

# Pauta Certamen 1, Programación II

*Prof. Rodrigo Olivares*  
*Ayud. Juan Carlos Tapia*  
*Abril 14, 2016*

---

## Instrucciones:

- El puntaje máximo del certamen es 100%, siendo el 60% el mínimo requerido para aprobar.
- Responda cada pregunta en la hoja indicada, agregando su nombre. Si no responde alguna pregunta, debe entregar la hoja con su nombre e indicar que **no responde**.
- El certamen es **individual**. Cualquier intento de copia, será sancionado con nota **1,0**.

1. *30pts.* De las siguientes afirmaciones, encierre en un círculo la o las alternativas correctas.

- |   |  |
|---|--|
| i. El paradigma de la orientación a objeto se basa en:                                    | vi. Un objeto es:  |
| <input checked="" type="radio"/> (a) El uso de objetos y su interacción.                  | <input type="radio"/> (a) Un tipo de dato de la clase.   |
| <input checked="" type="radio"/> (b) El uso de clases como categorización de instancias.  | <input checked="" type="radio"/> (b) La instancia de una clase.                                |
| <input type="radio"/> (c) El uso de un enfoque imperativo.                                | <input checked="" type="radio"/> (c) Una abstracción del mundo real.                           |
| <input checked="" type="radio"/> (d) El uso de un enfoque reusable.                       | <input type="radio"/> (d) Un sub-conjunto de atributos y métodos de la clase.                  |
| <input checked="" type="radio"/> (e) Principios como herencia y abstracción.              | <input type="radio"/> (e) Siempre estático.  |
| ii. En cuanto a la programación orientada a objeto:                                       | vii. El principio de ocultamiento:   |
| <input type="radio"/> (a) Se apoya en el paradigma estructural.                           | <input checked="" type="radio"/> (a) Es una técnica que protege el estado de una entidad.      |
| <input checked="" type="radio"/> (b) Divide el programa en pequeñas unidades de código.   | <input checked="" type="radio"/> (b) Es indispensable en el paradigma de orientación a objeto. |
| <input checked="" type="radio"/> (c) Proporciona herramientas para modelar el mundo real. | <input checked="" type="radio"/> (c) En Java, se logra utilizando los modificadores de acceso. |
| <input type="radio"/> (d) Es una herramienta del lenguaje JAVA.                           | <input type="radio"/> (d) Es encapsular el conocimiento de una entidad.                        |
| <input type="radio"/> (e) Ninguna de las anteriores.                                      | <input type="radio"/> (e) Ninguna de las anteriores.   |
| iii. Una clase es:  | viii. Respecto a la herencia, las clases:  |
| <input type="radio"/> (a) Un arreglo de objetos.  | <input type="radio"/> (a) Heredan sólo los métodos privados.                                   |
| <input type="radio"/> (b) Una absorción del mundo real.                                   | <input checked="" type="radio"/> (b) Heredan el comportamiento completo de la clase padre.     |
| <input type="radio"/> (c) Una herramienta de programación.                                | <input type="radio"/> (c) Heredan sólo el comportamiento que se desea utilizar.                |
| <input type="radio"/> (d) Un punto a memoria.   | <input type="radio"/> (d) En Java, se implementan con la palabra implements.                   |
| <input checked="" type="radio"/> (e) Ninguna de las anteriores.                           | <input checked="" type="radio"/> (e) En Java, se implementan con la palabra extends.           |
| iv. Respecto a una clase:   | ix. El polimorfismo:   |
| <input type="radio"/> (a) Se declaran utilizando la palabra reservada clase.              | <input checked="" type="radio"/> (a) El mismo nombre implementa distintas funcionalidades.     |
| <input checked="" type="radio"/> (b) Debe tener el mismo nombre que el archivo.           | <input type="radio"/> (b) Una funcionalidad implementada con distintos nombres.                |
| <input type="radio"/> (c) Son declaradas estáticas con static.                            | <input type="radio"/> (c) Es una característica de JAVA.                                       |
| <input type="radio"/> (d) Todas deben incluir el método main.                             | <input checked="" type="radio"/> (d) Es una característica de POO.                             |
| <input type="radio"/> (e) Ninguna de las anteriores.                                      | <input checked="" type="radio"/> (e) Un ejemplo es el símbolo %.                               |
| v. Respecto a una clase:  | x. El método main:   |
| <input checked="" type="radio"/> (a) Puede o no tener atributos.                          | <input type="radio"/> (a) Puede no ser void.   |
| <input checked="" type="radio"/> (b) Puede o no tener métodos.                            | <input checked="" type="radio"/> (b) Debe ser static.  |
| <input checked="" type="radio"/> (c) Puede o no tener constructor.                        | <input type="radio"/> (c) Puede no llevar argumentos de entrada.                               |
| <input type="radio"/> (d) Puede o no tener un nombre.                                     | <input type="radio"/> (d) Debe retornar un valor.  |
| <input type="radio"/> (e) Puede o no tener un tipo.                                       | <input checked="" type="radio"/> (e) Debe incluirse en un programa.                            |

