
Ödev-04

1. Ödev-04

1.1. Prototype

Aşağıdaki örnek bir oyun içersindeki eşyalar ile ilgilidir.

Eşyalar ikiye ayrılmıştır.

- **Çevresel Eşyalar** : Kullanıcının sahip olmadığı hacimleri diğer örnekleri (instance) ile aynı olan eşyalardır (varil, kutu, vb.).
- **Kişisel Eşyalar** : Kullanıcının sahip olacağı ve hacimleri diğer örneklerinden farklı olabilircek eşyalardır. (zırh, silah, vb).

Hacim pointer olarak tanımlanmıştır.

Çevresel Eşyalar'da istenilen sonucun elde edilmesi için Shallow Copy yöntemi kullanılmıştır.

Kişisel Eşyalar'da istenilen sonucun elde edilmesi için Deep Copy yöntemi kullanılmıştır.

```
#include <iostream>

class Esys1{
public:
    int id;
    int *hacim;

    Esys1(){
        id = 0;
        hacim = new int(10);
    }

    virtual void yazdir() const
    {
        std::cout << "Esys1 esya: " << "ID: " << this->id << " Hacim: "
        << *this->hacim << std::endl;
    }
};
```

```
class Esya2{
public:
    int id;
    int *hacim;

    Esya2(){
        this->id = 0;
        this->hacim = new int(10);
    }

    // Copy Constructer
    Esya2(const Esya2& es)
    {
        this->id = es.id;
        this->hacim = new int(*es.hacim);
    }

    virtual void yazdir() const
    {
        std::cout << "Esya2 esya: " << "ID: " << this->id << " Hacim: "
        << *this->hacim << std::endl;
    }
};

// Shallow copy icin
class Cevresel_Esya : public Esya1{
public:

    Cevresel_Esya(const Esya1& esya1)
    {
        this->id = esya1.id;
        this->hacim = esya1.hacim;
    }

    void yazdir() const
    {
        std::cout << "Cevresel esya: " << "ID: " << this->id << " Hacim: "
        << *this->hacim << std::endl;
    }
};

// Deep copy icin
```

```

class Kisisel_Esya : public Esya2{
public:
    // Copy Constructer
    Kisisel_Esya(const Esya2& es)
    {
        this->id = es.id;
        this->hacim = new int(*es.hacim);
    }

    void yazdir() const
    {
        std::cout << "Kisisel esya: " << "ID: " << this->id << " Hacim: " << *this->hacim << std::endl;
    }
};

int main()
{
    // Esya1 esya11;
    // Esya1 esya12 = esya11;
    //
    // std::cout << "esya11: id:" << *esya11.id << " Hacim: " <<
    esya11.hacim << std::endl;
    // std::cout << "esya12: id:" << *esya12.id << " Hacim: " <<
    esya12.hacim << std::endl;
    //
    // std::cout << std::endl;
    // *esya11.id = 5;
    // esya11.hacim = 12;
    //
    // std::cout << "esya11: id:" << *esya11.id << " Hacim: " <<
    esya11.hacim << std::endl;
    // std::cout << "esya12: id:" << *esya12.id << " Hacim: " <<
    esya12.hacim << std::endl;
    //
    // std::cout << std::endl;
    // std::cout << "-----" <<
    std::endl;
    // std::cout << std::endl;
    //
    // Esya2 esya21;
    // Esya2 esya22 = esya21;
    //

```

```
//      std::cout << "esya21: id:" << *esya21.id << " Hacim: " <<
esya21.hacim << std::endl;
//      std::cout << "esya22: id:" << *esya22.id << " Hacim: " <<
esya22.hacim << std::endl;
//
//      std::cout << std::endl;
//      *esya21.id = 5;
//      esya21.hacim = 12;
//
//      std::cout << "esya21: id:" << *esya21.id << " Hacim: " <<
esya21.hacim << std::endl;
//      std::cout << "esya22: id:" << *esya22.id << " Hacim: " <<
esya22.hacim << std::endl;

Esys1 esya3;
Cevresel_Esys cev = esya3;

std::cout << "esya3: ";
esya3.yazdir();
std::cout << "cev: ";
cev.yazdir();

std::cout << std::endl;
esya3.id = 5;
*esya3.hacim = 12;

std::cout << "esya3: ";
esya3.yazdir();
std::cout << "cev: ";
cev.yazdir();

std::cout << std::endl;
std::cout << "-----" << std::endl;
std::cout << std::endl;

Esys2 esya4;
Kisisel_Esys kis = esya4;

std::cout << "esya4: ";
esya4.yazdir();
std::cout << "kis: ";
kis.yazdir();

std::cout << std::endl;
esya4.id = 5;
```

```

*esya4.hacim = 12;

std::cout << "esya4: ";
esya4.yazdir();
std::cout << "kis: ";
kis.yazdir();

return 0;
}

```

- ❶ Esya1 sınıfı tanımlanır.
- ❷ Esya2 sınıfı tanımlanır.
- ❸ Esya2 sınıfı için tanımlanan Copy Constructer. Bununla beraber '=' operatörü ile bir tanımlama gerçekleşiyorsa bu fonksiyon çalışacaktır.
- ❹ Cevresel_Esya sınıfının tanımlanması.
- ❺ Bir Cevresel_Esya örneği (instance) oluşturulurken girdi olarak Esya1 sınıfının örneğinin kullanılabilmesi için oluşturulan Constructer.
- ❻ Kisisel_Esya sınıfının tanımlanması..
- ❼ Kisisel_Esya sınıfı için tanımlanan Copy Constructer. Bu fonksiyonda girdi olarak Esya2 sınıfının bir örneği (instance) kullanılmaktadır.
- ❽ Burada Copy Constructer çağırılmamaktadır. Dolayısıyla pointer değişken kopyalanmayacaktır, atanacaktır.
- ❾ Burada Copy Constructer çağırılmaktadır. Dolayısıyla pointer değişken kopyalanmaktadır. Böylece yeni bir pointer oluşturulmaktadır.

UML Diagramı



