

TED UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

SENG 453

SENG453/SENG311

Software Quality Assurance

Assignment 2 Report

Basis-path testing

Hilal Yurtoğlu

17386054060

**Table of Content**

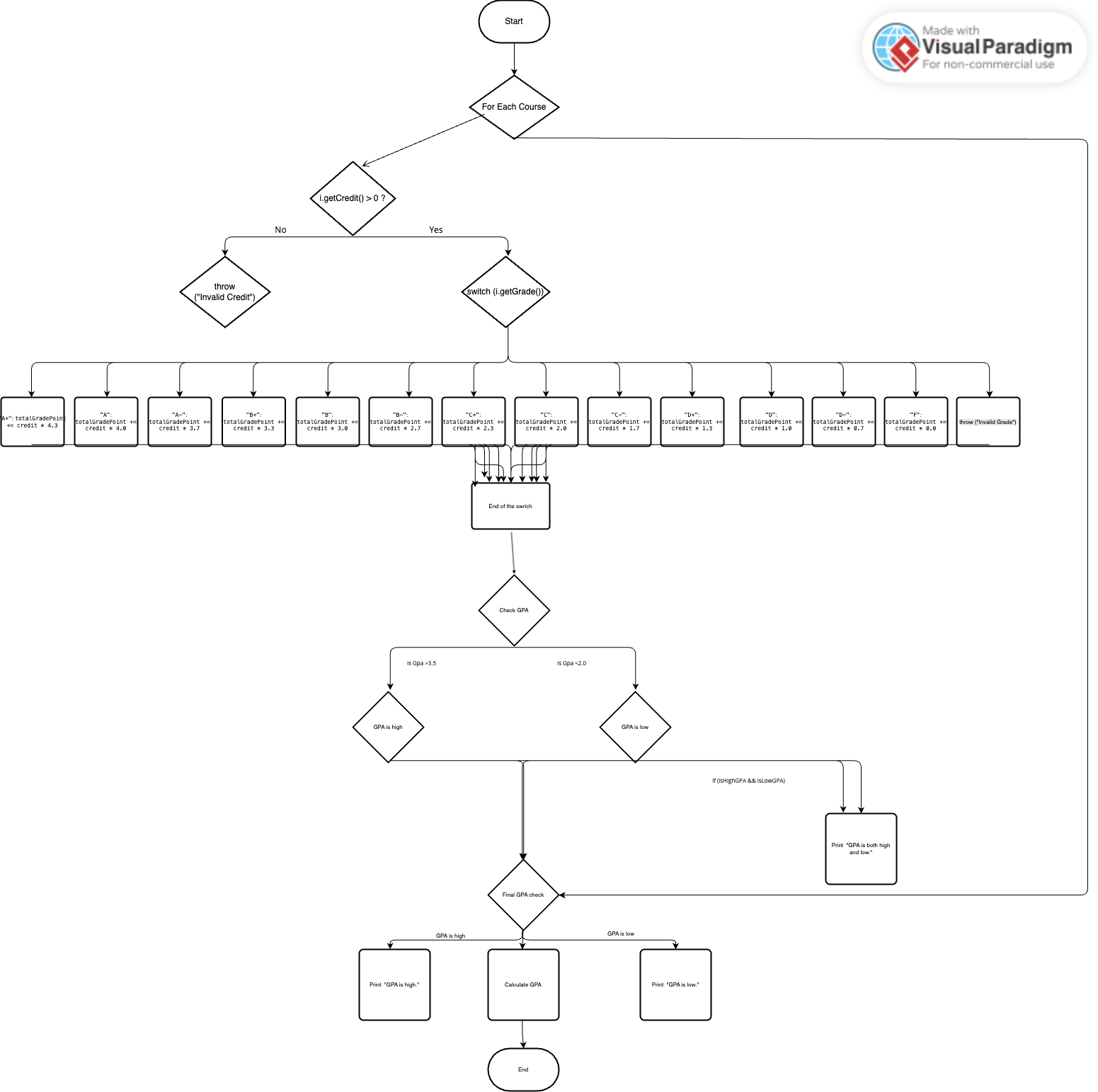
[Flowchart 2](#_Toc194342858)

[Basis Path 2](#_Toc194342859)

[Mutant Codes 2](#_Toc194342860)

[Table-based Test Scenario 2](#_Toc194342861)

