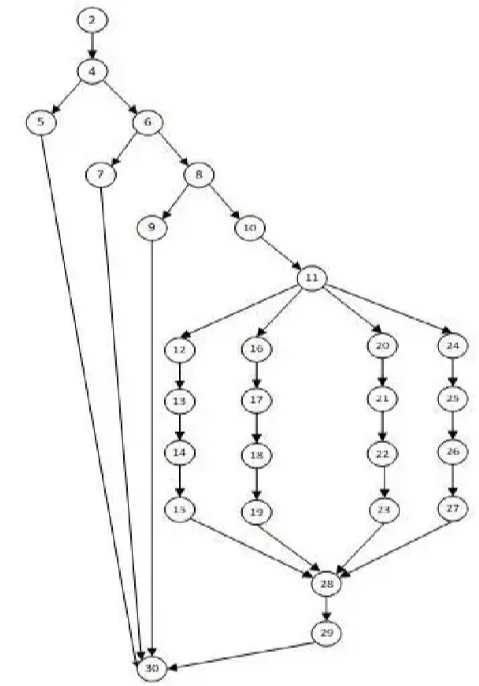
白盒测试-第1题

#### **伪代码：**

参数说明：work\_task——[int 类型,工作完成情况的得分], work\_capacity——[int 类型,工作能力的得分], work\_attitude——[int 类型,工作态度的得分], work\_total——[int 类型,员工总得分], work\_bonus——[int 类型(也可为 float 类型),员工的年终奖], rate——[String 类型,员工等级]。

1. int work\_total, work\_bonus, Total=20000;
2. String rate;
3. if(work\_task<0|| work\_task>100)
4. return "工作完成情况的输入有误";
5. if(work\_capacity<0|| work\_capacity>20)
6. return "工作能力的输入有误";
7. if(work\_attitude<0|| work\_attitude>30)
8. return "工作态度的输入有误";
9. work\_total=(int)(work\_task\*50%+ work\_capacity+ work\_attitude);
10. switch(work\_total){
11. Case 1: 85 <= work\_total <= 100:
12. work\_bonus = Total\*(1+10%);
13. rate="A";
14. break;
15. Case 2: 60 <= work\_total < 85:
16. work\_bonus = Total;
17. rate="B";
18. break;
19. Case 3: 40 <= work\_total < 60:
20. work\_bonus = Total\*(1-20%);
21. rate="C";
22. break;
23. Case 4: 0 < work\_total < 40:
24. work\_bonus = 0;
25. rate="D";
26. break;
27. }
28. return "该员工等级为" + rate + ",年终奖为" + work\_bonus;

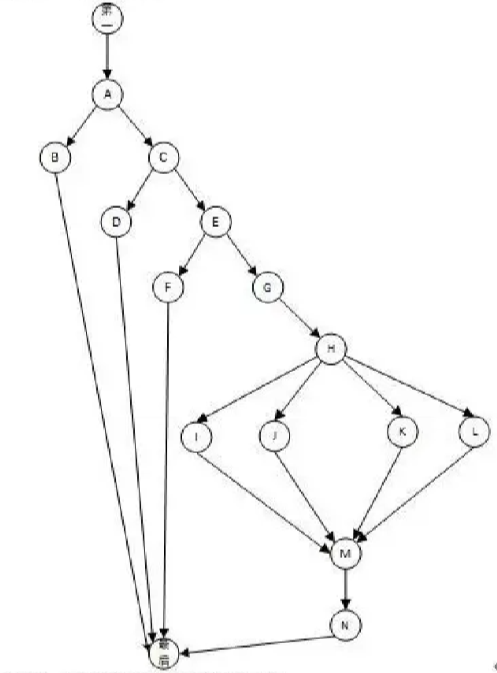
#### **程序图：**



#### **计算年终奖程序的DD路径：**

|  |  |  |
| --- | --- | --- |
| **程序图节点** | **DD路径名称** | **定义情况** |
| 2 | 第一 | 1 |
| 4 | A | 3 |
| 5 | B | 4 |
| 6 | C | 3 |
| 7 | D | 4 |
| 8 | E | 3 |
| 9 | F | 4 |
| 10 | G | 4 |
| 11 | H | 3 |
| 12-15 | I | 5 |
| 16-19 | J | 5 |
| 20-23 | K | 5 |
| 24-27 | L | 5 |
| 28 | M | 3 |
| 29 | N | 4 |
| 30 | 最后 | 2 |

