**Interface Segregation Prensibi**

Inteface Segregation (Arayüz Ayrımı) prensibi sorumlulukların hepsini bir arayüze toplamak yerine sorumluluklara özgü birçok arayüz oluşturmayı önermektedir. Bu prensip büyük olan arayüzleri daha küçük ve daha özel arayüzlere böler, böylece alt sınıflar sadece ilgili arayüzü uygularlar. Böylelikle kod karmaşıklığı ve bağımlılığı da azaltılmış olur.

**Uygulama**

public abstract class Kullanici {  
 String mail, sifre;  
  
 public String getMail() {  
 return mail;  
 }  
  
 public void setMail(String mail) {  
 this.mail = mail;  
 }  
  
 public String getSifre() {  
 return sifre;  
 }  
  
 public void setSifre(String sifre) {  
 this.sifre = sifre;  
 }  
  
 abstract boolean girisYap(String mail, String sifre);  
 abstract void urunGoruntule(Urun urun);  
  
}

public interface SatisYapabilen {  
 void urunEkle(Urun urun);  
}

public class Satici extends Kullanici implements SatisYapabilen {  
  
 private Veritabani veritabani;  
  
 public Satici(String mail, String sifre) {  
 veritabani = new Veritabani();  
 setMail(mail);  
 setSifre(sifre);  
 }  
  
 @Override  
 boolean girisYap(String mail, String sifre) {  
 return veritabani.getSifreFromMail(mail).equals(sifre);  
 }  
  
 @Override  
 void urunGoruntule(Urun urun) {  
 System.*out*.println(getMail() + ", " + urun.getIsim() + ", " + urun.getFiyat() + " inceliyor.");  
 }  
  
 @Override  
 public void urunEkle(Urun urun) {  
 veritabani.urunEkle(urun);  
 }  
}

public class Musteri extends Kullanici {  
  
 private Veritabani veritabani;  
  
 public Musteri(String mail, String sifre) {  
 veritabani = new Veritabani();  
 setMail(mail);  
 setSifre(sifre);  
 }  
  
 public Veritabani getVeritabani() {  
 return veritabani;  
 }  
  
 public void setVeritabani(Veritabani veritabani) {  
 this.veritabani = veritabani;  
 }  
  
 @Override  
 public boolean girisYap(String mail, String sifre) {  
 return veritabani.getSifreFromMail(mail).equals(sifre);  
 }  
  
 @Override  
 public void urunGoruntule(Urun urun) {  
 System.*out*.println(getMail() + ", " + urun.getIsim() + " inceliyor.");  
 }  
}

public class Urun {  
  
 private String isim;  
 private double fiyat;  
  
 public Urun() { }  
 public Urun(String isim, double fiyat) {  
 this.isim = isim;  
 this.fiyat = fiyat;  
 }  
  
 public String getIsim() {  
 return isim;  
 }  
  
 public void setIsim(String isim) {  
 this.isim = isim;  
 }  
  
 public double getFiyat() {  
 return fiyat;  
 }  
  
 public void setFiyat(double fiyat) {  
 this.fiyat = fiyat;  
 }  
}

public class Veritabani {  
  
 private ArrayList<Kullanici> kullanicilar;  
 private ArrayList<Urun> urunler;  
  
 public Veritabani() {  
 kullanicilar = new ArrayList<Kullanici>();  
 urunler = new ArrayList<Urun>();  
 }  
  
 public String getSifreFromMail(String mail) {  
  
 for(Kullanici kullanici : kullanicilar) {  
 if(kullanici.getMail().equals(mail)) return kullanici.getSifre();  
 }  
 return "";  
 }  
  
 public void kullaniciEkle() {  
 Kullanici musteri = new Musteri("sefaaras@ktu.edu.tr", "123456");  
 Kullanici satici = new Satici("mhbozkurt@ktu.edu.tr", "654321");  
 kullanicilar.add(musteri);  
 kullanicilar.add(satici);  
 }  
  
 public void urunEkle(Urun urun) {  
 urunler.add(urun);  
 }  
}

public class Main {  
  
 public static void main(String[] args) {  
  
 Musteri musteri = new Musteri("sefaaras@ktu.edu.tr", "123456");  
 Satici satici = new Satici("mhbozkurt@ktu.edu.tr", "654321");  
 musteri.getVeritabani().kullaniciEkle();  
  
 if(musteri.girisYap("sefaaras@ktu.edu.tr", "123456")) {  
 System.*out*.println("Sisteme giriş yapıldı.");  
 }  
  
 Urun urun = new Urun("Bilgisayar", 7500);  
 satici.urunEkle(urun);  
  
 musteri.urunGoruntule(urun);  
 }  
}