23.05.2015

**1Z0-803 NOTLAR**

**Chapter-1**

1. Bu bir hata değil, derlenir!! Ama maini bulamaz. Maini bulamadım der!( it throws a java.lang.NoSuchMethodError.)

public class TestClass{  
  public static long main(String[] args){  
     System.out.println("Hello");  
     return 10L;  
  }  
}

1. Import static ile diğer sınıfların sadece static değişken ve metodları alınır. Sınıfı bu şekilde import edemeyiz ama yıldız kullanarak içindeki bütün static değişkenleri alabiliriz.
2. Since the question says, "...an instance of the class is not needed...", the method has to be static.
3. this = new Test( ); //this değiştirilemez!! Sınıfın kendisi demektir aslında. Static metodlardan this e erişilemez çünkü this bir object field dır.
4. Since the code is using IOException, the java.io package (or just java.io.IOException class) must be imported. Only java.lang package is imported automatically.
5. //LOCAL DEĞİŞKENLER OTOMATİK İNİTALİZE OLMAZLAR!!!!! int localDegisken ; //Bu satır hata vermez //localDegisken++; //BU SATIR HATA VERİR! (Burada bir durum hariçtir. Eğer local değişken bir if içinde initalize oluyor ise hata vermez. Veya bir swich içinde. Ama herhangi bir durumda swich içinde initalize olamayacağı bi durum var ise de compiler hata verir.)
6. New ile yaratılan bir nesne bşka bir değişkene eşitlenir ise o nesne yaşamaya devam eder. GC tarafından toplanmaz. Mantık new ile yaratılan nesneyi sürekli birileri işaret ederek gösteriyor. Eğer onu işaret eden kalmaz ise ondan sonra GC tarafından toplnanır.
7. A top level class (i.e. a class not defined inside any other class) can only be public or have default access. (ve abstract , final alabilir sadece)
8. Note: a class declaration can have only final, abstract and public as modifiers, unless it is a nested class, in which case, it can be declared private or  protected as well. Within a method, a local variable may be declared as final.
9. Bir java programı parametresiz çağrılırsa args dizisi null değil elemanı 0 olan dizi olur. public static void main( String args[ ] ){        if (args.length == 0 )
10. Remember that the args array is never null. If the program is run without any arguments, args points to a String array of length 0
11. float density = 20.0; Bu satır hata verir sayının yanına f yazmak mecburidir !!!!!
12. You can suggest the JVM to perform garbage collection by calling System.gc();

