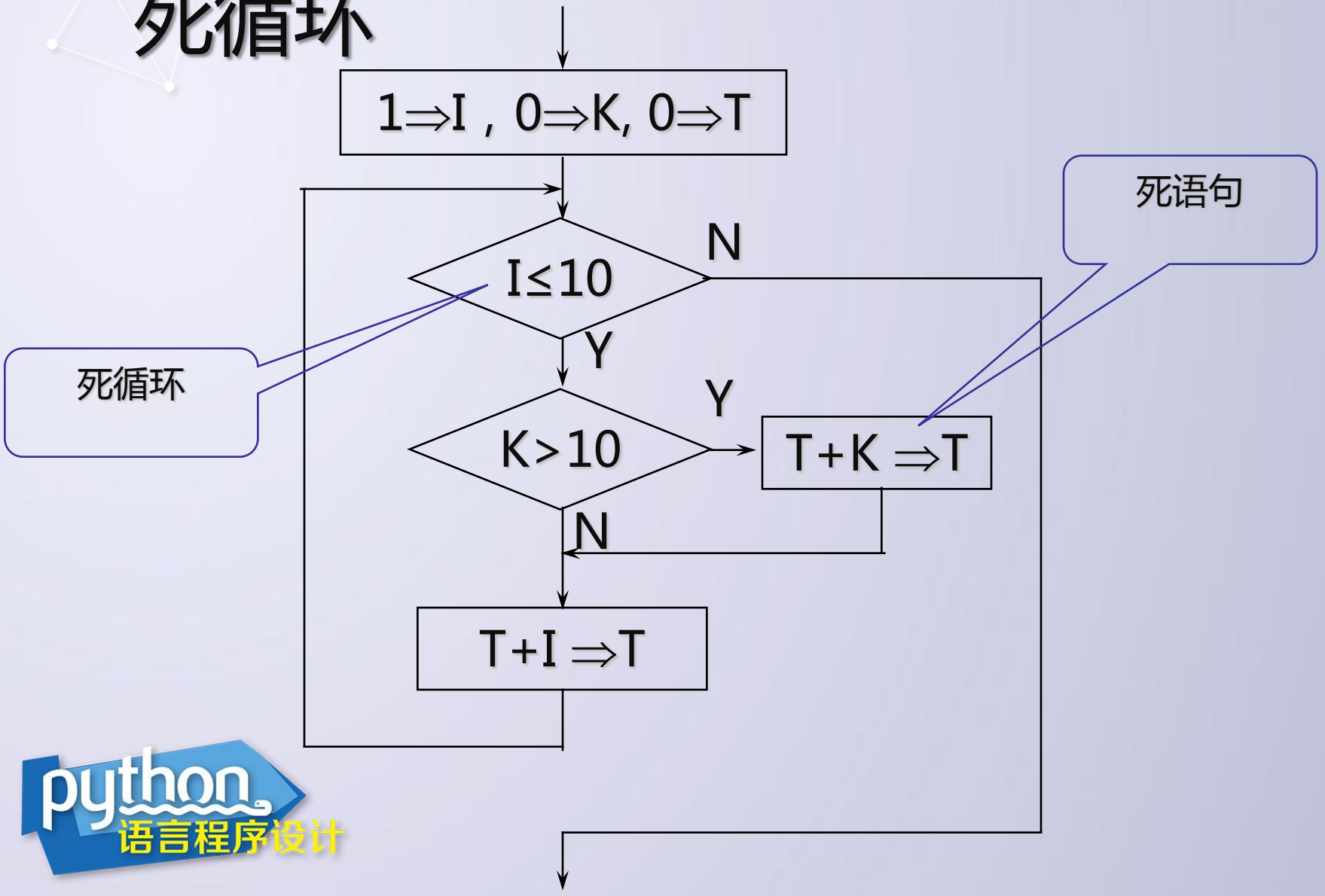
死循环/嵌套循环

礼 欣

北京理工大学

死循环



python 语言程序设计

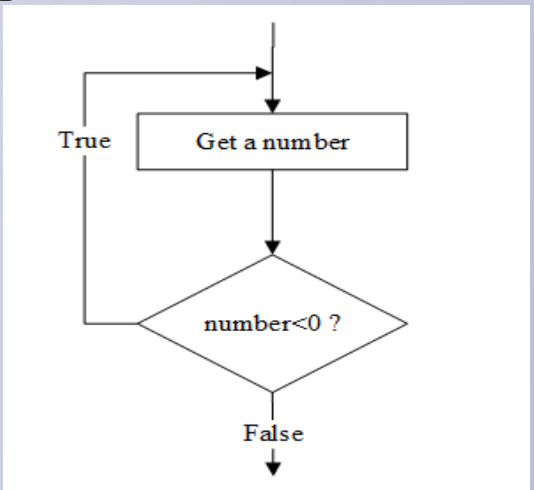
死循环的使用

- 死循环并非一无是处, c语言中死循环while true或while 1是单片机编程的普遍用法，死循环一直运行等待中断程序发生，然后去处理中断程序
- 在python 中我们也可以利用死循环完成特定功能

```
while True:
    try:
        x = int(input("Please enter a number:"))
        break
    except ValueError:
        print("Oops, that was no valid number. Try again ...")
```

后测循环

- 假设程序需要用户输入一个非负数
- 如果输入错误，重新提示用户输入直到得到一个有效值
- 伪代码如下：
  - 重复执行以下语句
  - 　接受外部输入数据
  - 直到外部输入为负值为止
- 循环不断接收输入，直到接受到合法的值为止
- 条件判断在循环体后面，称之为后测循环
- 后测循环至少执行一次循环体



python 语言程序设计

后测循环实现

- Python没有后测循环语句，但可以通过while间接实现
- 思想是设计一个循环条件，直接进入循环体，循环至少执行一次，相当于后测循环

```
number = -1
while number < 0:
    number = eval(input("Enter a positive number: "))
```

后测循环实现

- break语句也可以用来实现后测循环

```
while True:
    number = eval(input("Enter a positive number: "))
    if x >= 0: break
    # 如果数字有效则跳出循环
```

- while语句体永远执行，if条件决定循环退出
- 另外：if语句体只包含一个语句时，break可以跟if在同一行

