



Boujon Ignacio
Boujon Sebastián
Lavallen Lucas
Pereyra C. David
Scheffler Agustin

Facultad de Ciencia y Tecnología
Universidad Autónoma de Entre Ríos
Trabajo Practico 3
Programación Orientada a Objetos

Tournour Alberto Adrián
Pérez Alejandro

26 /Abril / 2024

1). De las siguientes listas de objetos indique que tienen en común y escriba un concepto que las defina. Dentro de cada colección forme subconjuntos variando el concepto.

a) Bicicleta, barco, coche, camión, aeroplano, planeador, motocicleta, caballo.

b) Diario, canal de TV, revista, radio AM, radio FM, semanario.

Para identificar el concepto que tienen en común estas listas de objetos y formar subconjuntos, primero necesitamos encontrar una característica que comparten los elementos de cada lista. Luego, podemos agruparlos en subconjuntos según esa característica común. Veamos:

a) Bicicleta, barco, coche, camión, aeroplano, planeador, motocicleta, caballo.

Concepto: Medios de transporte.

Subconjuntos:

- Vehículos terrestres: Bicicleta, coche, camión, motocicleta, caballo.

- Vehículos acuáticos: Barco.

- Vehículos aéreos: Aeroplano, planeador.

b) Diario, canal de TV, revista, radio AM, radio FM, semanario.

Concepto: Medios de comunicación.

Subconjuntos:

- Medios impresos: Diario, revista, semanario.

- Medios electrónicos: Canal de TV, radio AM, radio FM.

En ambos casos, hemos identificado un concepto común que engloba los objetos de la lista y luego hemos formado subconjuntos basados en esa característica común. ¿Hay algo más con lo que necesites ayuda?

2). Sobre la siguiente figura de un equipo de audio defina los atributos que representen su estado, y su comportamiento.

1. Botones: Pueden representarse como atributos para controlar las funciones del equipo, como reproducción, pausa, adelantar, retroceder, etc.
2. Radio FM y AM: Estos podrían ser atributos que indican si el equipo tiene capacidades de sintonización de radio FM y AM.
3. Volumen: Un atributo que almacena el nivel de volumen actual del equipo.
4. Dial: Podría ser un atributo que almacena la posición del dial para ajustar la sintonía de la radio.
5. Parlante: Este podría ser un atributo que almacena la información sobre el sistema de altavoces del equipo, como el número de parlantes, su ubicación, potencia, etc.

Cada instancia de la clase "EquipoDeAudio" tendría sus propios valores para estos atributos, lo que representaría el estado específico de cada equipo en un momento dado. Estos atributos permiten modelar y manipular las características y el comportamiento del equipo de audio en el contexto de un programa orientado a objetos.

3).

a) Factores importantes al decidir comprar una rueda para el auto:

- Tamaño: Debe ser compatible con el tamaño del automóvil y los otros neumáticos.
- Costo: Debe ajustarse al presupuesto del comprador.
- Duración esperada: La vida útil de la llanta es importante para la relación costo-beneficio.

b) Factores importantes para alguien que simula el rendimiento de un sistema antideslizamiento por computadora para coches:

- Material: Debe tener en cuenta la resistencia al deslizamiento en diferentes condiciones de superficie.
- Estructura interna: La estructura de la llanta puede afectar cómo interactúa con el sistema antideslizamiento.

- Dibujo: La forma y el diseño del dibujo pueden influir en la capacidad de agarre y tracción.
- Tamaño y peso: Deben ser compatibles con el diseño general del vehículo y no agregar una carga excesiva.

c) Factores importantes para alguien que está construyendo una hamaca para un niño:

- Material: Debe ser suave, seguro y duradero para la piel sensible de un niño.
- Costo: Debe ajustarse al presupuesto del comprador.

4. Identifique clases y objetos. La universidad desea almacenar la información de todos los profesores, las materias de cada profesor, y los alumnos que cursan cada materia.

