Unidad 3
Programación Orientada a Objetos. Programación JAVA. Cuestiones I:

Alumno: _____

1. Corregir los errores del siguiente código JAVA. (0.5p):

```
// Obtener el menor divisible por 3 y el mayor divisible por 5 de un vector
public class XX {
    public static void main(String[] args) throws IOException {
        int vector[];
        int mayor= Integer.MAX_VALUE;
        int menor= Integer. MIN_VALUE;
        BufferedReader in=new BufferedReader(new InputStreamReader(System.in));

        System.out.println("¿.....?");
        int longitud=Integer.valueOf(in.readLine()).intValue();
        vector=new int[];

        for(int i=0 ; i<vector.length ; i++)
        {
            System.out.println(".....:");
            vector[i] = Integer.valueOf(in.readLine()).intValue();
            if ((vector[i] % 5) = 0) && (vector[i] > mayor))
                    mayor = vector[i];
            if ((vector[i] % 3) == 0) &&  (vector[i] < menor))
                    menor = vector[i];
        }

        System.out.println("....: "+mayor);
        System.out.println("....: "+menor);
    }
}
```

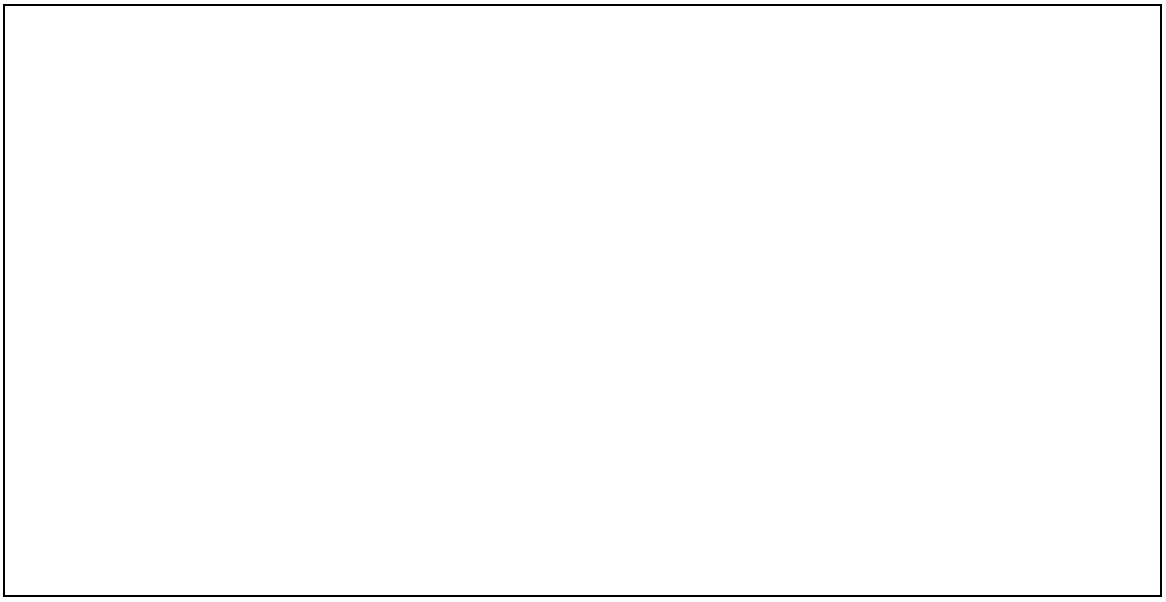
2. Corregir los errores del siguiente código JAVA (0.5p):

```
import java.io.*;
public class XX {
public static void main(String[] args) throws IOException
{
	float longitud;
	int [] vector;
	BufferedReader in=new BufferedReader(new InputStreamReader(System.in));

	System.out.println("¿.....?");
	longitud=Integer.valueOf(in.readLine()).intValue();
	vector=new int[longitud];

	for(i=0 ; i<longitud ;)
	{
		try
		{
			System.out.println(".....:");
			vector[i] = Integer.valueOf(in.readLine()).intValue();
		  }
		catch(NumberFormatException e)
		{
			System.out.println(".....:");

		}
		catch(Exception )
		{
			System.out.println(".....:");
		}
    }
}
}
```



3. Supongamos una máquina que dispone de un conjunto de N pulsadores numerados de 0 a N-1, cada uno de los cuales puede estar activo o no, en un instante de tiempo. Necesitamos una clase en JAVA (Class Pmaquina), que permita almacenar la información de cuáles de esos pulsadores están o no activos en un instante dado. La clase necesitará los siguientes métodos (1p – 0.25p/método):
    - **activarPulsador** : que recibirá un número de pulsador y hará que figure como activo.
    - **desactivarPulsador** : que recibirá un número de pulsador y hará que figure como NO activo.
    - **activado :** que recibirá un número de pulsador y devolverá si este está activo o no.
    - **numActivos** : que devolverá el número de pulsadores que están activados.

