



- ✓ Soru kâğıdına **adınız**, **soyadınız** ve **numaranız** dışında başka hiçbir şey yazmayınız.
- ✓ Sınav süresi **100** dakikadır.
- ✓ Sorular **eşit** puanlıdır.

```
public class AnaAktivite extends Activity {
    private Button mDugme;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.ana_aktivite);
        mDugme = (Button)findViewById(R.id.dugme);
        mDugme.setOnClickListener(new OnClickListener() {

            @Override
            private void onClick(View v) {
                Toast.makeText(AnaAktivite.this, "Merhaba", Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Yukarıda Android projesine ait Java programlama dilinde yazılmış bir kod parçası verilmiştir. Buna göre aşağıdaki soruları cevaplayınız.

- Soru 1:**
AnaAktivite alt sınıfı sadece Activity süper sınıfından türetilmiştir. Bu bağlamda, Java programlama dilinin niçin çoklu mirası desteklemediğini kısaca açıklayınız. Java programlama dilinde, çoklu miras yaklaşımı yerine kullanılmak üzere başka nasıl bir yaklaşım geliştirilmiştir? Örnek bir program kod parçası üzerinden açıklayınız.
- Soru 2:**
AnaAktivite alt sınıfı, public erişim belirteci kullanılarak tanımlanmıştır. Bu türden tanımlama yapılmasının ana nedenini kısaca açıklayınız. AnaAktivite alt sınıfı hiçbir erişim belirteci kullanılmadan tanımlansaydı ne türden bir sorunla karşılaşılabilirdi? Kısaca açıklama yapınız.
- Soru 3:**
onCreate metodu @Override olarak tanımlanmıştır. Bu türden tanımlama yapılmasının ana sebebini, örnek bir program kod parçası üzerinden kısaca açıklayınız.
- Soru 4:**
Program kod parçasında, üst sınıfın onCreate metodu çağrılmış ve Bundle tipindeki bir parametre metoda aktarılmıştır ve bu bağlamda uygulamanın istenmeyen bir nedenden dolayı aniden kapatılması durumunda uygulamaya ait kaydedilen bazı bilgiler tekrardan uygulamaya aktarılmaktadır. Bu durumda super anahtar kelimesinin kullanım nedenini, örnek bir program kod parçası üzerinden kısaca açıklayınız.
- Soru 5:**
Android projelerinde, ekran tasarımı için gerekli olan kodlama, klasik Java projelerinin aksine xml uzantılı dosyalarda yapılmaktadır. Bunun ana sebebi ne olabilir? Kısaca açıklama yapınız.
- Soru 6:**
Android projelerinde, derlenmiş kodların yorumlamasında, Java sanal makinesi yerine niçin Dalvik sanal makinesi kullanılmaktadır? Kısaca açıklayınız.
- Soru 7:**
Program kod parçasında, onCreate metodu, Activity süper sınıfında final olarak tanımlanabilir miydi? Örnek bir program kod parçası üzerinden kısaca açıklayınız.
- Soru 8:**
Program kod parçasında, mDugme üye değişkeni Button tipinde bildirilmiştir. Bu türden tanımlı Android sınıfları, ilgili projeye nasıl dâhil edilebilir? Ayrıca mDugme üye değişkeni, final olarak tanımlansaydı uygulamanın koşulmasında herhangi bir sorunla karşılaşılır mıydı? Örnek bir program kod parçası üzerinden kısaca açıklayınız.
- Soru 9:**
Yukardaki program kod parçasında, anonim bir iç sınıf tanımlaması yapılmıştır. Java programlama dilinde iç sınıf tanımlamasının klasik sınıf tanımlamalarına göre farklılıkları nelerdir? Örnek bir program kod parçası üzerinden kısaca açıklayınız.
- Soru 10:**
Yukardaki program kod parçasında, onClick metodu View tipinde bir parametre kullanılarak tanımlanmıştır ve View tipindeki parametre metodun gövdesinde kullanılmamıştır. Aşlında kod parçasında herhangi bir düğmeye tıklama olayı gerçekleştirildiğinde ekrana bir ileti yazılmaktadır. Bu metodun, birden çok düğmenin tıklanması olaylarında çağrıldığı düşünülürse, View tipindeki bu parametrenin metodun gövdesinde nasıl kullanılabileceğini, örnek bir program kod parçası üzerinden kısaca açıklayınız.



KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
Bilgisayar Mühendisliği Bölümü
2015-2016 Eğitim-Öğretim Güz Yarıyılı
Windows Programlama Arasınnav Soruları



Cevap 1:

Java programlama dili basit ve tam anlamıyla bir nesne tabanlı bir programlama dili olduğundan karışıklığı önlemek açısından çoklu mirası desteklemez. Ama, benzer bir yaklaşımla çoklu arabirimi destekler.

