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

| **Clase** | **Plantilla utilizada para realizar objetos Java.** |
| --- | --- |
| **This** | Palabra clave opcional para tener acceso a los miembros y los métodos de una clase. |
| **Objeto** | Instancia de una clase. |
| **= new** | Operador utilizado para crear una instancia de una clase. |
| **Objeto de referencia** | Función incorporada en la máquina virtual Java que libera memoria como objetos que ya no son necesarios o que no se hacen referencia. |
| **Setter** | Método que cambia el estado de un objeto. |
| **Getter** | Método que devuelve información sobre un objeto al programa de llamada. |
| **Mutacion** | Procedimiento (cambia el estado de un objeto) o función (devuelve información sobre un objeto) encapsulado como parte de una clase. |
| **Creación de una instancia** | Verbo utilizado para describir el acto de crear un objeto de clase mediante la palabra clave new (nuevo). |
| **Inicialización** | Proceso de asignar un valor predeterminado a una variable. |
| **Null** | Referencia a un objeto que no se ha instanciado. |
| **Finalize** | Método opcional llamado justo antes de que el recolector de basura elimina un objeto. |
| **Referencia** | Nombre de una variable asociada con un objeto. |
| **Constructor** | Método especial utilizado para crear una instancia de una clase. |

1. Identifique las partes clave de la Clase Java a continuación. Coloque asteriscos junto a todas las variables de instancias. Coloque un casillero junto a cada constructor. Haga un círculo en la firma de los métodos diferentes al método de constructor. Coloque triángulos alrededor de los parámetros. Subraye los tipos de retornos de los métodos.

public class Animal {

int weight, height;

double speed;

Animal() {

weight = 50;

height = 4;

speed = 2; // miles per hour

}

Animal(int w, int h, int s) {

weight = w;

h = height;

speed = s;

}

public double getTime(double miles) { // gets the number of hours to go these miles return miles/speed;

}

public int getWeight() {

return weight;

}

public int getHeight() {

return height;

}

public double getSpeed() {

return speed;

}

}

solucion

public class Animal {

\* int weight, height;

\* double speed;

Animal() {

weight = 50;

height = 4;

speed = 2; // miles per hour

}

[ ]Animal(△int w△, △int h△, △int s△) {

weight = w;

h = height;

speed = s;

}

(o)public \_double\_ getTime(△double miles△) { // gets the number of hours to go these miles return miles/speed;

}

(o)public \_int\_ getWeight() {

return weight;

}

(o)public \_int\_ getHeight() {

return height;

}

(o)public \_double\_ getSpeed() {

return speed;

}

}

3 Escriba el código para crear dos instancias de la plantilla de clase Animal incluida en el problema número 2. Asegúrese de utilizar ambos constructores proporcionados. Luego, agregue un código Java que imprimirá lo siguiente: a. El animal 1 tiene una velocidad de \_\_\_. b. El animal 2 tiene una velocidad de \_\_\_.

**public** **class** PuntoCuatro {

**public** PuntoCuatro(){

Animal animal1 = **new** Animal();

Animal animal2 = **new** Animal();

System.***out***.println("El animal 1 tiene una velocidad de " + animal1.getSpeed());

System.***out***.println("El animal 2 tiene una velocidad de " + animal2.getSpeed());

}

}