
Ödev-07

1. Ödev-07

1.1. Memento Pattern

Bu örnekte bir oyunun kaydetme süreci anlatılmaktadır. Kaydetme işlemi Memento tasarım kalıbı kullanılarak gerçekleştirilmiştir.

Burada Memento olarak kullanılan sınıf Save sınıfıdır.

```
#include <iostream>
#include <vector>

class Karakter{
public:
    int can;
    std::string isim;
    int pozisyonX;
    int pozisyonY;
    int pozisyonZ;

public:
    Karakter(int can, const std::string &isim, int pozisyonX, int
    pozisyonY, int pozisyonZ) : can(can), isim(isim),

        pozisyonX(pozisyonX),

        pozisyonY(pozisyonY),

        pozisyonZ(pozisyonZ) {}

    Karakter() : can(0), isim(""),
        pozisyonX(0),
        pozisyonY(0),
        pozisyonZ(0) {}

    void hasarAl(int hasar)
    {
        if(can - hasar > 0)
        {
            can -= hasar;
        }
    }
}
```

1

```
        } else{
            can = 0;
        }
    }
};
```

```
class Save{
public:
    int genislik;
    int yukseklik;
    std::string tur;
    Karakter karakter;
    Save(int genislik, int yukseklik, const std::string &tur, const
    Karakter &karakter) : genislik(genislik),

        yukseklik(yukseklik),

        tur(tur),

        karakter(karakter) {}

    friend class Harita;
};
```

```
class Harita{
public:
    int genislik;
    int yukseklik;
    std::string tur;
    Karakter karakter;

    Harita(int genislik, int yukseklik, const std::string &tur, const
    Karakter &karakter) : genislik(genislik),

        yukseklik(yukseklik),

        tur(tur),

        karakter(karakter) {}

    Save save()
```

```
{
    return {genislik,yukseklik,tur,karakter};
}
```

5

```

void load(const Save& s)
{
    genislik = s.genislik;
    yukseklik = s.yukseklik;
    tur = s.tur;
    karakter.can = s.karakter.can;
    karakter.isim = s.karakter.isim;
    karakter.pozisyonX = s.karakter.pozisyonX;
    karakter.pozisyonY = s.karakter.pozisyonY;
    karakter.pozisyonZ = s.karakter.pozisyonZ;
}

friend std::ostream &operator<<(std::ostream &os, const Harita
&harita) {
    os << "genislik: " << harita.genislik <<
    " yukseklik: " << harita.yukseklik <<
    " tur: " << harita.tur << std::endl <<
    "karakter: " <<
    " can: " << harita.karakter.can <<
    " isim: " << harita.karakter.isim <<
    " pozisyonX: " << harita.karakter.pozisyonX <<
    " pozisyonY: " << harita.karakter.pozisyonY <<
    " pozisyonZ: " << harita.karakter.pozisyonZ << std::endl;
    return os;
}
};

int main()
{
    Harita h1(100,100,"yesillik", Karakter(100, "Kahraman", 3,2,1));

    std::cout << h1 << std::endl;

    auto s1 = h1.save();

    h1.karakter.hasarAl(5);

    std::cout << h1 << std::endl;

    h1.load(s1);

    std::cout << h1 << std::endl;

    return 0;
}

```

}

- ❶ Karakter sınıfının tanımlanması.
- ❷ Save sınıfının tanımlanması. Memento olarak kullanılan sınıftır.
- ❸ Harita sınıfının tanımlanması. Kaydedilecek sınıftır.
- ❹ save() fonksiyonunun tanımlanması. Kaydetme işlemini gerçekleştiren fonksiyondur.
- ❺ load() fonksiyonunun tanımlanması. Yükleme işlemini gerçekleştiren fonksiyondur.

UML Diagramı

