ANTES DE COMENZAR A REALIZAR LA PRUEBA DEBE LEER LAS SIGUIENTES NORMAS

(Solamente debe entregar al tribunal, una hoja de lectura óptica con sus datos y respuestas)

Material: NINGUNO

IMPORTANTE

Si encuentra alguna anomalía en el enunciado, indique ésta en el reverso de la hoja de lectura óptica (o si fuera estrictamente necesario en una hoja adjunta) y argumente la solución adoptada al efecto. Estos comentarios serán de gran importancia ante posibles reclamaciones en la revisión de exámenes. Sólo el Equipo Docente podrá anular preguntas del examen.

1. Deberá entregar UNICAMENTE la hoja de lectura óptica con sus datos y respuestas.
2. La prueba consta de un test de 20 preguntas a contestar en una hoja de lectura óptica. Lea atentamente las instrucciones que figuran en la hoja de lectura óptica (no olvide poner su DNI, Código de Carrera, Código de Asignatura y Tipo de Examen).
3. Para superar la prueba se deberá obtener una puntuación mínima de 5 puntos. En cada pregunta del test se proponen cuatro respuestas de las cuales sólo una es correcta. Únicamente puntuarán las preguntas contestadas. Si la respuesta es correcta la puntuación será de 0.5 puntos y si es incorrecta restará 0.25 puntos.

PREGUNTAS

1. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
class prueba {
 int i;
  public prueba() {
   i=0;} }
public class ejer1 {
  public static void main(String args[]) {
    prueba uno = new prueba();
    prueba dos = new prueba();
    uno = dos;
    uno.i= 20;
    dos.i= 30;
    System.out.println(uno.i);
    }
  }
```

A) 50
B) 20
C) 30
D) Ninguna de ellas.

2. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
class Elemento{
  private String nombre;
    Elemento(String nombre){
    this.nombre=nombre;
    numElementos++;
  }
  static private int numElementos = 0;

  static int numeroDeElementos(){
    return numElementos;
  }
}
public class PruebaElemento {
  public PruebaElemento() {
  }
  public static void main(String[] args) {
```

```
    Elemento H = new Elemento("Hidrogeno");
    Elemento He = new  Elemento ("Helio");
    Elemento Li = new Elemento ("Litio");

    System.out.println("Creados:  " + Elemento.numeroDeElementos());
  }
}
```

A) Creados: 0
B) Creados: 3
C) Creados:
D) No se ejecuta, no se puede hacer referencia a una variable estática desde un contexto estático.

3. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
abstract class Actor {
  abstract void actuar();
}

class ActorFeliz extends Actor {
  public void actuar() {
    System.out.println("ActorFeliz");
  }
}

class ActorTriste extends Actor {
  public void actuar() {
    System.out.println("ActorTriste");
  }
}

class Escenario {
  Actor a = new ActorFeliz();
  void change() { a = new ActorTriste(); }
  void go() { a.actuar(); }
}

public class Transformar {
  public static void main(String[] args) {
    Escenario s = new Escenario();
    s.go();
    s.change();
    s.go();
  }
}
```

A) ActorFeliz
ActorFeliz

B) ActorTriste
ActorTriste

C) ActorFeliz
ActorTriste

D) ActorTriste
ActorTriste

4. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
import java.util.*;

public class ComparaArrays {
  public static void main(String[] args) {
    int[] a1 = new int[10];
    int[] a2 = new int[10];
    Arrays.fill(a1, 47);
    Arrays.fill(a2, 47);
    System.out.println(Arrays.equals(a1, a2));
    a2[3] = 11;
    System.out.println(Arrays.equals(a1, a2));
    String[] s1 = new String[5];
    Arrays.fill(s1, "Hi");
    String[] s2 = {"Hi", "Hi", "Hi", "Hi", "Hi"};
    System.out.println(Arrays.equals(s1, s2));
  }
}
```

A) true
false
true

B) true
true
true

C) false
true
false

D) false
false
false

5. ¿A continuación se muestra el código de la clase ListaDeArrays2. Se han numerado las líneas para facilitar la referencia de las mismas.

```
1     package p5;
2     import java.util.*;
3     public class ListaDeArrays2 {
4           public static void main(String args[]) {
5                 ArrayList a1   = new ArrayList();
6                 int       suma = 0;
7                 for (int i=0; i<3; i++)
8                       a1.add(new Integer(i));
9                 a1.add("a");
10                      a1.add("b");
11                      for (int i=0; i<a1.size(); i++)
12                            System.out.print(a1.get(i));
13
14                }
15          }
```

Señale cuál de las siguientes afirmaciones es correcta:

A) Se produce un error en la línea 8.
B) Se produce un error en la línea 9 y un error en la línea 10.
C) Las dos afirmaciones anteriores son correctas.
D) El código de la clase no contiene errores. El resultado de la ejecución de su método *main* es la impresión en el flujo de salida estándar de: 012ab

6. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?. Se han numerado las líneas para facilitar la referencia a las mismas.

```
1. public class ejer1 {
2.    static int count = 0;
3.    public static void main(String [] args) {
4.       int x = 1;
5.       for ( test('1'); test('2') && (x < = 2); test('3') ) {
6.         x++;
7.         test('4');
8.       }
9.       System.out.println(" Cuenta = " + count);
10.    }
```

```
11.     static boolean test(char num) {
12.       System.out.print(" "+num);
13.       count++;
14.       return true;
15.     }
16. }
```

A) El código es incorrecto. Produce un error de compilación.
B) Escribe en el flujo de salida estándar `FIN`
C) Escribe en el flujo de salida estándar `Clase`
D) Ninguna de las anteriores

7. Dada la siguiente orden de ejecución en línea de comandos

```
Java Clase param1 param2 param3
```

¿Dónde se almacenan inicialmente los valores de los parámetros `param1`, `param2` y `param3`?

A) Todos se almacenan en una única cadena denominada arg
B) Se almacenan en un array indexado entre 0 y 2
C) Se almacenan en un array indexado entre 0 y 1
D) Ninguna de las anteriores

8. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
class A{
  void callme(){
    System.out.println("llama al método callme dentro de A");
  }
}
class B extends A{
  //Sobreescribo callme()
  void callme(){
    System.out.println("Llama al método callme desde B");
  }
}
class C extends A{
  void callme(){
    System.out.println("Llama el método callme dentro de C");
  }
}
public class Dispatch {
  public Dispatch() {
  }
  public static void main(String[] args) {
    A a=new A();
    B b=new B();
    C c=new C();
    A r;
    r=a;
    r.callme();
    r=b;
    r.callme();
    r=c;
    r.callme();
  }

}
```

A) Llama al método callme dentro de A
   Llama al método callme desde B
   Llama el método callme dentro de C
B) Llama al método callme dentro de B
   Llama al método callme desde A

Llama el método callme dentro de C

C) Llama al método callme dentro de C
Llama al método callme desde B
Llama el método callme dentro de C

D) Llama al método callme dentro de A
Llama al método callme desde A
Llama el método callme dentro de A

9. En Java para poder guardar un objeto de manera persistente es necesario implementar la interfaz

A) Runnable
B) Seriarizable
C) Localizable
D) Ninguna de las anteriores

10. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
public class Class1 {
  public static void main(String[] args) {

    int[] a1 = { 1, 2, 3, 4, 5 };
    int[] a2={0,0,0,0,0};
    a2 = a1;
    for(int i = 0; i < a2.length; i++)
      ++a2[i];
    for(int i = 0; i < a1.length; i++)
     {
      System.out.print(
        "a1[" + i + "] = ");
      System.out.println(a1[i]);
     }

     try{
         System.in.read();
        }catch(Exception e){}

  }
}
```

A) a1[0] = 2
a1[1] = 3
a1[2] = 4
a1[3] = 5
a1[4] = 6

B) a1[0] = 1
a1[1] = 2
a1[2] = 3
a1[3] = 4
a1[4] = 5

C) a1[0] = 0
a1[1] = 0
a1[2] = 0
a1[3] = 0
a1[4] = 0

D) Ninguna de ellas. Existe un error de compilación.

11. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
class Balance{
  String nombre;
  double bal;

  Balance(String n, double b){
    nombre=n;
    bal=b;
  }
  void show(){
    if(bal<0)
      System.out.print("-->");
      System.out.println(nombre + ": &" + bal);
  }
}
```

```
public class CuentaBalance {
  public CuentaBalance() {
  }
  public static void main(String[] args) {
    Balance actual[] = new Balance[3];

    actual[0]=new Balance("Pepe",100);
    actual[1]=new Balance("Luis",-100);

    for(int i=0;i<2;i++) actual[i].show();
  }
}
```

A) Pepe: &100.0
-->Luis: &-100.0

B) Luis: &100.0
-->Pepe: &-100.0

C) Pepe: &-100.0
-->Luis: &-100.0

D) Ninguna de ellas. Existe un error de compilación.

12. La especificación de la URL que hay que componer para acceder a una base de datos denominada `notas` mediante JDBC utilizando el subprotocolo ODBC es:

A) jdbc:odbc: notas.
B) jdbc-odbc:notas.
C) jdbc.odbc:notas.
D) Ninguna de las anteriores.

13. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
public class ThreadEscritor implements Runnable{
  int interno;

  public ThreadEscritor(int msg) {
    super();
    interno=msg;

  }

   synchronized  public void run(){
    while(true){
      System.out.println(interno);
      Thread.currentThread().yield();
    }
  }
  public static void main(String[] args) {
    for(int i=0;i<5;i++){
      Thread t=new Thread(new ThreadEscritor(i));
      t.start();}


    try{
      Thread.currentThread().sleep(1000);
    }
    catch(InterruptedException e){

    }
    finally{
      System.exit(0);
    }
  }

}
```

A) La secuencia ordenada de salida 01234 durante un cierto tiempo
B) La secuencia ordenada de salida 1234 durante un cierto tiempo

C) La secuencia ordenada de salida 0123 durante un cierto tiempo
D) No se puede saber, el orden con el que la CPU atiende los hilos es indeterminado.

14. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
class ProductoLimpieza{
  private String s = new String("Producto Limpieza");
  public void aniadir(String a){s +=a;}
  public void diluir() {aniadir("diluir()");}
  public void aplicar() {aniadir("aplicar");}
  public void escribir() {System.out.println(s);}

  public static void main(String[] args){
    ProductoLimpieza x= new ProductoLimpieza();
    x.diluir();
    x.aplicar();
    x.escribir();
  }
}

public class Detergente extends ProductoLimpieza{

  // Cambiando un método
  public void aplicar(){
    aniadir("Detergente.frotar()");
    super.aplicar();
  }
  public static void main(String[] args) {
    Detergente detergente1 = new Detergente();
    detergente1.aniadir("a");
    detergente1.diluir();
    detergente1.aplicar();
    detergente1.escribir();

    System.out.println("Probando la clase base");
    ProductoLimpieza.main(args);
  }

}
```

A) Probando la clase base
Producto Limpiezaadiluir()Detergente.frotar()aplicar
Producto Limpiezadiluir()aplicar

B) Producto Limpiezaadiluir()Detergente.frotar()aplicar
Probando la clase base
Producto Limpiezadiluir()aplicar

C) Producto Limpiezadiluir()aplicar
Producto Limpiezaadiluir()Detergente.frotar()aplicar
Probando la clase base

D) Ninguna de ellas, la clase Detergente no puede heredar de ProductoLimpieza

15. Para fijar el layout a null en un objeto elemento1 de tipo java.awt.Frame y en un objeto elemento 2 del tipo java.awt.Panel ¿cuál de las secuencias de código es la correcta?

A)
```
elemento1.setLayout (null);
elemento2.setLayout (null);
```

B)
```
elemento1.getContentPane().setLayout (null);
elemento2.getContentPane().setLayout (null);
```

C)
```
elemento1.getContentPane().setLayout (null);
elemento2.setLayout (null);
```

D) Ninguna de las anteriores.

16. En el siguiente fragmento de programa, ¿cuántas veces se ejecuta la sentencia `System.out.println` ?

```
a = 9;
for (int i = 0; i < 100; i++)
if ((a % 4 == 0) || (i % 2) == 0)
   System.out.println(a + " " + i);
```

A) 50
B) 0
C) 25
D) 100

17. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
public class matriz2 {
  public matriz2() {
  }
  public static void main(String[] args) {
    Integer[] arrayInteger = new Integer[5];
    arrayInteger[1]= new Integer(3);
    arrayInteger[2]= new Integer(4);
    arrayInteger[3]= new Integer(5);
    arrayInteger[4]= new Integer(6);

    for ( int i = 0; i < arrayInteger.length; i++)
    { System.out.print(arrayInteger[i]);
    }
  }
}
```

A) 01234
B) null34567
C) 3456null
D) null3456

18. ¿Cuál de las siguientes afirmaciones es cierta?

A) *StringTokenizer* devuelve todos los símbolos contenidos en una cadena de caracteres uno a uno
B) *StringTokenizer* devuelve todos los símbolos contenidos en una cadena de caracteres ordenados según su código ASCII
C) *StringTokenizer* devuelve todos los símbolos contenidos en una cadena de caracteres según una función de conversión de claves.
D) Ninguna de las anteriores.

19. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
import java.util.*;
public class Iterador {

  public static void main(String[] args) {
    HashSet conjunto = new HashSet();
    conjunto.add("Esto ");
    conjunto.add("es ");
    conjunto.add("una ");
    conjunto.add("prueba ");
    conjunto.add("prueba ");

    Iterator itr = conjunto.iterator();

    while(itr.hasNext()){
```

```
        System.out.print(itr.next());
      }
    }
  }
```

A) Esto es una prueba prueba
B) Esto es una prueba
C) En principio no se sabe. Depende de la función de conversión de claves.
D) Al compilar da un error. No se pueden añadir dos elementos iguales a un objeto `HashSet`.

20. ¿Cuál es la salida del siguiente fragmento de código al ejecutar el método *main*?

```
import java.io.*;
public class Valores {
  public Valores() {
  }
  public static void main(String[] args) {
   byte[] byteArray={0,1,2,3,4,5,6,7,8,9};
   ByteArrayInputStream flujoArrayByte=new ByteArrayInputStream(byteArray);
   while(flujoArrayByte.available() !=0){
         byte leido = (byte)flujoArrayByte.read();
         System.out.print(leido);
         flujoArrayByte.skip(1);
   }
  try {
    flujoArrayByte.close();
  }
  catch (IOException ex) {
  }
  }
}
```

A) 2468
B) 0123456789
C) 9876543210
D) 02468