

Java

Los candidatos para este examen son desarrolladores de aplicaciones que trabajan con Java 6 SE o posterior, estudiantes de desarrollo de software de secundaria y postsecundaria inmediata, o desarrolladores de software de nivel inicial.

Los candidatos deben tener al menos 150 horas de instrucción o experiencia práctica con Java, estar familiarizados con sus características y capacidades y comprender cómo escribir, depurar y mantener código Java bien formado y documentado.

Para aprobar el examen, también se espera que el candidato tenga los siguientes conocimientos y habilidades previos:

- Habilidades de lectura de octavo grado
- Álgebra II
- Habilidades básicas en sistemas operativos de computadora. • Experiencia en interfaz de línea de comandos.

1. Fundamentos de Java

1.1 Describir el uso de main en una aplicación Java

- Firma de main, cómo consumir una instancia de tu propia clase, argumentos de línea de comando

1.2 Realizar entradas y salidas básicas utilizando paquetes estándar

- Imprimir declaraciones, importar y usar la clase Scanner

1.3 Evaluar el alcance de una variable

- Declarar una variable dentro de un bloque, clase o método

1.4 Programas de comentarios y documentos.

- Evaluar la sintaxis de Javadocs, escribir código sintácticamente correcto. comentarios

2. Tipos de datos, variables y expresiones

2.1 Declarar y utilizar variables de tipo de datos primitivos

- Tipos de datos, incluidos byte, char, int, double, short, long, float y boolean; identificar cuando se pierde precisión; inicialización; en qué se diferencian los primitivos de los tipos de objetos contenedores como enteros y booleanos

2.2 Construir y evaluar código que manipule cadenas

- Clases de cadenas y literales de cadenas, comparaciones, concatenación, mayúsculas y minúsculas. longitud; Métodos String.format; operadores de cadenas; la naturaleza inmutable de las cuerdas; inicialización; nulo

2.3 Construir y evaluar código que crea, itera y manipula matrices y listas de matrices

- Matrices unidimensionales y bidimensionales, incluyendo inicialización, nulo, tamaño, elementos iterantes y elementos de acceso; listas de matrices, incluida la adición y eliminación de elementos, atravesando la lista

2.4 Construir y evaluar código que realice análisis, conversión y conversión

- Transmitir entre tipos de datos primitivos, convertir tipos primitivos en tipos de objetos equivalentes, analizar cadenas en números, convertir tipos de datos primitivos en cadenas

2.5 Construir y evaluar expresiones aritméticas

- Operadores aritméticos, asignación, operadores de asignación compuesta, precedencia de operadores

3. Implementación del control de flujo

3.1 Construir y evaluar código que utiliza declaraciones de ramificación

- si, si no, si no, cambiar; línea única versus bloque; anidación; operadores lógicos y relacionales

3.2 Construir y evaluar código que utiliza bucles

- mientras, para, para cada uno, hacer mientras; romper y continuar; anidación; lógico, relacional, y operadores unarios

4. Programación orientada a objetos

4.1 Construir y evaluar definiciones de clases

- Constructores, sobrecarga de constructores, una clase por archivo .java, esta palabra clave, herencia básica y anulación

4.2 Declarar, implementar y acceder a miembros de datos en clases

- privado, público, protegido; miembros de datos de instancia; miembros de datos estáticos; use static final para crear constantes; describir la encapsulación

4.3 Declarar, implementar y acceder a métodos

- privado, público, protegido; parámetros del método; tipo de retorno; vacío; valor de retorno; métodos de instancia; métodos estáticos; sobrecargar

4.4 Crear instancias y usar objetos de clase en programas

- Creación de instancias, inicialización, nulo, acceso y modificación de miembros de datos, acceso. métodos, acceder y modificar miembros estáticos, importar paquetes y clases

5. Compilación y depuración de código

5.1 Solucionar errores de sintaxis, errores lógicos y errores de tiempo de ejecución

- Imprimir sentencias, salida de comandos javac, errores lógicos, excepciones de consola, evaluación de seguimiento de pila

5.2 Implementar manejo de excepciones

- intentarlo, atraparlo, finalmente; Clase de excepción; tipos de clases de excepción; mostrar excepción información

