NYP Uygulama 4

Java’da Büyük/Küçük Harf Kullanımı

Öğe Türü	İsimlendirme Kuralı	Örnek
Sınıf / Arayüz	PascalCase	BankAccount, UserProfile, DataProcessor
Metot / Değişken	camelCase	calculateInterest(), customerName, accountBalance
Sabitler (final)	BÜYÜK HARF, "_" ile ayrılmış	MAX_USERS, DEFAULT_TIMEOUT, PI
Paketler	küçük harf ve nokta ile ayrılmış	com.example.utils, org.myproject.service
Enum	PascalCase (türü), BÜYÜK HARF (sabitleri)	enum Day { MONDAY, TUESDAY, FRIDAY }

```
public class Main {
    public static void main(String[] args) {
        BankaHesabi hesap1=new BankaHesabi("Ali",1000);
        BankaHesabi hesap2=new BankaHesabi("Ayse",500);
        System.out.println(BankaHesabi.getHesapSayisi());
        System.out.println(BankaHesabi.toplamPara);
    }
}
```

Çıktı:

```
2
1500.0
```

SORU:

Üstte verilen kodun yandaki çıktığı sağlayacağı şekilde BankaHesabi sınıfını kodlayın.

İPUCU:

Statik metod ve alanlar kullanın.

```
class BankaHesabi{
    private String isim;
    private double bakiye;
    private static int hesapSayisi=0;
    public static double toplamPara=0;
    public BankaHesabi(String isim, double bakiye){
        this.isim = isim;
        this.bakiye = bakiye;
        hesapSayisi++;
        toplamPara+=bakiye;
    }
    public static int getHesapSayisi(){
        return hesapSayisi;
    }
}
```

```
public class Main {
    public static void main(String[] args) {
        System.out.println("30 °C = "+ Donusturucu.santigrattanFahrenheita(30)+"°F");
        System.out.println("20 km = "+ Donusturucu.kmdenMile(20)+" mil");
        System.out.println("r=2 => çevre= "+ Donusturucu.yariCaptanCevre(2));
        System.out.println(Donusturucu.kucukHarftenBuyukHarfe("Bugün gökyüzü masmavi, " +
                "çiçekler rengârenk ve kuşlar neşeyle şarkı söylüyor."));
    }
}
```

Donusturucu sınıfını yazın. Türkçe karakterler düzgün şekilde büyük harfe çevrilmelidir.

F = (C * 9 / 5) + 32;
mil = km * 0.621371

30 °C = 86.0°F

20 km = 12.42742 mil

r=2 => çevre= 12.56

BUGÜN GÖKYÜZÜ MASMAVİ, ÇİÇEKLER RENGÂRENK VE KUŞLAR NEŞEYLE ŞARKI SÖYLÜYOR.

```java
import java.util.Locale;
class Donusturucu {
    private static final double PI = 3.14;

    public static double santigrattanFahrenheita(double celsius) {
        return (celsius * 9 / 5) + 32;
    }



    public static double kmdenMile(double km) {
        return km * 0.621371;
    }
    public static double yariCaptanCevre(double r) {
        return 2*PI*r;
    }



    public static String kucukHarftenBuyukHarfe(String text) {
        return text.toUpperCase(Locale.forLanguageTag("tr-TR"));
    }
}
```

Factory

```
public class DatabaseFactory {
    public static Database getDatabase(String type) {
        if (type.equalsIgnoreCase("MySQL")) {
            return new MySQLDatabase();
        } else if (type.equalsIgnoreCase("PostgreSQL")) {
            return new PostgreSQLDatabase();
        } else {
            return null;
        }
    }
}
```

Factory

```
public class KarakterFabrikasi {
    public static Karakter karakterOlustur(String tur) {
        if (tur.equalsIgnoreCase("Savasci")) {
            return new Savasci();
        } else if (tur.equalsIgnoreCase("Buyucu")) {
            return new Buyucu();
        }
        return null;
    }
}
```

Factory

- Java dilinde, kullanıcı nesnelerinin doğrudan oluşturulmasını engellemek amacıyla private constructor kullanan bir 'Kullanici' sınıfı tasarlayınız. Bu sınıf, kullanıcı ismi ve rol bilgisini saklamalı ve bilgileri ekrana yazdıran bir metoda sahip olmalıdır.
- Nesne üretimini kontrol altına almak için 'KullaniciFabrikasi' adında iç statik bir sınıf tanımlayarak, 'Yönetici' ve 'Standart' kullanıcılar için ayrı üretim metotları oluşturunuz. Ayrıca, bu yapının kullanımını örnekleyen bir ana sınıf (Main) yazınız.

```
public class Main {
    public static void main(String[] args) {
        // Kullanıcıları fabrika metodları ile oluşturuyoruz
        Kullanici yonetici = Kullanici.KullaniciFabrikasi.yoneticiOlustur("Ahmet");
        Kullanici standartKullanici = Kullanici.KullaniciFabrikasi.standartKullaniciOlustur("Mehmet");

        // Kullanıcı bilgilerini gösterme
        yonetici.kullaniciBilgileriniGoster();
        standartKullanici.kullaniciBilgileriniGoster();
    }
}
```

```
class Kullanici {
    private String isim;
    private String rol;

    // Constructor private: Nesne doğrudan oluşturulamaz!
    private Kullanici(String isim, String rol) {
        this.isim = isim;
        this.rol = rol;
    }

    public void kullaniciBilgileriniGoster() {
        System.out.println(isim + " ==> Kullanıcı Türü: " + rol );
    }

    // Üretici metodları içeren sınıf
    public static class KullaniciFabrikasi {
        public static Kullanici yoneticiOlustur(String isim) {
            return new Kullanici(isim, "Yönetici");
        }

        public static Kullanici standartKullaniciOlustur(String isim) {
            return new Kullanici(isim, "Standart");
        }
    }
}
```