**Isim:**

**Numara:**

**25.05.2017**

|  |  |  |
| --- | --- | --- |
|  | PAMUKKALE ÜNİVERSİTESİ**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ** **Nesne Yönelimli Programlama  2016-2017 EĞİTİM ÖĞRETİM YILI BAHAR DÖNEMİ DÖNEM SONU SINAVI** |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Soru** | **1 (30p)** | **2 (15p)** | **3 (25p)** | **4 (30p)** | **Toplam** |
| **Puan** |  |  |  |  |  |

**SORU 1(30p)** Tür ve birim fiyat örnek değişkenleri (instance variableları) olan ve nesnesi türetilemeyen bir Kahve sınıfı oluşturmanız isteniyor. Kahve sınıfında bodysi bu sınıfta implement (doldurulmayacak) edilmeyecek void dönüş tipli bir Component metodu oluşturunuz. Bu sınıftan türeyen iki sınıf daha hazırlamanız istenmektedir. Bu sınıflardan ilki, içerisinde *instance variable* olarak süt miktarı, şeker miktarı ve kahve miktarı instance variablelarını tutan Latte sınıfıdır. Bu sınıfın bir de ToString() metodu bulunmalıdır. ToString metodunun içerisinde “Latte nesnesiyim” yazılmalıdır. İkinci sınıf ise kahve miktarı, çikolata miktarı ve süt miktarı verileri için instance variableları bulunan Mocha sınıfıdır. Bu sınıfta da bir ToString() metodu bulunmalıdır. ToString metodunun içerisinde “Mocha nesnesiyim” yazılmalıdır. Mocha sınıfının içerisinde ayrıca Component() metodunun doldurulması istenmektedir. Bu metodun içerisinde nesnenin kahve miktarı, süt ve çikolata miktarı ekrana yazdırılmalıdır.

1. (15p) Gerekli instance variable, constructor ve metodlarıyla Kahve, Latte ve Mocha sınıflarını yazınız.

**public abstract class Kahve {**

**private String tur;**

**private int fiyat;**

**public Kahve(String tur, int fiyat) {**

**setTur(tur);**

**setFiyat(fiyat);**

**}**

**//setters**

**public void setTur(String tur) {**

**this.tur = tur;**

**}**

**public void setFiyat(int fiyat) {**

**this.fiyat = fiyat;**

**}**

**//getters**

**public String getTur() {**

**return this.tur**

**}**

**public int getFiyat() {**

**return this.fiyat;**

**}**

**public void component(){**

**}**

**}**

**public class Latte extends Kahve {**

**private double sutMiktari;**

**private double sekerMiktari;**

**private double kahveMiktari;**

**public Latte(double sutMiktari, double sekerMiktari, double kahveMiktari, int fiyat) {**

**super(“Latte”, fiyat);**

**this.sutMiktari = sutMiktari;**

**this.sekerMiktari = sekerMiktari;**

**this.kahveMiktari = kahveMiktari;**

**}**

**@Overrine**

**public String toString() {**

**return “Latte nesnesiyim”;**

**}**

**}**

**public class Mocha extends Kahve {**

**private double kahveMiktari;**

**private double cikolataMiktari;**

**private double sutMiktari;**

**public Mocha(double kahveMiktari, double cikolataMiktari, double sutMiktari, int fiyat) {**

**super(“Mocha”, fiyat);**

**this.kahveMiktari = kahveMiktari;**

**this.cikolataMiktari = cikolataMiktari;**

**this.sutMiktari = sutMiktari;**

**}**

**@Override**

**public String toString() {**

**return “Mocha nesnesiyim”;**

**}**

**@Override**

**public void component() {**

**System.out.printf(“Kahve: %.2f, Cikolata: %.2f, Sut: %.2f”, this.kahveMiktari, this.cikolataMiktari, this.sutMiktari);**

**}**

**}**

1. (10p) Main metodu içerisinde Kahve tipinde 4 elemanlı adı kahveArray olan bir array oluşturunuz. Bu array’in her bir elemanına %70 olasılıkla Latte ve %30 olasılıkla Mocha nesnesi atayınız. Ardından bu arayin elemanlarını foreach ile gezerek o iterasyondaki nesne eğer Mocha tipinde ise (çalışma zamanında karar verdiriniz) o nesnenin Component metodunu çalıştırınız.

**public class KahveTest {**

**import java.util.SecureRandom;**

**public static void main(String[] args) {**

**SecureRandom rand = new SecureRandom();**

**Kahve[] kahveArray = new Kahve[4];**

**for(int counter = 0; counter < 4; counter++) {**

**int ihtimal = rand(10) + 1;**

**if(1 <= ihtimal && ihtimal <= 7)**

**kahveArray[counter] = new Latte(0.2, 0.3, 0.5, 8);**

**else**

**kahveArray[counter] = new Mocha(0.3, 0.2, 0.5, 9);**

**}**

**for(Kahve kahve : kahveArray) {**

**if(kahve instanceof latte)**

**kahve.component();**

**}**

**}**

**}**

1. (5p) Interface ve Abstract sınıflar ne işe yarar? Hangi durumlarda Interface hangi durumlarda Abstract sınıf kullanmak mantıklı olur açıklayınız.

