MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy
讲师：高焕堂（台湾）

http://www.microoh.com

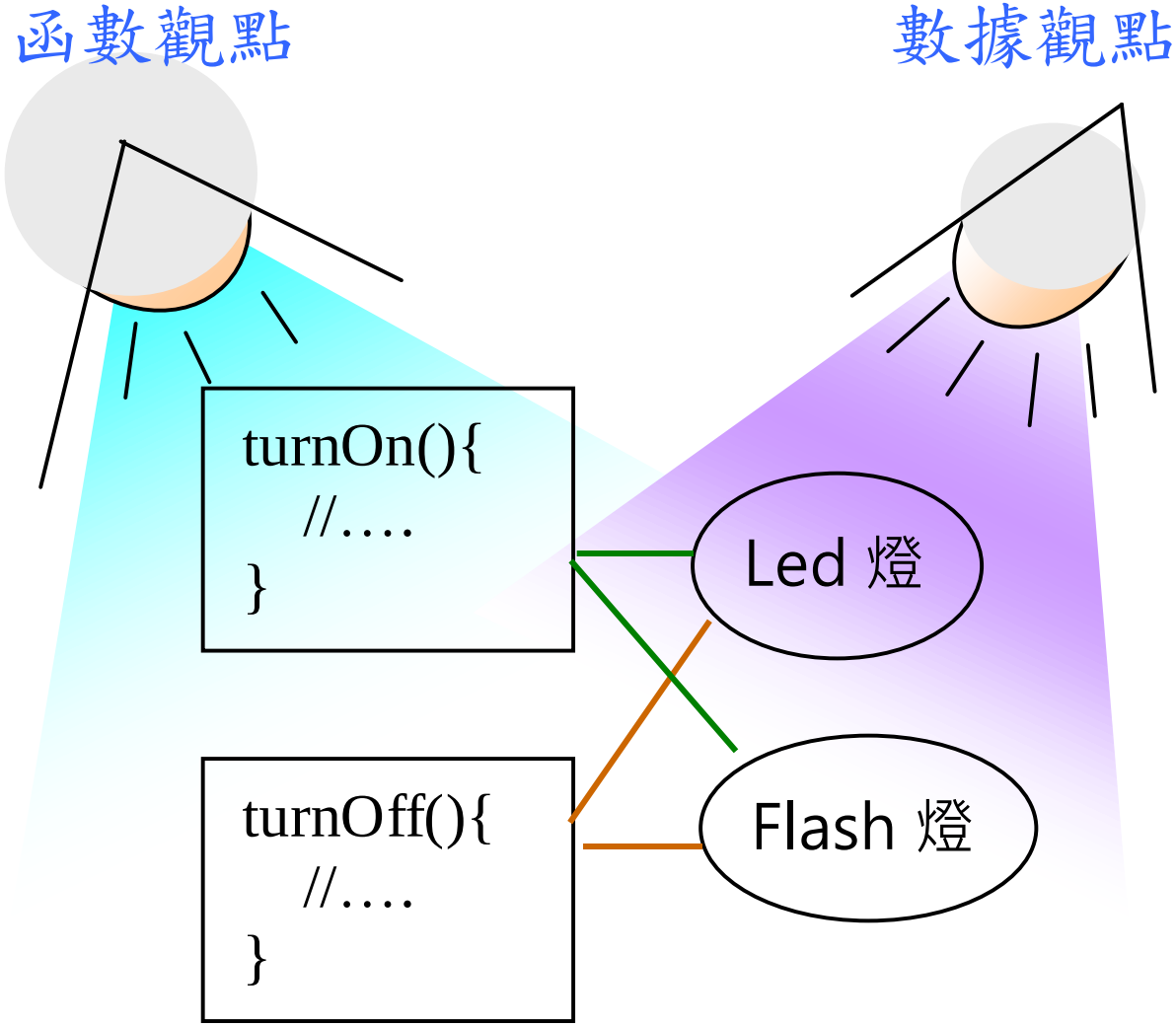
C01_d

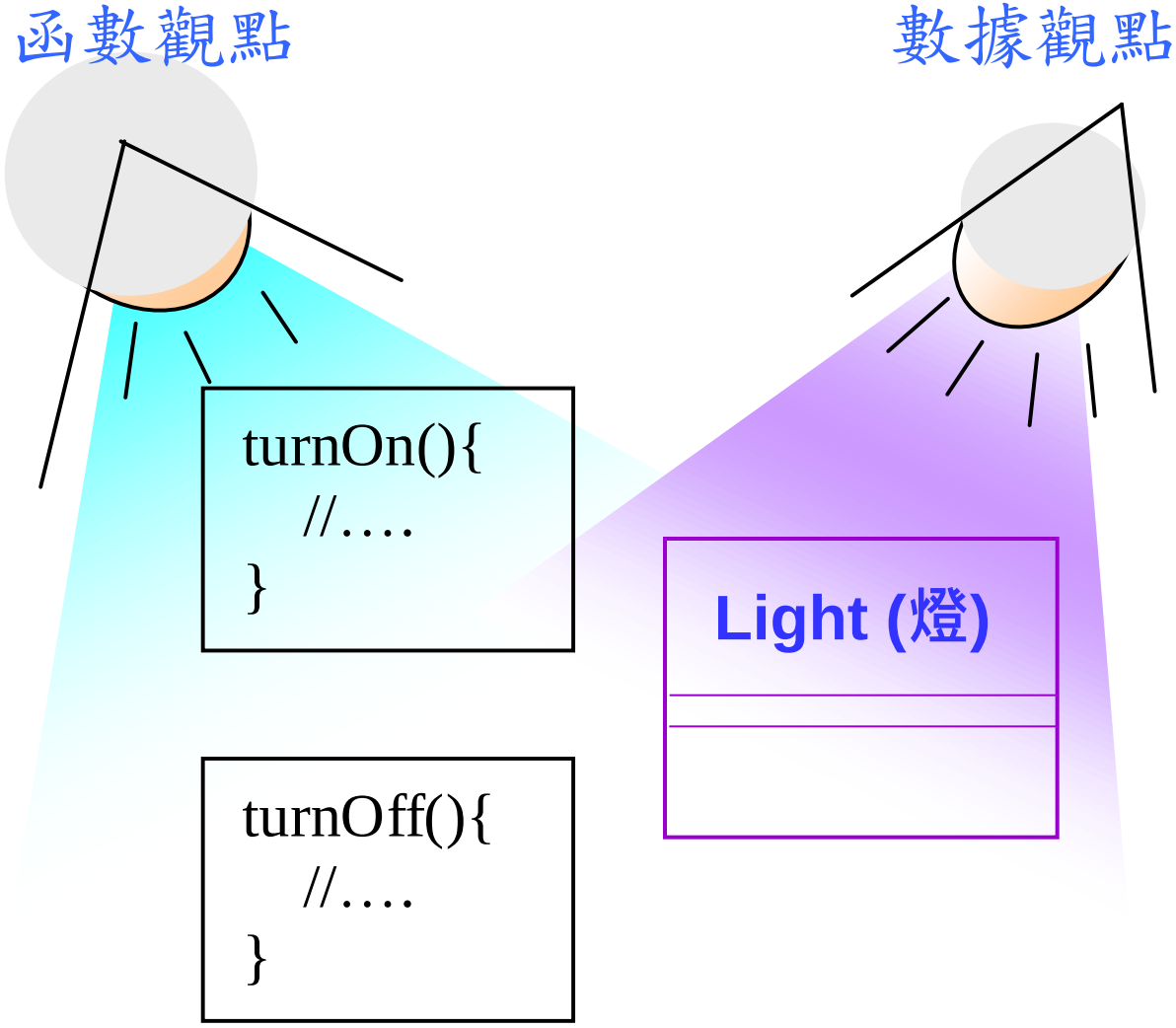
JNI架构原理：Java与C的对接(d)

By 高煥堂

4、 EIT造形的C和Java组合实现

前言


函數觀點
數據觀點
turnOn(){
//….
}
turnOff(){
//….
}
Led 燈
Flash 燈


函數觀點
數據觀點
turnOn(){
//….
}
turnOff(){
//….
}
Light (燈)

C定義Light類

```
// C代碼
```

對接

C函數

```
// a.so 檔案(File)
void turnOn( Light *obj )
   {   obj->state = 1;   }
void turnOff( Light *obj )
   {   obj->state = 0;   }
```