**Chapter-2**

1. stringBuilder sb.setLenght(5) karaketer sayısını 5’e set eder.
2. sb.ensureCapacity(100); sb’nin uzunluğunu en az 100 yapar.
3. Wrapper objects are always immutable. Fonksiyona geçildiği zaman değeri değişmez, aynı string gibi metod bu nesneyi döndürmedikçe değeri değişmez.
4. Remember that wrapper classes (java.lang.Boolean, java.lang.Integer, java.lang.Long, java.lang.Short etc.) are also final and so they cannot be extended.
5. java.lang.Number, however, is not final. Integer, Long, Double etc. extend Number.
6. java.lang.System is final as well.
7. Actually, String class itself is final and so all of its methods are implicitly final.
8. StringBuilder extends Object
9. String, StringBuilder, and StringBuffer - all are final classes.
10. String nesnesinde equalsIgnoreCase(String), intern() metodları vardır.
11. String de intern diye bi metod vardır. Bir string oluştururken, string pool da var ise pool’dakini döndürür yok ise yeni oluşturur.
12. String de toLowerCase metodu vardır.
13. String de compareTo metodu soldaki string’in unicode değerinden sağdakini çıkarır ve kalanı döndürür. Farklı harf bulana kadar devam eder. Eşit ise stringler 0 döndürür.
14. Append string de yoktur. This method is in StringBuffer and StringBuilder but not in String.
15. String deki replace yeni bir string nesnesi yaratır fakat eğer parametreler aynı ise yeni string yaratmaz aynı nesneyi döndürür.
16. Stringler soldan sağa toplanarak gelir başka bir string bulana kadar. (Expressions evaluate soldan sağa)
17. String e null değeri atayıp ona ekleme yapar isen her zaman içinde string olarak null yazar.
18. Bir metodda s = s.replace('j', 'l'); komutu var ise bu s = new String(s.replace(a,a)) demektir. Medota giden string objelerini asla değiştiremezsin. Anca metodun dönüş tipini string yapıp metod çağırırkenn bu objeye eşitlersen bir nevi değiştirmiş olursun.
19. StringBuilder’ın nesneyi değiştirmeyen metodu = ?
20. All operands of type byte, char or short are promoted AT LEAST to an int before performing mathematical operations. If one of the operands is larger than an int then the other one is promoted to the same type. Note that System.out.println((float)5/4); will print 1.25. If you remove the explicit cast (float), it will print 1.(5/4) performs integer division because both 5 and 4 are integers, resulting in the value 1.Both operands in the expression ( 'a' + 1 ) will be promoted to int => 97 + 1 = 98
21. byte b = -128 ; kabul edilir ama +128 sıgmaz!!
22. byte and int both hold signed values.
23. int i = 0, j = 10 OK doğru bir atamadır.
24. char c = '\t'; ile char için tab karakteri konabilir.
25. Byte char alabilir. Byte ın sınırları şöyledir; byte bbb = 'm' ; bbb = 127; bbb = -128;
26. A floating point number written in binary or hex cannot use any suffix for float. 0b10\_000D; D ve F harflerini 0b ve 0x önde varsa almaz.
27. This is invalid because the floating point suffices f, F or d, D are used only when using decimal system and not while using binary. However, since f is a valid digit in hexadecimal system, a hex number may end with an f although it will not be interpreted as float but as the digit f. Thus, float x = 0x10\_000f; and float x = 10\_000f; are valid because they are written in hex and decimal respectively but float x = 0b10\_000f;  is invalid because is written in binary. Note that a floating point number cannot be written in Octal. Therefore, float x = 010\_000f; is valid but it is not octal even though it starts with a 0. It is interpreted in decimal.
28. Note that when you specify the bit pattern using binary or hex, an explicit cast is not required even if the number specified using the bit pattern is larger than what a float can hold.( float x = 0b10\_000;)

**Chapter-3**

1. The expression b1 = i1 == i2 will be evaluated as b1 = (i1 == i2) because == has higher precedence than =
2. You should understand here that instanceof operator returns true even if the Right Hand Side is a super class.
3. o1 instanceof o1 (HATALI! \_o1 sınıf ismi olmalıdır.) ilk kısımdaki değişkende object olmalı, primitive type olamaz.
4. A instanceof b operatörü a ile b arasında is-a ilişkisi yok ise compile time hatası verir.
5. long, float, double, and boolean can never be used as a switch variable.
6. switch(x){} derlenir ama sadece switch(5) hata verir. Switch’in gövdesi olmalıdır, içinde hiçbir case olmasa da çalışır.

**Chapter-4**

1. getArray()[index=2]++; Parantez içindeki atama işlemi gerçekleşir önce. Sonra getArray fonk. Çağrılır ve en son ++ uygulanır. Burada çift boyutlu dizi olsaydı soldan sağa doğru üretilecekti.
2. ArrayList extends java.util.AbstractList ve implements Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, RandomAccess
3. RandomAccess is a marker interface used by List implementations to indicate that they support fast (generally constant time) random access.
4. For ArrayList, You can sort its elements using Collections.sort() method. This include Collections methods such as sort, reverse, and shuffle.
5. Collection c = new ArrayList(); derlenir ama arraylist alan fonksiyona collection alamaz.
6. ArrayList içinde primitive type saklanamaz ama add ile ekler isen hata vermez, ilgili wrapper class’a dönüştürülür!!!
7. ArrayList de istediğin pozisyona eleman eklemek için o posizyona önceden eleman eklenmiş olmalıdır yoksa IndexOutOfBoundsException hatası fırlatır..
8. System.arraycopy(soruce, 2, dest, 0, 2); bu komut source dizisinin 2. Index elemanı dahil olmak üzere bu elemandan, dest dizisinin 0. Elemanı üzerinden başlayarak 2 adet dizi elemanını dest’e yazar. Eğer destin uzunluğu küçük veya source da eleman yok ise ArrayIndexOutOfBoundsException nullpointerException fırlatır.
9. ArrayList de sublist(a,b) a dan itibaren b hariç döndür demek. List döndürür.
10. String sa[ ] = { "a ", " b", "c"};
11. String sa[ ] = new String[ ]{"a", "b", "c"};
12. Object[] sa = { }; sa null değil, boş bir obje uzunluk 0
13. int[] array4[], array5[]; Bunların hepsi 2 boyutlu diziler. Array5 de boyut işareti olmasaydı sadece tek boyutlu dizi olacaktı.
14. Object arr[][] = **new** Object[][] {**new** String[]{""}, {} };
15. Object arr2[] = **new** Object[] {**new** String[]{""}, {} };
16. 13 nolu geçerli bir tanım ama 14 nolu geçersiz bir tanım…

