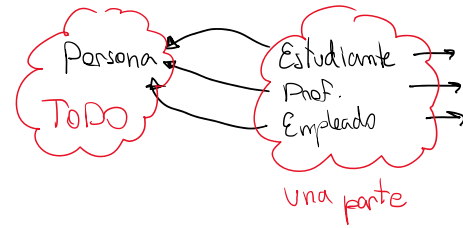


Programación Orientada a Objetos

Clases \leftrightarrow objetos
 ↑ ↑
 Todo parte

tipo de animal: \rightarrow `class Perro {`
`{`
 tipo de perro
`Perro chihuahua;`

.h main



Perro
 Atributos
 nombre
 edad
 Métodos
 correr()
 comer()

Int, double, str...
 Datos
 void(), int()
 Funciones

vector<int> \leftarrow hum
`class Perro {` \leftarrow perr
`{`

Encapsulamiento

public, private, protected herencia

`class Perro {`
`string name;` ✓
`int edad;` ✓
`void correr();` ✓
`{`

`class Perro {`
`private:`
`string name;`
`int edad;`
`public:`
`void correr();`

`int main() {`
`Perro pitbul;`
`cout << pitbul.name << endl;`

`int main() {`
`Perro pitbul;`
`pitbul.correr();`

Constructores y destructores

Rubén Rivas \rightarrow 6 tipos de constr.

Por defecto / default

`class Perro {`
`str name;`
`int age;`

`public:`
`Perro() = default;`
`Perro() {};`

`~Perro() {};`

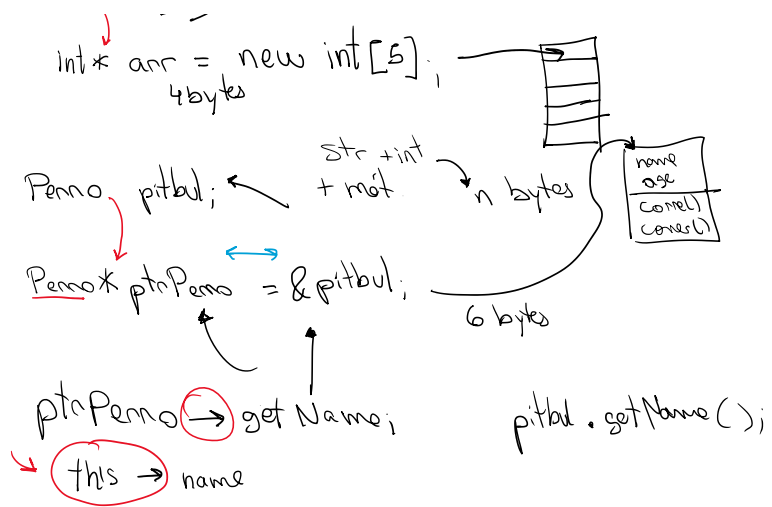
Asignación ✓

Copia / Copy

`Perro (&Perro p) {`
`name = p.name;`
`age = p.age;`
`{`

Clases dinámicas (Punteros a Objetos)

`int arr[5];` \leftarrow `int = 4`
`int* arr = new int[5];` \leftarrow `4 * 5 = 20`
 4 bytes



Contenedores de objetos

```

int arr[5];
arr[0] = 1;
  
```