# Flowchart



Estimated Cyclomatic Complexity: V(G) = E - N + 2P

47-27+2=22

# Basis Path

public double calculateGPA(ArrayList<Course> courseList)  
{  
 int totalCredit = 0;  
 double totalGradePoint = 0.0;  
 double gpa;  
  
 boolean isHighGPA = false;  
 boolean isLowGPA = false;  
  
 for(Course i: courseList) {  
 if (i.getCredit() > 0) {  
 totalCredit += i.getCredit();  
 switch (i.getGrade()) {  
 case "A+":  
 totalGradePoint += i.getCredit() \* 4.3;  
 break;  
 case "A":  
 totalGradePoint += i.getCredit() \* 4;  
 break;  
 case "A-":  
 totalGradePoint += i.getCredit() \* 3.7;  
 break;  
 case "B+":  
 totalGradePoint += i.getCredit() \* 3.3;  
 break;  
 case "B":  
 totalGradePoint += i.getCredit() \* 3;  
 break;  
 case "B-":  
 totalGradePoint += i.getCredit() \* 2.7;  
 break;  
 case "C+":  
 totalGradePoint += i.getCredit() \* 2.3;  
 break;  
 case "C":  
 totalGradePoint += i.getCredit() \* 2;  
 break;  
 case "C-":  
 totalGradePoint += i.getCredit() \* 1.7;  
 break;  
 case "D+":  
 totalGradePoint += i.getCredit() \* 1.3;  
 break;  
 case "D":  
 totalGradePoint += i.getCredit() \* 1;  
 break;  
 case "D-":  
 totalGradePoint += i.getCredit() \* 0.7;  
 break;  
 case "F":  
 totalGradePoint += i.getCredit() \* 0;  
 break;  
 default:  
 throw new IllegalArgumentException("Invalid Grade");  
 }  
  
 if (totalGradePoint / totalCredit > 3.5) {  
 isHighGPA = true;  
 } else if (totalGradePoint / totalCredit < 2.0) {  
 isLowGPA = true;  
 }  
 }  
 else  
 {  
 throw new IllegalArgumentException("Invalid Credit");  
 }  
 }  
  
 if (isHighGPA && isLowGPA) {  
 System.*out*.println("GPA is both high and low.");  
 } else if (isHighGPA) {  
 System.*out*.println("GPA is high.");  
 } else if (isLowGPA) {  
 System.*out*.println("GPA is low.");  
 }  
  
 gpa = totalGradePoint / totalCredit;  
 return gpa;  
}

In this assignment, a flowchart was created for the calculateGPA(ArrayList<Course> courseList) method in the GPACalculator.java class and the basic paths were determined.

Cyclomatic Complexity:

