

# Certamen Recuperativo, Programación II

Prof. Rodrigo Olivares

---

## Instrucciones:

- El puntaje máximo del certamen es 100%, siendo el 60% el mínimo requerido para aprobar.
- El certamen es **individual**. Cualquier intento de copia, será sancionado con nota **1,0**.

1. 100pts. De las siguientes afirmaciones, encierre en un círculo la o las alternativas correctas (4pts c/u).

- |   |   |
|---|---|
| i. La orientación a objeto es:                                | viii. Un constructor:                                     |
| (a) Un paradigma de programación procedural.                  | (a) No siempre tiene el mismo nombre de la clase.         |
| (b) Un paradigma de programación estructurado.                | (b) Puede o no incluirse en la clase.                     |
| (c) Una herramienta de programación.                          | (c) Debe tener parámetros de entrada.                     |
| (d) Un lenguaje de programación.                              | (d) No puede ser sobrecargado.                            |
| (e) Ninguna de las anteriores.                                | (e) Debe incluir el tipo de dato de retorno.              |
| ii. Algunos enfoques de la orientación a objeto son:          | ix. El método <i>main</i> :                               |
| (a) El enfoque de reusabilidad.                               | (a) Siempre debe ser void.                                |
| (b) El enfoque revolucionario.                                | (b) Siempre debe ser static.                              |
| (c) El enfoque evolutivo.                                     | (c) Siempre debe llevar argumentos de entrada.            |
| (d) El enfoque imperativo.                                    | (d) Puede retornar un valor.                              |
| (e) El enfoque procedural.                                    | (e) Puede no incluirse en un programa.                    |
| iii. En cuanto a la programación orientada a objeto:          | x. Una clase abstracta:                                   |
| (a) Se apoya en el paradigma procedural.                      | (a) Tiene al menos un método impementado.                 |
| (b) Divide el programa en pequeñas unidades de código.        | (b) Tiene al menos un método abstracto.                   |
| (c) Proporciona técnicas para modelar el mundo real.          | (c) Tiene todos sus métodos abstractos                    |
| (d) Es un lenguaje de programación.                           | (d) Es factible de ser implementada.                      |
| (e) Es una herramienta de programación.                       | (e) Es factible de ser extendida.                         |
| iv. Una clase es:   | xi. Una interface:  |
| (a) Una colección de objetos.                                 | (a) Tiene al menos un método impementado.                 |
| (b) Una abstracción del mundo real.                           | (b) Tiene al menos un método abstracto.                   |
| (c) Una herramienta de programación.                          | (c) Tiene todos sus métodos abstractos                    |
| (d) Un tipo de dato.  | (d) Es factible de ser implementada.                      |
| (e) Ninguna de las anteriores.                                | (e) Es factible de ser extendida.                         |
| v. Respecto a una clase:                                      | xii. Respecto a la herencia, las clases:                  |
| (a) Sólo pueden ser públicas.                                 | (a) Heredan sólo los métodos de igual nombre.             |
| (b) Debe tener el mismo nombre que el archivo.                | (b) Heredan el comportamiento completo de la clase padre. |
| (c) Debe tener al menos un constructor.                       | (c) Heredan sólo el comportamiento que se desea utilizar. |
| (d) Su constructor debe tener el mismo nombre.                | (d) En Java, se implementan con la palabra implements.    |
| (e) Puede o no tener atributos.                               | (e) En Java, se implementan con la palabra extends.       |
| vi. Un objeto es:   | xiii. La herencia múltiple:                               |
| (a) Una abstracción del mundo real.                           | (a) Permite heredar diverso compartimiento.               |
| (b) Un tipo de dato.  | (b) Apoya el principio ocultamiento.                      |
| (c) La instancia de una clase.                                | (c) Apoya el principio de encapsulamiento.                |
| (d) Un conjunto de atributos y métodos.                       | (d) En Java se desarrolla implementado clases abstractas. |
| (e) Una plantilla para generar más objetos.                   | (e) En Java se desarrolla implementado interfaces.        |
| vii. El principio de ocultamiento:                            | xiv. Un TDA Bean:   |
| (a) Es una técnica que protege el estado de una entidad.      | (a) No requiere constructor.                              |
| (b) Es útil en enfoques procedurales.                         | (b) Puede tener sobre carga de constructor.               |
| (c) En Java, se logra utilizando los modificadores de acceso. | (c) Apoya el principio ocultamiento.                      |
| (d) Es encapsular el conocimiento de una entidad.             | (d) Apoya el principio de encapsulamiento.                |
| (e) Ninguna de las anteriores.                                | (e) Tiene métodos privados y atributos públicos.          |

- xv. Una clase estática:
- (a) Requiere instanciación.
  - (b) No requiere instanciación.
  - (c) No es viable crear instancias.
  - (d) Sus atributos y métodos son miembros del objeto.
  - (e) Ninguna de las anteriores
- xvi. Respecto a las interfaces gráfica en Java:
- (a) Swing sustituye a AWT.
  - (b) AWT sustituye a Swing.
  - (c) Swing y AWT poseen los mismos componentes.
  - (d) Swing incorpora los AWT Components.
  - (e) Ninguna de las anteriores
- xvii. Referente a JFrame:
- (a) Habitualmente se usa para crear la ventana principal.
  - (b) Su método getContentPane() obtiene el panel principal.
  - (c) Su método add() permite agregar componentes al panel.
  - (d) Su método setSize() permite dimensionar la ventana.
  - (e) Ninguna de las anteriores
- xviii. Para realizar acciones desde un botón Se requiere:
- (a) Crear una clase que implemente un `ActionEvent`.
  - (b) Crear una clase que implemente un `ActionListener`.
  - (c) Sobreescibir el método `actionList(ActionPerformed)`
  - (d) Sobreescibir el método `actionPerformed(ActionEvent)`
  - (e) Agregar la instancia de la clase oyente, al botón.
- xix. Algunos JComponents :
- (a) `JPanel`, `JScrollPane`, `JDialog`.
  - (b) `JPanel`, `JScrollPane`, `JDialog`.
  - (c) `JFileChooser`, `JScrollPane`, `JLabel`.
  - (d) `JList`, `JButton`, `TextField`.
  - (e) `JPassword`, `JFrame`, `TextArea`.
- xx. *4pts.* Sobre los JLabel:
- (a) Su constructor recibe texto.
  - (b) Su constructor recibe imágenes.
  - (c) Su constructor recibe texto e imágenes, al mismo tiempo.
  - (d) Su constructor recibe el tamaño del texto a mostrar.
  - (e) No pueden ser modificados en ejecución.