

Yazılım Testi Dersi 2020-2021 Bahar Dönemi Ara Sınavı

SINAV KURALLARI

Sınav Başlama Saati: 14 Nisan Çarşamba Günü Saat: 16:00

Sınav Bitiş ve SABİS'e Son Yükleme Saati: 14 Nisan Çarşamba Günü Saat: 16:50

- Cevap kağıdınızda, **Adınız, Soyadınız, Numaranız, Şubeniz ve İmzanız** mutlaka olmalıdır.
- Cevaplar kurşun kalem ile A4 kağıdına el yazısı ile yazılıp daha sonra taranıp SABİS'e yüklenmelidir.
- Herhangi bir soru cevabının kopya olması durumunda her iki tarafta ara sınav notundan sıfır alacaktır.
- Mail üzerinden kesinlikle gönderim kabul edilmemektedir.
- Hangi sorunun cevabının yazıldığı cevap kağıdında açıkça belirtilmelidir.

1) Öğrenci numaranızın son rakamına göre aşağıdaki soruyu cevaplayınız. (20p)

Öğrenci Numarası Son Rakamı	Soru
0	Yazılımdaki tasarım ikilemi nedir açıklayınız?
1	Test case, test fixture, test terimlerini Google test çerçevesinde açıklayınız.
2	Test teorilerinde "Tam Test"e yaklaşım nasıl olmuştur?
3	Soyutlama ve arayüzlerin yazılım testi açısından önemini açıklayınız.
4	Taklit nesnelere ihtiyaç duyulacak bir senaryoyu anlatınız.
5	Yazılacak birim testlerinin private, public ve protected erişim niteleyicileri açısından açıklayınız.
6	Debug sürecinin yazılım testi açısından önemini belirtiniz.
7	Çevik yazılım sürecini ve yazılım testi ile ilişkisini açıklayınız.
8	Kod okunabilirliğini yazılım testi açısından önemini açıklayınız.
9	Dinamik ve statik testlerin yazılım testi açısından anlamını ve önemini açıklayınız.

2) Öğrenci numaranızın son rakamına göre aşağıdaki soruyu cevaplayınız. (20p)

Öğrenci Numarası Son Rakamı	Soru
0	Tek Sorumluluk İlkesini ihlal eden ve derste verilen örnekten farklı olacak şekilde Java dilinde yazıp açıklayınız.
1,2	Açık kapalılık İlkesini ihlal eden ve derste verilen örnekten farklı olacak şekilde Java dilinde yazıp açıklayınız.
3	Liskov Değiştirme İlkesini ihlal eden ve derste verilen örnekten farklı olacak şekilde Java dilinde yazıp açıklayınız.
4,5	Tek Sorumluluk İlkesini ihlal eden ve derste verilen örnekten farklı olacak şekilde C++ dilinde yazıp açıklayınız.
6,7	Açık kapalılık İlkesini ihlal eden ve derste verilen örnekten farklı olacak şekilde C++ dilinde yazıp açıklayınız.
8,9	Liskov Değiştirme İlkesini ihlal eden ve derste verilen örnekten farklı olacak şekilde C++ dilinde yazıp açıklayınız.

3) Aşağıda verilmiş olan metotlardan öğrenci numaranızın son rakamına göre birim testi yazınız. Öğrenci numaranızın sondan bir önceki rakamı tek ise sol taraftaki kodları dikkate alınız. Çift ise sağ taraftaki kodları dikkate alınız. Testi yazacağınız metodu seçmek için öğrenci numaranızın son rakamına göre hareket ediniz. (25p)

<pre> public class Ogrenci { private double genelNot; private double toplamNot; private int kredi; private Map<String, Double> aldigiDersler; public Ogrenci() { genelNot=0; toplamNot=0; kredi=0; aldigiDersler = new HashMap<String, Double>(); } ... } </pre>	<pre> class Ogrenci{ private: double genelNot; double toplamNot; int kredi; string *aldigiDersler; double *dersNotlari; int dersSayisi; bool stajYaptimi; public: Ogrenci() { genelNot=0; toplamNot=0; kredi=0; dersSayisi=0; aldigiDersler = new string[50]; dersNotlari = new double[50]; stajYaptimi=false; } ... }; </pre>
--	--

