

**Escuela Superior Politécnica del Litoral**  
**Programación Orientada a Objetos**  
**Examen Parcial - 2019 1T**

Nombre: \_\_\_\_\_

Paralelo: \_\_\_\_\_

**1. Responda Verdadero o Falso según corresponda (5 puntos)**

Enunciado	Respuesta
Es obligatorio que todos los métodos <small>abstractos de una clase</small> sean implementados por una clase no abstracta que la implemente.	
En una jerarquía de herencia, se ejecuta primero el constructor de la clase hija antes que el constructor de la clase padre.	
Una clase con modificador final puede tener una o más clases hijas en una relación de herencia.	
Si no se incluye la notación @override antes de la sobre-escritura de un método, se genera un error de compilación.	
Es posible crear instancias del tipo de una clase abstracta	

**2. Análisis de código (6 puntos)**

Un programador definió las siguientes dos clases en distintos paquetes

```
package pk1;
public class Alphaclass{
    int num;
    public Alphaclass(int num){
        this.num = num;
    }
}
```

```
package pk2;
public class Betaclass extends Alphaclass{
    int p;
    public Betaclass(int num, int p){
        this.num = num;
        this.p = p;
    }
    public static void main(String[] args){
        Betaclass obj1 = new Betaclass(2,3);
        System.out.println(obj1.num);
    }
}
```

Explique en detalle por qué este código no compilará y sugiera la o las modificaciones que harían que el código compile y que los valores de las variables de instancia **num** y **p** de obj1 tengan los valores 2 y 3 respectivamente.

### 3. Dado el siguiente código (18 puntos)

```
public abstract class Electronico{
    int getConsumoElectrico(); //retorna el consumo de energia de un electronico
}

public abstract class Computadora implements Electronico{
    public void printDescription() {
        System.out.println("Computadora");
    }
}

public class Laptop extends Computadora{
    public int getConsumoElectrico() {
        return 500;
    }
    public void printDescription() {
        System.out.println("Laptop");
    }
    public void printDescription(String s) {
        System.out.println(s);
    }
}

public class TV implements Electronico{
    public void printDescription() {
        System.out.println("TV");
    }
    public int getConsumoElectrico() {
        return 1000;
    }
}

public class Test {
    //su codigo
    public static void main(String [] args){
        Laptop la = new Laptop();
        System.out.println(esEficiente(la)); //true
        TV tv = new TV();
        System.out.println(esEficiente(tv)); //false
    }
}
```

- Respecto al código mostrado, indique al menos un caso en el que ocurre sobre-escritura y uno en el que ocurre sobre-carga. En su respuesta incluya el nombre del método, las clases involucradas y por qué es sobre-escritura y sobre-carga. (6 puntos)
- La clase **Computadora** es abstracta, ¿**Por qué debe ser abstracta para que el código compile?** (2 puntos)
- En la clase **Test** implemente el método **esEficiente** para que funcione como se ve en el método main. El método debe retornar verdadero si el objeto consume menos de 800 kw de energía. (6 puntos)

- d. Para cada uno de las siguientes líneas indique si el código se ejecutará o no. En caso de compilar y ejecutarse cuál será la salida en pantalla. En caso de no poder ejecutarse cuál es la razón (4 puntos)

Computadora c1 = new Computadora(); c1.printDescription();	
Computadora c2 = new Laptop(); c2.getConsumoElectrico();	
c2.printDescription("Laptop");	
c2.printDescription();	

#### 4. Elabore el diagrama de clases del siguiente escenario (25 puntos)

El sistema nacional de Justicia es el encargado de procesar y ofrecer justicia de muchos casos a través de un juicio. Todos los juicios corresponden a un caso determinado, tienen un acusado y una fecha de inicio. El acusado es una persona natural de los cuales se necesita su nombre y cédula.

En los juicios actúan: un juez a cargo, un abogado defensor y un abogado de la fiscalía. Todos ellos deben estar registrados ante la ley como abogados de la república con un código de registro y la fecha de inicio de sus actividades, adicional a datos básicos como nombre y cédula.

El jurado de un juicio está conformado por 7 **miembros**. Cada uno de estos miembros es una persona de los cuales se necesita su nombre y su cédula. Al final del juicio cada miembro del jurado da su veredicto del acusado: CULPABLE o INOCENTE. El juez contabiliza estos votos, para definir el veredicto final del juicio (CULPABLE o INOCENTE) por mayoría simple. Finalmente, el juez dicta una sentencia si el veredicto final es CULPABLE.

Antes de iniciar un juicio la fiscalía registra las evidencias que pueden ser: registros médicos, policiales, testimonios, fotografías y grabaciones. Cada evidencia debe estar registrada con un código numérico único, una descripción y el tipo. Si la defensa sugiere que alguna evidencia ha sido manipulada, puede solicitar que su eliminación.