4. ¿Qué resuelve este código JAVA? (0.5p):

```
import java.io.*;
public class Main {

   public static void main(String[] args) throws IOException{
      int vector[];
      int aux=Integer. MIN_VALUE;
      String valor;

      BufferedReader in=new BufferedReader(new InputStreamReader(System.in));

      System.out.println("¿.....?");
      valor= in.readLine();
      int longitud=Integer.valueOf(valor).intValue();
      vector=new int[longitud];

      for (int i=0 ; i<vector.length ; i++){
         System.out.println(".....: ");
          valor= in.readLine();
         vector[i]=Integer.valueOf(valor).intValue();
      }
      for (int i=0 ; i<vector.length ; i++){
         if ((i % 2== 0) && (vector[i] > aux))
               aux = vector[i];
      }
      System.out.println(".....: "+aux);
   }
}
```

5. Escriba en la clase Java Gpoligono el siguiente método (0.75p):

Atributos:

- ✓ Números de lados (*int*). int nlados;
- ✓ Vector de (*class* ***Punto***). Punto [] posicionamiento;
- ✓ Color de relleno (*class* ***Rgb***). Rgb color;

Métodos:

- ✓ public gpoligono(){// Constructor por defecto}
- ✓ public gpoligono(int num, Rgb c){/* Mediante este constructor sobrecargado se proporciona el número de lados y su color. Dentro del cuerpo de este método te preguntará por cada uno de los puntos (X,Y) que formarán su posicionamiento en el espacio.*/}
- ✓ ……..
- ✓ public boolean compareTopoligono(Gpoligono a){/* Método que dado un polígono determina si es igual a él. Son iguales cuando coinciden el color, número de lados y su posicionamiento. */}

Nuevo Método:

- ✓ public boolean igualesTopoligonos(Gpoligono [] a){/* Método que dado un vector de polígonos determina si todos los Gpoligonos son iguales. */}

```
public boolean igualesTopoligonos(Gpoligono [] a) {




} // Fin del new método
```

6. Escriba los nuevos métodos de la clase Java que represente un Viaje (1.5p):

Atributos:

- ✓ Números de trayectos (int). int ntra;
- ✓ Vector de (class Trayecto). Trayecto [] sectrayectos;

Clase Trayectoria

```
public class Trayecto {
        public String origen;
        public String destino;
        public double distancia;
        .....}
```

Métodos:

- ✓ public viaje(){// Constructor por defecto}
- ✓ public viaje(int num){/* Mediante este constructor sobrecargado se proporciona el número de trayectos. Dentro del cuerpo de este método te preguntará cada uno de los trayectos que forman el viaje.*/}
- ✓ public Trayecto mayortrayecto()){/* Devuelve el mayor trayecto en longitug.*/}
- ✓ public double distanciaviaje(){// Resultado de la suma de los trayectos}
- ✓ public boolean combinables(Viaje v1) {/*Si coincide el destino de último trayecto de v1 con el origen del primer trayecto de this, los trayectos son combinables true*/}.

Nuevo Método (0.75p):

- ✓ public viaje maximadistancia(Viaje [] a){/* Método que dado un vector de viajes devuelve aquel viaje con mayor distancia. */} ***// Nota Double.MIN_VALUE***

```
public Viaje maxdistancia(Viaje [] a){
```

} // Fin del new método

Nuevo Método (0.75p):

- ✓ public Trayecto mayortrayectoviajes(Viaje [] a){/* Método que dado un vector de viajes devuelve el trayecto más largo. */}

public Trayecto mayortrayectoviajes(Viaje [] a)

} // Fin del new método

7. Cuestionario tipo test (1.25p)*(2 mal quitar 1 bién)*:

1. Dada la siguiente definición: ¿Cuál debería ser el contenido del constructor?

```
class Temperatura {
double t;
Temperatura(double t) { ¿? }
}
```

a) t=t;
b) double t=t;
c) this.t=t;
d) No se puede llamar igual el parámetro del constructor que el atributo de la clase

2. Dado el siguiente fragmento de programa: Indique que afirmación es cierta:

```
int k;
for (k=5 ; k>0 ; k--)
System.out.print(k);
System.out.print(k);
```

a) Se imprime 543210
b) Se imprime 5432100
c) Se imprime 554433221100
d) Se imprime 543210-1

3. ¿Qué es cierto respecto de este fragmento de programa?

```
Temperatura p[] = new Temperatura[12];
p[2].calentar(3);
```

a) Funciona correctamente.
b) Funciona correctamente, aplicándose el método a la 3ª clase Temperatura.
c) Funciona correctamente pero hay que capturar las excepciones provocadas en los arrays de objetos.
d) Produce un error de ejecución, ya que no puede invocarse un método sobre una referencia null o una variable no inicializada.