#### **DD路径图：**



#### **计算年终奖程序的可执行路径：**

|  |  |
| --- | --- |
| **编号** | **可行路径** |
| P1 | 第一 - A - B - 最后 |
| P2 | 第一 - A - C - D - 最后 |
| P3 | 第一 - A - C - E - F - 最后 |
| P4 | 第一 - A - C - E - G - H - I - M - N - 最后 |
| P5 | 第一 - A - C - E - G - H - J - M - N - 最后 |
| P6 | 第一 - A - C - E - G - H - K - M - N - 最后 |
| P7 | 第一 - A - C - E - G - H - L - M - N - 最后 |

#### **语句覆盖测试用例**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **输入** | | | **通过路径** |
| **work\_task** | **work\_capacity** | **work\_attitude** |
| 1 | 110 | 20 | 15 | P1 |
| 2 | 80 | 25 | 15 | P2 |
| 3 | 80 | 15 | -1 | P3 |
| 4 | 95 | 20 | 15 | P4 |
| 5 | 80 | 15 | 10 | P5 |
| 6 | 60 | 15 | 5 | P6 |
| 7 | 40 | 10 | 5 | P7 |

#### **判定覆盖测试用例**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **work\_task** | **work\_capacity** | **work\_attitude** | **通过路径** | **判定取值** |
| 1 | 110 | 20 | 15 | P1 | R1 L2 L3→ |
| 2 | 80 | 25 | 15 | P2 | L1 R2 L3→ |
| 3 | 80 | 15 | -1 | P3 | L1 L2 R3→ |
| 4 | 95 | 20 | 15 | P4 | L1 L2 L3 S1 |
| 5 | 80 | 15 | 10 | P5 | L1 L2 L3 S2 |
| 6 | 60 | 15 | 5 | P6 | L1 L2 L3 S3 |
| 7 | 40 | 10 | 5 | P7 | L1 L2 L3 S4 |

#### **条件覆盖**

整个程序的条件整理：

(1) 语句4 work\_task<0||work\_task>100：

* 条件 work\_task<0 取真为 T1，取假为 F1；
* 条件 work\_task>100 取真为 T2，取假为 F2；

(2) 语句6 work\_capacity<0||work\_capacity>20：

* 条件 work\_capacity<0 取真为 T3，取假为 F3；
* 条件 work\_capacity>20 取真为 T4，取假为 F4；

(3) 语句8 work\_attitude<0||work\_attitude>30：

* 条件 work\_attitude<0 取真为 T5，取假为 F5；
* 条件 work\_attitude>30 取真为 T6，取假为 F6；

(4) 语句11 switch(work\_total)，共有4个条件：S1：85 <= work\_total <= 100，S2：60 <= work\_total < 85，S3：40 <= work\_total < 60，S4：0 < age < 40。

则满足条件覆盖的测试用例如表5所示：

**计算年终奖程序的条件覆盖测试用例**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **work\_task** | **work\_capacity** | **work\_attitude** | **通过路径** | **条件取值** |
| 1 | 110 | 20 | 15 | P1 | T2 F3 F5 |
| 2 | 80 | 25 | 15 | P2 | F1 T4 F5 |
| 3 | 80 | 15 | -1 | P3 | F1 F3 T5 |
| 4 | 80 | 15 | 35 | P3 | F1 F3 T6 |
| 5 | 80 | 15 | 15 | P2 | F1 F3 F5 |
| 6 | 80 | 15 | -1 | P3 | F1 F3 T5 |
| 7 | 95 | 20 | 15 | P4 | F1 F3 F5 S1 |
| 8 | 80 | 15 | 10 | P5 | F1 F3 F5 S2 |
| 9 | 60 | 15 | 10 | P6 | F1 F3 F5 S3 |
| 10 | 40 | 10 | 5 | P7 | F1 F3 F5 S4 |

