

---

# Ödev-05

## 1. Ödev-05

### 1.1. Builder Pattern

Bu örnekte Builder Pattern ile oyun için karakter modeli oluşturulması ve üretilmesi anlatılmaktadır.

İnsan, köpek ve kuş olacak şekilde 3 tip model oluşturulmaktadır.

```
#include <iostream>

class Model {
public:
    void turAta(const std::string& tur1)
    {
        this->tur=tur1;
    }

    void kafayiOlustur(int kafa_uzunluk_x1, int kafa_uzunluk_y1, int kafa_uzunluk_z1)
    {
        this->kafa_uzunluk_x = kafa_uzunluk_x1;
        this->kafa_uzunluk_y = kafa_uzunluk_y1;
        this->kafa_uzunluk_z = kafa_uzunluk_z1;
    }

    void govdeyiOlustur(int govde_uzunluk_x1, int govde_uzunluk_y1, int govde_uzunluk_z1)
    {
        this->govde_uzunluk_x = govde_uzunluk_x1;
        this->govde_uzunluk_y = govde_uzunluk_y1;
        this->govde_uzunluk_z = govde_uzunluk_z1;
    }

    void kollariOlustur(int kol_sayisi1, int kol_uzunluk_x1, int kol_uzunluk_y1, int kol_uzunluk_z1)
    {
        this->kol_sayisi = kol_sayisi1;
        if(this->kol_sayisi <= 0)
        {
```

```
        this->kol_sayisi = 0;
        this->kol_uzunluk_x = 0;
        this->kol_uzunluk_y = 0;
        this->kol_uzunluk_z = 0;
        return;
    }
    this->kol_uzunluk_x = kol_uzunluk_x1;
    this->kol_uzunluk_y = kol_uzunluk_y1;
    this->kol_uzunluk_z = kol_uzunluk_z1;
}

void bacaklariOlustur(int bacak_sayisi1, int bacak_uzunluk_x1, int
bacak_uzunluk_y1, int bacak_uzunluk_z1)
{
    this->bacak_sayisi = bacak_sayisi1;
    if(this->bacak_sayisi <= 0)
    {
        this->bacak_sayisi = 0;
        this->bacak_uzunluk_x = 0;
        this->bacak_uzunluk_y = 0;
        this->bacak_uzunluk_z = 0;
        return;
    }
    this->bacak_uzunluk_x = bacak_uzunluk_x1;
    this->bacak_uzunluk_y = bacak_uzunluk_y1;
    this->bacak_uzunluk_z = bacak_uzunluk_z1;
}

void kanatlariOlustur(int kanat_sayisi1, int kanat_uzunluk_x1, int
kanat_uzunluk_y1, int kanat_uzunluk_z1)
{
    this->kanat_sayisi = kanat_sayisi1;
    if(this->kanat_sayisi <= 0)
    {
        this->kanat_sayisi = 0;
        this->kanat_uzunluk_x = 0;
        this->kanat_uzunluk_y = 0;
        this->kanat_uzunluk_z = 0;
        return;
    }

    this->kanat_uzunluk_x = kanat_uzunluk_x1;
    this->kanat_uzunluk_y = kanat_uzunluk_y1;
    this->kanat_uzunluk_z = kanat_uzunluk_z1;
}
```

```

std::string modelVerileri() {
    return ("Tur: " + this->tur + '\n' +
        "Kafa: \n" +
        "x: " + std::to_string(this->kafa_uzunluk_x) + ' ' +
        "y: " + std::to_string(this->kafa_uzunluk_y) + ' ' +
        "z: " + std::to_string(this->kafa_uzunluk_z) + '\n' +
        "Govde: \n" +
        "x: " + std::to_string(this->govde_uzunluk_x) + ' ' +
        "y: " + std::to_string(this->govde_uzunluk_y) + ' ' +
        "z: " + std::to_string(this->govde_uzunluk_z) + '\n' +
        "Kollar: \n" +
        "Sayi: " + std::to_string(this->kol_sayisi) + ' ' +
        "x: " + std::to_string(this->kol_uzunluk_x) + ' ' +
        "y: " + std::to_string(this->kol_uzunluk_y) + ' ' +
        "z: " + std::to_string(this->kol_uzunluk_z) + '\n' +
        "Bacaklar: \n" +
        "Sayi: " + std::to_string(this->bacak_sayisi) + ' ' +
        "x: " + std::to_string(this->bacak_uzunluk_x) + ' ' +
        "y: " + std::to_string(this->bacak_uzunluk_y) + ' ' +
        "z: " + std::to_string(this->bacak_uzunluk_z) + '\n' +
        "Kanatlar: \n" +
        "Sayi: " + std::to_string(this->kanat_sayisi) + ' ' +
        "x: " + std::to_string(this->kanat_uzunluk_x) + ' ' +
        "y: " + std::to_string(this->kanat_uzunluk_y) + ' ' +
        "z: " + std::to_string(this->kanat_uzunluk_z) + '\n'
    );
}