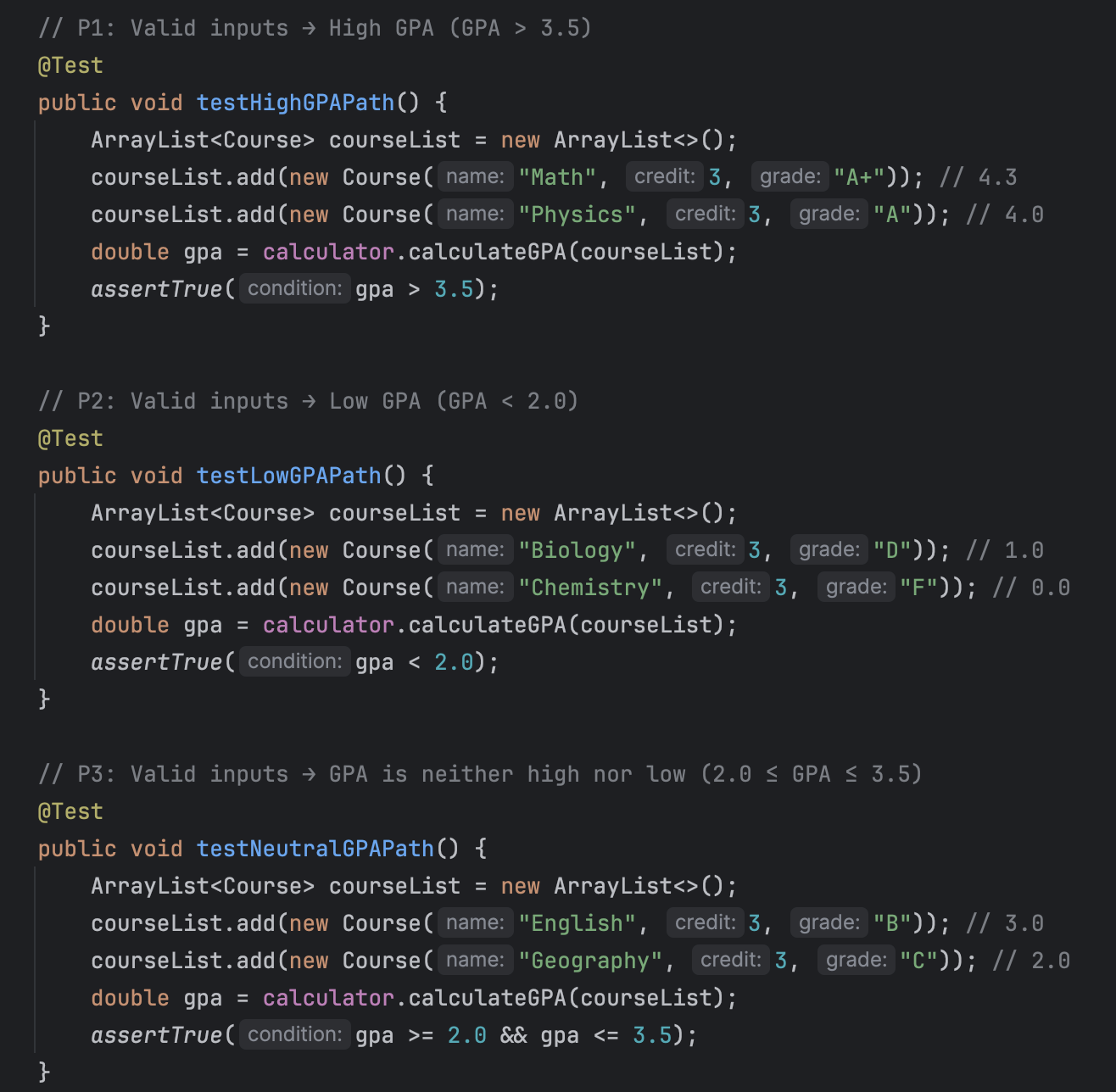
A for loop, multiple if and switch-case structures are used in the calculateGPA() method.

