

HW_8_SONUC_RAPORU

MÜH.TĞM. MUHAMMET YURTSEVEN

Soru 1- Cevap D

Soru 2 Cevap B: Doğru blok sırası try,catch, ve finally .

Soru 3- Cevap

Soru 4 - Cevap A : Exception ve RuntimeException Java uygulamalarında sık sık yakalanırken, Hata yakalanmamaları önerilir. Bir Hata genellikle JVM'nin kurtarılamayan bir arızasını gösterir. Bu nedenle, Seçenek A doğrudur

Soru 5 - Cevap D

Soru 6 – Cevap B: IOException, kullanıldığında işlenmesi veya bildirilmesi gereken, Seçenek B'yi doğru yanıt haline getiren, denetlenen bir exceptiondır.

Soru 7 – Cevap A : Throws anahtar sözcüğü yöntem bildirimlerinde kullanılırken,throw anahtar sözcüğü, çevresindeki A işlemine bir exception atmak için kullanılır ve Seçenek A'yı doğru yanıtlayıcı yapar. Catch anahtar sözcüğü, exceptionları işlemek için kullanılır, bunları oluşturmak için değil veya bir yöntemin bildiriminde kullanılır.

Soru 8 – Cevap B : IOException, Exception'ın bir alt sınıfıdır, bu nedenle ilgili catch bloklarında ilk olarak görünmelidir. Exception'ın IOException öncesinde görüncekse, atılan herhangi bir IOException zaten Exception catch bloğu tarafından işlendiği için IOException bloğuna erişilemez kod olarak kabul edilir. Bu nedenle, Seçenek B doğrudur.

Soru 9- Cevap D

Soru 10- Cevap C

Soru 11- Cevap B: NullPointerException ve ArithmeticException ögelerinin her ikisi de işaretlenmeyen özel durumlar olan ve atıldıkları yöntemde işlenmeleri veya bildirilmeleri gerekmeyen RuntimeException özelliğini genişletir. Öte yandan, Exception kontrol edilmiş bir exceptiondır ve atıldığı yöntemle ele alınmalı veya bildirilmelidir. Bu nedenle, Seçenek B doğru cevaptır.

Soru 12- Cevap A

Soru 13- Cevap B: Sonunda ifade { } parantez gerektirir.

Soru 14 Cevap C

Soru 15- Cevap C : Bir try deyimi bir catch veya bir finally blok gerektirir. Bunlardan biri olmadan kod derlenmeyecektir; Bu nedenle, Seçenek D yanlıştır. Bir try deyimi hem catch hem de finally blok ile kullanılabilir, bu da Seçenek C'yi doğru cevap haline getirir.

Soru 16- Cevap B : Seçenek A, Exceptionlar genellikle işler yanlış gittiğinde veya beklenen yoldan saptığında kullanılır. Seçenek B yanlış ifadedir ve doğru cevaptır. Bir exception atan bir uygulama exceptionı ele almayı ve feshedilmeyi önleyebilir. Seçenek C doğru bir ifadedir. Örneğin, bir NullPointerException nesnesini kullanmaya çalışmadan önce null olup olmadığını test ederek null bir nesne üzerinde önlenebilir. Seçenek D de doğru bir ifadedir. Beklenmedik sorunlardan kurtulmaya çalışmak, uygun exception işlemenin önemli bir özelliğidir.

Soru 17- Cevap D

Soru 18- Cevap B : Exception, PrintException'dan daha geniş bir kontrol edilen exception olduğundan, Seçenek B'ye izin verilmez ve doğru seçimdir.

Soru 19- Cevap D: Bu sınıfların üçü de java.lang paketine aittir, bu nedenle Seçenek C doğru cevaba benziyor. Java derleyicisi, java.lang varsayılan olarak, bu nedenle bu üç sınıfı kullanmak için hiçbir import deyimi gerekmez ve bu da D Seçeneği'nin doğru cevabı yapar.