后测循环代码1

- 在前面的while 版本的后测循环代码中添加一个if语句，使得在有效输入时不显示警告
- 修改代码如下：

```
number = -1
    while number < 0:
        number = eval(input("Enter a positive number: "))
        if number < 0:
            print("The number you entered was not positive")
```

后测循环代码2

- 限定合法性检验只判断一次，需为if 添加匹配的 else语句来实现
- 修改后的代码如下：

```
while True:
    number = eval(input("Enter a positive number: "))
    if x >= 0:
        break # 如果数字有效则跳出循环
    else:
        print("The number you entered was not positive")
```

半路循环

- 运用break中途退出循环，循环出口在循环体中部，被称为半路循环

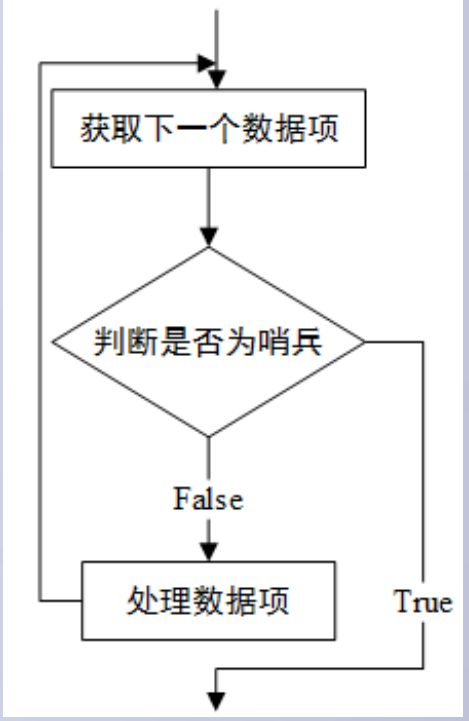
```
while True:
    number = eval(input("Enter a positive number: "))
    if x >= 0: break     # 跳出循环
    print("The number you entered was not positive")
```

半路循环-哨兵

- 半路循环退出实现哨兵循环的一般模式

```
while True:
    Get next data item
    if the item is the sentinel: break
    process the item
```

- 在程序中是否使用break语句，跟个人编程风格有关。
- 应避免在一个循环体内使用过多的break语句。因为当循环有多个出口的时候，程序逻辑就显得不够清晰了。



python 语言程序设计