



APRENDERAPROGRAMAR.COM

INDICE DEL CURSO
“APRENDER
PROGRAMACIÓN JAVA
DESDE CERO.
PROGRAMACIÓN
ORIENTADA A
OBJETOS”(CU00601B)

Sección: Cursos

Categoría: Curso “Aprender programación Java desde cero”

Fecha revisión: 2029

Resumen: Entrega nº1 del curso Aprender programación Java desde cero. Índice del curso.

Autor: Alex Rodríguez

INDICE DEL CURSO

1. CUESTIONES BÁSICAS SOBRE JAVA

- 1.1. ¿Qué es Java?
- 1.2. ¿Es Java un lenguaje ideal para aprender a programar?
- 1.3. ¿Es Java software libre?
- 1.4. ¿Cuáles son las versiones y distribuciones Java y cuál usar?
- 1.5. ¿Qué sistema operativo debo usar para programar Java?
- 1.6. ¿Qué son las actualizaciones de Java?

2. INSTALAR JAVA EN NUESTRO ORDENADOR Y DAR LOS PRIMEROS PASOS

- 2.1. Descargar (download) Java: obtener el instalador
- 2.2. Instalación de Java en Windows (en nuestro pc)
- 2.3. Configuración de Java en Windows: JAVA_HOME, PATH
- 2.4. Máquina virtual JVM, compilador e intérprete Java. Bytecode.
- 2.5. NetBeans, Eclipse, JCreator, JBuilder... ¿Cuál es el mejor entorno de desarrollo (IDE) Java?
- 2.6. Descargar (download) BlueJ. Instalación en Windows.
- 2.7. ¿Qué es un proyecto Java? Creación del primer proyecto.
- 2.8. La ventana del editor de BlueJ
- 2.9. Escribir código (una clase) en Java con un IDE. Primer programa.
- 2.10. Comentarios en lenguaje Java y bloques en Java

3. CONCEPTOS BÁSICOS DE PROGRAMACIÓN ORIENTADA A OBJETOS. CLASES Y OBJETOS.

- 3.1. Visualizar clases y objetos con BlueJ e invocar métodos
- 3.2. Tipos de datos (variables) en Java
- 3.3. Ejemplos de uso de tipos de datos (variables) en Java. Declaración y asignación de contenido.
- 3.4. ¿Qué es una clase? Atributos, constructor y métodos
- 3.5. Métodos tipo procedimiento (...void) y tipo función (...return).
- 3.6. Métodos con y sin parámetros
- 3.7. Métodos consultores o de acceso (getters) y métodos modificadores (setters)
- 3.8. Estado de un objeto
- 3.9. Parámetros formales y parámetros actuales
- 3.10. Comprender la filosofía de métodos y clases en Java
- 3.11. Signatura de un método. Interfaz o interface.
- 3.12. Guardar los proyectos Java. Copias de seguridad.

4. ESCRITURA BÁSICA DE CÓDIGO Y ESTRUCTURAS DE DECISIÓN

- 4.1 Imprimir por pantalla con System.out. Concatenar cadenas. Notación de punto.
- 4.2 Operadores aritméticos en Java.
- 4.3 Operadores lógicos principales en Java.
- 4.4 Sentencia de asignación. Asignación compuesta.
- 4.5 Condicional if else / if else if en Java
- 4.6 Condicional de selección con switch en Java

5. ESTRUCTURAR EL CÓDIGO EN JAVA

- 5.1 Variables locales a un método o constructor. Sobrecarga de nombres.
- 5.2 Cómo crear constructores en Java. Ejemplos.
- 5.3 Clases con dos o más constructores. Sobrecarga de constructores o métodos.
- 5.4 Clases que utilizan objetos preexistentes: relación de uso entre clases. Diagrama de clases.
- 5.5 Paso de objetos como parámetros a un método o constructor
- 5.6 La sentencia new como invocación de un constructor.

6. LIBRERÍAS DEL API DE JAVA. BIBLIOTECAS DE CLASES.

- 6.1 Otra definición de clase: un paquete de código. Objetos del mundo real y objetos abstractos.
- 6.2 ¿Qué es y para qué sirve el API de Java?
- 6.3 Organización y forma de nombrar las librerías en el API de Java
- 6.4 Importar y usar clases del API de Java. Ejemplo: clase Math
- 6.5 Interface de clase. Ejemplo: clase String
- 6.6 Explorar un método. El método substring de la clase String.
- 6.7 Usar métodos para evitar errores. Ejemplo método length de la clase String

7. CONSTRUIR UN PROGRAMA CON JAVA

- 7.1. Concepción de programas mediante abstracción y modularización
- 7.2 Un ejemplo de código Java muy elemental. Crear clases con campos, constructor y métodos.
- 7.3 Definición de método interno y método externo en Java
- 7.4 La palabra clave this. Contenido null por defecto de un objeto.
- 7.5 Clase con el método main: clase principal, iniciadora o "programa" principal.
- 7.6 Sintaxis y código ejemplo de uso del método main.
- 7.7 Pedir datos por consola (teclado) en java. Escape con barra invertida backslash. Print y salto de línea con \n.

