

Указатель на функцию

```
// Programming 1. Sample at 2017-03-02-13-55.55
#include <stdio.h>

struct Person
{
    const char *name;
    int age;
} person;

void print_nice(const struct Person *p)
{
    printf("Person %s age %d\n", p->name, p->age);
}

void print_ugly(const struct Person *p)
{
    printf("[%s,%d]\n", p->name, p->age);
}

void print(const struct Person *obj,
           void (*printer)(const struct Person*))
{
    printer(obj);
}

int main(/*int c, char **v */)
{
    person.name = "Vasya";
    person.age = 15;

    print(&person, print_nice);
    print(&person, print_ugly);

    return ;
}
```

Полиморфный принтер

```
// Programming 1. Sample at 2017-03-02-13-55.55
#include <stdio.h>

struct Person
{
    const char *name;
    int age;
    void (*print)(const struct Person*);
    void (*printer)(const struct Person*);
}
```

```
} person1, person2;

void print_nice(const struct Person *p)
{
    printf("Person %s age %d\n", p->name, p->age);
}

void print_ugly(const struct Person *p)
{
    printf("[%s,%d]\n", p->name, p->age);
}

void print(const struct Person *obj)
{
    obj->printer(obj);
}

int main(/*int c, char **v */)
{
    person1.name = "Vasya";
    person1.age = 15;
    person1.print = print;
    person1.printer = print_nice;

    person2.name = "Petya";
    person2.age = 51;
    person2.print = print;
    person2.printer = print_ugly;

    person1.print(&person1);
    person2.print(&person2);

    return ;
}
```

Universal search

```
// Programming 1. Sample at 2018-02-15-14-55.05
#include <stdio.h>

struct S{
    int a;
    int b;
};

void* search( void *key,
              void *arr,
              int n,
```

```
        int size,
        int (*comp)(const void *, const void *)
    ){

    for(int i=0;i<n;i++){
        if(comp(key, arr+i*size) == 0){
            return arr+i*size;
        }
    }
    return NULL;
}

// if both fields are equal, the structures are equal
// otherwise compare by the sum of two fields
int comp(const void *a, const void *b){
    struct S *aa = (struct S*)a;
    struct S *bb = (struct S*)b;
    if(aa->a == bb->a && aa->b == bb->b){
        return 0;
    }
    if((aa->a + aa->b) - (bb->a + bb->b) > 0){
        return 1;
    }
    return -1;
}

int main(){
    struct S arr[]={1,2},{3,4},{5,6},{6,7}};
    struct S* value;
    struct S key={5,6};

    value = search(&key, arr, 4, sizeof(struct S), comp);
    if(value){
        printf("%zu\n", value - arr);
    }else{
        printf("nooooo\n");
    }
    return 0;
}
```

From:
<http://se.moevm.info/> - **se.moevm.info**

Permanent link:
<http://se.moevm.info/doku.php/start:prog2:lectures:code2>

Last update: **2022/12/10 09:08**

