ii



# Conceptos fundamentales de Java 4-2: Clases de objeto y piloto Actividades prácticas

**Objetivos de la lección:**

* Describir la forma general de un programa Java
* Describir la diferencia entre una clase de objeto y una clase de controlador
* Acceder a un mínimo de dos API de clase Java
* Explicar y dar ejemplos de palabras clave de Java
* Crear clases de objeto
* Crear clases de controlador



Copyright © 2020 Oracle y/o sus filiales. Todos los derechos reservados. Oracle y Java son marcas comerciales registradas de Oracle y sus filiales. Todos los demás nombres pueden ser marcas comerciales de sus respectivos propietarios.

**Vocabulario**

Identifique el término correspondiente a cada una de las definiciones a continuación.

|  |  |
| --- | --- |
| Paquete | Un grupo de clases Java relacionadas. |
| Bloque de código | Secciones de código que se escriben entre llaves. {} |
| UpperCamelCase | Primera letra mayúscula y mayúscula inicial en la primera letra de cada palabra interna. Ejemplo: CuentaCorriente |
| Variable | Valor con nombre que no cambia. |
| LowerCamelCase | Primera letra minúscula y mayúscula inicial en la primera letra de cada palabra interna. Ejemplo: nombreEstudiante |
| Main class | Clase que contiene un método principal. |
| Sentencia de importación | Sentencia de código en un archivo de clase Java que incluye código Java de otro paquete o clase. |
| Clase de objeto | Clase que define las instancias de objetos que hay que usar en otra clase. |
| Comentario | Código precedido por //. Los comentarios se utilizan para aclarar la lógica de programación. El compilador ignora los comentarios. |
| Palabra clave | Palabra que tiene una función especial en el lenguaje Java y no puede utilizarse como nombre de clase, método o variable. |
| Librería | Biblioteca de clases Java disponible para importar en una clase creada por el programador. |
| Clase | Estructura de un objeto, incluidas las variables de clase, los constructores y los métodos. |
| Método constructor | Tipo especial de método que es una plantilla para un objeto. |
| Argumentos | Valores que se envían a un método o constructor para que se utilicen en un cálculo o para sustituirlos por valores de la clase. |
| Tipo de datos | Valores como números, caracteres o booleanos. Referencias a objetos como un objeto CuentaBancaria. |
| Modificador de acceso | Palabras clave para especificar la accesibilidad de una clase (o tipo) y sus miembros. Por ejemplo: public, private, protected, default |
| Método | Bloque de código dentro de una clase que se usa para cambiar la información de la clase o acceder a ella. |

**Inténtelo/resuélvalo:**

1. Enumere los componentes que forman un archivo .java. Enumere los componentes en el orden en que esperaría verlos en un programa Java.

R//

1. Declaración del paquete
2. Importaciones
3. Declaración de la clase
4. Variables de clase
5. Constructores
6. Métodos
7. Métodos de clase
8. Variables de instancia

1. Describa la diferencia entre el formato CamelCase en mayúsculas y en minúsculas y proporcione un ejemplo de cuándo los usaría.

El formato CamelCase es una convención de nombrado utilizada en programación para nombrar variables, funciones, clases y otros elementos de código. La diferencia entre CamelCase en mayúsculas y en minúsculas es que en el primero, la primera letra de cada palabra es mayúscula, excepto la primera palabra, mientras que en el segundo, la primera letra de la primera palabra es minúscula y la primera letra de cada palabra siguiente es mayúscula. Se utiliza CamelCase en mayúsculas para nombrar clases y CamelCase en minúsculas para nombrar variables y funciones, dependiendo del contexto y la convención de nombrado del lenguaje de programación o del proyecto en el que se está trabajando.

lowerCamelCase variable = edadEstudiantes

UpperCamelCase = NombreEstudiante

1. ¿Qué sintaxis se utiliza para importar todo el paquete de utilidades de Java? Y si importase todo el paquete, ¿también necesitaría importar las clases adicionales en el mismo paquete por separado?