**Abstract: Eğer bir sınıf direkt olarak kullanılamasın, yalnızca türetilerek kullanılsın istiyorsak abstract olarak tanımlarız. Örneğin hayvan diye bir nesnenin oluşturulabilmesini istemeyiz. Kedi, köpek, kuş.. gibi nesnelerin hayvan sınıfından türetilmesini bekleriz. Bu gibi durumlarda abstract sınıf kullanılır. Eğer bir method abstract olarak tanımlanırsa, bu method’un çocuk tarafından kesinlike override edilmesi gerekir.**

**Interface: Sınıflarda kesinlikle kullanılacak belli başlı method’lar var ise bunlar interface’e yazılır. Sınıflar bu interface’den implements edilerek kullanılır. Bu sayede class’ların içersinde o method’ların kesinlikle yazılması sağlanır.**

**Soru 2(15p)** Şıklarda istenen bilgileri verecek SQL komutlarını yazınız.

*tblOgrenci*

|  |  |  |
| --- | --- | --- |
| **ogrenciId** | **isim** | **Ev Adresi** |
| 10000 | Ali | Denizli |
| 20000 | Veli | Denizli |
| 30000 | Ayse | İzmir |
| 40000 | Fatma | Afyon |

*tblDers*

|  |  |  |
| --- | --- | --- |
| **dersId** | **isim** | **kontenjan** |
| 100 | Algoritmalar | 32 |
| 200 | Ayrık Matematik | 40 |
| 300 | Paralel Programlama | 60 |

*tblDersKaydı*

|  |  |  |
| --- | --- | --- |
| **kayitID** | **dersId** | **ogrenciId** |
| 1 | 100 | 20000 |
| 1 | 200 | 20000 |
| 2 | 100 | 10000 |
| 3 | 300 | 30000 |

1. Denizli’de yaşayan öğrencilerin isimleri

**select isim from tblOgrenci where Ev Adresi = “Denizli”**

1. İsmi Ali olan öğrencinin seçmiş olduğu derslerin Id’leri

**select dersId from tblDersKaydı where ogrenciId = (select ogrenciId from tblOgrenci where isim = ‘Ali’)**

1. Algoritmalar dersi alan öğrencilerin isimleri

**select isim from tblOgrenci where ogrenciId = (select ogrenciId from tblDersKaydı where dersId = (select dersId from tblDers where isim = ‘Algoritmalar’))**

1. Kontenjanı miktarı 30’dan yüksek olan derslerin isimleri

**select isim from tblDers where kontenjan > 30**

1. Paralel Programlama dersinin kontenjanını 70 olarak güncelleyen komut

**update tblDers set kontenjan = ‘70’ where isim = ‘Paralel Programlama’**

**SORU 3(25p)** Aşağıdaki programlarda Hata var mı? Varsa doğrusu nedir?

**a)** public Interface IsSomethingWrong **YANLIS**  
 {  
 private static boolean buyukMu(int x, int y)**~~{~~**

**~~return (x > y);~~**

**~~}~~ ;**

public static void Main(String[] args){

int v1 = 14;

int v2 = 9; if (buyukMu(v1, v2))

Console.WriteLine(v1 + " is greater ");

else

Console.WriteLine(v2 + " is greater");

}

}

**b)**  abstract class IsSomethingWrong { **YANLIS**

public **abstract** void aMethod(int aValue);

}

**c)** public abstract class IsSomethingWrong **YANLIS**

{

public static void main(String[] args)

{

Student s **= new Student()**;

//student class has a “int grade()” method

Console.WriteLine("students grade.. " + s.grade());

}

}

**d)** class IsSomethingWrong **DOGRU**

{

static int i;

Object k;

void aMethod()

{

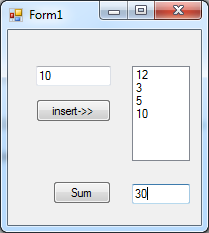
i++;

k.ToString();

}

}

**Soru 4. (30p)** Yandaki şekilde 5 adet kontrol bulunmaktadır, girisTextBox, toplaTextBox, numbersListBox, ekleBtn and toplaBtn. Aşağıdakileri istenenler için gerekli kodu yazınız.



a) ekleBtn basıldığında, girisTextBox’ta bulunan değeri numbersListBox’a ekleyin. Yalnızca integer değerler eklenmelidir, integer olmayan değerler için hata mesajı veriniz(try-catch kullanın).

private void ekleBtn\_Click(object sender, EventArgs e)

{

**try {**

**int value = Integer.valueOf(this.girisTextBox.getText());**

**this.numberListBox.addItem(“\n” + value);**

**} catch(Exception ex) {**

**JoptionPane.showMessageDialog(this, “Do not use any value that is not a integer”);**

**}**

}

b) toplaBtn’a basıldığında (Sum) listboxta bulunan tek sayıları toplasın ve alttaki toplaTextBox’a yazsın.

private void toplaBtn\_Click(object sender, EventArgs e)

{

**int sum = 0;**

**String[] values = numberListBox.getText().split(“\n”);**

**for(value : values) {**

**if((value%2) != 0) {**

**sum += value;**

**}**

**}**

**toplaTextBox.setText(String.valueOf(sum));**

}