#### 判定-条件覆盖

判定覆盖和条件覆盖的综合，条件覆盖的测试用例满足判定-条件覆盖。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **work\_task** | **work\_capacity** | **work\_attitude** | **通过路径** | **条件取值** |
| 1 | 110 | 20 | 15 | P1 | T2 F3 F5 |
| 2 | 80 | 25 | 15 | P2 | F1 T4 F5 |
| 3 | 80 | 15 | -1 | P3 | F1 F3 T5 |
| 4 | 80 | 15 | 35 | P3 | F1 F3 T6 |
| 5 | 80 | 15 | 15 | P2 | F1 F3 F5 |
| 6 | 80 | 15 | -1 | P3 | F1 F3 T5 |
| 7 | 95 | 20 | 15 | P4 | F1 F3 F5 S1 |
| 8 | 80 | 15 | 10 | P5 | F1 F3 F5 S2 |
| 9 | 60 | 15 | 10 | P6 | F1 F3 F5 S3 |
| 10 | 40 | 10 | 5 | P7 | F1 F3 F5 S4 |

#### 条件组合覆盖

条件组合覆盖需要使每个判断的所有可能的条件取值组合至少执行一次。根据程序的实际情况，得到如下的取值组合：

* **work\_task = -1** 作 T1F2
* **work\_task = 110** 作 F1T2
* **work\_task = 80** 作 F1F2
* **work\_capacity = -1** 作 T3F4
* **work\_capacity = 25** 作 F3T4
* **work\_capacity = 15** 作 F3F4
* **work\_attitude = -1** 作 T5F6
* **work\_attitude = 35** 作 F5T6
* **work\_attitude = 25** 作 F5F6

语句8 **switch(work\_total)**，共有4个条件：S1: 85 <= **work\_total** <= 100，S2: 60 <= **work\_total** < 85，S3: 40 <= **work\_total** < 60，S4: 0 <= **work\_total** < 40 取各自的条件即可。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **编号** | **work\_task** | **work\_capacity** | **work\_attitude** | **通过路径** | **条件取值** |
| 1 | 110 | 20 | 15 | P1 | T2 F3 F5 |
| 2 | 80 | 25 | 15 | P2 | F1 T4 F5 |
| 3 | 80 | 15 | -1 | P3 | F1 F3 T5 |
| 4 | 80 | 15 | 35 | P3 | F1 F3 T6 |
| 5 | 80 | 15 | 15 | P2 | F1 F3 F5 |
| 6 | 80 | 15 | -1 | P3 | F1 F3 T5 |
| 7 | 95 | 20 | 15 | P4 | F1 F3 F5 S1 |
| 8 | 80 | 15 | 10 | P5 | F1 F3 F5 S2 |
| 9 | 60 | 15 | 10 | P6 | F1 F3 F5 S3 |
| 10 | 40 | 10 | 5 | P7 | F1 F3 F5 S4 |

#### 路径覆盖

由于年终奖程序中不存在循环，共存在7条路径。语句覆盖的测试用例满足路径覆盖。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **输入** | | | **通过路径** |
| **work\_task** | **work\_capacity** | **work\_attitude** |
| 1 | 110 | 20 | 15 | P1 |
| 2 | 80 | 25 | 15 | P2 |
| 3 | 80 | 15 | -1 | P3 |
| 4 | 95 | 20 | 15 | P4 |
| 5 | 80 | 15 | 10 | P5 |
| 6 | 60 | 15 | 5 | P6 |
| 7 | 40 | 10 | 5 | P7 |