



Objetos parte 2

Clase #24

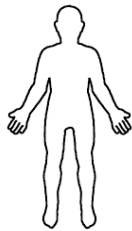
IIC1103 – Introducción a la Programación

El plan de hoy es...

- Seguir con OOP (object-oriented programming)
- Próx semana: haremos repaso (ambas clases)

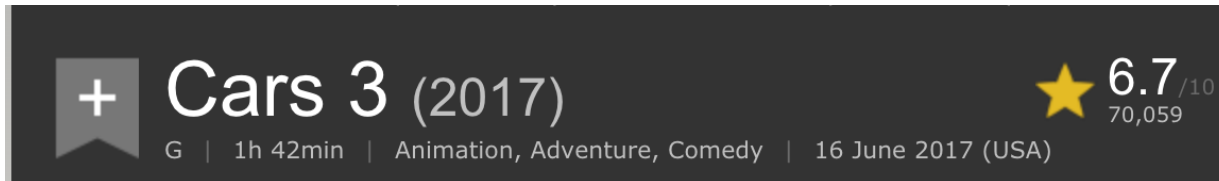
Resumen de última clase

- Tenemos los siguientes tipos:
 - int → representa un número entero
 - float → representa un número real
 - bool → representa un valor de verdad (True/False)
 - str → representa un string (texto)
 - list → representa una lista de elementos
- Quiero definir un tipo propio para representar una Persona



nombre → str
apellido → str
edad → int
rut → int

class Pelicula (menti)



Objeto Pelicula
(o: Instancia de la clase Pelicula)



Objeto Pelicula
(o: Instancia de la clase Pelicula)

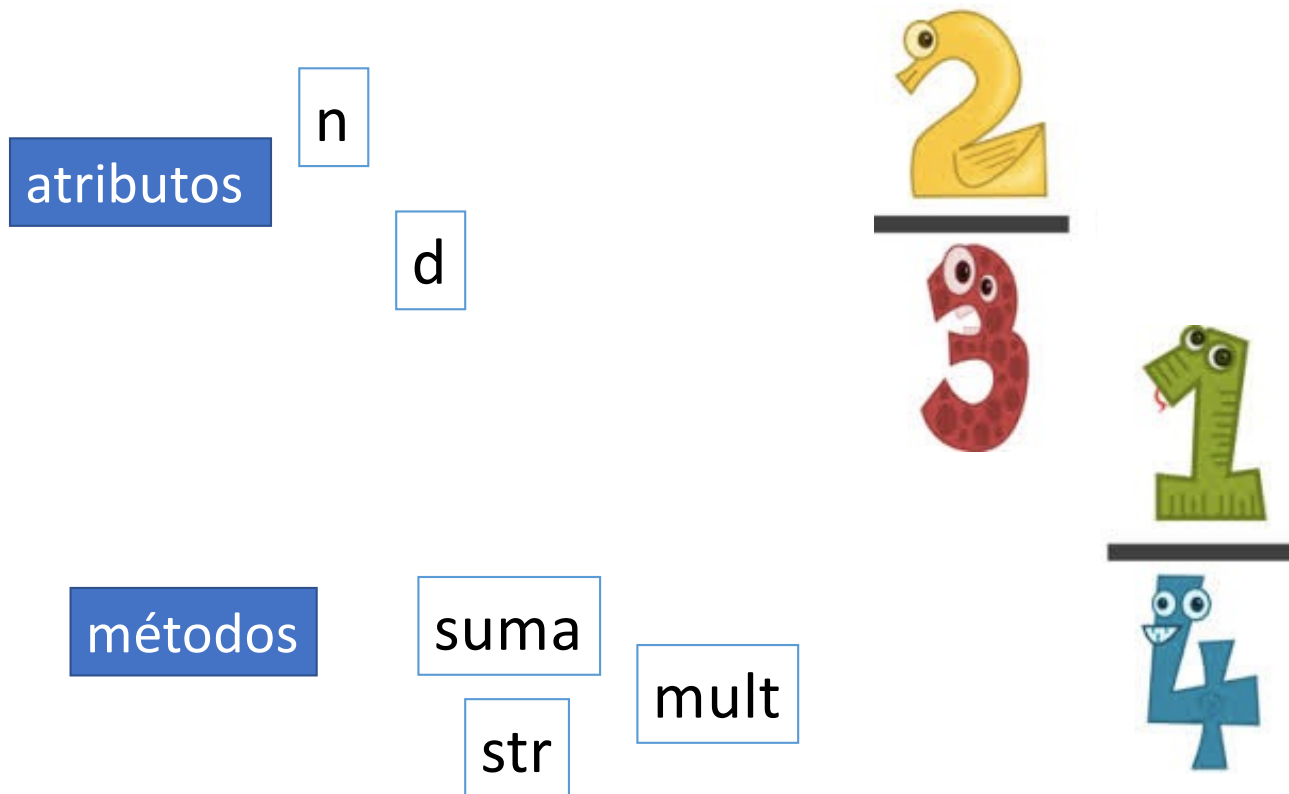
Resumen de última clase

- `class <Nombre de Clase>:`
- `def __init__(self, parámetros):`
- `self.atributo1 = ...`
- `def método(self,...):`
- `...`
- `def __str__(self):`
- `...`

Problema #1: Definir un tipo Fracción

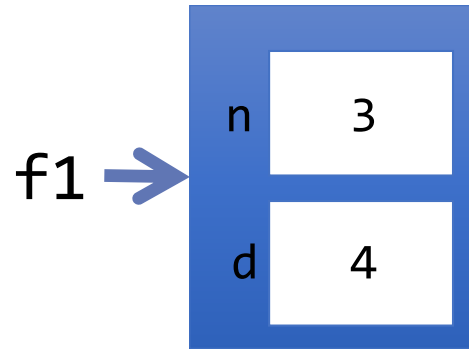
- `f = Fraccion(120,60)`
 `g = Fraccion(1,2)`
 `h = f.suma(g)`
- `i = h.mult(3)`
- `print(h)`
- `print(i)`