**Chapter-5**

1. Note that both the exceptions are RuntimeExceptions so there is no need to catch these
2. A catch must have a exception: catch(SomeException se){ }
3. Done will be followed by an exception. Finally is always executed (Only exception is System.exit();) Catch blogunda bir hata oluşsa bile finally çalışır ama hata da fırlatılır.

**Chapter-6**

1. To construct an instance of a sub class, its super class needs need to be constructed first. Extend etmek için base class ın const. Nasıl ise o şekilde çağrılmalıdır.
2. Sub class manuel olarak contructor tanımlayıp gövdesini boş bıraksa bile super kelimesi ekleniyor. Enteresan gövdeye bişey yazarsan da zaten super ekleniyormuş…Her constructor’ın yazsanda yazmasanda super() ekleniyor otomatik olarak..
3. Constructor önüne static, final, synchronized, native and abstract alamaz.
4. A constructor cannot be final, static or abstract. Ve contsructor kendi nesnesinden bi değişken alabilir.. Ayrıca her bir constructor ilk satırında super i çağırır. Birden fazla constructor var ise hepsinde çağrılmak zorundadır.
5. Always remember: Methods are overridden and variables are shadowed.
6. First, static statements/blocks are called IN THE ORDER they are defined.  
   Next, instance initializer statements/blocks are called IN THE ORDER they are defined.  
   Finally, the constructor is called. So, it prints a b c 2 3 4 1.
7. Note that the cast (long) applies to 'by' not to the whole expression.  
   ( (long) by ) / d \* 3; == return (long) by/d\*3;
8. abstract public void add(){ } Bu hatali abstract metodda body olmaz!!
9. When you program to an interface, your return type is usually the interface name and not the class name

**Chapter-7**

1. The overriding method cannot decrease the accessibility. The concept here is that an overriding method cannot make the overridden method more private. private->'no modifier'->protected->public
2. Override edilen metodların dönüş tipi aynı olmalıdır, fakat In case of objects, the return type of the overriding method may be a subclass of the return type of the overridden method. Primitive type larda ise tamamen aynı tip olmalıdır.int-long bile olmaz.
3. Covariant returns are allowed in Java 1.5, which means that an overriding method can change the return type of the overridden method to a subclass of the original return type. Here, C is a subclass of B.( This would have been true prior to Java 1.5. But from Java 1.5, an overriding method is allowed to change the return type to any subclass of the original return type, also known as covariant return type. This does not apply to primitives, in which case, the return type of the overriding method must match exactly to the return type of the overridden method.)
4. The overriding method may choose to have no throws clause even if the overridden method has a throws clause.
5. The overriding method can throw a subset of the exception or subclass of the exceptions thrown by the overridden class. Having no throws clause is also valid since an empty set is a valid subset. Yani override eden metod daha sub class exception’ı fırlatabilir.( An overriding method only needs to declare that it can throw a subset of the exceptions the overridden method can throw.)
6. Classes do not extend interfaces, they implement interfaces. Interfaces do not implement anything, they can extend multiple interfaces.
7. constructors are not inherited..
8. Override edilen metodların base’inde throws var ise bunu yeni metodda kullanmak zorunda değiliz. Ama base class değişken tipinde bir değişkenimiz bu metodu çağırırsa handle etmemiz lazımdır.
9. Since the original (overridden) method does not have any throws clause, the overriding method cannot declare any checked exceptions.( It may throw only those exceptions that are declared in the throws clause of the superclass's method or exceptions that are subclasses of the declared exceptions. It may also choose NOT to throw any exception.)
10. Abstract sınıflar normal sınıfları extend edebilir.
11. Observe that the rule for overriding a method is opposite to the rule for constructors. An overriding method cannot throw a superclass exception, while a constructor of a subclass cannot throw subclass exception (Assuming that the same exception or its super class is not present in the subclass constructor's throws clause).
12. A subclass may have a static method with the same signature as a static method in the base class but it is not called overriding. It is called shadowing because the concept of polymorphism doesn't apply to static members.
13. Only methods that are accessible can be overridden. A private method cannot, therefore, be overridden in subclasses, but the subclasses are allowed to define a new method with exactly the same signature.
14. You can assign a subclass object reference to superclass reference without a cast but to assign a super class object reference to a subclass (or interface) reference you need an explicit cast as in option 2.( Runnable r = (Runnable) o; )
15. Bir sınıfın içinde abstract metod varsa sınıf da abstract olmalıdır.

