

Objetivos del curso Java Foundations

Visión general

Este curso de estudio está destinado a alumnos con poca experiencia en programación. Se les ofrecerá una introducción a los conceptos, la terminología y la sintaxis orientados a objetos, y a los pasos necesarios para crear programas Java básicos mediante actividades prácticas y participativas. Los alumnos aprenderán los conceptos de la programación Java, a diseñar aplicaciones orientadas a objetos con Java y a crear programas Java mediante actividades prácticas y participativas.

Idiomas del plan de estudios disponibles:

- Árabe, chino simplificado, inglés, francés, japonés, portugués brasileño, ruso, español, indonesio

Duración

- Duración total del curso recomendada: 90 horas*
- Horas de créditos de formación profesional para los educadores que completan la formación de Oracle Academy: 30

** La duración del curso incluye fase de instrucción, autoestudio/deberes, prácticas, proyectos y evaluación.*

Asistentes

Educadores

- Técnicos, educadores de formación profesional y miembros del profesorado de universidades y escuelas superiores de titulaciones de 2 y 4 años que imparten clases de programación informática, tecnología de comunicaciones de la información (ICT) o una asignatura relacionada a un nivel básico.
- Profesores de secundaria y formación profesional que imparten clases de programación informática.

Alumnos

- Alumnos que desean aprender programación Java y crear su propia experiencia en la programación orientada a objetos con Java.
- Este curso es una clase fundamental para estudiantes de ciencias de la computación y, cuando se imparte junto con la programación Java, se puede emplear para preparar a los alumnos para el examen de Ciencias de la computación AP-A.

Requisitos previos

Recomendados

- Taller de Oracle Academy: Introducción a Java mediante Alice
- Taller de Oracle Academy: Creación de programas Java con Greenfoot
- Curso de Oracle Academy: Java Fundamentals

Próximos cursos sugeridos

- Curso de Oracle Academy: Java Programming

Temas lección por lección y objetivos

Sección 1: Introducción

- 1-1 Acerca del curso
 - Identificar los objetivos del curso y su finalidad
 - Comprender el entorno del curso
 - Describir la estrategia de aprendizaje del curso
- 1-2 Una breve historia
 - Mostrar ejemplos de cómo las personas interactúan con Java en su vida diaria
 - Resumir la historia de Java
 - Describir los grupos de productos de tecnología Java
- 1-3 Configuración de Java
 - Describir la diferencia entre el JDK y el JRE
 - Describir la diferencia entre archivos .java y .class
 - Describir las ventajas de un entorno de desarrollo integrado (IDE)
 - Agregar un archivo .java existente a un proyecto Java

Sección 2: Desarrollo de Software en Java

- 2-1 Proceso de desarrollo de software
 - Describir el modelo espiral de desarrollo
 - Reconocer las tareas y subtareas del modelo espiral
 - Reconocer qué ocurre cuando se ignoran los pasos
 - Identificar funciones de software
 - Describir cómo se implantan gradualmente las funciones
- 2-2 ¿Qué hace mi programa?
 - Describir cómo se lee Java línea a línea
 - Definir y utilizar puntos de ruptura
 - Finalizar sentencias con punto y coma (;)
 - Organizar el código con espacios en blanco y otras convenciones
 - Crear comentarios
- 2-3 Introducción a los conceptos de programación orientada a objetos
 - Diferenciar entre la programación de procedimiento y orientada a objetos
 - Describir una clase como un plano de un objeto
 - Describir una clase que se utiliza para crear instancias de un objeto
 - Modelar objetos como una combinación de:
 - Propiedades (campos de datos)
 - Comportamientos (métodos)