Para modelar la información de los profesores, las materias que enseñan y los alumnos que cursan cada materia, podemos identificar las siguientes clases y objetos:

1. Clase: Profesor

- Objetos:
 - Profesor 1 (por ejemplo, Juan Pérez)
 - Profesor 2 (por ejemplo, María Gómez)
 - ...

2. Clase: Materia

- Objetos:
 - Materia 1 (por ejemplo, Matemáticas)
 - Materia 2 (por ejemplo, Historia)
 - ...

3. Clase: Alumno

- Objetos:
 - Alumno 1 (por ejemplo, Carlos Rodríguez)
 - Alumno 2 (por ejemplo, Ana López)
 - ...

Ahora, para establecer las relaciones entre estas clases, podemos utilizar asociaciones.
Por ejemplo:

- Cada Profesor puede enseñar varias Materias, y cada Materia puede ser enseñada por varios Profesores. Esta es una relación de "muchos a muchos".

- Cada Materia puede tener varios Alumnos matriculados, pero cada Alumno puede estar matriculado en varias Materias. Esta también es una relación de "muchos a muchos".

Entonces, podríamos tener algo así:

4. Relación: Enseñanza

- Objetos:
 - Enseñanza 1 (por ejemplo, Juan Pérez enseña Matemáticas)
 - Enseñanza 2 (por ejemplo, María Gómez enseña Historia)
 - ...

5. Relación: Matriculación

- Objetos:
 - Matriculación 1 (por ejemplo, Carlos Rodríguez se matricula en Matemáticas)
 - Matriculación 2 (por ejemplo, Ana López se matricula en Historia)
 - ...

Cada objeto de estas relaciones puede contener más detalles, como las horas de clase, la sala, las calificaciones, etc., dependiendo de los requisitos específicos del sistema que se esté desarrollando.

5) Atributos.

- 1) Base: Define base del rectángulo.
- 2) Altura: Define la altura del rectángulo.
- 3) Área: el Comportamiento define la base por altura.
- 4) Color_F: Define color del fondo.
- 5) Color_L: Define color de lado.

6)

Un objeto es una entidad dentro de un lenguaje de programación que contiene al mismo tiempo un estado (dado por los valores que toman sus atributos) y un comportamiento (dado por una serie de métodos que permiten comunicarse con dicho objeto). (Universidad de la rioja) Una instancia es un entorno de gestor de bases de datos lógicas donde se catalogan las bases de datos y se establecen los parámetros de configuración. (IBM).

7)

Si, el estado de un objeto puede cambiar a medida que se realizan operaciones sobre él, el estado de un objeto representa las características de objeto en un momento dado, si nuestro objeto es un sartén, su estado será el material, la marca, si tiene mango, si tiene tapa, etc.

- ¿Qué etapas tiene la vida de una instancia de una clase?

La vida de una instancia de una clase en un programa típicamente sigue un ciclo que incluye varias etapas. Aquí hay una descripción general de estas etapas:

Creación (Instanciación): En esta etapa, se crea una nueva instancia de la clase utilizando el operador `new` u otro método de creación proporcionado por el lenguaje de programación. Durante esta etapa, se asigna memoria para almacenar los datos de la instancia y se inicializan los atributos con los valores predeterminados o los especificados en el constructor de la clase.

Inicialización: Después de la creación, la instancia puede pasar por una fase de inicialización donde se establecen los valores iniciales de sus atributos. Esto puede ocurrir en el constructor de la clase o mediante métodos específicos de inicialización.

Utilización: Durante esta etapa, la instancia se utiliza en el programa para realizar diversas operaciones. Esto puede implicar la invocación de métodos de la instancia, la modificación de sus atributos o la interacción con otras partes del programa.

Modificación: En algunos casos, una instancia puede cambiar su estado a lo largo del tiempo mediante la modificación de sus atributos. Esto puede suceder a través de métodos que actualizan los valores de los atributos o en respuesta a eventos específicos del programa.

