

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.

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

Inténtelo/resuélvalo:

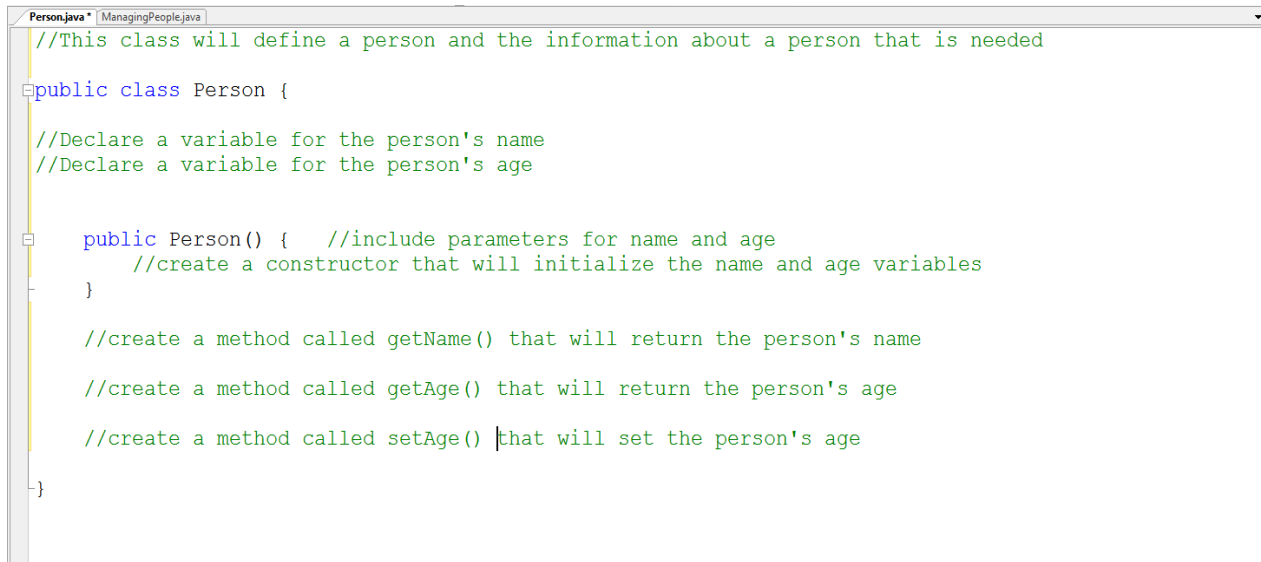
1. Enumere los componentes que forman un archivo .java. Enumere los componentes en el orden en que esperaría verlos en un programa Java.
2. Describa la diferencia entre el formato CamelCase en mayúsculas y en minúsculas y proporcione un ejemplo de cuándo los usaría.
3. ¿Qué sintaxis se utiliza para importar 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?

4. 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.

5. Escriba el código para una clase de controlador que cree un objeto Student y muestre la información sobre el objeto en la pantalla.
6. Enumere 10 palabras clave de Java de esta lección.
7. Complete la clase de objeto creada por el programador que aparece a continuación. Tenga en cuenta las instrucciones de los comentarios.



```
Person.java * | ManagingPeople.java
//This class will define a person and the information about a person that is needed

public class Person {

    //Declare a variable for the person's name
    //Declare a variable for the person's age

    public Person() {    //include parameters for name and age
        //create a constructor that will initialize the name and age variables
    }

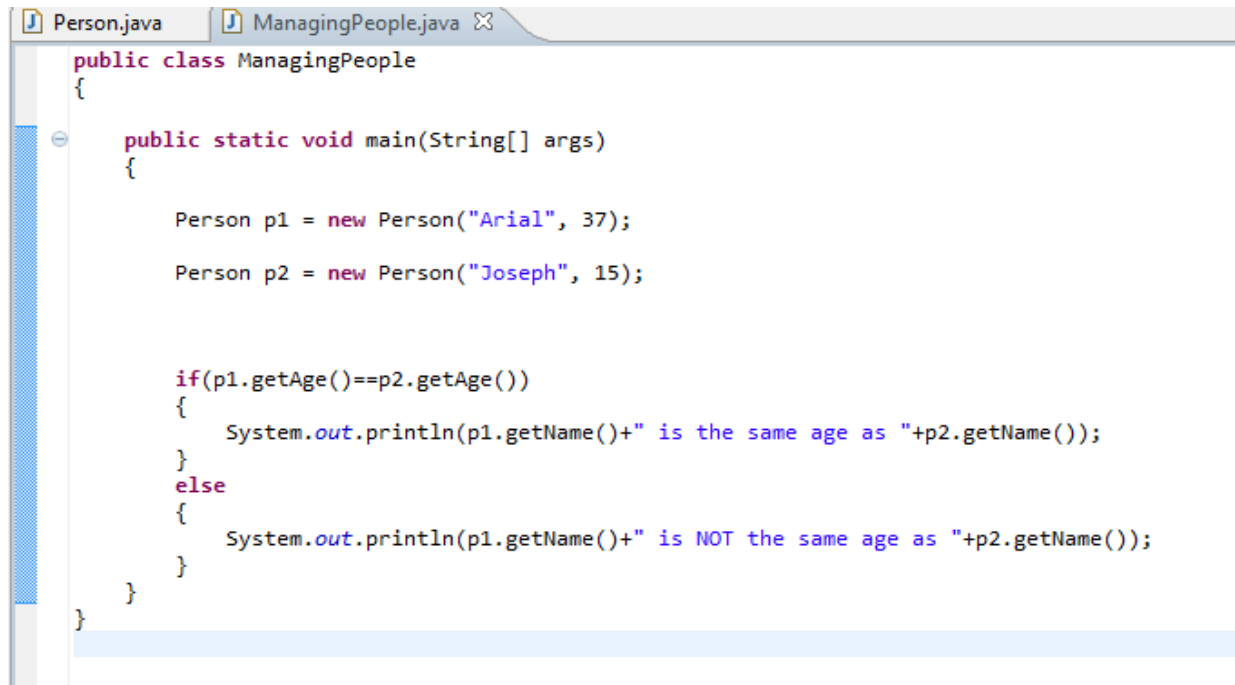
    //create a method called getName() that will return the person's name

    //create a method called getAge() that will return the person's age

    //create a method called setAge() that will set the person's age

}
```

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



```
Person.java ManagingPeople.java X
public class ManagingPeople
{
    public static void main(String[] args)
    {
        Person p1 = new Person("Arial", 37);
        Person p2 = new Person("Joseph", 15);

        if(p1.getAge()==p2.getAge())
        {
            System.out.println(p1.getName()+" is the same age as "+p2.getName());
        }
        else
        {
            System.out.println(p1.getName()+" is NOT the same age as "+p2.getName());
        }
    }
}
```