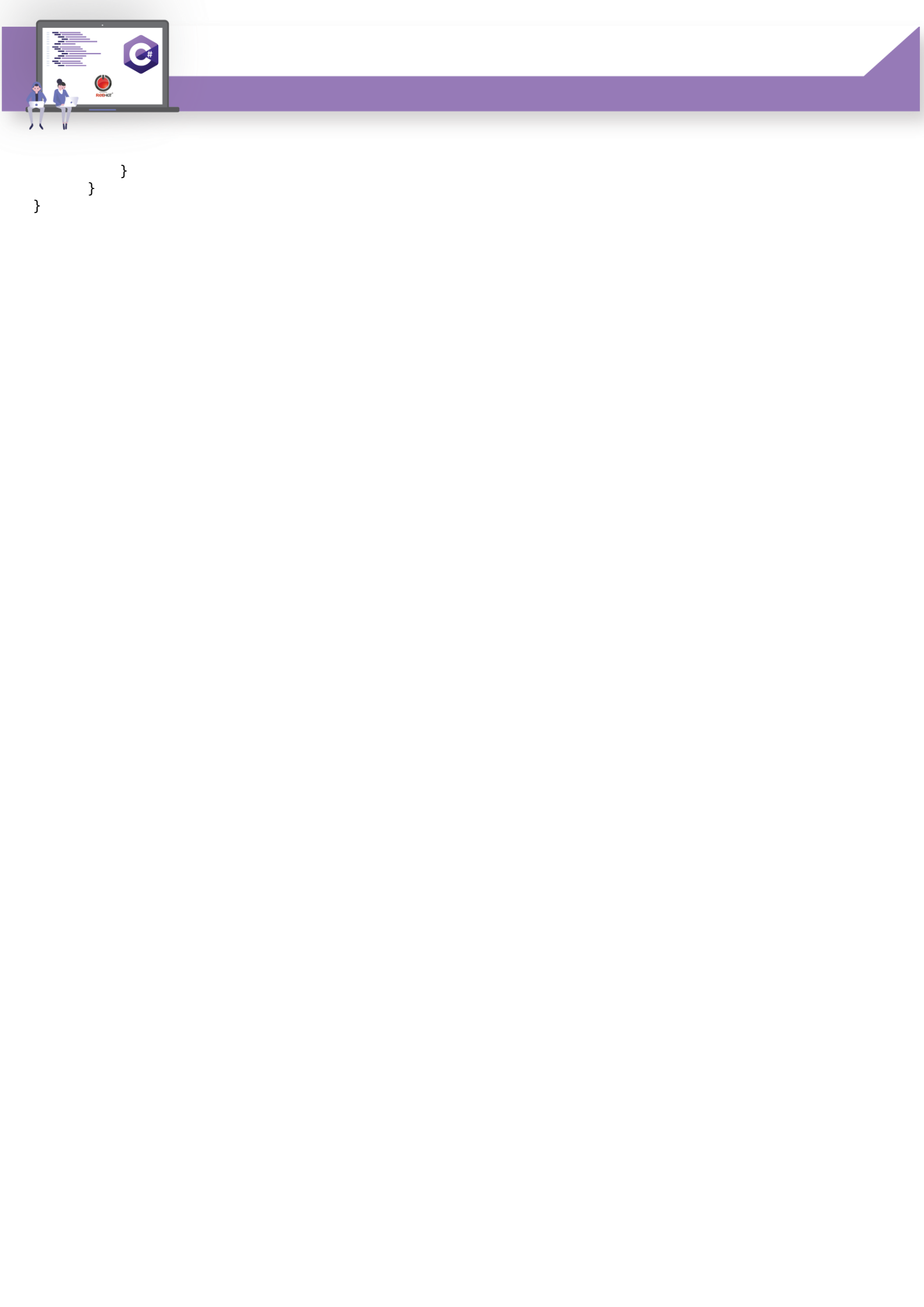


```
bool[] BagControl = new bool[Bag.Length]; // bag's letters location control
// Information Messages
string Cinko = "First Çinko - Player{0} wins the smallest prize";
string Tombala = "Tombala - Player{0} wins the grand prize";
string NotWinner = "The game is over without any winner (tie)";
Random RD = new Random();
int RandomNumber = 0;
// RESET CONTROL ARRAYS
for (byte i = 0; i < Letters.Length; i++) LettersC[i] = false;
for (byte i = 0; i < Bag.Length; i++) BagControl[i] = false;
int temp1 = 0, temp2 = 0;
for (byte k = 0; k < 4; k++) // CREATE CARS, RANDOM LETTERS ARE LOCATED
{
    for (byte l = 0; l < 4; l++)
    {
        if (k % 2 == 0) { temp1 = 0; temp2 = 13; } else { temp1 = 14; temp2 = Letters.Length; }
        RandomNumber = RD.Next(temp1, temp2);
        if (LettersC[RandomNumber] == false)
        {
            Cards[k, l] = Letters[RandomNumber];
            LettersC[RandomNumber] = true;
        }
        else l--;
    }
}
// DRAW SCREEN
Console.WriteLine(@" CARD 1 CARD 2

||| || ||| |||
||| || ||| |||");
int GetLetter = 0; byte Counter = 0, CounterLetter = 0;
do
{
    byte top = 2;
    Console.SetCursorPosition(0, 7); Console.WriteLine(" ");
    // SHOW LETTERS
    for (byte k = 0; k < 4; k++)
    {
        if (k < 2) Console.SetCursorPosition(5, top); else Console.SetCursorPosition(28, top);
        for (byte l = 0; l < 4; l++)
        {
            Console.Write(Cards[k, l] + " ");
        }
        top += 2;
        if (k == 1) top = 2;
    }
    Console.SetCursorPosition(0, 6);
    GetLetter = RD.Next(0, Bag.Length); // GET A NEW LETTERS IN BAG
    if (BagControl[GetLetter] == false)
    {
        Counter++;
        BagControl[GetLetter] = true;
        if (Bag[GetLetter] != '#') // NORMAL LETTER
        {
            for (byte k = 0; k < 4; k++)
            {
                for (byte l = 0; l < 4; l++)
                {