Sección 3: Tipos de dato Java

- 3-1 ¿Qué es una variable?
 - Describir las ventajas de las variables
 - Identificar cuatro tipos principales de variables:
 - (boolean, int, double, String)
 - Declarar y asignar valores a variables
 - Asignar un nombre a las variables según las convenciones
- 3-2 Datos numéricos
 - Diferenciar los tipos de dato entero (byte, short, int, long)
 - Diferenciar los tipos de dato de coma flotante (float, double)
 - Manipular y hacer cálculos con datos numéricos
 - Utilizar los paréntesis y el orden de las operaciones

- 3-3 Datos textuales
 - Utilizar el tipo de dato char
 - Utilizar cadenas
 - Concatenar cadenas
 - Describir secuencias de escape
 - Describir mejor las sentencias print
- 3-4 Conversión entre tipos de datos
 - Aprovechar la promoción automática
 - Cuándo se debe tener cuidado con las promociones
 - Convertir variables a otros tipos de dato
 - Cuándo se debe tener cuidado con la conversión
 - Analizar cadenas como valores numéricos
- 3-5 Entrada del teclado
 - Describir la entrada del usuario
 - Crear un JOptionPane para recopilar la entrada del usuario
 - Utilizar un escáner para recopilar la entrada de la consola
 - Utilizar un escáner para recopilar la entrada de un archivo
 - Describir cómo un escáner maneja los tokens y los delimitadores

Sección 4: Clases de biblioteca y métodos Java

- 4-1 ¿Qué es un método?
 - Estructurar el código dentro de una clase
 - Instanciar el objeto
 - Describir las ventajas de los métodos
 - Utilizar el operador de punto (.) para acceder a los campos y métodos de un objeto
 - Proporcionar argumentos a un método
 - Devolver valores desde un método
- 4-2 Declaración import y paquetes
 - Acceder a una clase mediante su nombre totalmente cualificado
 - Describir la función de la sentencia import
 - Utilizar la sentencia import para acceder a una clase en un paquete
 - Describir la finalidad de un asterisco en una sentencia import
 - Identificar los paquetes que se importan automáticamente
- 4-3 Clase String
 - Buscar la clase String en la documentación de la API de Java
 - Describir los métodos de la clase String
 - Comparar dos objetos String de forma lexicográfica
 - Buscar la ubicación de una subcadena en un objeto String
 - Extraer una subcadena de un objeto String
- 4-4 Clase Random
 - Describir la finalidad y los usos de números aleatorios en la programación Java
 - Identificar los métodos de la clase Random que obtienen números aleatorios
 - Obtener números aleatorios en un rango de números
 - Describir la finalidad del valor inicial de los números aleatorios
- 4-5 Clase Math
 - Describir los métodos de la clase Math
 - Utilizar métodos de la clase Math para realizar cálculos matemáticos
 - Utilizar campos de la clase Math

Sección 5: Sentencias de decisión

- 5-1 Expresiones booleanas y construcciones if/else
 - Declarar, inicializar y utilizar variables booleanas
 - Comparar las expresiones booleanas mediante operadores relacionales
 - Crear una sentencia if
 - Crear construcciones if/else
 - Comparar cadenas

- 5-2 Descripción de ejecución condicional
 - Describir la ejecución condicional
 - Describir los operadores lógicos
 - Describir la evaluación de "cortocircuito" de los operadores lógicos
 - Crear construcciones if encadenadas
- 5-3 Sentencia SWITCH
 - Crear una estructura de control de cambio
 - Comparar construcciones IF/ELSE con estructuras de control de cambio
 - Describir la finalidad de la palabra clave BREAK

Sección 6: Construcciones de Iteración

- 6-1 Ciclos FOR
 - Describir los componentes del ciclo for estándar
 - Describir cómo crear y utilizar un ciclo for
 - Describir el ámbito de variable
 - Describir las técnicas de depuración
 - Explicar cómo se producen los ciclos infinitos en Java
- 6-2 Ciclos WHILE y DO-WHILE
 - Utilizar un ciclo WHILE en un programa Java (anterior a la prueba)
 - Utilizar un ciclo DO-WHILE en un programa Java (posterior a la prueba)
 - Describir cuándo un tipo de ciclo puede ser más beneficioso que otro
- 6-3 Uso de sentencias BREAK y CONTINUE
 - Utilizar una sentencia BREAK para salir de un ciclo
 - Utilizar una sentencia CONTINUE para omitir parte de un ciclo
 - Explicar la necesidad de comentarios de Ciclo

