第二部分 软件测试

2.13 对循环的测试

本节教学目标

- 理解常规循环结构有哪些
- 掌握各种循环结构的测试方法
- 重难点：各种循环结构的测试方法

内容回顾

- 白盒测试基础知识：
  - 白盒测试关注的对象：源代码和程序结构
  - 动态白盒测试的方法：
    - 逻辑覆盖：语句、判定、条件、条件判定、条件组合、路径覆盖
    - 独立路径测试：
      - 画出程序图
      - 确定环复杂度（3种方式）
      - 找出独立路径
      - 去掉不可行路径+补充路径
      - 转化成测试用例

内容回顾

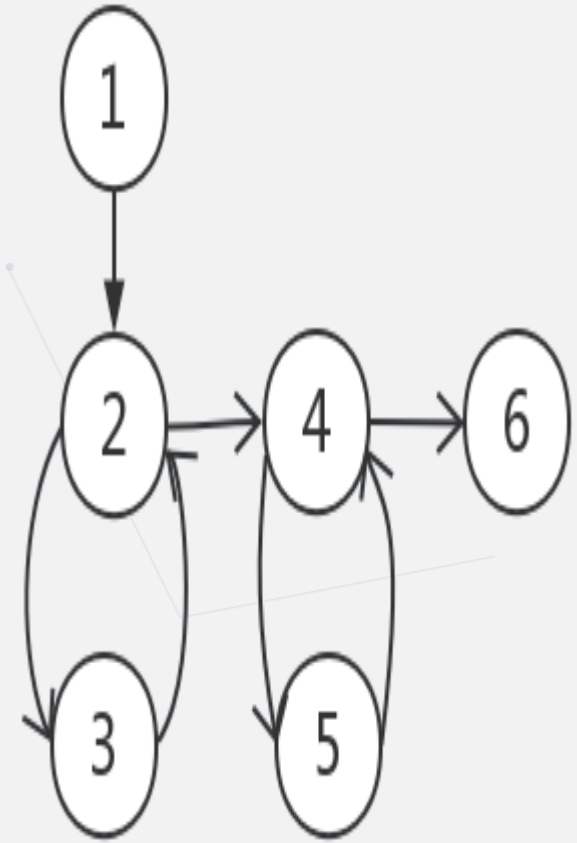
- 计算环复杂度的3种方法的注意事项：
  - 直观观察法
  - 公式计算法：V(G) = e - n + 1，需要注意：
    - 程序图不包含孤立节点
    - 程序图必须是一个强连通图（可以对图进行改造来完成）
  - 判定节点法：V(G) = P + 1，需要注意：
    - 由switch所引起的多分支，先修改为if……else if……else形式