La sintaxis para importar todo el paquete de utilidades de Java es la siguiente: import java.util.\*;

Al importar todo el paquete de utilidades de Java con la sintaxis anterior, no es necesario importar las clases adicionales en el mismo paquete por separado. Sin embargo, si hay alguna clase adicional en el paquete que no se encuentra en java.util, es necesario importarla por separado.

1. Escriba la sintaxis de una clase de objeto Java simple con el nombre Student y el siguiente formato:

Student Name: Lisa Palombo

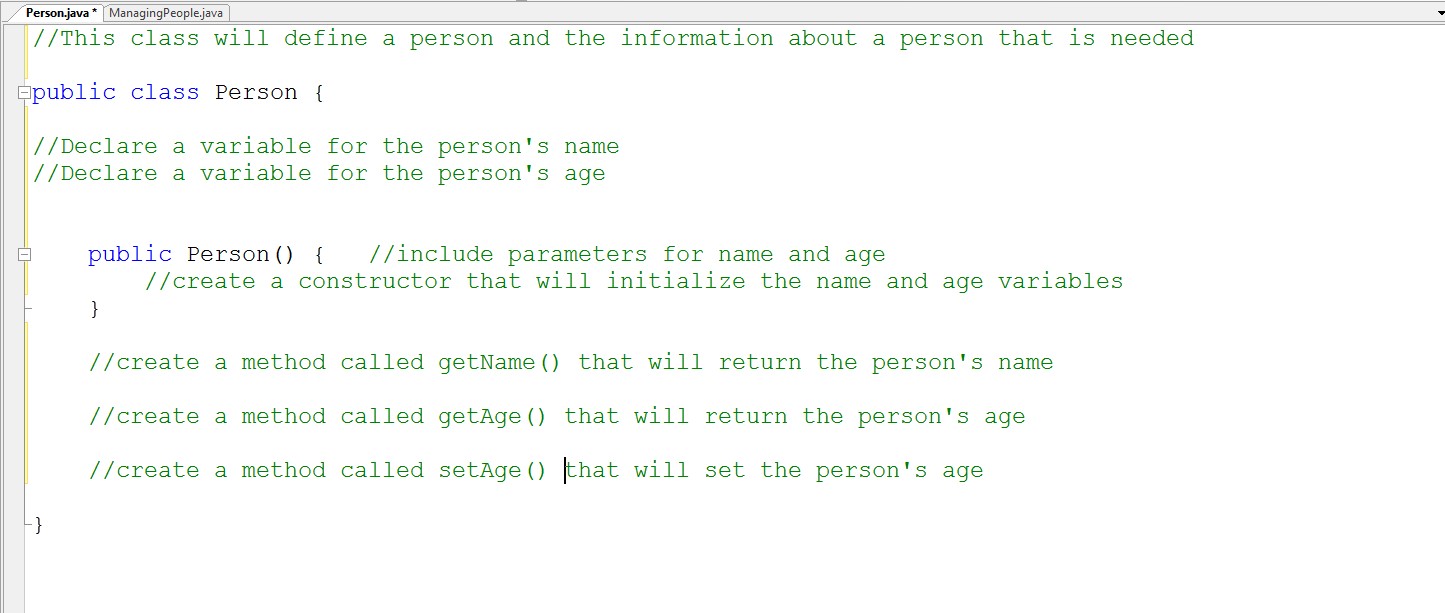
Student ID: 123456789 Student Status: Active

La información del estudiante se almacenará en las siguientes variables:

fName, lName, stuId, stuStatus.

1. Escriba el código para una clase de controlador que cree un objeto Student y muestre la información sobre el objeto en la pantalla.
2. Enumere 10 palabras clave de Java de esta lección.
3. class
4. public
5. private:
6. static
7. void
8. This
9. System
10. Main
11. PrintLn
12. Out

1. Complete la clase de objeto creada por el programador que aparece a continuación. Tenga en cuenta las instrucciones de los comentarios.



1. Complete la siguiente clase de controlador. Tenga en cuenta las instrucciones de los comentarios.

