Sesión 8

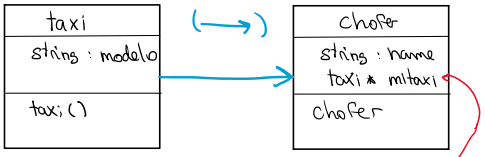
sábado, 3 de junio de 2023 14:02

Relaciones entre clases

Asociación

- Dos clases se relacionan literalmente
- Conviven separados
- "USA UN"
- (Taxi - Chofer)

UML (Unified Model Language)




```
class taxi {
private:
    String modelo;
public:
    Taxi();
};
```

usa un

```
class chofer {
    string name
    taxi * mitaxi;
    chofer(Taxi& ){}{
    }
```

existe ⟷ existe

Contextos diferentes

Relación débil

Composición

- Una objeto se crea dentro de otro objeto
- El constructor del objeto relacionado se define en el constructor del objeto mayor

"ES PARTE DE"

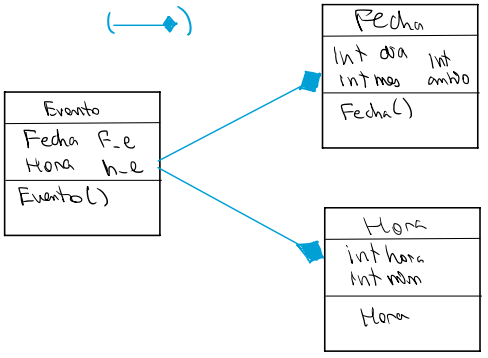
```
class Evento {
    obj Fecha  f_evento
        Hora   h_evento
};

class Fecha {
    int dia, mes, anhio;
{

class Hora {
    int hora, minutos;
{
```

```
Evento(dia, mes, anhio, hora, minutos) {
    f_evento = Fecha(dia, mes, anhio);
    h_evento = Hora(hora, minutos);
{
```

Relación Fuerte



Agregación

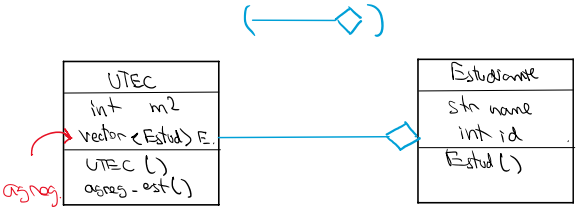
- Vamos objetos dentro de otro objeto
  - un contenedor.
- "Tiene Un"

```
class UTEC {
    int mes2;
    vector<Estudiante> Estudiantes;
UTEC(int m2){}{
void agregar_estudiante(Estudiante& E){
    Estudiantes.push_back(E);
{
```

múltiples asociadores

agregando

```
class Estudiante {
    str name,
    int id,
Est(str, int){}{
```



Herencia

- Crea una Jerarquía
- Super clase (padre) — Sub clases (hijas)
  - GENERALIZA
  - ESPECIALIZA

hija

padre

```
class Figura2D{
protected:
    int lados = 4;
    int vertices = 4;
public:
    Figura2D(){};
{
```

Solo las hijas tienen acceso

hija

padre

```
class Cuadrado : public Figura2D{
```

hereda: lados = 4, vertices = 4

```
    str color;
public:
    Triangulo(int l, int v, str c){}{
{
```