

C. Sonuç

Süre: 10 dk.

Materyal: [EK 3. Hafta 9 Süreli Ödev](#)

Hazırlık: EK 3 öğrencilere ÖYS ortamında süreli ödev olarak ders öncesinde açılmalıdır.

Ders içi uygulama: Dersin son 10 dk.'sında öğrencilere ödevin nasıl yapılacağı ve gönderileceği açıklanır. Gelecek hafta “Nesne Tabanlı Programlamanın Prensipleri” konusunun işleneceğini bildirir.

Ders dışı uygulama (50 dk.): Bu uygulama asenkron olarak ÖYS üzerinden yürütülecektir. Öğrenciler ders sonunda istedikleri zaman başlatabilecekleri ancak süreli olacak şekilde tamamlayacakları görevleri ders dışında ödev olarak ÖYS’den yapacaktır. EK 3 ödev sırasında öğrencilere açılır. Ödevin süresi 50 dk. olarak belirlenir. Öğrenciler görevleri istedikleri sırada ve sayıda kendi tercihlerine bağlı olarak verilen süre içinde asenkron olarak uygulamaya başlar ve ÖYS ortamında ödev olarak gönderir. Bir görevi doğru yapan öğrenciye, o göreve ilişkin beceri rozeti ÖYS üzerinden atanacaktır. Ödevler ileildikten sonra eğitmen görevleri ve yanıtlarını GitHub üzerinden öğrencilere gönderir. Doğru yanıtların buradan kontrol edilmesi için Github ortamını gösterir. Doğru yanıtlanan görevler ile ilgili rozetler öğrenci hesabına gönderilir. Süreli ödevler ile öğrencilerin performanslarının değerlendirilmesi, kalıcı öğrenme ve transferin sağlanması amaçlanmaktadır.

Süreli Ödev Yanıtlar

Tasarlayıcı: Bir basketbol takımının koçu, oyuncularının ad-soyad, forma numarası ve attığı basket bilgilerini tutmak için bir program hazırlamak ister. Koç bunun için örnek bir program yazar. Programda örnek olarak Basketbolcu sınıf yapısı ve bu sınıfa ait 2 basketbolcu nesnesi bulunmaktadır. Koçun yazdığı örnek kodu tasarlayınız.

```
#include <iostream>
#include <string>
using namespace std;
class Basketbolcu
{
public:
    string ad_soyad;
    int forma_no;
    int basket_sayisi;
    Basketbolcu(string x_ad_soyad, int x_forma_no, int x_basket_sayisi){
        ad_soyad = x_ad_soyad;
        forma_no = x_forma_no;
        basket_sayisi = x_basket_sayisi;
    }
};
```

```

    }
};

int main()
{
    Basketbolcu b1("Mirsad Turkcan", 5 , 21);
    Basketbolcu b2("Semih Erden", 9, 15);

    cout << b1.ad_soyad << " " << b1.forma_no << " " << b1.basket_sayisi << "\n";
    cout << b2.ad_soyad << " " << b2.forma_no << " " << b2.basket_sayisi << "\n";

    return 0;
}

```

Kodun Çıktısı:

```

Mirsad Turkcan 5 21
Semih Erden 9 15

```

Kodlayıcı: Grafik programlarında kullanmak üzere nokta nesnelerini tanımlamak için bir Nokta sınıfı oluşturalım. Noktalar iki boyutlu düzlemde yer alacağından özellik olarak x ve y koordinatları olmak üzere iki adet koordinat bilgisine sahiptir. Programınızda noktaların sahip olması gereken yetenekler (davranışlar) ise şunlar olmalıdır:

- Noktalar, düzlemde herhangi bir yere konumlanabilmeli: git fonksiyonu
- Noktalar bulundukları koordinatları ekranda gösterebilmeli: goster fonksiyonu
- Noktalar, sıfır (0,0) koordinatında olup olmadıkları sorusunu yanıtlayabilmeli: sifir_mi fonksiyonu

```

#include <iostream>

using namespace std;

class Nokta{
    int x,y;
public:
    void git(int, int);
    void goster();
    void sifir_mi();
};

void Nokta::git(int yeni_x, int yeni_y)
{
    x = yeni_x;
    y = yeni_y;
}

```

```

}

void Nokta::goster()
{
    cout << "X noktasi: " << x << ", Y noktasi: " << y << endl;
}

void Nokta::sifir_mi()
{
    if ((x == 0) && (y == 0))
        cout << "n1 su anda sifir noktasindadir." << endl;
    else
        cout << "n1 su anda sifir noktasinda degildir." << endl;
}

int main() {
    Nokta n1,n2;
    n1.git(78,34);
    n1.goster();
    n1.git(61,35);
    n1.goster();
    n1.sifir_mi();
    n2.git(0,0);
    n2.sifir_mi();
    return 0;
}

```

Kodun Çıktısı:

X noktasi: 78, Y noktasi: 34
 X noktasi: 61, Y noktasi: 35
 n1 su anda sifir noktasinda degildir.
 n1 su anda sifir noktasindadir.