C誕生對象& 調用函數

```
// C代碼
```

C定義Light類

```
typedef struct Light Light;
struct Light {
    int state;
    void (*turnOn)(Light*);
    void (*turnOff)(Light*);   };
```

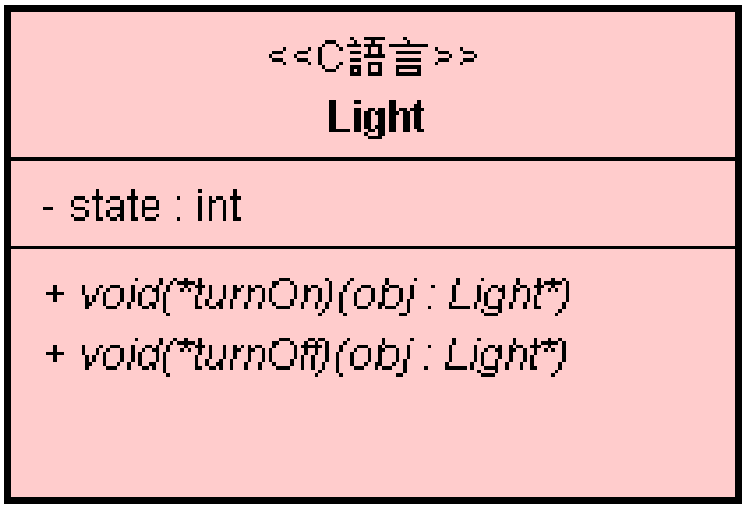
對接

C函數

```
// a.so 檔案(File)
void turnOn( Light *obj )
   {   obj->state = 1;   }
void turnOff( Light *obj )
   {   obj->state = 0;   }
```

C誕生對象&調用函數

```
struct Light *LightNew(){
   struct Light *t = (Light *)
            malloc(sizeof(Light));
   return (void*) t;
 }

void main() {
  Light *led = (Light*)LightNew();
  t->turnOn=turnOn; /*裝配C函數 */
  t->turnOff = turnOff;
  led->turnOn( led ); /* 啟動執行 */
  led->turnOff( led );    }
```




```
static void turnOn( Light *cobj ){
        cobj->state = 1;
        printf( “ON” );
 }
static void turnOff( Light *cobj ) {
         cobj->state = 0;
         printf( “OFF” );
 }
```

Java定義Light類

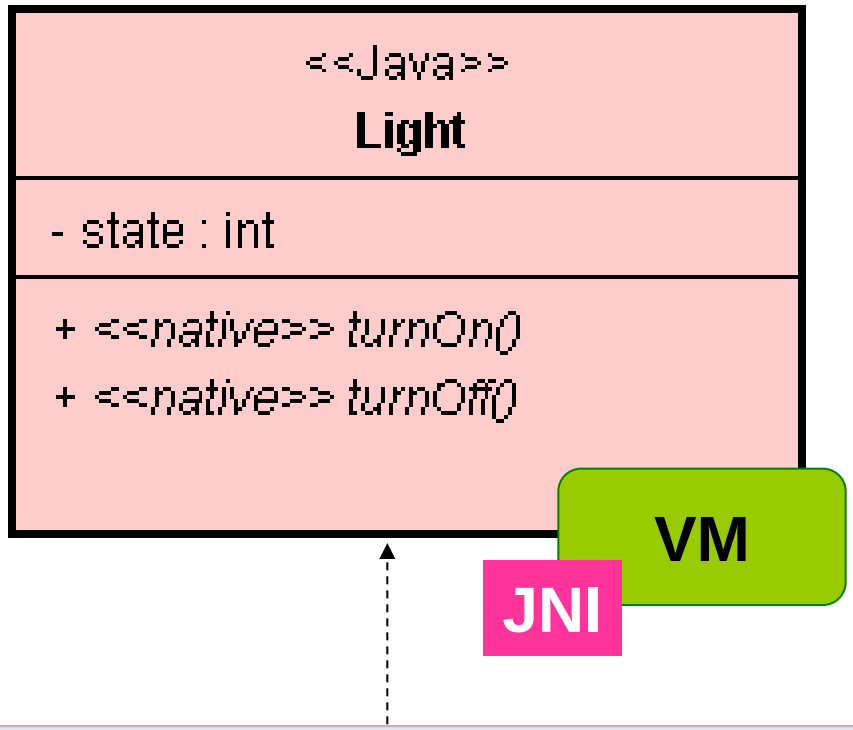
```
// Java代碼
```

C函數

```
// a.so 檔案(File)
void turnOn( Light *obj )
   {   obj->state = 1;   }
void turnOff( Light *obj )
   {   obj->state = 0;   }
```

