**UNIVERSIDAD TECNICA PARTICULAR DE LOJA**

**Nombre**: Andres Hinojosa

**Materia**: Programacion Orientada a Objetos

**Fecha**: 30/07/2024

1. **La implementación de clases, objetos, constructor**

**Clases**

Clases:

* Una clase en Java es como un plano o prototipo que define cómo deben ser los objetos que se creen a partir de ella.
* Define las propiedades (características) y los métodos (comportamientos) que estos objetos tendrán.
* En nuestro ejemplo de la fábrica de galletas, la clase sería uno de los moldes para galletas, junto con los métodos para colorearlas y la receta (método para prepararlas).
* En Java, primero debemos crear una clase antes de poder crear instancias u objetos de esa clase.

Objetos:

* Un **objeto** en Java es básicamente una instancia de una clase.
* Siguiendo con el ejemplo de la fábrica de galletas, los objetos serían cada una de las diferentes galletas obtenidas de los moldes definidos (clases).
* Los objetos se crean utilizando el comando *new*, que nos permite crear un nuevo objeto de una clase específica y enviarle los parámetros según el constructor.
* Por ejemplo, si tenemos una clase llamada Galleta, podemos crear un objeto de esa clase así: *Galleta miGalleta = new Galleta();.*

Constructores:

* Un **constructor** es un método especial dentro de una clase que se utiliza para crear e inicializar un objeto de esa clase.
* Se invoca automáticamente cuando se crea un nuevo objeto a partir de una clase.
* El constructor define cómo se construirá el objeto y cómo se inicializarán sus atributos.
* En el ejemplo de la fábrica de galletas, el constructor sería el proceso mediante el cual se crea una galleta específica a partir del molde y la receta definidos en la clase.

**Uso de estructuras de datos en lenguaje de programación Python (list y tuple)**

1. **Listas en Python:**

* Las **listas** son colecciones **ordenadas** y **mutables** de elementos.
* Se definen utilizando corchetes [ ] y pueden contener elementos de diferentes tipos (números, cadenas, objetos, etc.).
* Algunos métodos útiles para trabajar con listas son:
  + - append(x): Agrega un elemento al final de la lista.
    - extend(iterable): Extiende la lista agregando todos los elementos del iterable.
    - insert(i, x): Inserta un elemento en una posición dada.
    - remove(x): Elimina el primer elemento de la lista con el valor x.
    - pop([i]): Elimina y devuelve el elemento en la posición especificada (si no se especifica, elimina el último elemento).
    - clear(): Elimina todos los elementos de la lista.
    - index(x[, start[, end]]): Devuelve el índice del primer elemento con el valor x.
    - count(x): Cuenta cuántas veces aparece x en la lista.
    - sort(): Ordena los elementos de la lista.
    - reverse(): Invierte los elementos de la lista.

Ejemplo:

**Python**

*fruits = ['orange', 'apple', 'pear', 'banana', 'kiwi', 'apple', 'banana']*

*fruits.count('apple') # 2*

*fruits.index('banana') # 3*

*fruits.reverse()*

*fruits.append('grape')*

*fruits.sort()*

Código generado por IA. Revisar y usar cuidadosamente. [Más información sobre preguntas frecuentes](https://www.bing.com/new#faq).

1. **Tuplas en Python:**
   * Las **tuplas** son similares a las listas, pero son **inmutables**, lo que significa que no se pueden modificar una vez creadas.
   * Se definen utilizando paréntesis ().
   * Aunque no tienen tantos métodos como las listas, son útiles cuando necesitas almacenar datos que no deben cambiar.
   * Ejemplo:

*coordinates = (10, 20)*