

```

public interface IEnemy
{
    void useGun();
    void run();
    void enemyName(String name);
}

```

```

public class EnemyRobot
{
    public void Punch()
    {

    }

    public void Run()
    {

    }

    public void RobotName(String name)
    {

    }
}

```

```

public class EnemyTank : IEnemy
{
    public void enemyName(string name)
    {
        throw new NotImplementedException();
    }

    public void run()
    {
        throw new NotImplementedException();
    }

    public void useGun()
    {
        throw new NotImplementedException();
    }
}

```

```

public class Adapter : IEnemy
{
    EnemyRobot enemyRobot;

    public Adapter(EnemyRobot e)
    {
        enemyRobot = e;
    }

    public void enemyName(string name)
    {
        enemyRobot.RobotName(name);
    }

    public void run()
    {
        enemyRobot.Run();
    }

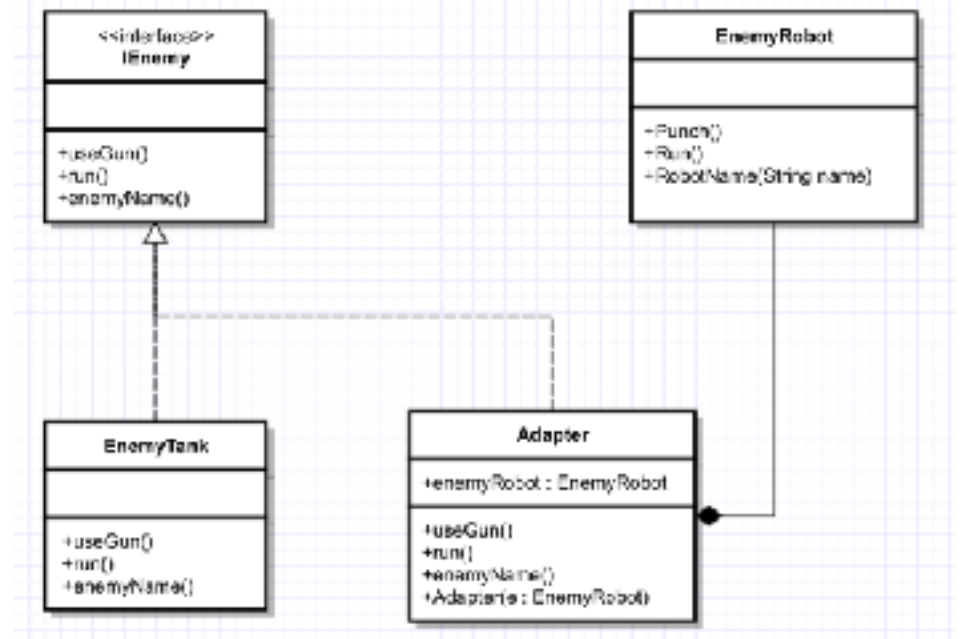
    public void useGun()
    {
        enemyRobot.Punch();
    }
}

```

YAZILIM MİMARİSİ VE TASARIMI

ÖDEV2

-142805009 BERTUĞ YILMAZ



Adapter Design Pattern, farklı interface'lerin sahip sınıflarının birbiriyle çalışmasını sağlamak amacıyla kullanılır. Kod tekrarını önlemek ve var olan modülleri adapte etmemize yol gösterir.

Projemde EnemyTank ile EnemyRobot arasında Adapter sınıfı kullandım. Adapter sınıfı IEnemy interface sine EnemyRobot sınıfını adapte etmemi sağlıyor.