Java誕生對象&調用函數

```
// Java代碼
```




```
static void turnOn( jobject* thiz ){
        // thiz->state = 1;
        // printf("ON");
 }
static void turnOff( jobject* thiz ) {
        // thiz->state = 0;
        // printf("OFF"); }
```

目前对象(Current Object)指针

- 无论是C或Java都必须将目前对象(Current Object)指针传给C函数。
- 让C函数可指向目前对象，以便存取对象的内部属性质或调用类里的其它函数。

定义Light类

```
typedef struct Light Light;
struct Light {
    int state;
    void (*turnOn)(Light*);
    void (*turnOff)(Light*);
};
```

撰写构造式

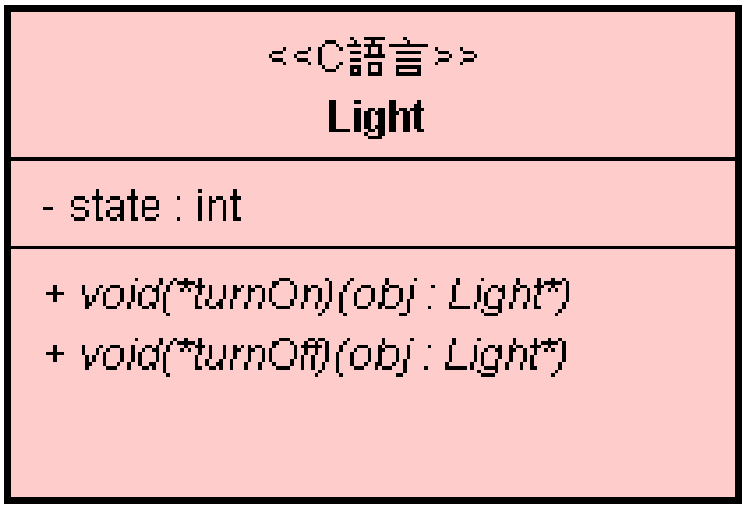
```
struct Light *LightNew(){   // 構造式
      struct Light *t;
      t = (Light *)malloc(sizeof(Light));
      return (void*) t;
 }
```

撰写C函数

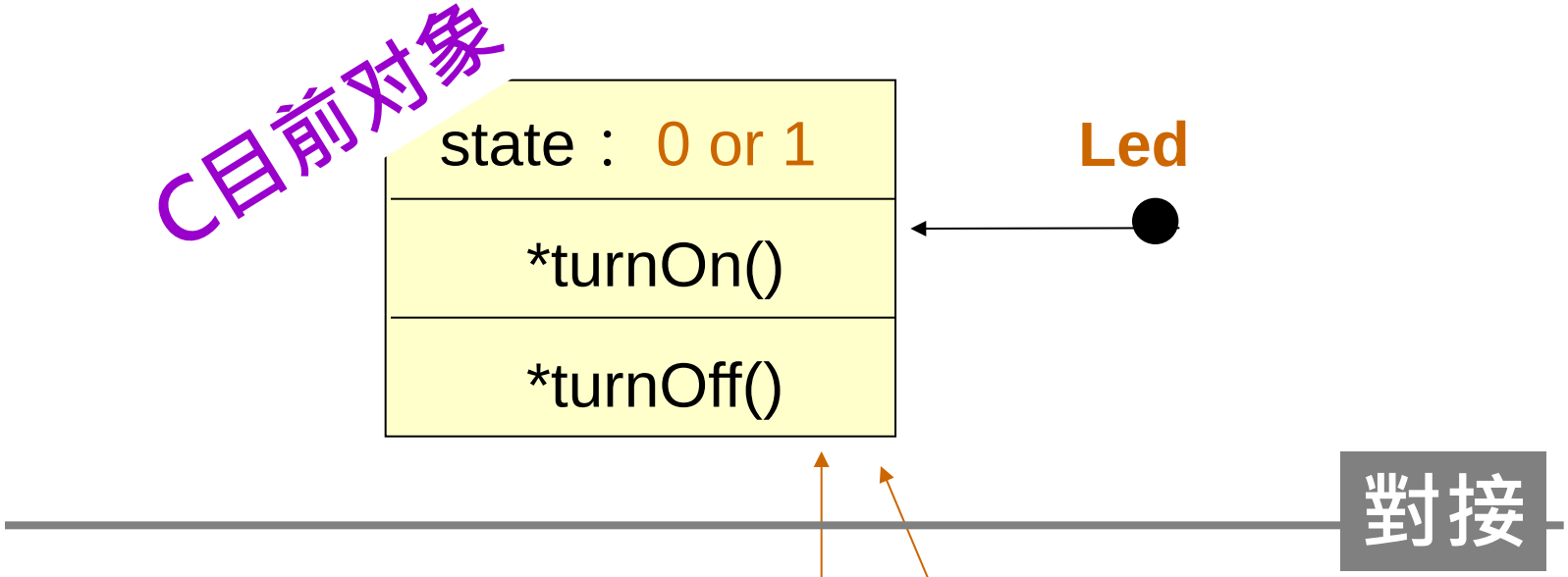
```
static void turnOn( Light *cobj ){
        cobj->state = 1;
        printf( “ON” );
 }
static void turnOff( Light *cobj ) {
         cobj->state = 0;
         printf( “OFF” );
 }
```

