

# Certamen 2, Programación II

Prof. Rodrigo Olivares  
Ayud. Juan Carlos Tapia

Noviembre 3, 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. *30pts.* De las siguientes afirmaciones, encierre en un círculo la o las alternativas correctas.

- |   |  |
|---|--|
| i. Un constructor:  | vi. Con respecto al paso de parámetros:                          |
| (a) No necesariamente debe ir en una clase.                   | (a) Puede ser por valor.   |
| (b) Al ser private sólo instancia objetos dentro de la clase. | (b) Puede ser por omisión.                                       |
| (c) No puede ser sobrecargado.                                | (c) Puede ser por referencia.                                    |
| (d) Si no se declara, se crea uno en tiempo de compilación    | (d) Puede ser por convención.                                    |
| (e) Si no se declara, se crea uno en tiempo de ejecución      | (e) Puede ser por default.                                       |
| ii. La lectura de datos de la entrada estándar:               | vii. El relación a la auto-referencia:                           |
| (a) Puede realizarse con la clase FileReader.                 | (a) Cumple la misma función que el operador <i>super</i> .       |
| (b) Puede realizarse con la clase Scanner.                    | (b) Permite referenciar el constructor de la clase.              |
| (c) Sólo puede ser de tipo String.                            | (c) Permite referenciar los atributos de la clase.               |
| (d) Permite leer los argumentos del método <i>main</i> .      | (d) Permite referenciar los método de la clase.                  |
| (e) Ninguna de las anteriores                                 | (e) Ninguna de las anteriores.                                   |
| iii. En relación a los arreglos:                              | viii. Sobre la herencia:   |
| (a) No almacenan los elementos de tipo char.                  | (a) Se realiza con la palabra reservada <i>extends</i> .         |
| (b) Pueden almacenar tipos de datos primitivos.               | (b) Se realiza con la palabra reservada <i>include</i> .         |
| (c) Pueden almacenar objetos.                                 | (c) Todas las clases heredan de la clase <i>Object</i> .         |
| (d) Se instancian sólo con el operador <i>new</i> .           | (d) La clase padre hereda el comportamiento de la clase hija.    |
| (e) Su dimensión puede ser determinada en ejecución.          | (e) La clase hija hereda el comportamiento de la clase padre.    |
| iv. Una clase abstracta:                                      | xi. Respecto a la herencia múltiple:                             |
| (a) Puede instanciar objetos.                                 | (a) No existe en Java.   |
| (b) Posee sólo métodos abstractos.                            | (b) Existe en Java.  |
| (c) Posee al menos un método abstracto.                       | (c) Se puede emular con clases abstractas.                       |
| (d) No permite extender una interfaz.                         | (d) Se puede emular con interfaces.                              |
| (e) Ninguna de las anteriores.                                | (e) Se puede emular con clases estáticas.                        |
| v. La interfaz <i>List</i> .                                  | x. En relación a la manipulación de archivos.                    |
| (a) Puede ser implementada por la clase <i>ArrayVector</i> .  | (a) Se lee un archivo con la instancia <i>FileWriter</i> .       |
| (b) Puede instanciar objetos.                                 | (b) Se lee un archivo con la instancia <i>FileReader</i> .       |
| (c) Posee sólo métodos abstractos.                            | (c) No es factible agregar contenido a un archivo existente.     |
| (d) Puede ser extendida en una clase hija.                    | (d) <i>StringTokenizer</i> se usa para archivos con delimitador. |
| (e) No posee atributos.                                       | (e) Ninguna de las anteriores.                                   |

2. *30pts.* Desarrolle un programa en Java que permita llenar un array con 10 nombres de personas, ingresados desde la entrada estándar. A partir de ese arreglo, construya un nuevo arreglo que almacene el largo de cada nombre. Por último, muestre el nombre y su largo, desde los arreglos ya creados.

3. *40pts.* El BancoPais le ha solicitado implementar un sistema de seguridad de transferencias bancarias, utilizando un “digipass”. Este dispositivo trabaja con una combinación aleatoria de 3 pares de números (entre el 10 y el 99, inclusive) de un total de 50 posibles valores, no necesariamente distintos. Estos números deben estar almacenados en una fuente de datos permanente, único por cliente. Diseñe un programa en Java que gestione una transferencia bancaria, incluyendo el monto a transferir e ingresando un código de “digipass” y lo compare con cualquier combinación de los valores almacenados. Debe informar al usuario (por la salida estándar) si la transferencia se realizó correcta o incorrectamente, informando para este caso, los errores cometidos.