Sección 7: Creando clases

- 7-1 Creando de una clase
 - Crear una clase principal/de prueba Java
 - Crear una clase Java en el IDE
 - Utilizar condicionales en los métodos
 - Convertir especificaciones o una descripción en campos y comportamientos
- 7-2 Instanciación de objetos
 - Describir las consecuencias en la memoria de la instanciación de objetos
 - Describir referencias de objetos
 - Describir la diferencia entre la memoria de pila y de montón
 - Describir cómo las cadenas son objetos especiales
- 7-3 Constructores
 - Describir los valores por defecto
 - Hacer que falle el programa con una referencia null
 - Describir el constructor por defecto
 - Escribir un constructor que acepta argumentos
 - Inicializar los campos con un constructor
 - Utilizar esto como una referencia de objeto
- 7-4 Sobrecarga de métodos
 - Describir los efectos de varios constructores en una clase
 - Definir la sobrecarga de un método
 - Explicar la firma de método
 - Describir cuándo es posible o no la sobrecarga
- 7-5 Interacción de objeto y encapsulación
 - Describir la interacción de objeto con mayor detalle
 - Utilizar el modificador privado para definir variables de clase
 - Describir la finalidad de los métodos getter
 - Describir la finalidad de los métodos setter
- 7-6 Variables y métodos estáticos
 - Describir una variable estática y demostrar su uso dentro de un programa
 - Describir un método estático y demostrar su uso dentro de un programa
 - Describir cómo utilizar la palabra clave final con variables estáticas

Sección 8-1: Arreglas y Excepciones

- 8-1 Arreglas unidimensionales
 - Crear e inicializar arreglas unidimensionales
 - Modificar un elemento de arreglo
 - Recorrer una arreglo unidimensional mediante un ciclo for
 - Identificar la causa de `ArrayIndexOutOfBoundsException`
- 8-2 ArrayLists
 - Crear una ArrayList
 - Manipular una ArrayList con sus métodos
 - Recorrer una ArrayList con iteradores y ciclos for-each
 - Utilizar las clases wrapper y el empaquetado automático para agregar tipos de dato primitivos a una ArrayList
- 8-3 Manejo de excepciones
 - Explicar la finalidad del manejo de excepciones
 - Manejar excepciones con una construcción try/catch
 - Describir excepciones comunes devueltas en Java
- 8-4 Conceptos y técnicas de depuración
 - Probar y depurar un programa Java
 - Identificar los tres tipos de errores
 - Aplicar técnicas de depuración
 - Sentencias `print`
 - Uso del depurador de su IDE
 - Aplicar algunos consejos y técnicas de depuración

Sección 9: JavaFX

- 9-1 Introducción a Java FX
 - Crear un proyecto de JavaFX
 - Explicar los componentes del proyecto de JavaFX por defecto
 - Describir los diferentes tipos de nodos y paneles
 - Explicar el gráfico de escena, el nodo raíz, las escenas y las etapas
- 9-2 Colores y formas
 - Crear y utilizar colores personalizados
 - Crear formas y explicar sus propiedades y comportamientos
 - Consulte la documentación de la API de JavaFX
- 9-3 Gráficos, audio y MouseEvents
 - Crear y utilizar una imagen de JavaFX e ImageView
 - Crear y utilizar audio de JavaFX
 - Crear y utilizar MouseEvents
 - Describir expresiones Lambda en las aplicaciones de GUI

Para buscar y registrarse en los eventos programados en su zona, visite el [calendario de eventos de Academy](#).