



Examen de Metodología de la Programación Curso 2020-2021, Convocatoria Febrero

Sección Informativa:

Duración del examen: 2 horas 30 minutos

1- (6 Puntos). C++.

1) Dada la siguiente clase genérica:

```
class Fecha {  
    int dia;  
    int mes;  
    int anio;  
public:  
    Fecha(int d, int m, int a);  
    ...  
};
```

```
template <class T>  
class Jugador {  
    const int dni;  
    Fecha fecha;  
    T id;  
public:  
    Jugador(int d, Fecha f, T id);  
    ~Jugador();  
};
```

Indica exclusivamente cómo sería el código del constructor y del destructor para que sirva y funcione correctamente para los tipos `<int>`, `<float>`, `<char>` y `<char*>` especializando aquellos métodos que consideres necesarios. A partir de Jugador crea una clase derivada JugadorPlus que tenga un atributo `const int edad` e indica cómo sería su constructor y su destructor. (1.50 puntos)

2) Dado el siguiente código:

```
class Error: public exception {  
    char *mensaje;  
public:  
    Error(char *m): exception() {  
        mensaje=new char[strlen(m)+1];  
        strcpy(mensaje, m);  
    }  
    const char *what() const throw() { return mensaje; }  
};  
  
class Fecha { //clase immutable  
    int dia, anio;  
    char *mes;  
    static int diasValidos[];  
    static const char* mesesValidos[];  
public:  
    Fecha(int d, char* m, int a) { dia=d; mes=m; anio=a; }  
    int getDia() { return this->dia; }  
    char* getMes() { return this->mes; }  
    int getAnio() { return this->anio; }  
};  
  
int Fecha::diasValidos[]={31,28,31,30,31,30,31,31,30,31,30,31};  
const char *Fecha::mesesValidos[]{"ene", "feb", "mar", "abr", "may", "jun", "jul", "ago",  
                                   "sep", "oct", "nov", "dic"};
```

Corrige los errores de la clase Fecha y haz que sea robusta, segura y sirva para cualquier `main()` posible. La clase Fecha solo debe permitir crear fechas válidas. Si la fecha no es válida debe lanzar una excepción indicando si el error está en el mes o en el día (lanza la primera que se produzca, en ese orden). Para simplificar las cosas suponed que ningún año es bisiesto. (2.00 puntos)

3) Dado el siguiente código:

```
class Fecha {
    int dia, anio;
    char *mes;
public:
    Fecha(int d, char* m, int a)
    ...
};

class Cliente {
    char *nombre, *dni;
    Fecha fecha;
public:
    Cliente(const char *d, const char *nom, Fecha f);
    Cliente(const Cliente& c);
    ...
};
```

```
class ClienteVIP: public Cliente {
    ...
};

class ClienteNOVIP: public Cliente {
    ...
};

class Empresa {
    Cliente **clientes;
    int ncli, nmaxcli;
public:
    Empresa();
    void alta(Cliente *c);
    void bajaCliente(char *dni);
    ...
};
```

Corrige los errores del método `bajaClientesVIPFecha()` e implementa todos los métodos necesarios en cada clase respectiva para que dicho método pueda funcionar. **(2.50 puntos)**

```
void Empresa::bajaClientesVIPFecha(Empresa e, Fecha f) {
    for(i=0; i<100; i++) {
        if (clientes[i]->fecha==f) {
            bajaCliente(clientes[i].dni); //al eliminar se corren los siguientes hacia atras
        }
    }
}

int main() {
    Fecha f1(2, "ene", 2017);
    Empresa ING;
    ING.alta(new ClienteVIP("juan", ...));
    ING.alta(new ClienteNOVIP("eva", ...));
    ...
    Empresa::bajaClientesFecha(ING, f1); //elimina de ING y de la memoria los Clientes VIP
    //datos de alta la fecha f1
}
```

2- (4 Puntos). Java:

1) Indica los posibles errores del siguiente código justificando el motivo:

(1.00 puntos)

```
public class A {
    private final int at = 5;

    public static int met() { return at; }
    public void proceso() { System.out.println(at); at++; }
    public final void saludo() { System.out.println("hola"); }
}

public class B extends A {
    public void saludo() { System.out.println(super.saludo()+" bienvenido"); }
}
```

2) ¿Cuál será el resultado de la ejecución del método *main*? Justifica la respuesta:

(0.75 puntos)

```
public class HiloEjecucion {
    public static int[] datos = new int[3];

    public static String met(int i) {
        String salida = "";
        salida = salida + datos[++i];
        try {
            salida += datos[i+1];
            salida += "OK";
        } catch (Exception e){
            salida += "Excepcion";
        } finally {
            salida += "Finally";
        }
        salida += "---";
        return salida;
    }

    public static void main(String [] args) {
        System.out.println(met(-1));
        System.out.println(met(1));
        System.out.println(met(2));
    }
}
```

- 3) Dado la siguiente clase implementa los métodos **estrictamente necesarios** para que el main() que aparece a continuación pueda ejecutarse y produzca la salida indicada: **(2.25 puntos)**

```
package libClases;

public class Persona implements Cloneable {
    private int edad;
    private int [] telefonos; //capacidad inicial 2 ampliable dinámicamente
    private int n;
    private String nombre;

    public Persona(String nombre, int e) {
        this.nombre = nombre;
        edad = e;
        n = 0;
        telefonos = new int[2];
    }

    public void setEdad(int e) { edad=e;      }

    //A RELLENAR POR EL ALUMNO
}
```

```
package libPruebas;

import libClases.*;

public class Prueba1 {
    public static void main(String[] args) {
        Persona a=new Persona("juan", 23), b=(Persona)a.clone();
        a.setEdad(46);
        System.out.println(a);
        a.agregar(959217388); a.agregar(959217373);
        Persona c=(Persona)a.clone();
        c.agregar(959444444);
        System.out.println(a);
        System.out.println(b);
        System.out.println(c);
    }
}
```

Salida:

juan tiene 46 años y 0 telefonos	//a
juan tiene 46 años y 2 telefonos (959217388 959217373)	//a
juan tiene 23 años y 0 telefonos	//b
juan tiene 46 años y 3 telefonos (959217388 959217373 959444444)	//c