Costa Rica Institute of Technology

Computer Engineering

tarea 1

Lee Sang Cheol

19 de febrero de 2025

1. ¿Qué es un Puntero y una Referencia?

1.1. Puntero

Imagina que tienes un mapa del tesoro. En el mapa hay una "X" que indica dónde está el tesoro. Pero la "X" no es el tesoro, solo te dice dónde encontrarlo.

En programación, el puntero funciona igual. No guarda el objeto en sí, sino la dirección de dónde está en la memoria.

Ejemplo: - Supongamos que tienes una caja llamada `CajaDeJuguetes`. Dentro de la caja no hay juguetes, sino una nota que dice "Busca en el estante número 3". - Cuando quieres el juguete, sigues la nota hasta el estante 3 y lo encuentras.

En código C, esto se vería así:

```
int *puntero;  // Declaras un puntero
int juguete = 5;
puntero = &juguete;  // El puntero guarda la dirección de '
    juguete'
```

Aquí, el puntero no guarda el número 5, sino la ubicación de dónde está guardado.

1.2. Referencia

Imaginamos que tienes un peluche de nombre "kimbab", pero a veces también lo llamas "tteok". Ambos nombres se refieren al mismo peluche, no hay dos peluches, solo uno con dos nombres.

En programación, una referencia es como un segundo nombre para una variable u objeto.

```
Peluche kimbab = new Peluche();
Peluche tteok = kimbab;
```

Tanto `kimbab` como `tteok` se refieren al mismo peluche.

2. Palabra Reservada `new`

Imaginamos que tiene una tienda de juguetes mágica. Cada vez que dices la palabra mágica `new`, aparece un nuevo peluche en su cuarto.

Por ejemplo, si quiere un peluche de **mandu**, solo tiene que decir:

```
Peluche mandu = new Peluche();
```

¡Y pum! Ahora tienes un nuevo peluche llamado `mandu` en su cuarto.

Pero, ¿qué pasa si dice la palabra mágica otra vez?

```
Peluche otroPeluche = new Peluche();
```

Ahora tiene dos peluches. ¡No es el mismo dos veces, son diferentes! Cada vez que usas `new`, creas un peluche completamente nuevo que ocupa su propio espacio en la memoria. Es como si en la tienda mágica cada pedido fuera único y diferente.

Así es como funciona `new` en la programación: crea nuevos objetos en la memoria, igual que la tienda de juguetes mágica

3. ¿Qué es la Instancia?

La instancia es el objeto específico creado a partir de la clase.

supongamos que tiene un molde para hacer galletas con forma de estrella. Cada vez que usa el molde, hace la galleta. Cada galleta es una instancia diferente del mismo molde.

En código Java, esto se vería así:

```
Galleta chocoGalleta = new Galleta("Chocolate");
Galleta kimchiGalleta = new Galleta("Kimchi");
```

Aquí, `chocoGalleta` y `kimchiGalleta` son dos galletas diferentes hechas con el mismo molde.

4. Qué es una Clase, Método, Atributo, Constructor, Clase de Auto-Referencia y Nodo

4.1. Clase

La clase es como el plano para construir casas. Define cómo serán los objetos, pero no es el objeto en sí.

```
class Casa {
    int habitaciones = 3;
    int banos = 2;
}
```

4.2. Método

El método es como una acción que un objeto puede hacer. Si tiene un robot de juguete, encenderlo o apagarlo sería ese método.

```
robot.encender();
robot.apagar();
```

4.3. Atributo

el atributo es la característica del objeto. Por ejemplo, un coche puede tener color, modelo y velocidad.

```
class Coche {
    String color;
    String modelo;
    int velocidad;
}
```

4.4. Constructor

Un constructor es como una receta para crear el objeto.

```
class Coche {
    Coche(String color) {
        this.color = color;
    }
}
```

4.5. Clase de Auto-Referencia

Es la clase que se refiere a sí misma. Imaginamos un tren donde cada vagón está conectado al siguiente.

```
class Vagon {
    Vagon siguiente;
}
```

4.6. Nodo

el nodo es como un punto en una cadena. supongamos una cadena de papel donde cada eslabón está conectado al siguiente.

```
class Nodo {
    int valor;
    Nodo siguiente;
}
```

5. Citas y Bibliografía

5.1. Citas

- "Los punteros permiten una manipulación eficiente de la memoria en C y C++"[1].
- .En Java, los objetos se manejan mediante referencias, no punteros"[2].

5.2. Bibliografía

Referencias

[1] Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley.

[2] Deitel, P., Deitel, H. (2017). *Java: How to Program*. Pearson.

[3] Knuth, D. (1997). *The Art of Computer Programming*. Addison-Wesley.

[4] Kernighan, B. W., Ritchie, D. M. (1988). *The C Programming Language*. Prentice Hall.

[5] *Documentación oficial de Java*. Disponible en: `https://docs.oracle.com/en/java/`