裝配&执行

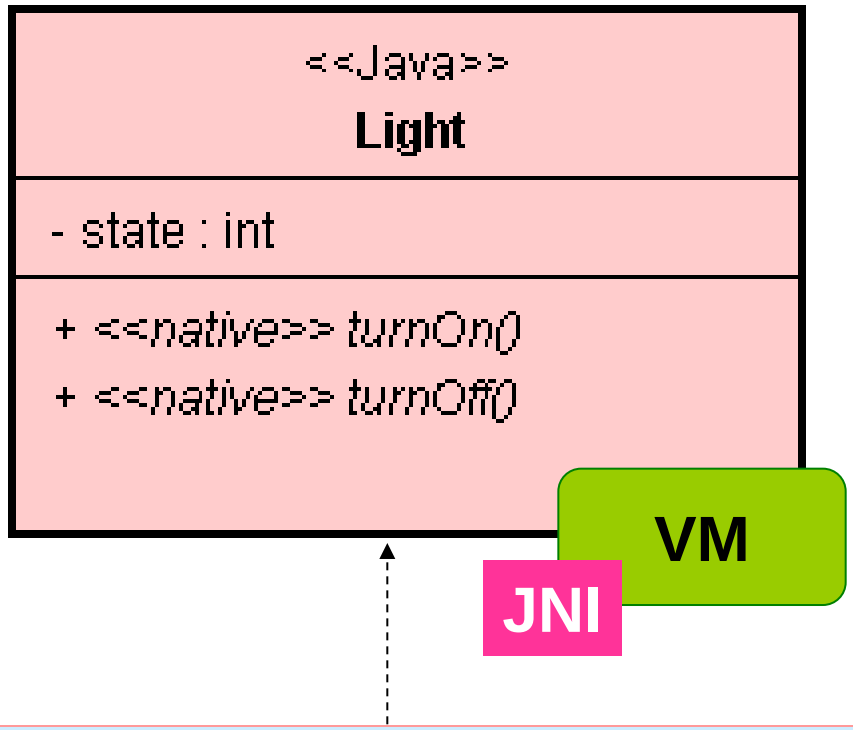
```
void main() {
    Light *led = (Light*)LightNew();

    t->turnOn = turnOn; /* 裝配C函數 */
    t->turnOff = turnOff;

    led->turnOn( led ); /* 啟動執行 */
    led->turnOff( led );
}
```