```



```
        if (Cards[k, 1] == Bag[GetLetter])
        {
            Cards[k, 1] = ' ';
        }
    }
}
else // JOKER
{
    int count = 0, d = Letters.Length - 1;
    while (true) // START THE LAST INDEX OF LETTERS TO FIRST INDEX SEARCH ON CARDS
    {
        for (byte k = 0; k < 4; k++)
        {
            for (byte l = 0; l < 4; l++)
            {
                if (Cards[k, l] == Letters[d])
                { Cards[k, l] = ' '; if (k == 0 || k == 1) k = 2; else k = 0; d =
Letters.Length - 1; count++; continue; }
            }
            if (count == 2) break; // JUST 2 LETTERS MUST BE DELETED
            d--;
        }
    }
    Console.WriteLine("{0}. Selected Letter is {1}", Counter, Bag[GetLetter]);
    // CONTROL PRIZES, FIRST OR PRIZE
    for (byte k = 0; k < 4; k++)
    {
        for (byte l = 0; l < 4; l++)
        {
            if (Cards[k, l] == ' ')
            {
                CounterLetter++;
            }
            if (CounterLetter == 4)
                Control[k] = true;
        }
        CounterLetter = 0;
    }
    Console.WriteLine();
    // CONTROLS AND FINISH
    if ((Control[2] && Control[3]) && (Control[0] && Control[1])) if (!Written[4]) {
Console.WriteLine(NotWinner); Written[3] = true; break; }
    if ((Control[0] || Control[1])) if (!Written[0]) { Console.WriteLine(Cinko, "1");
Written[0] = true; }
    if ((Control[0] && Control[1])) if (!Written[1]) { Console.WriteLine(Tombala, "1");
Written[1] = true; break; }
    if ((Control[2] || Control[3])) if (!Written[2]) { Console.WriteLine(Cinko, "2");
Written[2] = true; }
    if ((Control[2] && Control[3])) if (!Written[3]) { Console.WriteLine(Tombala, "2");
Written[3] = true; break; }
    for (byte c = 0; c < 4; c++) Control[c] = false;
    Console.ReadKey();
}
}
while (true);
Console.WriteLine("The game is over after {0} steps \nGood BYE !", Counter);
Console.ReadKey();
```



```
}
    }
}
```