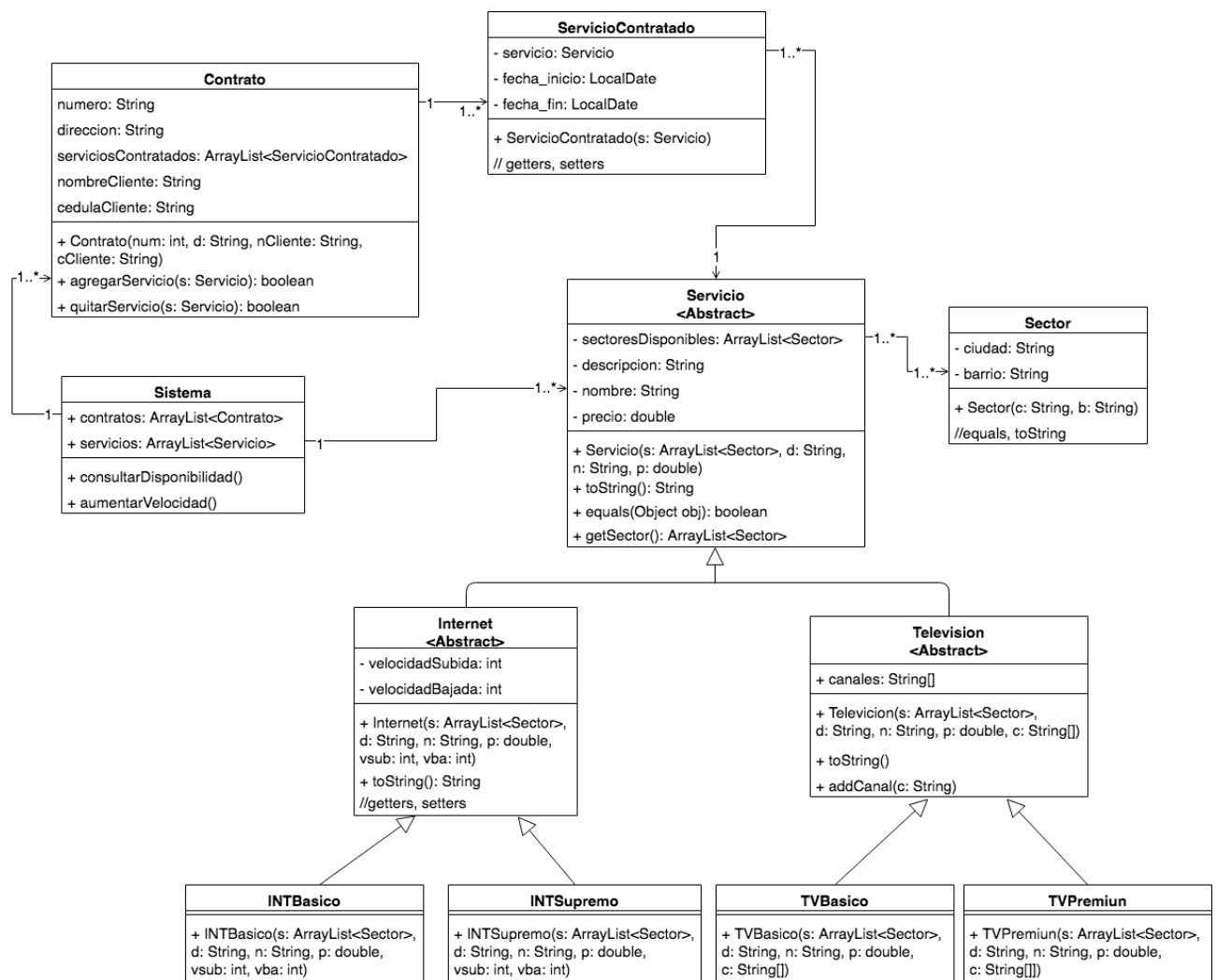
El abogado de la defensa y el abogado de la fiscalía presentan sus testigos al juez, y estos son citados formalmente por el juez a la fecha del juicio para ofrecer sus testimonios. Es importante saber la fecha en que se generó la cita y la fecha de recepción del documento ya que los testigos que se nieguen a dar su testimonio entran en delito de desacato. Los testigos son personas naturales. (En cada juicio hay al menos un testigo de la defensa y un testigo de la fiscalía).

## 5. Conteste brevemente las siguientes preguntas y/o enunciados (4 puntos)

- Mencione una diferencia entre clases abstractas e interfaces (2 punto)
- Si se quisiera compartir el valor de una variable entre todas las instancias de una clase, ¿cómo se debería declarar esta variable y por qué? (2 puntos)

## 6. Tema desarrollo (42 + 5 puntos)

Una empresa dedicada a brindar servicios de internet y televisión pagada tiene el siguiente diagrama de clases.