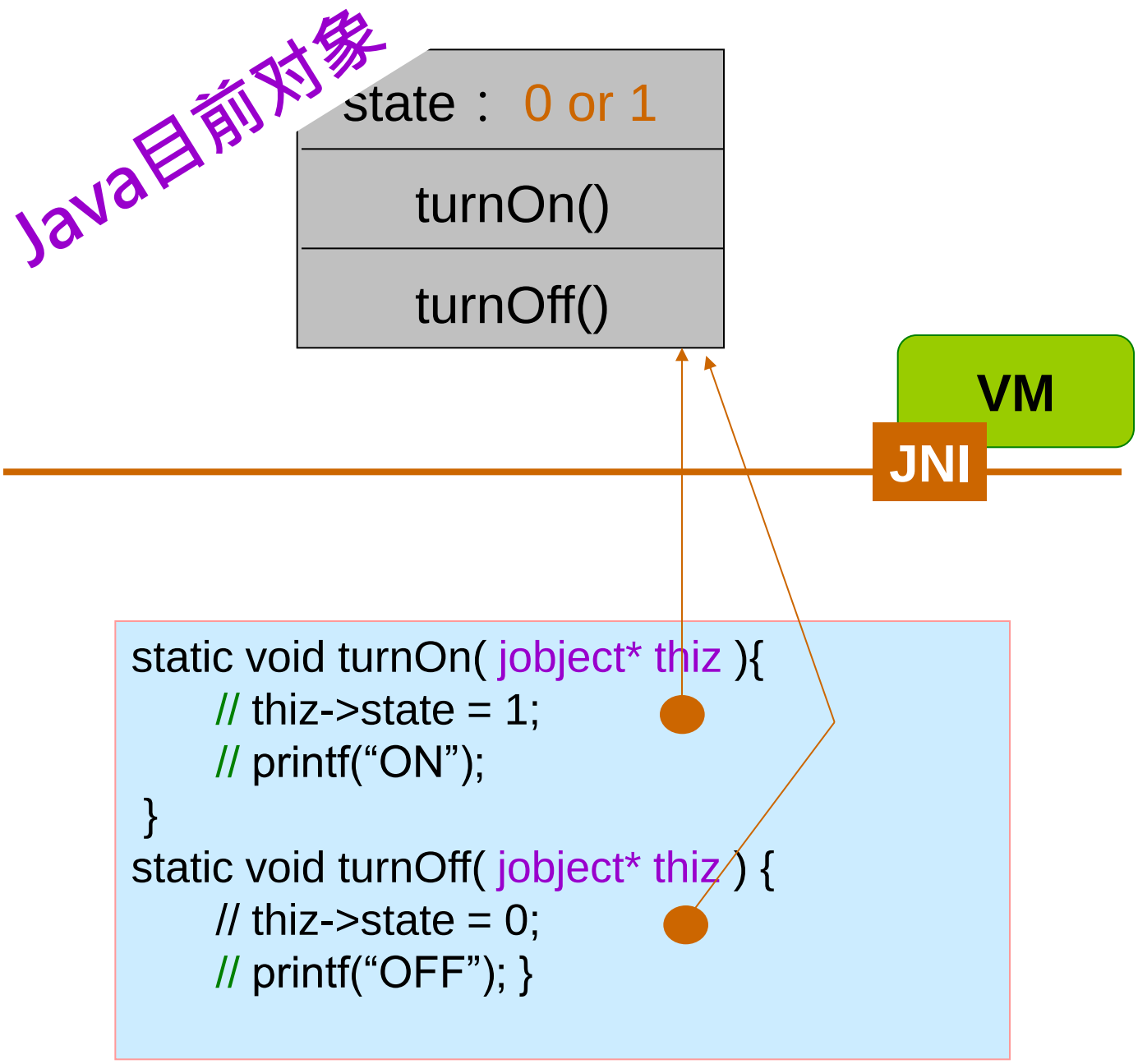

```
static void turnOn( Light *cobj ){
        cobj->state = 1;
        printf( “ON” );
 }
static void turnOff( Light *cobj ) {
         cobj->state = 0;
         printf( “OFF” );
 }
```




```
static void turnOn( Light *cobj ){
        cobj->state = 1;
        printf( “ON” );
 }
static void turnOff( Light *cobj ) {
        cobj->state = 0;
        printf( “OFF” );
 }
```




```
static void turnOn( jobject* thiz ){
        // thiz->state = 1;
        // printf("ON");
 }
static void turnOff( jobject* thiz ) {
        // thiz->state = 0;
        // printf("OFF"); }
```


Java目前对象
state： 0 or 1
turnOn()
turnOff()
VM
JNI
static void turnOn(jobject* thiz){
// thiz->state = 1;
// printf("ON");
}
static void turnOff(jobject* thiz) {
// thiz->state = 0;
// printf("OFF"); }

~ Continued ~