```
interface Arabirim1 { /*...*/ }  
interface Arabirim2 { /*...*/ }
```

```
class Ornek implements Arabirim1, Arabirim2 { /*...*/ }
```

Cevap 2:

Paket dışından erişim olması için public olarak tanımlanmıştır. Erişim belirteci kullanılmıyorsa, sınıfa, sadece paket içi erişim olurdu.

Cevap 3:

Miras alma işlemlerinde, bir metod süper sınıfta tanımlı değilse, program koşulmadan da hata iletisi verilmesi için, metodu @Override ile birlikte tanımlamak daha mantıklıdır.

```
class A {  
    void f() { /*...*/ }  
}
```

```
class B extends A {  
    @Override  
    void f() { /*...*/ }  
}
```

Cevap 4:

super anahtar kelimesi, üstsınıfın aynı adlı üye değişkenlerine ve metotlarına erişim için kullanılmaktadır.

```
class A {  
    void f() { /*...*/ }  
}
```

```
class B extends A {  
    @Override  
    void f() { super.f(); }  
}
```

Cevap 5:

Uygulamadaki program kod parçalarıyla ekran tasarımı için gerekli olan kodlamaların birbirinden ayrıştırılması, kodun hem bakımı hem de anlaşılabilirliği açısından önemlidir.

Cevap 6:

Uygulamanın kodunun Android cihazlara uygunluğunu en iyileştirmek için Dalvik sanal makinesi kullanılmaktadır.

Cevap 7:

Miras işlemlerinde bir metoda aşım bindirme yapılacaksa, o metod final ve private olarak tanımlanamaz.

```
class A {  
    final private void f() { /*...*/ }  
}
```

```
class B extends A {  
    @Override  
    void f() { super.f(); } //hata  
}
```

Cevap 8:

import ile projeye kütüphaneler dâhil edilmektedir. mDugme, final olarak tanımlıysaydı atama işleminde sorun oluşurdu.

```
import android.widget.Button;  
  
public class AnaAktivite extends Activity {  
    private final Button mDugme; //hata.  
    //...  
}
```

Cevap 9:

Bazı farklılıklar vardır. Örneğin, dış sınıf, iç sınıfın üye değişkenlerine erişemez, ama tersi geçerlidir. İç sınıf, private olarak tanımlanabilir. Dış sınıfın erişimi açısından bir sorun oluşturmaz.

```
private class Tikla implements OnClickListener {  
    @Override  
    public void onClick(View v) { /*...*/ }  
}
```



KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
Bilgisayar Mühendisliği Bölümü
2015-2016 Eğitim-Öğretim Güz Yarıyılı
Windows Programlama Arasınava Soruları



Cevap 10:

if yapısıyla View tipindeki parametrenin hangi düğmeye tıklandığına dair bilgisi kontrol edilir ve ona göre işlem yapılabilir.

```
public void onClick(View v) {  
    if(v == mDugme1){  
        //...  
    }  
    else if(v == mDugme2){  
        //...  
    }  
    //...  
}
```