

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

# 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. {} |
| CamelCase | Primera letra mayúscula y mayúscula inicial en la primera letra de cada palabra interna. Ejemplo: CuentaCorriente |
| Constante | Valor con nombre que no cambia. |
| camelCase | Primera letra minúscula y mayúscula inicial en la primera letra de cada palabra interna. Ejemplo: nombreEstudiante |
| Main Class | Clase que contiene un método principal. |
| Import | Sentencia de código en un archivo de clase Java que incluye código Java de otro paquete o clase. |
| Model Class | 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 reservada | Palabra que tiene una función especial en el lenguaje Java y no puede utilizarse como nombre de clase, método o variable. |
| API | 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. |
| Constructor | Tipo especial de método que es una plantilla para un objeto. |
| Parámetros | 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. |
| Modificadores de acceso | Palabras clave para especificar la accesibilidad de una clase (o tipo) y sus miembros. Por ejemplo: public, private, protected, default |
| Métodos | 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.
   1. Comentarios
   2. Declaraciones de paquetes (opcional)
   3. Declaraciones de importación
   4. Declaración de clase
   5. Variables de instancia
   6. Constructor
   7. Métodos de instancia
   8. Métodos estáticos
   9. Variables estáticas
   10. Método principal (main)
2. Describa la diferencia entre el formato CamelCase en mayúsculas y en minúsculas y proporcione un ejemplo de cuándo los usaría.

Este formato consiste en escribir palabras compuestas juntas sin espacios y con la primera letra de cada palabra en mayúscula. Hay dos variaciones del formato CamelCase. La primera letra es en mayúscula o minúscula, estas son:

* UpperCamelCase: También conocido como PascalCase, en este formato, la primera letra de cada palabra se escribe en mayúscula, incluyendo la primera palabra. Ejemplo: "PhoneNumber”, " AccountBalance ". Este formato se usa a menudo para nombrar clases o tipos de datos.
* lowerCamelCase: En este formato, la primera letra de la primera palabra se escribe en minúscula, pero la primera letra de las palabras subsiguientes se escribe en mayúscula. Ejemplo: "phoneNumber", "accountBalance". Este formato se usa a menudo para nombrar variables y métodos.

La diferencia entre ambos formatos es principalmente que UpperCamelCase se utiliza para nombrar clases y tipos de datos, mientras que lowerCamelCase se utiliza para nombrar variables y métodos.

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?

Para importar todo el paquete de utilidades de Java, se usa la siguiente sintaxis:

import java.util.\*;

Si se importa todo el paquete utilizando la sintaxis anterior, no es necesario importar las clases adicionales en el mismo paquete por separado, ya que todas las clases del paquete estarán disponibles para su uso.

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

Student Name: Lisa Palombo

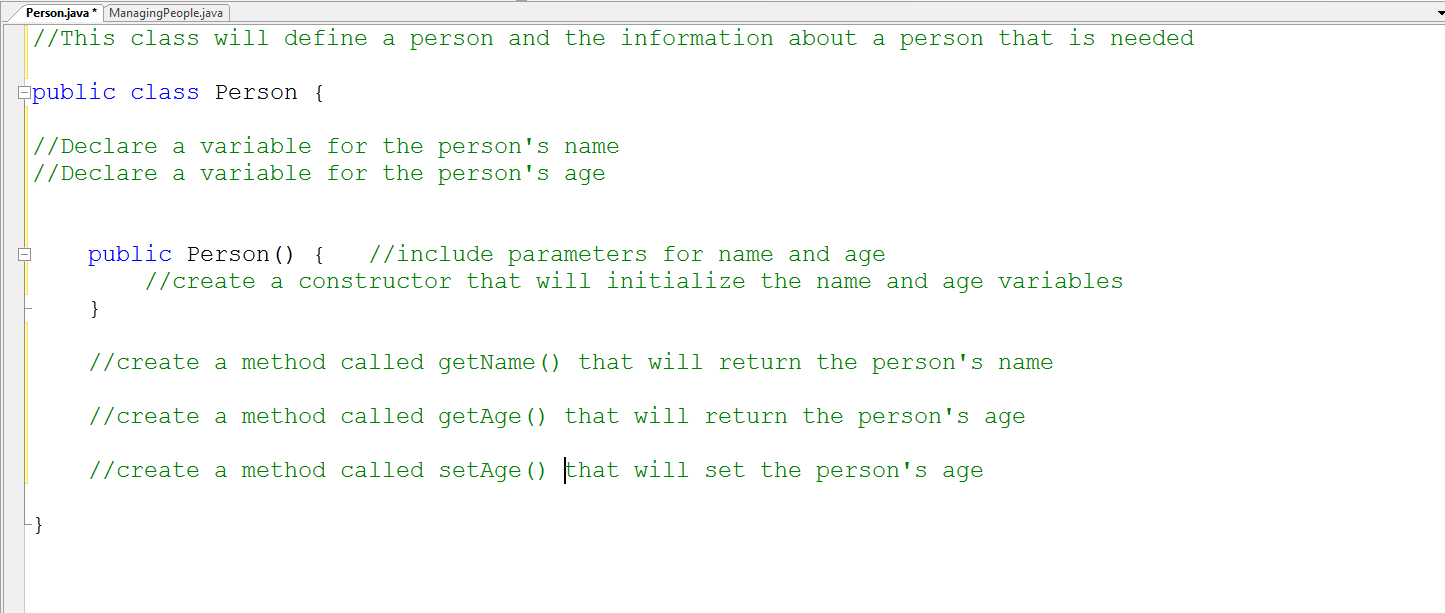
Student ID: 123456789

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.
   1. public
   2. class
   3. static
   4. void
   5. new
   6. private
   7. this
   8. String
   9. int
   10. return
3. 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.