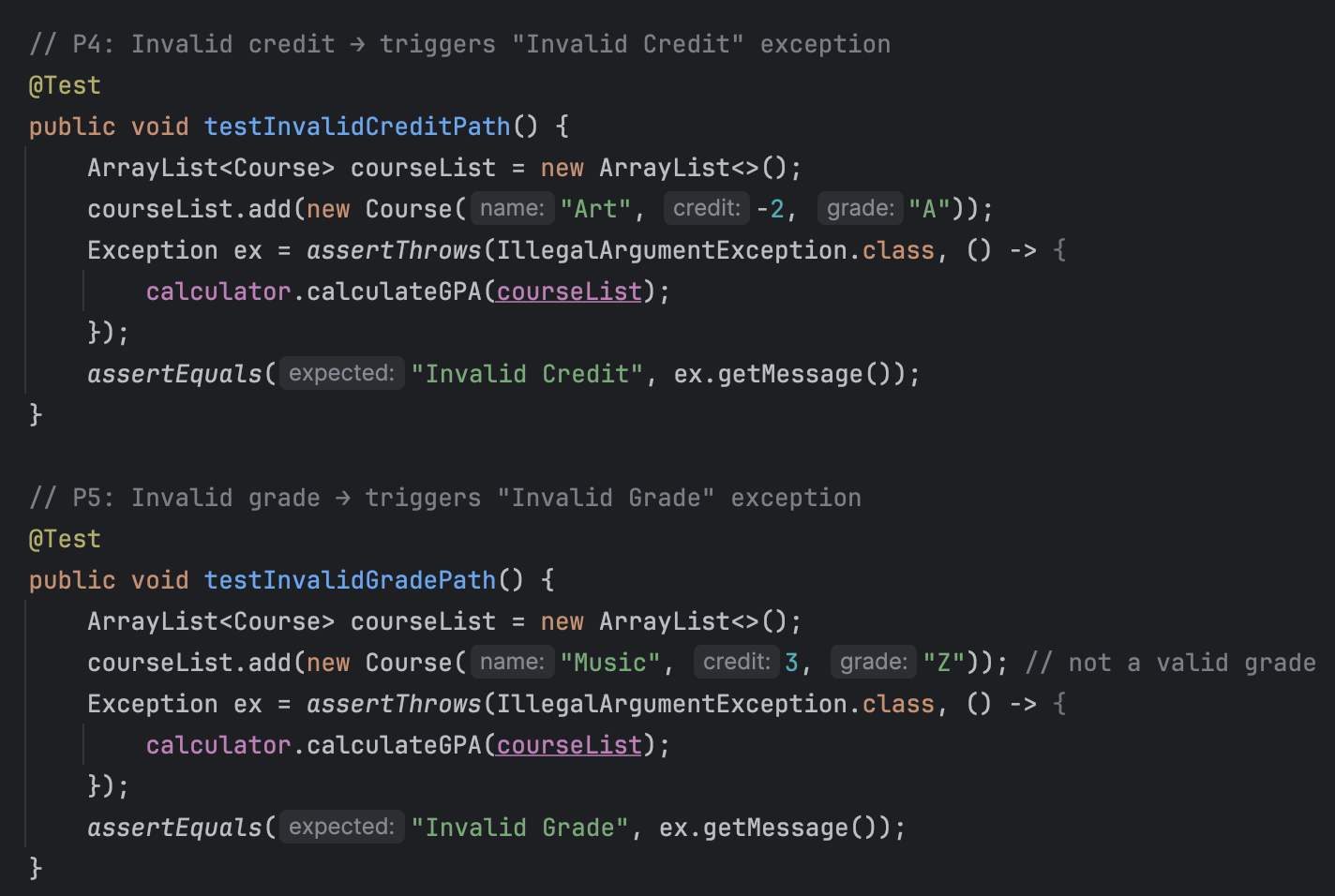
The cyclomatic complexity (CC) calculated by McCabe's formula is ≥ 4, and this method is suitable for basis path testing.

