

Casillas

- Clase **padre** *Casilla*: funcionalidad común
 - atributo *nombre*
 - métodos *informe()*, *recibeJugador()*, *getNombre()* y *toString()*
- Clases **hijas**: cada una con su funcionalidad propia y **redefiniendo** *recibeJugador()* y *toString()*
 - *CasillaCalle*
 - *CasillaSorpresa*
- Casting en Java en *CivitasJuego::comprar()*

Sorpresas

- Clase **padre** *Sorpresa*: **Abstracta**
 - cabecera: *public **abstract** class Sorpresa { ... }*
 - atributos *texto* y *valor*
 - método **abstracto** *aplicarAJugador()*:
abstract void aplicarAJugador(int iactual, ArrayList<Jugador> todos);
 - métodos *informe()* y *toString()*
- Clases **hijas**: todas **implementan** *aplicarAJugador()*
 - *SorpresaPagarCobrar*
 - *SorpresaPorCasaHotel*

JugadorEspeculador

- Clase **padre** *Jugador*:
 - nuevo método: ***protected Jugador convertir() { ... }***
- Clase **hija**: *JugadorEspeculador*
 - los atributos se construyen utilizando constructor **copia** del padre
 - **redefine** los métodos: *paga()*, *getCasasMax()*, *getHotelesMax()* y *toString()*

JugadorEspeculador

- Nueva clase ***SorpresaConvertirme***
 - Hija de *Sorpresa*
 - implementar *aplicarAJugador(int iactual, ArrayList<Jugador> todos)*:
 - *ArrayList::set(pos, E)*: reemplaza el elemento del arraylist en la posición *pos* por el elemento *E*
- ***TestP4***: probar desde la clase *Jugador*, ya que el constructor tiene visibilidad de paquete