2. *70pts*. Como la asignatura Programación 2 tiene demasiados alumnos, el profesor dividió el curso en 2 grupos (A y B) equitativos, para tomar evaluaciones (cantidades semejantes en ambos grupos). En la asignatura se realizarán 3 evaluaciones. En la primera evaluación ingresan primero los del grupo A y luego del B. En la segunda evaluación es a la inversa. Para la tercera evaluación, el profesor decidió que para ser justo, los integrantes de los grupos serán reordenados de forma aleatoria, por lo cual le ha solicitado a Ud que desarrolle un programa que, seleccione randómicamente y sin repetición, los alumnos que formaran parte del grupo A y B y luego muestre los dos grupos en la salida estándar.

Considere:

- La lista total de los alumnos es un valor aleatorio mayor a 40 y menor a 100.
- La lista contiene alumnos (agregue la propiedad "id" de tipo entero que almacenará su identificador aleatorio, no repetible en entre los alumnos de la lista).
- Desarrolle el programa bajo el paradigma de orientación a objeto.

¿Cómo será evaluado en la pregunta 2?			
Tópico	Logrado	Medianamente logrado	No logrado
Construir entidades	<i>15pts</i> Crea la clase atómica con sus atributos/métodos.	<i>7pts</i> Crea la clase atómica sin los atributos o métodos.	<i>0pts</i> No crea la clase atómica.
Construir clase Lista y sus métodos	<i>25pts</i> Define e implementa correctamente la clase Lista, sus atributos y métodos:  Atributos: listas, tamaño, límites, etc. Métodos: llenar, generar id sin repetir, mostrar, etc.	<i>10pts</i> Define algunos atributos o algunos métodos, pero no todos los necesarios para el problema.	<i>0pts</i> No define ni los atributos ni métodos
Construir clase principal	<i>15pts</i> Define la clase con el método principal.	<i>10pts</i> Define el método principal en la misma clase.	<i>0pts</i> No define el método principal.
Paradigma Orientación a Objetos	<i>15pts</i> Resuelve el problema utilizando el POO.	<i>8pts</i> Utiliza parte del POO para resolver el problema.	<i>0pts</i> No utiliza el POO para dar solución al problema.
Total máximo puntaje pregunta 2	<i>70pts</i>	<i>35pts</i>	<i>0pts</i>

### Clase atómica - Alumno

```
public class Alumno {

    private int id;

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    @Override
    public String toString() {
        return String.format("AlumnoID: %d", id);
    }
}
```

## Clase Lista - ListaAlumno

```
import java.util.ArrayList;
import java.util.Random;

public class ListaAlumno {

    private ArrayList<Alumno> listaAlumnos = new ArrayList<>();
    private ArrayList<Alumno> listaAlumnosGrupoA = new ArrayList<>();
    private ArrayList<Alumno> listaAlumnosGrupoB = new ArrayList<>();
    private Random rnd;
    private int tamanio, MIN_LISTA = 40, MAX_LISTA = 75, MAX_ID = 100;

    public ListaAlumno() {
        rnd = new Random();
        tamanio = rnd.nextInt(MAX_LISTA - MIN_LISTA) + MIN_LISTA + 1;
    }

    public void llenarLista() {
        Alumno a;
        for (int i = 0; i < tamanio; i++) {
            a = new Alumno();
            a.setId(generarIDSinRepeticion());
            listaAlumnos.add(a);
        }
    }

    public void dividirListas() {
        Alumno a;
        for (int i = 0; i < tamanio; i++) {
            a = listaAlumnos.remove(rnd.nextInt(listaAlumnos.size()));
            if (rnd.nextBoolean()) {
                listaAlumnosGrupoA.add(a);
            } else {
                listaAlumnosGrupoB.add(a);
            }
        }
    }

    private int generarIDSinRepeticion() {
        int id;
        do {
            id = generarID();
        } while (id < 0);
        return id;
    }

    private int generarID() {
        int id = rnd.nextInt(MAX_ID) + 1;
        for (Alumno alumno : listaAlumnos) {
            if (alumno.getId() == id) {
                id = -1;
                break;
            }
        }
        return id;
    }
}
```

```

    public void mostrarLista() {
        System.out.println("Curso:");
        for (Alumno a : listaAlumnos) {
            System.out.println(a);
        }
    }
    public void mostrarGrupos() {
        System.out.println("Grupo A:");
        for (Alumno a : listaAlumnosGrupoA) {
            System.out.println(a);
        }
        System.out.println("Grupo B:");
        for (Alumno a : listaAlumnosGrupoB) {
            System.out.println(a);
        }
    }
}

```

### Clase Principal - OrdenarListaAlumno

```

public class OrdenListaAlumno {

    public static void main(String[] args) {
        ListaAlumno la = new ListaAlumno();
        la.llenarLista();
        la.mostrarLista();
        la.dividirListas();
        la.mostrarGrupos();
    }
}

```