# factory method（工厂方法模式）

## 简述：

定义一个**用于创建对象**的接口，让子类决定实例化哪一个类。工厂方法，使一个类的实例化延迟到其子类。

## 原理：

使用工厂方法，使得实例化延迟到子类。

## 接口：

```
/**
 * roujiamo_store.c
 */
typedef struct _store_t store_t;

typedef void (* fp_sell_produce)(store_t *thiz, int kind);
typedef roujiamo_t *(* fp_factory_make)(int kind);

typedef struct _store_t {
    fp_factory_make make_func;
    fp_sell_produce sell_func;
}store_t;
```

## 应用：

```
/**
 * dongguan_store.c
 */
#include "dongguan_store.h"

#include <stdio.h>
#include <stdlib.h>

static void dongguan_store_sell (store_t *thiz, int kind)        //实现sell_func接口
{
    roujiamo_t *p_roujiamo = thiz->make_func(kind);

    printf("you had bought %s roujiamo~\r\n", p_roujiamo->taste);

    p_roujiamo->prepare();
    p_roujiamo->fire();
    p_roujiamo->pack();

    printf("sale~!\r\n");
}
```

```
store_t *dongguan_store_create ()
{
    store_t *thiz = NULL;

    thiz = (store_t *)malloc(sizeof(store_t));

    thiz->sell_func = dongguan_store_sell;
    thiz->make_func = factory_make;

    return thiz;
}
```

```
/**
 * main.c
 */
#include <stdio.h>

#include "dongguan_store.h"
#include "roujiamo_store.h"

int main ()
{
    store_t *ming_store = dongguan_store_create();

    ming_store->sell_func(ming_store, 1);

    return 0;
}
```