8. BUCLES, COLECCIONES DE OBJETOS Y RECORRIDOS DE COLECCIONES

- 8.1 Concepto general de bucle
- 8.2 Bucle con instrucción for. Operador ++ y --. Sentencia break
- 8.3 Bucle con instrucción while. Ejemplo de uso de break.
- 8.4 Bucle con instrucción do ... while.
- 8.5 El debugger de BlueJ. Detener un programa en ejecución.

- 8.6 Pensar en objetos. Pensar una entrada de teclado como objeto.
- 8.7 El método equals. Diferencia entre igualdad e identidad entre objetos.
- 8.8 Asignación de igualdad con tipos primitivos y con objetos.
- 8.9 Repaso y ejemplos sobre igualdad, identidad y método equals.
- 8.10 Colecciones de objetos de tamaño flexible. Contenedores de objetos.
- 8.11 La clase ArrayList del API de Java.
- 8.12 Concepto de clase genérica (clase parametrizada)
- 8.13 El for extendido o bucles for each en Java.
- 8.14 Recorrer colecciones con objetos Iterator. Evitar errores tipo `java.util.ConcurrentModificationException`.
- 8.15 Resumen de tipos de bucles (ciclos) en Java.
- 8.16 Objetos con referencia null y excepciones tipo `java.lang.NullPointerException`.
- 8.17 Autoboxing y unboxing. Conversión automática de tipos envoltorio a primitivos y viceversa.
- 8.18 Objetos anónimos.
- 8.19 Colecciones de tamaño fijo: arrays, arreglos o formaciones.
- 8.20 Campo `length` para saber el número de elementos de un array
- 8.21 Uso de ciclos for each con arrays
- 8.22 Resumen de colecciones

9. MÁS CONCEPTOS Y CUESTIONES BÁSICAS DE JAVA

- 9.1 Conversión de tipos en Java
- 9.2 Método `valueOf` para conversión de tipos
- 9.3 Ejemplo de conversión de tipos. Tipo obtenido frente a tipo requerido. Métodos `get` y `remove` de `ArrayList`
- 9.4 Generar números aleatorios en Java. Clase `Random`.
- 9.5 Variables de clase o estáticas y constantes. Palabras clave `static` y `final`.
- 9.6 Organizar un proyecto Java en paquetes (packages). Cláusulas `package` e `import`
- 9.7 Formas de nombrar packages, jerarquización y visibilidad de clases. Los packages en BlueJ.
- 9.8 Copiar arrays y comparar arrays. Identidad e igualdad entre arrays.
- 9.9 La clase `Arrays` del API de Java. Métodos `equals` y `copyOf`.
- 9.10 Rellenar un array con un valor u objeto. Método `fill` de la clase `Arrays`.
- 9.11 Interfaz o interface en Java. Ampliación del concepto.
- 9.12 Polimorfismo en Java. Primera aproximación.
- 9.13 Transformar un array en una lista con el método `asList` de la clase `Arrays`. Constructores que usan colecciones.
- 9.14 Documentar un proyecto con `javadoc`. Comentarios, símbolos, tags.
- 9.15 Tipos enumerados (enum) en Java.
- 9.16 Método `values`. Enumerados clases con campos y constructores.
- 9.17 Métodos de clase o `static` frente a métodos de instancia. Comprender el método `main`.

10. HERENCIA EN JAVA. ¿QUÉ ES? ¿PARA QUÉ SIRVE?

- 10.1 ¿Qué es la herencia en programación orientada a objetos?
- 10.2 Jerarquías de herencia. Organización y acceso entre clases.

- 10.3 Ejemplo de herencia y uso de palabras clave extends y super. Constructores con herencia.
- 10.4 Ejemplo de herencia descendente o herencia simple.
- 10.5 Jerarquía de tipos. Subtipos. Polimorfismo y variables polimórficas.
- 10.6 Conversión de tipos (enmascaramiento). Hacer casting y ClassCastException.
- 10.7 Determinación del tipo de variables con instanceof.
- 10.8 Tipo estático y tipo dinámico de variables. Sobreescritura (redefinición) de métodos. Métodos polimórficos.
- 10.9 Ejercicio ejemplo de código con herencia, polimorfismo de variables y métodos, y sobreescritura de métodos.
- 10.10 Uso de la palabra clave super para llamar a métodos.
- 10.11 Modificadores de acceso public, private y protected.
- 10.12 Sobreescibir métodos de la clase Object: método toString.
- 10.13 Sobreescibir métodos de la clase Object: método equals.
- 10.14 Clases y métodos abstractos.
- 10.15 Clases abstractas en el API de Java
- 10.16 Herencia múltiple. Interfaces en Java.
- 10.17 Para qué sirven las interfaces en Java.
- 10.18 Ejemplo sencillo de interface en Java.
- 10.19 Implementar una interface del API de Java.
- 10.20 Resumen de herencia en Java.

11. IR MÁS ALLÁ EN JAVA: SWING, GESTIÓN DE ERRORES Y MÁS.

- 11.1 ¿Qué hemos aprendido y qué no hemos aprendido con este curso?