

# Introducción a Ciencias de la Computación 2020-2

## Práctica 6: Herencia

Pedro Ulises Cervantes González  
confundeme@ciencias.unam.mx

Emmanuel Cruz Hernández  
emmanuel\_cruzh@ciencias.unam.mx

Adriana Sánchez del Moral  
adrisanchez@ciencias.unam.mx

Víctor Zamora Gutiérrez  
agua@ciencias.unam.mx

Fecha límite de entrega: 24 de Mayo de 2020.  
Hora límite de entrega: 23:59.

### 1. Objetivo

Explotar las aplicaciones que tiene La herencia en código, tal como reutilizar código de manera inteligente a partir de la extensión de clases.

Conocer el comportamiento que toman los atributos y métodos de una clase padre sobre una clase hija. Además de explotar las cualidades del polimorfismo.

### 2. Actividad

Considera las clases *ZonaConstruccion*, *Construccion*, *Casa* y *Trabajador*.

#### 2.1. Actividad 1 (*1 punto*)

Cumple los siguientes puntos en la clase *Casa*.

1. Crea un método constructor para la clase *Casa*.
2. El precio de las casas es de \$500,000. Si tiene al lado un terreno baldío, su precio disminuye 10 %, pero si está entre dos terrenos baldíos, disminuye 25 %. Si no hay terrenos baldíos en la zona donde se encuentra la casa, aumenta 15 % su valor.

## 2.2. Actividad 2 (2 puntos)

Crea una clase *Edificio* que represente un edificio y extienda a la clase *Casa*.

1. Los edificios deben tener al menos 2 plantas (planta baja y primer piso). El precio de los edificios es de \$600,000, agregando \$200,000 por cada piso extra.
2. Los edificios modifican su valor del mismo modo que las casas por los terrenos baldíos, así que, en lugar de repetir código, reutiliza usando herencia algún método de la clase *Casa*.

## 2.3. Actividad 3 (2 puntos)

Crea una clase *Constructor* que extienda la clase *Trabajador*.

Los constructores crean una *Casa* o un *Edificio* en un terreno baldío de la zona donde trabajan. Cobra \$100,000 por cada planta que construye. Se considera que las casas sólo consisten de planta baja.

## 2.4. Actividad 4 (2 puntos)

Crea una clase *Demoledor* que extienda la clase *Trabajador*.

Los demoledores destruyen una *Casa* o un *Edificio* de la zona donde trabajan, dejando finalmente un terreno baldío. Cobran \$50,000 por cada planta destruida.

## 2.5. Actividad 5 (2 puntos)

Crea una clase *Remodelador* que extienda la clase *Trabajador*.

Los remodeladores solo trabajan en edificios de su zona. Pueden agregar o quitar pisos. Al quitar pisos, debe dejar al menos dos plantas en el edificio (un edificio posee al menos dos). Cobran \$120,000 por agregar plantas y \$60,000 por quitarlas.

## 2.6. Actividad 6 (1 punto)

Crea un método *main* en la clase *ZonaConstruccion* con una zona de construcción con al menos 5 terrenos baldíos, un constructor, un demoledor y un remodelador.

Tienes 2 opciones:

1. Que los trabajadores laboren en un orden aleatorio y realicen en conjunto 16 trabajos. Después de cada 4 trabajos debes mostrar cuál es el valor actual de cada construcción.
2. Hacer un menú en que brindes las opciones de elegir quién va a trabajar y mostrar el valor actual de cada construcción. El menú solo debe mostrar a los trabajadores que pueden laborar.

Después de que alguien trabaja, debes imprimir en pantalla quién trabajó, qué hizo y cuánto cobró. Deberás mostrar además como luce la zona de construcción actualmente.

### 3. Nota importante

Esta práctica se puede entregar en parejas. El formato de entrega es el siguiente:

- Apellido1Nombre1Apellido2Nombre2
  - src
    - Casa.java
    - Construccion.java
    - Trabajador.java
    - ZonaConstruccion.java
    - ...
  - Readme.txt

En caso de realizar la práctica en parejas, uno de los integrantes debe enviar un correo a `emmanuel_cruzh@ciencias.unam.mx` a más tardar un día antes de la fecha de entrega de la práctica, mencionando el nombre de ambos integrantes.

El archivo *Readme.txt* debe contener una breve descripción de qué hizo cada integrante y cuál fue la forma en que se organizaron para realizar la práctica. **Ambos integrantes deben contribuir en la implementación de la práctica.**

Sólo una persona debe subir y entregar la práctica en su versión final a Classroom, el otro integrante sólo debe marcar la práctica como entregada.

### 4. Créditos

Clases creadas por Lic. en Ciencias Computacionales Pedro Ulises Cervantes González para el curso de Introducción a las Ciencias de la Computación 2017-2.

### 5. Materiales para consultar

1. Ejemplo de herencia: <https://youtu.be/wqoyQ3BxK4A>
2. Ejemplo de herencia II: <https://youtu.be/rE0FpdI3HY0>
3. Polimorfismo: <https://youtu.be/BSw1MLEc4PQ>
4. Herencia con Minecraft: <https://youtu.be/yh8bTKqC0tU>
5. Polimorfismo con Minecraft: <https://youtu.be/bb1FTvuk4pY>

## 6. Reglas Importantes

- No se recibirán prácticas en las que estén involucrados más de dos integrantes.
- Cumple con los lineamientos de entrega.
- Todos los archivos deberán contener nombre y número de cuenta.
- Tu código debe estar comentado. Esto abarca clases, atributos, métodos y comentarios extra.
- Para cada clase solicitada, crea un nuevo archivo.
- Utiliza correctamente las convenciones para nombrar variables, constantes, clases y métodos.
- Sólo se permite el uso de las bibliotecas Scanner y Random.
- **SÍ** se reciben prácticas con retraso. Por cada día se restará 1 punto.
- En caso de no cumplirse alguna de las reglas especificadas, se restará 0.5 puntos en tu calificación obtenida.