Soru 20- Cevap C

Soru 21- Cevap B : Checked exceptionlar ele alınmalı veya bildirilmelidir, aksi takdirde program derlenmez, ancak işaretlenmeyen istisnalar isteğe bağlı olarak ele alınabilir. Öte yandan, java.lang.Error hiçbir zaman uygulama tarafından ele alınmamalıdır, çünkü genellikle JVM'de belleğin bitmesi gibi kurtarılamaz bir durumu gösterir. Bu nedenlerle, Seçenek B doğru cevaptır.

Soru 22- Cevap

Soru 23- Cevap A : Bu senaryoda, ilk türden atılan bir exception her ikisiyle de eşleşir.blokları yakalar, ancak yalnızca ilk yakalama bloğu yürütülür, çünkü daha spesifik eşleşmedir.

Soru 24- Cevap

Soru 25- Cevap D

Soru 26- Cevap B : Bir program sonsuz bir döngüye çok derin bir şekilde geri döndüğünde StackOverflowError oluşur. Bu bir hata olarak kabul edilir, çünkü JVM genellikle belleği yetersizdir ve kurtarılamaz. Bir NullPointerException, null başvurusunda bir örnek yöntemi veya değişkeni kullanıldığında oluşur. Bu nedenlerle, Seçenek B doğrudur. NoClassDefFoundError derleme zamanında kullanılabilir kod çalışma zamanında kullanılamadığında oluşur. Bir nesne uyumsuz bir başvuru türüne verildiğinde bir ClassCastException oluşur. Son olarak, bir IllegalArgumentException amethod'a geçersiz parametreler gönderildiğinde oluşur.

Soru 27- Cevap C: Exceptionların hiçbir zaman uygulamanın sonlandırılmasına neden olmamasını sağlamak için.

Soru 28- Cevap D

Soru 29- Cevap A : Sonunda bir bloğa sahip olmak için bir try ifadesi gerekli değildir, ancak eğer varsa, bir çoğunu olabilir. Ayrıca, bir try ifadesinde herhangi bir sayıda catch bloğu olabilir. Bu nedenlerden dolayı, Seçenek A doğru cevaptır.

Soru 30- Cevap D

Soru 31- Cevap B: Hem catch hem de finally bloklar bir exception atarsa, finally bloktan olanı arayan tutulur, catch bloğundan olanı bırakılır ve Seçenek B doğru cevap olur.

Soru 32- Cevap

Soru 33- Cevap

Soru 34- Cevap C : Java'daki tüm exceptionlar Throwable'dan devralır ve Seçenek C'yi doğru yanıt haline getirir.

Soru 35- Cevap

Soru 36- Cevap A: ClassCastException, RuntimeException ögesinin bir alt sınıfıdır. ClassCastException RuntimeException catch bloğu tarafından zaten ele alınmıştır.Bu nedenle, Seçenek A doğrudur.

Soru 37- Cevap C : A seçeneği yanlış. Bilgisayar yanıyorsa muhtemelen yardım gerekecektir. Derleme kodu yanlış çalıştığından ve bu nedenle exceptionlar atmadığından Seçenek B yanlıştır. Seçenek C en iyi yanıttır, çünkü bir arayanı eksik veya geçersiz verilerle uyarmak için bir IllegalArgumentException kullanılabilir. Seçenek D yanlıştır; daha erken bitirmek nadiren bir problem olarak kabul edilir.

Soru 38- Cevap C

Soru 39- Cevap

Soru 40- Cevap D

Soru 41- Cevap C

Soru 42- Cevap

Soru 43- Cevap

Soru 44- Cevap

Soru 45- Cevap C : Hem `IllegalArgumentException` hem de `ClassCastException`, `RuntimeException` ögesini devralır, ancak ikisi de diğerinin alt sınıfı değildir. Bu nedenle, Seçenek C'yi doğru ifade yaparak her iki sırada da listelenebilirler.

Soru 46- Cevap

Soru 47- Cevap D

Soru 48- Cevap C: Bir kullanıcı şifresini yanlış giriyor ve sonuç bu olacaktır.`java.lang.Error`

Soru 49- Cevap C

Soru 50- Cevap