**Testler**

1. Code unreachable lara dikkat et. Onlar compile time error veriyor. While(false) bu hatayı verir her zaman, while içindeki constant ise. Variable olsa idi hata vermezdi.
2. Final değişkenler sadece = you can initialize it in the constructor or an instance block.
3. System.out.println(true + null); hata verir. Burada en az 1 adet string olması lazımmış.
4. Super bir önceki üretilen sınıftaki metodlara ulaşır. Daha içtekilere ulaşamaz.
5. Casting syntax’ı şu şekilde; ((GoodSpeak)s).up(); şu yanlış syntax = (GoodSpeak)s.up();
6. C3 sınıfı C1’den extend oluyorsa, o3=o1 compile time hata verir. base class child ına atamanaz. Ancak casting ile atanır. Nesne casting yaparken cast edilen en son ne ise o sınıf gibi davranır, soruları bu şekilde çöz.
7. str1.equals(str1=str2) burada önce st1 in değeri yerine konur, sonra st2 str1 e eşitlerin. Yani false olur sonuç
8. { { } } ve block : { break block ; } satırları geçerlidir. Continue ise sadece for while do-while içinde olur. Break bunlara ek swich içinde ve yukardaki gibi labelled block lar içinde tanımlanabilir. (compound statement)
9. Bir metoda null değeri pass edilebilir, sorun olmaz. Ama ilgili metod overload edilmiş ise en child sınıf çağrılır. Eğer eşit değerde iki sınıf var ise de compile time error oluşur.
10. A class can have overloaded static methods.
11. 1 Evaluate Left-Hand Operand First
12. 2 Evaluate Operands before Operation
13. 3 Evaluation Respects Parentheses and Precedence
14. 4 Argument Lists are Evaluated Left-to-Right
15. For Arrays: First, the dimension expressions are evaluated, left-to-right. If any of the expression evaluations completes abruptly, the expressions to the right of it are not evaluated.
16. i<20 ? out1() : out2() bu kullanım için 2. Ve 3. Kısımdan dönüş tipi void olan fonksiyon tanımlanamaz. Ayrıca bu kısımların sonuçları = ile birbirine atanabilir olmalı, yoksa yine hata verir. Ayrıca operandın biri null diğeri ref. Type ise sonuç ref. Type olur. Son olarak referanslar is-a ilişkisi olan sınıflar da kullanılabilir. Burada ise sonuc ikisine de atanabilir olmalı yani base class olsa sorun yok.
17. Java has a java.lang.SecurityException. This exception extends RuntimeException and is thrown by the security issues. Mesela java.util diye package oluşturup sınıf yazarsak SecurityException fırlatılır.
18. The condition expression in a while header is required. while ın condition'ı boş olamaz
19. Sıfır' a bölümden ArithmeticException hatası fırlatılır.