

Certamen Recuperativo, Programación II

Prof. Rodrigo Olivares

Ayud. Diego Agullo

Julio 14, 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. 100pts. De las siguientes afirmaciones, encierre en un círculo la o las alternativas correctas (4pts c/u).

- La orientación a objeto es:
 - Un paradigma de programación procedural.
 - Un paradigma de programación estructurado.
 - Una herramienta de programación.
 - Un lenguaje de programación.
 - ☒ Ninguna de las anteriores.
- Algunos enfoques de la orientación a objeto son:
 - El enfoque de reusabilidad.
 - ☒ El enfoque revolucionario.
 - ☒ El enfoque evolutivo.
 - El enfoque imperativo.
 - El enfoque procedural.
- En cuanto a la programación orientada a objeto:
 - Se apoya en el paradigma procedural.
 - ☒ Divide el programa en pequeñas unidades de código.
 - ☒ Proporciona técnicas para modelar el mundo real.
 - Es un lenguaje de programación.
 - Es una herramienta de programación.
- Una clase es:
 - Una colección de objetos.
 - ☒ Una abstracción del mundo real.
 - ☒ Una herramienta de programación.
 - Un tipo de dato.
 - Ninguna de las anteriores.
- Respecto a una clase:
 - Sólo pueden ser públicas.
 - ☒ Debe tener el mismo nombre que el archivo.
 - Debe tener al menos un constructor.
 - ☒ Su constructor debe tener el mismo nombre.
 - ☒ Puede o no tener atributos.
- Un objeto es:
 - ☒ Una abstracción del mundo real.
 - Un tipo de dato.
 - ☒ La instancia de una clase.
 - ☒ Un conjunto de atributos y métodos.
 - Una plantilla para generar más objetos.
- El principio de ocultamiento:
 - Es una técnica que protege el estado de una entidad.
 - Es útil en enfoques procedurales.
 - ☒ En Java, se logra utilizando los modificadores de acceso.
 - Es encapsular el conocimiento de una entidad.
 - Ninguna de las anteriores.
- Un constructor:
 - No siempre tiene el mismo nombre de la clase.
 - ☒ Puede o no incluirse en la clase.
 - Debe tener parámetros de entrada.
 - No puede ser sobrecargado.
 - Debe incluir el tipo de dato de retorno.
- El método *main*:
 - ☒ Siempre debe ser void.
 - ☒ Siempre debe ser static.
 - ☒ Siempre debe llevar argumentos de entrada.
 - Puede retornar un valor.
 - Puede no incluirse en un programa.
- Una clase abstracta:
 - Tiene al menos un método impementado.
 - ☒ Tiene al menos un método abstracto.
 - Tiene todos sus métodos abstractos
 - Es factible de ser implementada.
 - ☒ Es factible de ser extendida.
- Una interface:
 - Tiene al menos un método impementado.
 - Tiene al menos un método abstracto.

- (c) Tiene todos sus métodos abstractos
- (d) Es factible de ser implementada.
- (e) Es factible de ser extendida.
- xii. Respecto a la herencia, las clases:
- (a) Heredan sólo los métodos de igual nombre.
- (b) Heredan el comportamiento completo de la clase padre.
- (c) Heredan sólo el comportamiento que se desea utilizar.
- (d) En Java, se implementan con la palabra implements.
- (e) En Java, se implementan con la palabra extends.
- xiii. La herencia múltiple:
- (a) Permite heredar diverso compartimiento.
- (b) Apoya el principio ocultamiento.
- (c) Apoya el principio de encapsulamiento.
- (d) En Java se desarrolla implementado clases abstractas.
- (e) En Java se desarrolla implementado interfaces.
- xiv. Un TDA Bean:
- (a) No requiere constructor.
- (b) Puede tener sobre carga de constructor.
- (c) Apoya el principio ocultamiento.
- (d) Apoya el principio de encapsulamiento.
- (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 actionPerformed(ActionEvent)
- (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, JTextField.
- (e) JPasswordField, JFrame, JTextArea.
- 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.