|  |  |
| --- | --- |
| Path | Explanation |
| P1 | All courses are valid and GPA ≥ 3.5 → “GPA is high” is printed |
| P2 | All courses are valid and GPA < 2.0 → “GPA is low” is printed |
| P3 | GPA exceeds both high and low limits → “both” status |
| P4 | Exception is thrown when invalid grade (like “X”) is entered |
| P5 | Exception is thrown when invalid credit (0 or negative) is entered |

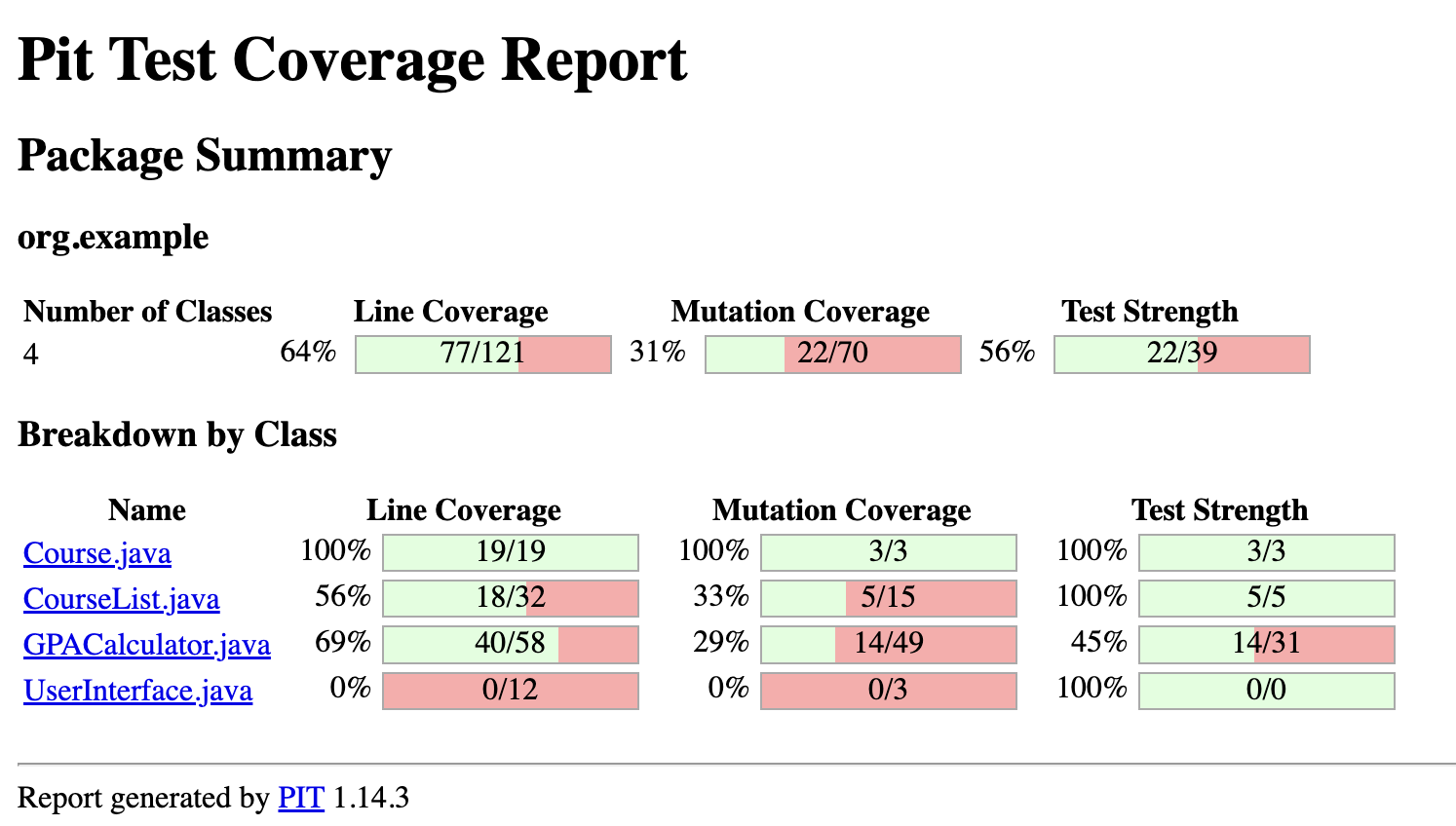
**Tests:**

For each of these paths, white-box tests were written using JUnit 5. Example:





# Mutant Codes

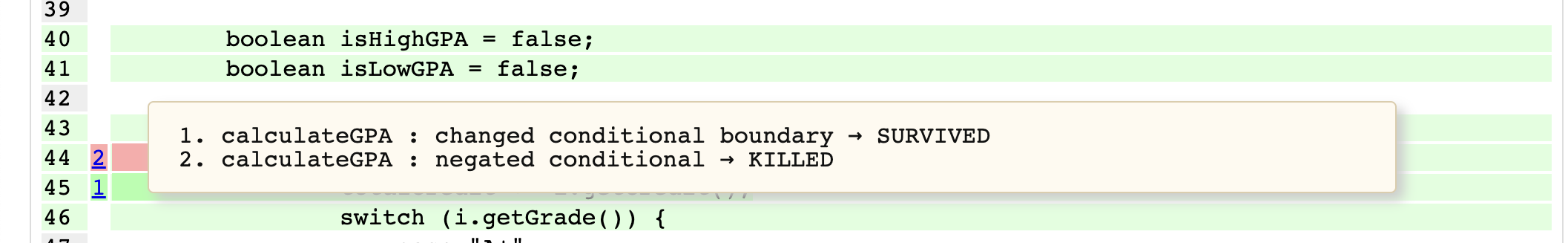


Mutation testing is a white-box testing technique used to measure the effectiveness of tests.

Small changes (mutants) are intentionally made to the code. The goal is to measure whether the tests can catch these errors.

If the tests detect the error, the mutant is killed.

If they cannot detect it, it survives.



# 

# Table-based Test Scenario

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Condition and Actions** | **TC1** | **TC2** | **TC3** | **TC4** | **TC5** |
| c1: Are the courses valid for credit? | T | T | T | T | **F** |
| c2: Are the grades valid? | T | T | T | **F** | T |
| c3: GPA > 3.5? | T | F | F | - | - |
| c4: GPA < 2.0? | F | T | F | - | - |
| a1: print "GPA is high" | X | - | - | - | - |
| a2: print "GPA is low" | - | X | - | - | - |
| a3: "GPA is both high and low" | - | - | X | - | - |
| a4: Exception: Invalid Grade | - | - | - | X | - |
| a5: Exception: Invalid Credit | - | - | - | - | X |

|  |  |
| --- | --- |
| **Test Case** | **Explanation** |
| **TC1** | GPA > 3.5, all credits and grades are valid → "GPA is high" is written |
| **TC2** | GPA < 2.0, all values ​​are valid → "GPA is low" is written |
| **TC3** | Some classes are too high, some are too low → GPA mixed → both |
| **TC4** | Invalid letter grade (example: "X") → Invalid Grade is thrown |
| **TC5** | Credit value is 0 or negative → Invalid Credit is thrown |