

Niřantařı Üniversitesi
EBLG205 Nesne Yönelimli Programlama Dersi Final Sınavı
Güz 2020

Öğretim Üyesi :Dr. Alper ÖZCAN

3 Şubat, 2021

Öğrenci Adı:

Öğrenci Numarası:

Önemli 3 Uyarı :

1. Bütün ödevler SanalKampüs sistemine zamanında yüklenmelidir , e-mail ile gönderilen ödevler kabul edilmeyecektir.
2. Gönderilen ödev dosyaları pdf veya word formatında olmalı dosyanın ismi öğrenci adı_soyadı_öğrenciNo şeklinde olmalıdır.
3. Bütün ödevler kopya intihal programı ile taratılacaktır ve ödevleri aynı olan öğrenciler hakkında işlem yapılacaktır.

[30 pts] Aşağıda verilen Class A, Class B, Class C sınıf tanımlamalarına göre verilen ifadeler hatalıysa, hatanın sebebini ve türünü (derleme zamanı, çalışma zaman) yazınız, hatalı değilse çıktısını yazınız. Sorunun a, b, c, d, e, f maddeleri ilgili maddelerin altına cevaplanmalıdır.

```
public class ClassA {  
  
    int n;  
  
    public ClassA () {  
        System.out.println("ClassA is created");  
    }  
  
    public ClassA(int n){  
        this.n = n;  
  
        System.out.println("ClassA is created with n: "  
            + this.n);  
  
    }  
  
    public void method1(){  
        System.out.println("Method1 is called with ClassA");  
    }  
}
```

```
public class ClassB extends ClassA {

    int m;

    public ClassB(){
        System.out.println("ClassB is created");
    }

    public ClassB(int n, int m){
        super(n);
        System.out.println("ClassB is created with m: " + m);
    }

    public void method1(){
        super.method1();
        System.out.println("Method1 is called with ClassB");
    }

    public void method2(){
        System.out.println("Method2 is called with ClassB");
    }
}
```

```
public class ClassC extends ClassA {

    public ClassC(){
        super();
        System.out.println("ClassC is created");
    }

    public void method1() {
        System.out.println("Method1 is called with ClassC");
    }

    public void method3(){
        method1();
        super.method1();
    }
}
```

(a) `ClassA instance1 = new ClassA();`
`instance1.method1();`

(b) `ClassA instance2 = new ClassB();`
`instance2.method1();`

(c) `ClassA instance2 = new ClassB();`
`instance2.method2();`

(d) `ClassA instance3 = new ClassC();`
`ClassB instance4 = (ClassB)(instance3);`

(e) `ClassA instance5 = new ClassB(2,3);`

(f) `ClassA instance6 = new ClassC();`
`((ClassC)instance6).method3();`

[45 pts] Aşağıdaki programların kodunu Java programlama dili ile yazınız.

a-)

Verilen bir metindeki kelimelerin harflerinin sıralamasını tersine çeviren ama kelimelerin sırlamasını değiştirmeyen bir program yazınız. Programın çıktısı aşağıdaki gibi olmalıdır:

```
Enter a text: Hello World
Reversed    : olleH dlroW
Enter a text: This week is going to be sunny
Reversed    : sihT keew si gniog ot eb ynnus
```

b-)

Parametre olarak bir metni String olarak alan ve bu metindeki her kelimenin ikinci küçük 'a' harfini büyüğüyle yani 'A' ile değiştiren bir metot yazınız. Eğer kelime 'a' ile başlıyorsa bu 'a' sayılmamalıdır. Metot değiştirilmiş metni String tipinde döndürmelidir. Programın çıktısı aşağıdaki gibi olmalıdır:

```
Enter a text: aabc
Converted    : aabc
Enter a text: deaa
Converted    : deaA
Enter a text: aeaa azAa
Converted    : aeaA azAa
```

c-)

Bir String nesnesindeki her '(' karakterine karşılık, daha sonra gelen bir ')' karakteri varsa o String nesnesine parantez dengelidir denir. Parametre olarak aldığı bir String nesnesinin parantez-dengeli olup olmadığını boolean tipinde döndüren bir metot yazınız. Bu metodu kullanarak çıktısı aşağıdaki gibi olan bir program yazınız.

```
Enter a String: ((a))
This string is not parenthesis-balanced.
Enter a String: )(a+b)
This string is parenthesis-balanced.
Enter a String: ((a+b)/(d-c))*e
This string is parenthesis-balanced.
```

[25 pts] Aşağıdaki programın kodunu Java programlama dili ile yazınız.

Parametre olarak iki nesne alan ve bu nesneleri `getWeight` metotlarına göre karşılaştırarak en ağır nesneyi döndüren `findHeaviest` metodunu yazınız. Bu metot tek tip parametrelili generic bir metot olmalıdır ve metodun tip parametresi için üst sınır `IPonderable` arayüzü olmalıdır. (Eğer parametre olarak aldığı iki nesnenin ağırlığı eşitse ikinci nesneyi döndürmelidir.)

```
public interface IPonderable {  
  
    double getWeight();  
}
```

```
public class Item1 implements IPonderable{  
  
    public double getWeight() {  
        return 80;  
    }  
  
    public String toString() {  
        return "Item1";  
    }  
}
```

```
public class Item2 implements IPonderable{  
  
    public double getWeight() {  
        return 50;  
    }  
  
    public String toString() {  
        return "Item2";  
    }  
}
```

Yukarıda verilen `Item1` ve `Item2` sınıflarına göre

```
Item1 item1 = new Item1();  
Item2 item2 = new Item2();  
System.out.println("Heaviest item is " + findHeaviest(item1, item2));
```

kod parçasının çıktısı

Heaviest item is Item1

olmalıdır.