

	IES LUIS VIVES - 1º DAM - EXAMEN 1ª EVALUACIÓN Módulo "Programación"	Curso 2024/2025 Fecha: dic 24
Nombre y Apellidos		Nota:

### Parte de Teoría (20%)

Cada pregunta tiene entre 1 y 4 respuestas válidas.

Cada pregunta vale 0,5 puntos y si en una pregunta hay n respuestas correctas y el alumno marca p respuestas correctas la puntuación obtenida en la pregunta será de  $p \cdot 0,5/n$  puntos. Teniendo en cuenta que se eliminará una respuesta correcta por cada respuesta incorrecta marcada.

Las puntuaciones negativas en una pregunta se considerarán como 0.

1.-Indicar que definición/definiciones de métodos sería válido para un método de la clase siguiente

```
public class Persona {
    String nombre;
    int edad;
    boolean casado;
}
```

- a) persona Persona (){}
- b) Persona (Persona a){edad=a.edad;}
- c) void aumentar (){edad= (casado)?edad+1:edad-1;}
- d) boolean igualEdad(Persona a){ if (edad==a.edad) return true; else return false;}
- e) Ninguna de las anteriores.

2.-Indicar que afirmaciones son CIERTAS para el lenguaje java

- a) En un programa en java declarar la variable `int variable=3;` en el método main me impide realizar después la instrucción `double Variable;` en dicho método.
- b) Después de ejecutar las instrucciones `int x=10, int y=4; double d=x/y;` la variable d tendría almacenado el valor 2.5
- c) En una aplicación java básica es necesario que exista un método main, al menos, en 1 de las clases que forman parte de la aplicación.
- d) Los comentarios en java se pueden escribir utilizando `<!-- comentario -->`
- e) Para crear un objeto de la clase Persona del ejercicio anterior, y suponiendo que está definido el constructor sin parámetros, se utilizaría la instrucción `Persona p=Persona();`
- f) Si en una clase A está declarado el método `void miMetodo(int x){}`, no podré declarar otro método como `void miMetodo(double x){}` en dicha clase.
- g) En el método main de una aplicación java el argumento tiene que ser de tipo `String []` y llamarse args.
- h) Ninguna de las anteriores.

3.-Indicar cuál es el resultado de la ejecución del siguiente programa

```
public class Fraccion {
    int num,den;
    Fraccion (int n, int d){
        num=n;
        den=d;
    }
    int modificarNumerador(int n){
        int n1=num;
        num=n;
        return n1;
    }
    public static void main(String[] args) {
        Fraccion f= new Fraccion (1,2);
        Fraccion f2=new Fraccion (3,4);
        f=f2;
    }
}
```

```
        int x=f2.modificarNumerador(5);
        System.out.println("x:" + x+ " f.num:" + f.num+ " f2.num:" + f2.num);
    }
}
```

- a) x:3 f.num:1 f2.num:5
- b) x:1 f.num:1 f2.num:5
- c) x:3 f.num:3 f2.num:5
- d) x:3 f.num:5 f2.num:5
- e) Ninguna de las anteriores

4.-Indicar que afirmaciones son ciertas cuando se ejecuta el siguiente fragmento de código y suponiendo accesible la clase Teclado utilizada en clase:

```
public static void main(String[] args) throws IOException {
    Teclado t= new Teclado();
    int n=1,i;
    for (i=2; n!=0; i++){
        n=t.leerInt();
        n=n%i;
        i++;
    }
    System.out.println(i);
}
```

- a) Si introduzco por teclado el número 4, el programa sólo realiza 1 iteración del bucle for.
- b) Si introduzco por teclado el número 5 y luego el número 8 el programa terminará saliendo del for con el valor de i igual a 4.
- c) Si realizó una única lectura y el valor introducido hace finalizar el bucle, el valor de i mostrado en pantalla será 3
- d) Si introduzco por teclado el número 0 el programa sólo realizará una iteración del bucle for.
- e) Ninguna de las anteriores es cierta.

## Solución

- 1.- b, c y d
- 2.- c
- 3.- d
- 4.- a y d

**Parte práctica (80%)**

**1.- (2 ptos)** Realizar un programa que lea números de teclado hasta que se de un 0 y muestre el número leído tantas veces como su valor absoluto. Solo se aceptarán números de un dígito. Si el número introducido es 0 no se escribirá nada.