目录

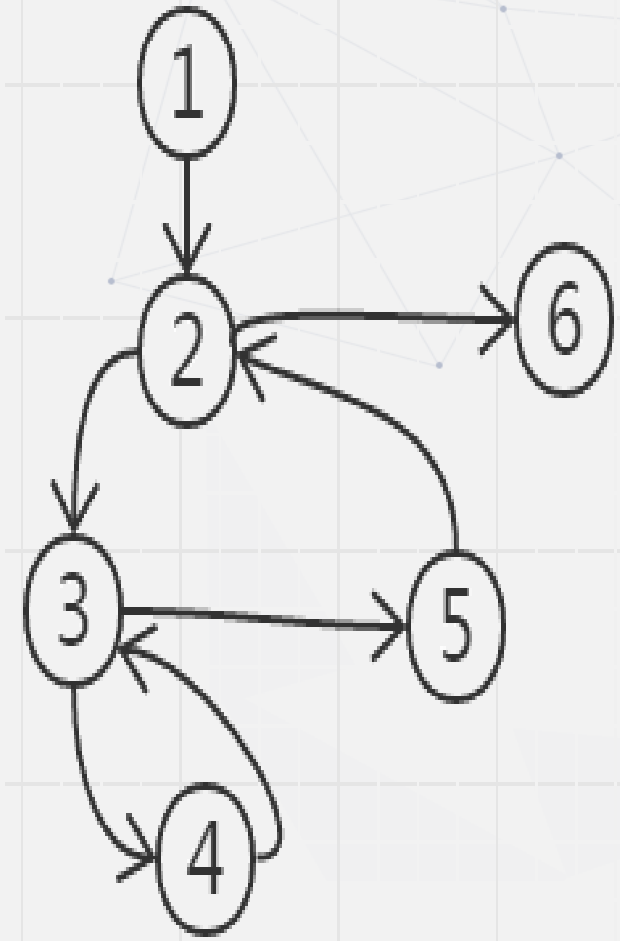
循环的测试背景知识

- 为什么进行循环测试
  - 循环结构是程序设计时一类重要结构
  - 重复多次循环可能导致内存泄漏
  - 循环到边界位置可能出现错误
- 对循环结构进行测试的重点关注点：
  - 循环过程的正确性
  - 循环的边界和界限内对循环体的执行过程

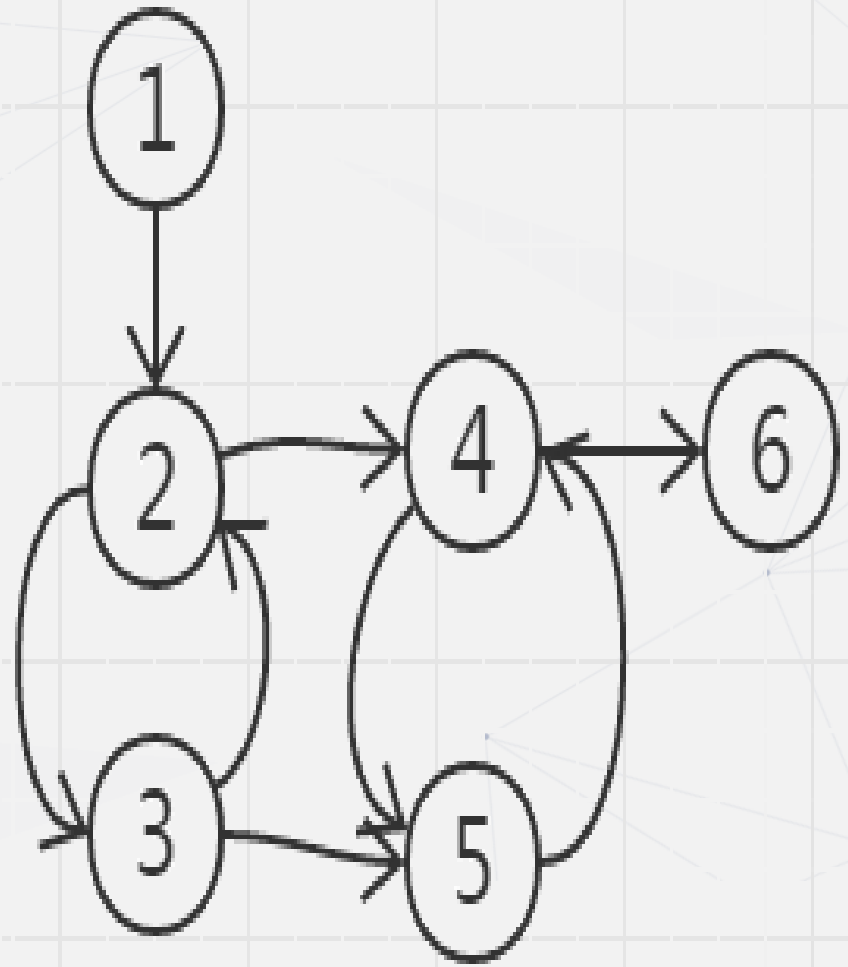
循环结构分类



循环节点串联



循环节点嵌套



非结构化的循环

循环结构测试难点分析

- 针对不同类型的循环结构，测试难点如下：
  - 单个循环节点，如何结合测试循环的边界进行测试
  - 单个循环节点，如何设计测试用例来保证循环的完整性
  - 串联的循环节点，如何保证测试的全面性
  - 对于非结构化的循环，如何进行测试

