

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de
ficheros

Encapsulación

Polimorfismo

Punteros a
funciones

Métodos

Constructor
vehicle

Constructor
car

Destruyores

Curso de programación en C moderno (II Edición)

Neira Ayuso, Pablo Falgueras García, Carlos

Tema 12

Objetos (II): Herencia

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de ficheros

Encapsulación

Polimorfismo

Punteros a funciones

Métodos

Constructor vehicle

Constructor car

Destruyores

1 Ejemplo de herencia

2 El objetivo

3 La forma

4 Herencia

- Estructura de ficheros
- Encapsulación

5 Polimorfismo

- Punteros a funciones
- Métodos
- Constructor vehicle
- Constructor car
- Destruyores

Ejemplo de herencia

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de ficheros

Encapsulación

Polimorfismo

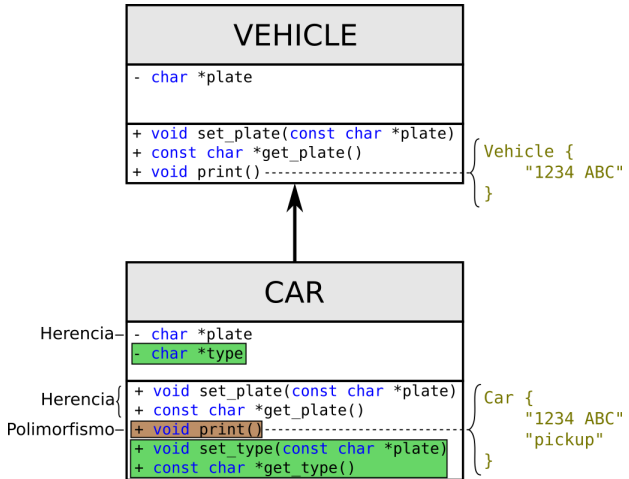
Punteros a funciones

Métodos

Constructor vehicle

Constructor car

Destruyores



El objetivo

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de archivos

Encapsulación

Polimorfismo

Punteros a funciones

Métodos

Constructor vehicle

Constructor car

Destruyores

```
1 #include "vehicle.h"
2 #include "car.h"
3
4 int main()
5 {
6     struct vehicle *v;
7     struct vehicle *c;
8
9     v = vehicle_alloc("1234 ABC");
10    c = (struct vehicle *)car_alloc("4321 CBA", "pickup");
11
12    vehicle_print(v);
13    vehicle_print(c);
14
15    return 0;
16 };
```

```
> ./vehicles
Vehicle {
    plate = "1234 ABC"
}
Car {
    plate = "4321 CBA"
    type = "pickup"
}
```

La forma

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de ficheros

Encapsulación

Polimorfismo

Punteros a funciones

Métodos

Constructor vehicle

Constructor car

Destructores

- **Encapsular** la estructura del objeto padre en la del objeto hijo para implementar la **herencia**
- Utilizar **punteros a funciones** para implementar el **polimorfismo**: Un puntero guardará una función u otra en función del tipo de objeto

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de
ficheros

Encapsulación

Polimorfismo

Punteros a
funciones

Métodos

Constructor
vehicle

Constructor
car

Destruyores

HERENCIA

Estructura de ficheros

Portada

Ejemplo

El objetivo

La forma

Herencia

**Estructura de
ficheros**

Encapsulación

Polimorfismo

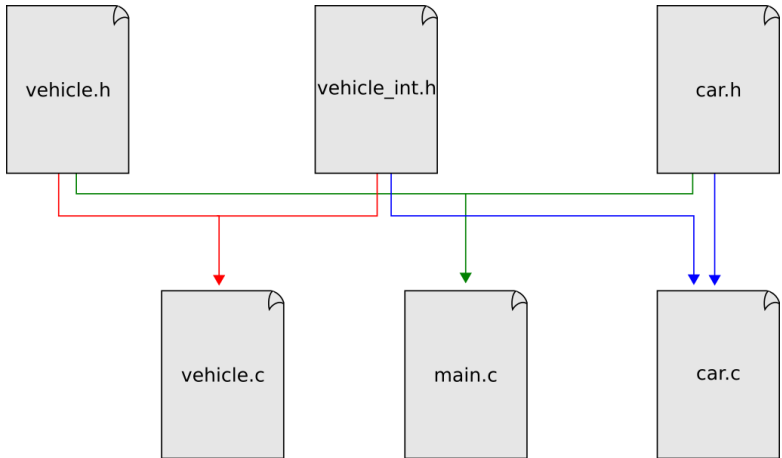
Punteros a
funciones

Métodos

Constructor
vehicle

Constructor
car

Destruyores



Encapsulación

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de archivos

Encapsulación

Polimorfismo

Punteros a funciones

Métodos

Constructor vehicle

Constructor car

Constructores

vehicle_int.h

```
1 #include <stdint.h>
2
3 struct vehicle
4 {
5     char *plate;
6     uint32_t flags;
7 };
8
9 enum vehicle_attr {
10     VEHICLE_PLATE,
11 };
```

car.c

```
1 #include "vehicle_int.h"
2
3 struct car {
4     struct vehicle super; /* Siempre el primero */
5     char *type;
6     uint32_t flags;
7 };
8
9 enum car_attr {
10     CAR_TYPE,
11 };
```