4. Se quiere definir el método 'prueba' que no recibirá ningún argumento y tampoco devolverá ningún valor. ¿Cuál de las siguientes declaraciones es correcta?

a) prueba(void)
b) void prueba(void)
c) prueba()
d) void prueba()

5. Indique la salida de:

```
int a= 7, b= 3;
System.out.println ((++a) * b);
```

a) 24
b) 21
c) 10
d) 73

6. Evalúe el valor final que toma la variable "s":

```
int n= 1; s= 0;
while (n <= 9) s+=n;
```

a) 45
b) 0
c) 9
d) el programa no termina nunca

7. En Java this

a) hace referencia al objeto que invoca al método
b) hace referencia sólo a los atributos propios de la clase en la que se encuentra.
c) hace referencia a la super-clase
d) hace referencia a las variables del método en el que se encuentra.

8. ¿Qué se imprimirá al ejecutar el siguiente bucle?

```
for (int i=0; i < 5; i++) {
if (i==3) { i=5; }
System.out.println (i + " ");
}
```

0 1 2 3 4
0 1 2 3
0 1 2 4
0 1 2

9. Dado el siguiente código, indique qué ocurriría al llamar al método wom():

```
class CExamen{
private int i=0;
public void wom(){
for (int i=0; i<5;i++)
System.out.println(this.i);
}
}
```

a) imprime 00000
b) imprime 01234
b) imprime infinitos ceros
c) se producirá un error en tiempo de compilación.

10. Dados los siguientes fragmentos de código:

```
class ClaseC {
public void fmet (int i) { ... }
public int fmet (int i) { ... }
}  ...
ClaseC c = new ClaseC();
c.fmet(4);
```

Se produce:
a) la llamada al primer método fmet.
b) la llamada a ningún método porque hay sobrecarga.
c) un error al compilar.
d) un error al ejecutar.

11. Indique qué valor imprime el siguiente programa:

```
static void proc (int x) {
x= 1;
}
public static void main (String [] arg) {
int x= 4;
proc (x);
System.out.print (x);
}
```

a) 4
b) 1
c) 0
d) El programa producirá un error

12. Dada el siguiente código:

```
class ClaseA {
public int campo;
}
class PruebaClaseA {
public static void main(String x[]){
ClaseA a1 = new ClaseA();
ClaseA a2 = new ClaseA();
ClaseA a3 = new ClaseA();
a1.campo=150;
a2.campo=150;
a3 = a2;
if (a1 == a2) { System.out.println(" UNO");}
if (a1 == a3) { System.out.println(" DOS");}
if (a2 == a3) { System.out.println(" TRES");}
}
```

El resultado sera:
a) UNO
b) UNO TRES
c) UNO DOS TRES
d) TRES

13. En una sentencia try-catch-finally:

a) Los bloques catch se pueden repetir tantas veces como excepciones de distinto tipo se desee atrapar. El bloque finally debe aparecer al menos una vez y se ejecuta siempre.
b) Los bloques catch se pueden repetir tantas veces como excepciones de distinto tipo se desee atrapar. El bloque finally no es opcional y se ejecuta siempre.
c) Los bloques catch se pueden repetir tantas veces como excepciones de distinto tipo se desee atrapar. El bloque finally es opcional y solo puede aparecer una vez. Este bloque se ejecuta siempre.
d) Todas las afirmaciones son falsas

14. Para utilizar componentes que están en otro paquete diferente se debe añadir una declaración de importación con la sintaxis:

a) include nombre-del-paquete
b) import nombre-del-paquete
c) package nombre-del-paquete
d) class nombre-del paquete

15. ¿Para qué se usan las clases asociadas a los tipos primitivos?

a) Para facilitar la programación en Java. Estas clases proporcionan métodos útiles para convertir un tipo primitivo en otro, para imprimir los números con diversos formatos .. etc
b) Para definir nuevos tipos simples
c) Para convertir cadenas de texto a otros tipos
d) Para poder leer valores proporcionados por el usuario.

