

# Certamen 1, Programación II

*Prof. Rodrigo Olivares*

*Ayud. Diego Aguyo*

*Abril 22, 2015*

---

## 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. *20pts.* De las siguientes afirmaciones, encierre en un círculo la o las alternativas correctas.

- |  |   |
|--|---|
| i. La orientación a objeto es:                         | vi. Un objeto es:   |
| (a) Un paradigma de programación.                      | (a) Una abstracción del mundo real.                           |
| (b) Un paradigma de programación estructurado.         | (b) Un tipo de dato.  |
| (c) Una herramienta de programación.                   | (c) La instancia de una clase.                                |
| (d) Un lenguaje de programación.                       | (d) Un conjunto de atributos y métodos.                       |
| (e) Ninguna de las anteriores.                         | (e) Una plantilla para generar más objetos.                   |
| ii. Algunos enfoques de la orientación a objeto son:   | vii. El principio de ocultamiento:                            |
| (a) El enfoque de reusabilidad.                        | (a) Es una técnica que protege el estado de una entidad.      |
| (b) El enfoque revolucionario.                         | (b) Es dispensable en el paradigma de orientación a objeto.   |
| (c) El enfoque evolutivo.                              | (c) En Java, se logra utilizando los modificadores de acceso. |
| (d) El enfoque imperativo.                             | (d) Es encapsular el conocimiento de una entidad.             |
| (e) El enfoque procedural.                             | (e) Ninguna de las anteriores.                                |
| iii. En cuanto a la programación orientada a objeto:   | viii. Respecto a la herencia, las clases:                     |
| (a) Se apoya en el paradigma procedural.               | (a) Heredan sólo los métodos de igual nombre.                 |
| (b) Divide el programa en pequeñas unidades de código. | (b) Heredan el comportamiento completo de la clase padre.     |
| (c) Proporciona técnicas para modelar el mundo real.   | (c) Heredan sólo el comportamiento que se desea utilizar.     |
| (d) Es un lenguaje de programación.                    | (d) En Java, se implementan con la palabra implements.        |
| (e) Ninguna de las anteriores.                         | (e) En Java, se implementan con la palabra extends.           |
| iv. Una clase es:                                      | ix. Un constructor:   |
| (a) Una colección de objetos.                          | (a) Tiene siempre el mismo nombre de la clase.                |
| (b) Una abstracción del mundo real.                    | (b) Puede o no incluirse en la clase.                         |
| (c) Una herramienta de programación.                   | (c) Debe tener parámetros de entrada.                         |
| (d) Un tipo de dato.                                   | (d) No puede ser sobrecargado.                                |
| (e) Ninguna de las anteriores.                         | (e) Debe incluir el tipo de dato de retorno.                  |
| v. Respecto a una clase:                               | x. El método <i>main</i> :                                    |
| (a) Sólo pueden ser públicas.                          | (a) Siempre debe ser void.                                    |
| (b) Debe tener el mismo nombre que el archivo.         | (b) Siempre debe ser static.                                  |
| (c) Una clase puede o no tener un constructor.         | (c) Siempre debe llevar argumentos de entrada.                |
| (d) Una clase puede o no tener atributos.              | (d) Puede retornar un valor.                                  |
| (e) Ninguna de las anteriores.                         | (e) Puede no incluirse en un programa.                        |

2. 25pts. La siguiente clase describe un cierto comportamiento para cálculos aritméticos:

```
public class Calculo {

    private ArrayList lista = new ArrayList();
    private Random rand = new Random();
    private int size;

    public void crearTamano() {
        size = rand.nextInt(99) + 1;
    }

    public void llenar() {
        for (int lista = 0; lista < size; lista++) {
            this.lista.add(rand.nextInt(99));
        }
    }

    public int suma() {
        int suma = 0;
        for (int lista = 0; lista < size; lista++) {
            suma += Integer.parseInt(this.lista.get(lista).toString());
        }
        return suma;
    }

    public float promedio() {
        return (this.suma() / size);
    }
}
```

- (a) 5pts. Construya el método *main* que permita crear el tamaño de la lista, luego la llene y muestre las operaciones aritméticas.
- (b) 7pts. Construya un *ruteo* para la clase anterior (incluyendo el método *main*).
- (c) 13pts. Agregue un método a la clase *Calculo* que permita obtener la *mediana* (el dato del medio, de una lista ordenada).

3. *25pts.* Desarrollo las clases necesarias para gestionar un arreglo de números de un tamaño determinado por teclado. El arreglo contendrá números aleatorios entre 1 y 1000. Se deberán mostrar todos aquellos números que terminen en un dígito que se indique por teclado (se debe controlar que el valor introducido sea correcto). Por ejemplo, en un arreglo de 10 elementos, indicamos mostrar los números terminados en 5. La salida sería: 155, 25, 5, etc.

4. *30pts.* Desarrolle las clases necesarias que permitan clasificar smart-phones dependiente de su sistema operativo (iOS, Andriod, Windows Phone, etc). Los datos asociados a los atributos de un smart-phone deben ser ingresados por la entrada estándar (considere al menos 3 atributos). Al momento de terminar el ingreso, se debe mostrar la lista de cada uno de ellos, indicando la cantidad. Recuerde utilizar todos los conceptos de orientación a objetos vistos en clase.