Encapsulación

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de ficheros

Encapsulación

Polimorfismo

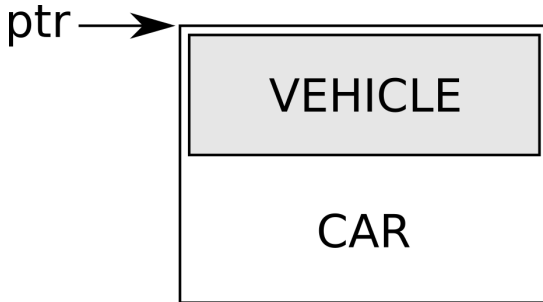
Punteros a funciones

Métodos

Constructor vehicle

Constructor car

Destruidores



```
struct vehicle *v = (struct vehicle *)car_alloc(...);
```

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de
ficheros

Encapsulación

Polimorfismo

Punteros a
funciones

Métodos

Constructor
vehicle

Constructor
car

Destructores

POLIMORFISMO

Punteros a funciones

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de archivos

Encapsulación

Polimorfismo

Punteros a funciones

Métodos

Constructor vehicle

Constructor car

Destruyores

```
1 #ifndef VEHICLE_INT_
2 #define _VEHICLE_INT_
3
4 #include <stdint.h>
5
6 struct vehicle
7 {
8     char *plate;
9
10    void (*set_plate)(struct vehicle *, const char *);
11    const char *(*get_plate)(const struct vehicle *);
12    void (*print)(const struct vehicle *);
13
14    uint32_t flags;
15 };
16
17 enum vehicle_attr {
18     VEHICLE_PLATE,
19 };
20
21 #endif
```

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de
archivos

Encapsulación

Polimorfismo

Punteros a
funciones

Métodos

Constructor
vehicle

Constructor
car

Constructores

```
1 static void default_print(const struct vehicle *v)
2 {
3     printf("Vehicle ");
4
5     if (ATTR_IS_SET(v->flags, VEHICLE_PLATE))
6         printf("\n\tplate = \"%s\"", v->plate);
7
8     printf("\n}\n");
9 }
10
11 void vehicle_print(const struct vehicle *v)
12 {
13     v->print(v);
14 }
```

Nota: Para crear un método/clase **abstract@** basta con no crear el/los método(s) por defecto

Constructor vehicle

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de archivos

Encapsulación

Polimorfismo

Punteros a funciones

Métodos

Constructor vehicle

Constructor car

Destruedores

```
1 void vehicle_init(struct vehicle *v, const char *plate)
2 {
3     v->flags = 0;
4
5     v->set_plate = default_set_plate;
6     v->get_plate = default_get_plate;
7     v->print = default_print;
8
9     default_set_plate(v, plate);
10 }
11
12 struct vehicle *vehicle_alloc(const char *plate)
13 {
14     struct vehicle *v;
15
16     v = (struct vehicle *)malloc(sizeof(struct vehicle));
17     if (!v)
18         return NULL;
19
20     vehicle_init(v, plate);
21
22     return v;
23 }
```

Constructor car

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de archivos

Encapsulación

Polimorfismo

Punteros a funciones

Métodos

Constructor vehicle

Constructor car

Destruyores

```
1 struct car *car_alloc(const char *plate, const char *type
2 )
3 {
4     struct car *c;
5
6     c = (struct car *)malloc(sizeof(struct car));
7     if (!c)
8         return NULL;
9
10    vehicle_init(&c->super, plate);
11
12    c->flags = 0;
13    c->super.print = car_printf;
14    car_set_type(c, type);
15
16    return c;
17 }
```

Destructores

Portada

Ejemplo

El objetivo

La forma

Herencia

Estructura de ficheros

Encapsulación

Polimorfismo

Punteros a funciones

Métodos

Constructor vehicle

Constructor car

Destructores

vehicle

```
1 void vehicle_free(struct vehicle *v)
2 {
3     if (ATTR_IS_SET(v->flags , VEHICLE_PLATE))
4         free(v->plate);
5
6     free(v);
7 }
```

car

```
1 void car_free(struct car *c)
2 {
3     if (ATTR_IS_SET(c->flags , CAR_TYPE))
4         free(c->type);
5     vehicle_free(&c->super);
6 }
```