Fraccion: atributos + metodos



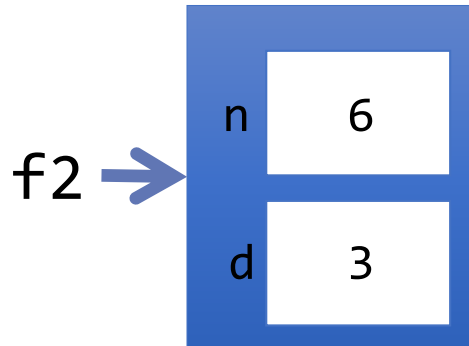
Definiendo nuestra propia Fraccion

- `f1 = Fraccion(3,4)`



Cada fracción tiene
dos variables internas
o atributos, llamados
n y d

- `f2 = Fraccion(6,3)`



Solución

- **class** Fraccion:

```
def __init__(self,n=0,d=1):  
    self.n = n  
    self.d = d
```

```
def suma(self,other):  
    return  
Fraccion(self.n*other.d+self.d*other.n,self.n*other.n)
```

- ```
def str(self):
 return str(self.n)+"/"+str(self.d)
```

## Problema #2



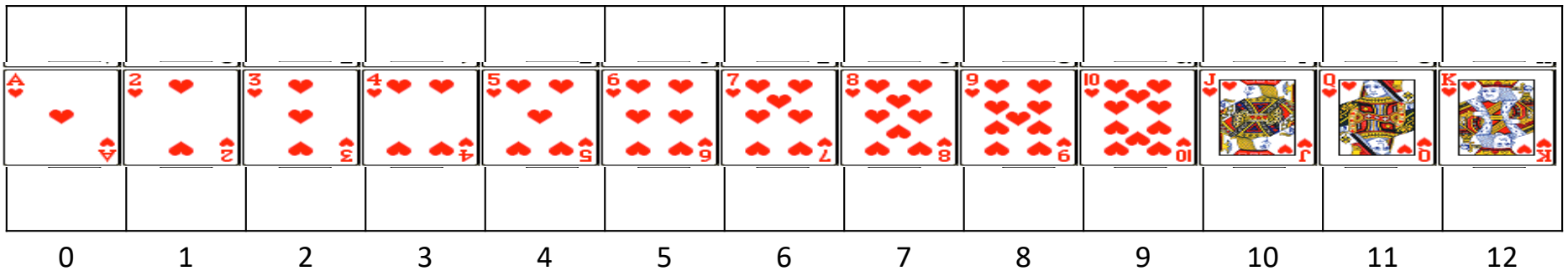
- ```
c1 = Carta(13,"C")
c2 = Carta(5,"D")
d = c1.diferencia(c2) #d es 8
print(d)
print(c1) #imprime KC
```

Un mazo es una lista de cartas

```
lista = []  
for i in range(1,14):  
    c = Carta(i,"C")  
    lista.append(c)
```

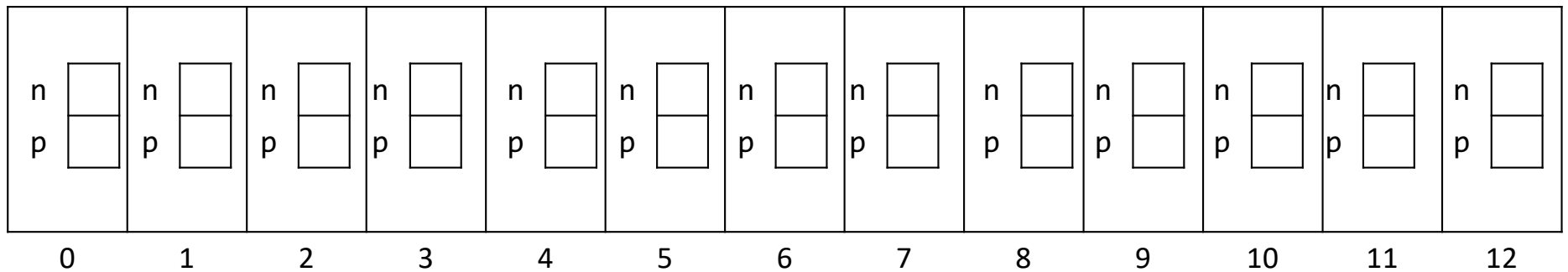
Un mazo es una lista de cartas

```
lista = []  
for i in range(1,14):  
    c = Carta(i,"C")  
    lista.append(c)
```



Un mazo es una lista de cartas

```
lista = []  
for i in range(1,14):  
    c = Carta(i,"C")  
    lista.append(c)
```



Un mazo es una lista de cartas

```
lista = []
for i in range(1,14):
    c = Carta(i,"C")
    lista.append(c)
```

The diagram illustrates a 13-bit shift register. It consists of 13 stages, indexed from 0 to 12. Each stage contains a flip-flop (labeled 'n' and 'p') and a combinatorial logic block (labeled 'C'). The flip-flop outputs are connected to the combinatorial logic blocks of the next stage. The combinatorial logic blocks are connected to the flip-flop inputs. The stages are arranged in a row, with the output of stage 12 being the final output of the register.

Problema #3: Mazo

- Define un Mazo, que permita obtener una Carta al azar del mazo (ya definimos la clase Carta 😊)