```

**private:**

```

std::string tur;
int kafa_uzunluk_x;
int kafa_uzunluk_y;
int kafa_uzunluk_z;
int govde_uzunluk_x;
int govde_uzunluk_y;
int govde_uzunluk_z;
int kol_sayisi;
int kol_uzunluk_x;
int kol_uzunluk_y;
int kol_uzunluk_z;
int bacak_sayisi;
int bacak_uzunluk_x;
int bacak_uzunluk_y;
int bacak_uzunluk_z;
int kanat_sayisi;

```

```
int kanat_uzunluk_x;  
int kanat_uzunluk_y;  
int kanat_uzunluk_z;  
};
```

```
class Builder {  
public:
```

②

```
//Abstract class  
Model modelOlustur() { return model; }
```

```
virtual void buildTur() = 0;  
virtual void buildKafa() = 0;  
virtual void buildGovde() = 0;  
virtual void buildKol() = 0;  
virtual void buildBacak() = 0;  
virtual void buildKanat() = 0;
```

```
protected:
```

```
Model model;  
};
```

```
class InsanBuilder : public Builder {  
public:
```

③

```
void buildTur() { model.turAta("Insan"); }  
void buildKafa() { model.kafayiOlustur(15,15,20); }  
void buildGovde() { model.govdeyiOlustur(40,20,60); }  
void buildKol() { model.kollariOlustur(2,10,10,60); }  
void buildBacak() { model.bacaklariOlustur(2,20,20,85); }  
void buildKanat() { model.kanatlariOlustur(0,0,0,0); }
```

```
};
```

```
class KopekBuilder : public Builder {  
public:
```

④

```
void buildTur() { model.turAta("Kopek"); }  
void buildKafa() { model.kafayiOlustur(15,20,15); }  
void buildGovde() { model.govdeyiOlustur(30,80,40); }  
void buildKol() { model.kollariOlustur(0,0,0,0); }  
void buildBacak() { model.bacaklariOlustur(4,6,6,30); }  
void buildKanat() { model.kanatlariOlustur(0,0,0,0); }
```

```
};
```

```
class KusBuilder : public Builder { ⑤
public:
    void buildTur() { model.turAta("Kus"); }
    void buildKafa() { model.kafayiOlustur(2,2,2); }
    void buildGovde() { model.govdeyiOlustur(5,5,10); }
    void buildKol() { model.kollariOlustur(0,0,0,0); }
    void buildBacak() { model.bacaklariOlustur(2,2,2,5); }
    void buildKanat() { model.kanatlariOlustur(2,5,1,5); }

};

class Creator { ⑥
public:

    void set(Builder *b) {
        if (builder)
            delete builder;

        builder = b;
    }

    void olustur() { ⑦
        builder->buildKol();
        builder->buildBacak();
        builder->buildGovde();
        builder->buildKafa();
        builder->buildKanat();
        builder->buildTur();
    }

    Model uret() { return builder->modelOlustur(); }

private:
    Builder *builder{nullptr};
};

int main() {
    Creator creator;
    creator.set(new InsanBuilder);
    creator.olustur();

    Model model1 = creator.uret();
    std::cout << model1.modelVerileri() << std::endl;
```

```
creator.set(new KopekBuilder);
creator.olustur();

Model model2 = creator.uret();
std::cout << model2.modelVerileri() << std::endl;

creator.set(new KusBuilder);
creator.olustur();

Model model3 = creator.uret();
std::cout << model3.modelVerileri() << std::endl;

return 0;
}
```

- ❶ Model sınıfının tanımlanması.
- ❷ Builder sınıfının tanımlanması. Alt builder sınıflar bu sınıftaki fonksiyonları kullanmaktadır.
- ❸ İnsanBuilder sınıfının tanımlanması.
- ❹ KopekBuilder sınıfının tanımlanması.
- ❺ KusBuilder sınıfının tanımlanması.
- ❻ Creator sınıfının tanımlanması
- ❼ Burada tanımlanan olustur() fonksiyonu ile Creator sınıfı örneği istenilen builderi oluşturur.

## UML Diagramı