Finalización (Destrucción): Cuando una instancia ya no es necesaria, generalmente se marca para su eliminación. En lenguajes de programación que emplean un recolector de basura, como Java o Python, la instancia será eliminada automáticamente cuando ya no haya referencias que la apunten. En otros lenguajes, como C++, es responsabilidad del programador liberar explícitamente la memoria ocupada por la instancia mediante la llamada al destructor o a métodos de liberación de memoria.

9)

Los constructores son métodos especiales dentro de una clase que se utilizan para inicializar objetos de esa clase. Tienen varias funciones importantes:

Inicialización de atributos: Los constructores permiten establecer valores iniciales para los atributos de un objeto cuando se crea una nueva instancia de la clase. Esto garantiza que el objeto tenga un estado coherente desde el principio.

Asignación de recursos: En algunas ocasiones, es necesario asignar recursos externos (como memoria, conexiones de red, archivos, etc.) cuando se crea un objeto. Los constructores pueden encargarse de esta asignación de recursos y liberarlos adecuadamente cuando el objeto es destruido.

Validación de datos: Los constructores pueden incluir lógica para validar los datos proporcionados al crear un objeto. Esto puede ayudar a garantizar que el objeto se inicialice correctamente y que los datos proporcionados cumplan con ciertos criterios.

- Describe en términos de objetos un número, Defina características y comportamiento

Características: signo (positivo, negativo), tipo (entero, decimal, real), valor (el número en sí)

Comportamiento: operaciones aritméticas, comparación, mayor que, menor que, factorial, etc.

- Describe en términos de objetos un empleado, Defina características y comportamiento

Características: altura, peso, color de pelo, sexo, domicilio, etc

Comportamiento: horas que trabaja, tareas que realiza, días a la semana que trabaja, horas extra, etc.

- Describe en términos de objetos la fórmula de velocidad ($v = d / t$, velocidades distancia sobre tiempo), Defina características y comportamiento

10)

1. **Valor:** El valor del número, que puede ser un entero, decimal, fracción, etc.
2. **Tipo:** El tipo de número, como entero, flotante, complejo, etc.
3. **Precisión:** La cantidad de dígitos significativos o decimales que tiene el número (en el caso de números decimales).
4. **Signo:** Indica si el número es positivo, negativo o neutro.
5. **Base:** En el caso de números no decimales, la base en la que está representado el número (por ejemplo, binario, octal, hexadecimal).

Comportamiento.

1. **Suma, resta, multiplicación, división:** Capacidad para realizar operaciones aritméticas básicas con otros números.
2. **Comparación:** Posibilidad de comparar el número con otros números para determinar si es mayor, menor o igual.
3. **Conversión de tipos:** Capacidad para convertir el número a otros tipos, como entero, flotante, cadena, etc.
4. **Redondeo:** Función para redondear el número a una cierta cantidad de decimales o cifras significativas.
5. **Formato de salida:** Posibilidad de representar el número en diferentes formatos de salida, como notación científica, formato decimal, etc.

11)

Nombre: El nombre del empleado.

Edad: La edad del empleado.

ID: Un identificador único para el empleado, como un número de identificación de empleado.

Cargo: El cargo que ocupa en la empresa, como gerente, desarrollador, diseñador, etc.

Departamento: El departamento al que pertenece el empleado dentro de la empresa.

Salario: El salario del empleado.

Comportamientos.

Calcular **salario**: Capacidad para calcular el salario del empleado, que puede depender de factores como horas trabajadas, bonificaciones, deducciones, etc.

Actualizar **información**: Posibilidad de actualizar la información del empleado, como el salario, la posición, el departamento, etc.

Calcular **antigüedad**: Función para calcular la antigüedad del empleado en la empresa, basada en la fecha de contratación.

Registrar **horas trabajadas**: Capacidad para registrar las horas trabajadas por el empleado.

12) Características: velocidad (V), distancia (D), tiempo (T).

Comportamiento: calcular velocidad del objeto, calcular tiempo que tarda en llegar el objeto, calcular distancia que recorre el objeto, validación (que el tiempo sea distinto de 0), formato de salida (que agregue el resultado más m/s si se calcula velocidad), etc.