Öğrenci Numarası Son Rakamı	Öğrenci numaranızın sondan bir önceki rakamı tek ise (JUnit5 kullanılmalıdır.) Metodun birkaç durumu test edilmelidir.	Öğrenci numaranızın sondan bir önceki rakamı çift ise (Google Test kullanılmalıdır.) Metodun birkaç durumu test edilmelidir.
0	<pre> public double mezuniyetOrtalamasi() { if(kredi<240 genelNot < 2 !aldigiDersler.containsKey("Staj")) throw new IllegalArgumentException(); return genelNot; } </pre>	<pre> string* dersleriGetir(){ return aldigiDersler; } </pre>
1	<pre> public void aldigiDersEkle(String dersAdi,int kredi,double not) { aldigiDersler.put(dersAdi,not); toplamNot+=not; this.kredi+=kredi; genelNot = toplamNot/kredi; } </pre>	<pre> void stajYap() { if(dersSayisi < 20 stajYaptimi) throw "IllegalArgumentException"; aldigiDersler[dersSayisi] = "Staj"; dersNotlari[dersSayisi] = 4; toplamNot+=4; kredi+=2; genelNot = toplamNot/kredi; stajYaptimi=true; dersSayisi++; } </pre>
2	<pre> public String[] dersleriGetir() { String []dersler = new String[aldigiDersler.size()]; aldigiDersler.keySet().toArray(dersler); return dersler; } </pre>	<pre> double MezuniyetOrtalamasi(){ if(kredi<240 genelNot < 2) throw "IllegalArgumentException"; return genelNot; } </pre>

3	<pre> public String enDusukNotDers() { double min = Double.MAX_VALUE; String ders=""; for (Map.Entry<String, Double> girdi : aldigiDersler.entrySet()) { if (girdi.getValue().compareTo(min) < 0) { min = girdi.getValue(); ders = girdi.getKey(); } } return ders; } </pre>	<pre> void derslerEkle(string *dersler,int *krediler,double *notlar,int dersSayi) { for(int i=0;i<dersSayi;i++) { aldigiDersEkle(dersler[i],krediler[i],notlar[i]); } } </pre>
4	<pre> public void stajYap() { if(aldigiDersler.size() < 20 aldigiDersler.containsKey("Staj")) throw new IllegalArgumentException(); aldigiDersler.put("Staj",4.0); toplamNot+=4; this.kredi+=2; genelNot = toplamNot/kredi; } </pre>	<pre> void aldigiDersEkle(string dersAdi,int kredi,double notu) { aldigiDersler[dersSayisi] = dersAdi; dersNotlari[dersSayisi] = notu; toplamNot+=notu; this->kredi+=kredi; genelNot = toplamNot/kredi; dersSayisi++; } </pre>
5	<pre> public void derslerEkle(String[] dersler, int[] krediler,double[] notlar) { for(int i=0;i<dersler.length;i++) { aldigiDersEkle(dersler[i],krediler[i],notlar[i]); } } </pre>	<pre> string enDusukNotDers() { double min = numeric_limits<double>::max(); string ders=""; for (int i=0;i<dersSayisi;i++) { if (dersNotlari[i] < min) { min = dersNotlari[i]; ders = aldigiDersler[i]; } } return ders; } </pre>
6	<pre> public String[] yuksekDersleriGetir(double not) { Map<String, Double> yuksekDersler = new HashMap<String, Double>(); for (Map.Entry<String, Double> girdi : aldigiDersler.entrySet()) { if (girdi.getValue() >= not) { yuksekDersler.put(girdi.getKey(), girdi.getValue()); } } String []dersler = new String[yuksekDersler.size()]; yuksekDersler.keySet().toArray(dersler); return dersler; } </pre>	<pre> string enYuksekNotDers() { double max = numeric_limits<double>::min(); string ders=""; for (int i=0;i<dersSayisi;i++) { if (dersNotlari[i] > max) { max = dersNotlari[i]; ders = aldigiDersler[i]; } } return ders; } </pre>
7	<pre> public String enYuksekNotDers() { double max = Double.MIN_VALUE; String ders=""; for (Map.Entry<String, Double> girdi : aldigiDersler.entrySet()) { if (girdi.getValue().compareTo(max) > 0) { max = girdi.getValue(); ders = girdi.getKey(); } } } </pre>	<pre> double dersNotu(string ders) { for(int i=0;i<dersSayisi;i++){ if(aldigiDersler[i] == ders) return dersNotlari[i]; } throw "IllegalArgumentException"; } </pre>