En base al diagrama de clases implemente lo siguiente:

### Clase Sector (7 puntos)

- Implemente el método equals utilizando los atributos de ciudad y barrio.

### Clase Servicio (3 puntos)

- Implemente el método `toString()` para que retorne la siguiente información:  
"nombre, descripcion, precio".

### Clase Internet (11 puntos)

- Implemente la clase Internet con sus atributos, constructores y métodos (incluyendo getters y setters).
- El método **`toString`** debe retornar un String que contenga la siguiente información:  
"nombre, descripcion, precio, velocidadSubida, velocidadBajada".

### Clase Contrato (10 puntos)

- Implemente el método **`agregarServicio`** que recibe un Servicio y lo agrega a la lista de servicios asociados al contrato. **Un cliente no puede tener más de un mismo tipo de servicio en un contrato (por ejemplo: en un mismo contrato NO PUEDE HABER el servicio INTBasico e INTSupremo)**. El método retorna *true* si se pudo realizar la acción o *false* caso contrario.
  - NOTA: La solución desarrollada debe seguir funcionando en caso que se agreguen al diagrama más tipos de servicios de internet o más tipos de servicios de televisión.

### Clase Sistema

- Implemente el método **`consultarDisponibilidad`** que pide al usuario la ciudad y barrio y muestra en pantalla cuáles son productos disponibles para él. Si no existe ningún servicio disponible para su zona el método muestra un mensaje de **"No hay disponibilidad en zona"**. (8 puntos)

A continuación, se muestra un ejemplo de la ejecución del programa:

```
Ingrese ciudad: Guayaquil
Ingrese barrio: Guayacanes
Los servicios disponibles son:
Internet Basico, internet para toda la familia, 15.00, 8.00 MB, 1.00 MB
TV BASICO, canales nacionales e infantiles, 20.00, Ecuavisa, ..
TV Premiun, las mejores pelis, 15.00, Ecuavisa, HBO, ..
```

NOTA:

- El ejemplo no muestra el servicio **INTSupremo** porque para el sector ingresado no llega ese servicio.
- De cada servicio se muestra lo devuelto por el metodo `toString`.
- La solución desarrollada debe seguir funcionando en caso que se agreguen al diagrama más planes de internet o televisión.
- Por el mes de Guayaquil se ha decidido aumentar 2 MB de subida a los planes de internet de los contratos. Implemente un método llamado **`aumentarVelocidad`** que aumente 2MB a los planes de internet de los contratos. (8 puntos)

**NOTA: TODO LO QUE NO SE PIDA IMPLEMENTAR YA ESTÁ IMPLEMENTADO Y LO PUEDE USAR.**

## Rúbrica

### 1.- Equal (7 puntos)

1. Firma correcta del método equals **1 punto**
2. Verifica si es null y retorna false **1 punto**
3. Verifica si son las mismas referencias **1 punto**
4. Verifica por la clase **1 punto**
5. Hace downcasting **1 punto**
6. Verifica por los atributos ciudad y barrio evaluando usando equals ya que son strings (**2 puntos**).

### 2.- Clase internet (9 puntos)

1. Definición clase Internet como abstracta y que extienda Servicio **2 puntos**
2. Constructor de la clase **3 puntos**
3. Sobrescritura método toString **3 puntos**
4. Getter y setter campos (**4 puntos**)

### 3.- Clase Contrato agregar Servicio ( 10 puntos)

1. Firma correcta del método de acuerdo a lo solicitado **1 punto**
2. Recorrer los servicios correctamente **1 punto**
3. Por cada servicio evaluar tv o internet dependiendo de lo que recibe **3 puntos**
4. Devolver false si ya hay un servicio de este tipo **2 puntos**
5. Crear un objeto contratoArticulo **1 punto**
6. Añadir objeto creado a la lista **1 punto**
7. Retornar true si la operación fue exitosa **1 punto**

### 5.- consultarDisponibilidad (8 puntos)

1. Pides los datos por teclado **2 punto**
2. Crear un sector **1 punto**
3. Recorrer la lista de servicios **1 punto**
4. Evalua si el sector esta en la lista de sectores del servicio **2 punto**
5. Presenta la información **1 punto**
6. Si no hay ninguno presenta el mensaje adecuado **1 punto**

### 6.- aumentarVelocidad (8 puntos )

1. Recorrer los contratos y los servicios contratados correctamente **3**
2. Preguntar si es instancia de internet **1**
3. Hacer downcasting **2**
4. Cambiar velocidad usando setter y getter **2 puntos**