目录

单个循环节点测试分析

- 针对单个循环节点循环次数的测试（假设循环N次）
- 测试点如下：
  - 循环0次（即不执行循环）
  - 循环1次
  - 循环2次
  - 循环正常次数（通常为最大次数的一半）
  - 循环n-1次
  - 循环n次

```
1 void SampleFunc4(int iteration)
2 {//iteration决定循环的次数
3     for (int i = 1;i < iteration;i++)
4     {
5         printf("i = %d\n",i);
6     }
7 }
```

单个循环节点测试分析

```
1 void SampleFunc4(int iteration)
2 {//iteration决定循环的次数
3       for (int i = 1;i < iteration;i++)
4       {
5           printf("i = %d\n",i);
6       }
7 }
```

测试项	输入条件	预期输出
循环0次	Iteration = 0	不进入循环体，i值不变，屏幕无显示
	Iteration = 1	不进入循环体，i值不变，屏幕无显示
	Iteration = 10	
循环1次		i = 1
循环2次		i = 2
循环正常次数		i = 5
循环n - 1次		i = 8
循环n次		i = 9

单个循环节点测试总结

- 循环的初始化
  - 控制循环过程的变量称为循环变量，对于初值设置是否正确，初值设置错误，则循环总次数必然受到影响
- 循环的迭代
  - 测试循环体内包含的语句执行过程
  - 测试增量的变化是否正确
  - 每次循环涉及到的变量的取值是否按预期规律发生变化
  - 重复多次循环是否导致误差累积

单个循环节点测试总结

- 多次循环是否对内存造成压力
- 是否存在continue，break语句，导致某些循环过程中强制跳过部分语句不执行，从而注入代码质量风险

- 循环的终止
  - 循环的终止条件是否存在边界错误
  - 退出循环的条件是否正确

- 如上例中需要测试的数据变量为循环变量i，增量1，最大值iteration和退出条件

串联循环节点的测试

- 当各循环节点串联，若各个判定节点相互独立，则仅需根据单个循环体的测试原则进行测试即可
- 对于串联循环节点存在相互关联，则不能孤立的测试每个循环节点，应结合测试

循环节点嵌套测试分析

- 当循环节点为嵌套形式，且判定节点相互独立时：

```
1 def test():
2     for i in range(1,10):
3         for j in range(1,10):
4             print("%d*%d=%2d" % (j,i,i * j),end=" ")
5         print("")
6 test()
```

  - 先测试最内层循环体，然后逐步外推，直至测试到最外层的循环体
  - 测试每层循环体时，仍根据单个循环体的测试原则进行测试

循环节点嵌套测试分析

- 考虑4种特殊组合：
  - 1）内层最小循环次数，外层最小循环次数组合，计算结果
  - 2）内层最小循环次数，外层最大循环次数，计算结果
  - 3）内层最大循环次数，外层最小循环次数，计算结果
  - 4）内层最大循环次数，外层最大循环次数，计算结果

非结构化循环结构测试

- 首先：建议修改代码
- 其次：如果不能修改代码，则先对单次循环体进行测试；兼顾嵌套循环条件下对循环次数的多种特殊组合

循环测试实例练习

- 根据如下代码，设计测试用例

```
1 def test():
2     for i in range(1,10):
3         for j in range(1,i+1):
4             print("%d*%d=%2d" % (j,i,i * j),end=" ")
5         print("")
6 test()
```

目录

循环测试总结

- 循环结构是程序中重要结构，必须重点测试
- 单个循环节点测试：
  - 边界，初始条件，最大条件，中间循环值，循环执行过程，变量变化，涉及变量的变化
- 串联循环结构：
  - 非关联循环节点：按单个循环节点依次测试；
  - 关联循环节点：结合测试

内容总结

- 循环测试背景知识
  - 为什么进行循环测试
    - 循环测试是程序结构中非常重要的一种结构
  - 循环结构分析：单循环节点、循环节点串联、循环节点嵌套
  - 不同的循环结构分别如何测试
    - 单循环节点
    - 串联循环节点
    - 嵌套循环

Question