	<pre> return ders; } </pre>	
8	<pre> public double dersNotu(String ders) { if(!aldigiDersler.containsKey(ders)) throw new IllegalArgumentException(); return aldigiDersler.get(ders); } </pre>	<pre> string* dusukDersleriGetir(double notu) { int kacAdet=0; for(int i=0;i<dersSayisi;i++){ if(dersNotlari[i] < notu) kacAdet++; } string *dusukDersler = new string[kacAdet]; for(int i=0,index=0;i<dersSayisi;i++){ if(dersNotlari[i] < notu){ dusukDersler[index] = aldigiDersler[i]; index++; } } return dusukDersler; } </pre>
9	<pre> public String[] dusukDersleriGetir(double not) { Map<String, Double> dusukDersler = new HashMap<String, Double>(); for (Map.Entry<String, Double> girdi : aldigiDersler.entrySet()) { if (girdi.getValue() < not) { dusukDersler.put(girdi.getKey(), girdi.getValue()); } } String []dersler = new String[dusukDersler.size()]; dusukDersler.keySet().toArray(dersler); return dersler; } </pre>	<pre> string* yuksekDersleriGetir(double notu) { int kacAdet=0; for(int i=0;i<dersSayisi;i++){ if(dersNotlari[i] >= notu) kacAdet++; } string *yuksekDersler = new string[kacAdet]; for(int i=0,index=0;i<dersSayisi;i++){ if(dersNotlari[i] >= notu){ yuksekDersler[index] = aldigiDersler[i]; index++; } } return yuksekDersler; } </pre>

4) Aşağıda verilmiş olan Game sınıfına **öğrenci numarasının sondan bir önceki rakamı tek olanlar Play1** isimli metoda, **çift olanlar Play2** isimli metoda test verisi, parametre olarak öğrenci numaralarını (b191210060) formatında gönderecekler. Daha sonra mutasyon testi için öğrenci numarası ile ilgili mutasyonları kullanacaklar (Toplamda 3 adet mutasyon). Mutasyon skoru kaçtır? Kaç adet mutant öldürülmüştür? (35p)

Öğrenci Numarası Son Rakamı	Play1 için Mutasyonlar Sondan bir önceki rakam tek olanlar	Play2 için Mutasyonlar Sondan bir önceki rakam çift olanlar
0,1,2	<pre> str.substring(1).toCharArray(); yerine str.substring(2).toCharArray(); int []sh = new int[bh.length]; yerine int []sh = new int[str.length()]; a = a%9; yerine a = a%8; </pre>	<pre> str = str.substring(1); yerine str = str.substring(2); String []s = str.split("0"); yerine String []s = str.split("1"); j[index] += yerine j[index] = </pre>
3,4,5,6	<pre> Arrays.stream(sh).sum()/a; yerine Arrays.stream(sh).sum(); sh[index++] = yerine sh[index++] += </pre>	<pre> str = str.substring(1); yerine str = str.substring(3); String []s = str.split("0"); yerine String []s = str.split("2"); </pre>

	return Arrays.stream(sh).sum()/a; yerine return (int)Arrays.stream(sh).count();	if(t > k) yerine if(t >= k)
7,8,9	str.substring(1).toCharArray(); yerine str.substring(3).toCharArray(); Character.getNumericValue(b); yerine Character.getNumericValue(2); a = a%9; yerine a = 3;	int []j = new int[s.length]; yerine int []j = new int[str.length()]; String []s = str.split("0"); yerine String []s = str.split("3"); index++; yerine index=1;

```

public class Game {
    public int Play1(String str) {
        if(str.length() != 10) return -1;
        char[] bh = str.substring(1).toCharArray();
        int []sh = new int[bh.length];
        int index=0;
        for(var b : bh) {
            sh[index++] = Character.getNumericValue(b);
        }
        int a = (int)Arrays.stream(sh).average().getAsDouble();
        a = a%9;
        return Arrays.stream(sh).sum()/a;
    }
    public int Play2(String str) {
        str = str.substring(1);
        String []s = str.split("0");
        int []j = new int[s.length];
        int index=0;
        for(var x : s) {
            if(x.isEmpty()) continue;

            for(var y : x.toCharArray()) {
                j[index] += Character.getNumericValue(y);
            }
            index++;
        }
        int k = Integer.MIN_VALUE;
        for(var t : j) {
            if(t > k) k=t;
        }
        return k;
    }
}

```