16. Dado el siguiente código cual es la salida por la consola para valores de la nota 4, 6, 3, 8.

```
public class Resultados {
public static void main(String[] args) {
int nota = 7;
System.out.println((nota>=5) ? (nota<8) ?"Entrevistar" : "Contratar" : "Rechazar");
}}
```

a) Contratar Entrevistar Rechazar Contratar
b) Rechazar Entrevistar Contratar Contratar
c) Rechazar Entrevistar Rechazar Contratar
d) Contratar Entrevistar Rechazar Contratar

17. La declaración Cliente[] clientes = new Cliente[5] corresponde a:

a) Un array de tipo base Cliente, de una dimensión y tamaño 5,
con identificador clientes
b) Un array de tipo base Cliente, de una dimensión, con identificador clientes, que aún no ha sido instanciado
c) Un array de tipo base Cliente, de una dimensión y tamaño 5,que puede almacenar objetos en las posiciones: clientes[1],clientes[2], clientes[3], clientes[4], clientes[5]
d) Todas las afirmaciones son falsas.

18. Declare un array de objetos de la clase Cliente de tamaño 10 con identificador misClientes. El segundo cliente con NIF 43658713X, nombre Juan y apellidos Fernández López, recuperándose el nombre del mismos:

```
public class Cliente {
private String nif;
public String nombre;
private String apellidos;
public Cliente(String nif, String nombre,
String apellidos) {
this.nif = nif;
this.nombre = nombre;
this.apellidos = apellidos;
}}
```

a)
```
misClientes[2]=new Cliente(“43658713X”,”Juan”,”Fernández”);
System.out.print(misClientes[1].nombre);
```
b)
```
Cliente [] misClientes= new Cliente[10];
misClientes[2]=new Cliente(“43658713X”,”Juan”,”Fernández”);
System.out.print(misClientes[2].nombre);
```
c)
```
Cliente [] misClientes= new Cliente[10];
misClientes[1]=new Cliente(“43658713X”,”Juan”,”Fernández”);
System.out.print(misClientes[1].nombre);
```
d)
```
Cliente [] misClientes= new Cliente(“43658713X”,”Juan”,”Fernández”);
System.out.print(misClientes[1].nombre);
```

Unidad 3
Programación Orientada a Objetos. Programación JAVA. Cuestiones II

Alumno: _____

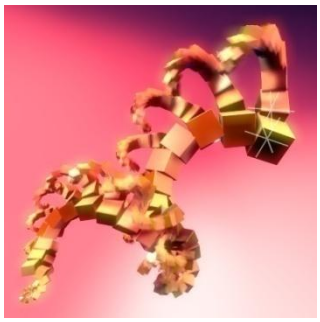
Desarrolla un proyecto JAVA llamado nombre_examen_Asir que contenga los siguientes elementos (4p):

- Un paquete Ejercicio1_Examen.
- Un paquete Ejercicio2_Examen.
- Una clase Main.java que se utilizará para probar de manera opcional los ejercicios resueltos. Es evidente que el examen estará dividido en dos ejercicios y cada una de ellos se resolverá con una , dos ..etc clases que serán definidas en su correspondiente paquete.

Ejercicios 1 (2 puntos).- (Paquete Ejercicio1_Examen)

Clase ArrayAsir

/ clase que moldea y contiene las operaciones relacionadas con un vector de enteros (int v[]) */*



Atributos *(el alumno decidirá si un atri. es público o privado)*:

int [] elementos;

int num_elemente;

Métodos (cada método 0,5p):

public ArrayAsir(int num,int [] a);

/* Contructor que inicializa cada uno de los atributos. */

public Estado compareToArrayAsir(ArrayAsir a);

/* Método que devuelve un objeto de la Clase Estado con el valor 1 (no tienen el mismo número de elementos),2 (tienen el mismo número de elementos pero tienen algún valor discordante) o 3(tienen el mismo número de elementos y tienen los mismos valores y en las mismas posiciones) . */

public int suma_max_min_ArrayAsir();

/* Método que devuelve la suma del mayor y menor. */

public boolean iguales(ArrayAsir [] a);

/* Método que devuelve true si todos los ArrayAsir son iguales en número de elementos y contienen el mismos valores en las mismas posiciones. */

Ejercicios 1 (2 puntos).- (Paquete Ejercicio2_Examen)

Clase Hucha

Atributos *(el alumno decidirá si un atri. es público o privado)*:

int [] cantidad_tipo_moneda ;

int [] tipo_modena;

Métodos (cada método 0,5p):

public Hucha(int [] ctm,int [] tm);

/* Contructor que inicializa cada uno de los atributos. */

public int total();

/* Devolverá el valor total del contenido de la hucha. */

public boolean ingresar(int tipo,int cantidad) ;

/* Recibirá un tipo de moneda y cantidad de moneda para añadirla a la hucha. Si ese tipo de moneda no está considerada en la hucha devolverá false y si se ha realizado el proceso devolverá true */

public boolean existeTipo(int tipo) ;

/* Recibirá un tipo de moneda y devolverá true si existe ese tipo de moneda y false si no existe. */