Ejemplo

Número leído	1	-4	3	23
Se mostrará:	1	44444	333	No Válido

**2.- (2.5 ptos)** Se desea realizar una aplicación para controlar las cuentas corrientes de los clientes de un banco. Se dispone de las clases siguientes:

Clase **Cuenta** que almacena los datos de una cuenta corriente. Por simplificación sólo se almacenará el último movimiento realizado en la misma.

```
public class Cuenta {
    String nombreTitular;
    double saldo;
    Movimiento ultimoMovimiento;

    public Cuenta(String nombreTitular) {
        this.nombreTitular = nombreTitular;
        this.saldo = 0;
        this.ultimoMovimiento = null;
    }
}
```

Clase **Movimiento** que almacena los datos de un movimiento de una cuenta corriente.

```
public class Movimiento {
    double cantidad;
    boolean ingreso; //true para ingreso, false para reintegro
    private Fecha fecha; // fecha del movimiento;
    int tipo; // 1- para cálculo de intereses, 2- para retirada en cajero, etc...
}
```

Clase **Fecha** para almacenar fechas.

```
public class Fecha {
    private int dia, mes, agno;

    Fecha(){}
    Fecha (int d, int m, int a){
        dia=d;
        mes =m;
        agno =a;
    }
    /*Coloca en los atributos del objeto la fecha actual obtenida del sistema operativo*/
    void fechaActual(){
        agno = Calendar.getInstance().get(Calendar.YEAR);
        mes=1+ Calendar.getInstance().get(Calendar.MONTH);
        dia=Calendar.getInstance().get(Calendar.DAY_OF_MONTH);
    }
}
```

- (0,5 ptos) Definir el **constructor** de la **clase Movimiento**. El constructor sólo recibirá como parámetros la cantidad, si es un ingreso o no y el tipo del movimiento. Al atributo fecha se le asignará la fecha actual.
- (0,5 ptos) Definir un **método mismoAño** en la **clase Fecha** que reciba una fecha como parámetro y devuelva verdadero si la fecha utilizada para llamar al método es del mismo año que la fecha del parámetro.
- (1,5 pto) Definir un **método calcularInteres** en la **clase Cuenta** que calcule el interés de la cuenta menos las comisiones, añada el resultado al saldo de la cuenta y cree un movimiento con dicho cálculo que asignará al último movimiento de la cuenta. El resultado se calculará teniendo en cuenta que se dará un interés de un 1% para los primeros 1000€ de saldo, 1,5 para las

cantidades entre 1000 y 5000 y 1,8 para el resto. Por ejemplo, un saldo de 7000€ dará un interés de  $1 \cdot 1000 + 1,5 \cdot 4000 + 1,8 \cdot 2000$ . Además, se cobrarán 5€ de mantenimiento de cuenta si el último movimiento no se realizó en el año actual. El método devolverá -1 si el nombre del titular está a null y 0 en caso contrario.

**3.- (3 ptos)** Realizar una aplicación en java para gestionar las habitaciones de un hospital

Las habitaciones pueden tener 1 o 2 camas y están identificadas por un número entre 1 y 50. Las habitaciones de la 10 a la 20 son dobles, las demás son individuales.

Cuando ingresa un paciente se le asigna un identificador (número entero) y una habitación libre. Los ingresos pueden ser de 2 tipos: urgentes o planeados.

Cuando ingresa un paciente se le asigna automáticamente un identificador único, el tipo de ingreso y se le asigna una habitación y cuando se da de alta se libera la habitación utilizada por el paciente.

Crear una aplicación para la gestión de las habitaciones de dicho hospital con un menú con las siguientes opciones:

- 1.- Ingreso de paciente: se le asigna un identificador, que no se puede repetir para otro paciente, y se pide por teclado el tipo de ingreso (1-urgente, 2-planeado). Se le asigna una cama libre. Si no hay camas libres se indicará con un mensaje por pantalla
  - 2.- Alta de paciente: se pide el identificador del paciente y se libera la cama que ocupaba.
  - 3.- Indicar el número total de habitaciones libres y un listado del número de camas libres de cada habitación con alguna cama libre. Se indicará el número de habitación seguido del número de camas libres (ejemplo Habitación 3: 2 camas libres). Solo aparecerán en el listado las habitaciones con camas libres.
  - 4.- Listar los identificadores de los pacientes y las habitaciones en las que se encuentran con el formato del siguiente ejemplo: Paciente: 32 en Habitación 2 con tipoIngreso: Urgente
- No es necesario que el listado esté ordenado.
- 0.- Fin de la aplicación

Se utilizará una clase Habitación en la que tengas como atributo, el identificador de la habitación y un array de objetos de la clase Cama.

La clase Cama tendrá como atributo un objeto de tipo Paciente

El Paciente tiene como atributos el identificador y el tipo de ingreso.

Se tendrá una clase Hospital con un atributo que será el array de habitaciones y que se creará, con las características indicadas en el enunciado.

En todos los ejercicios se podrá utilizar la clase Teclado vista en clase o la clase Scanner.

No se deberá leer o escribir en métodos que no estén expresamente dedicados a lecturas o escrituras.

Se valorará la correcta elección de las estructuras y clases.