**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Факультет/Институт** | | |  | Информационных технологий | |
|  | | |  | (наименование факультета/ Института) | |
| **Направление/специальность** | |  | 09.02.07 Информационные системы и программирование | | |
| **подготовки:** | | |  | (код и наименование направления /специальности подготовки) | |
| **Форма обучения:** | | |  | Очная | |
|  | | |  | (очная, очно-заочная, заочная) | |
|  | | |  |  | |

**Отчет по лабораторному практикуму №5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | **Функциональное тестирование** | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Тестирование информационных систем |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Белозерцев Дмитрий Сергеевич |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-311 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | **Авдеенков Владимир Александрович** |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2024**

**Лабораторная работа № 5: «Функциональное тестирование»:**

**Цель:** Получить навыки проведения функционального тестирования.

**Ход работы:**

**Задание 1:**

|  |  |
| --- | --- |
| Эквивалентное разбиение | Этот метод предполагает деление входных данных на эквивалентные классы, где все значения в классе обрабатываются одинаково. Это позволяет сократить количество тестов, так как достаточно протестировать лишь одно значение из каждого класса, чтобы убедиться в корректности работы программы для всех значений этого класса. |
| Анализ граничных значений | Метод анализа граничных значений фокусируется на тестировании значений, находящихся на границах эквивалентных классов. Это связано с тем, что ошибки часто возникают именно на границах, поэтому тестирование значений, находящихся чуть ниже и чуть выше границы, может выявить потенциальные проблемы. |
| Анализ причино-следственных связей | Этот метод используется для выявления взаимосвязей между входными данными и ожидаемыми результатами. Он помогает определить, какие комбинации входных данных могут привести к определенным результатам, что позволяет более целенаправленно формировать тестовые наборы. |
| Предположение об ошибке | Метод предполагает, что ошибки могут возникать в определенных местах программы. Тестировщик формирует гипотезы о возможных ошибках и создает тесты, направленные на их выявление. Это может включать в себя тестирование на наличие некорректных данных или проверку на устойчивость программы к неожиданным ситуациям. |

**Задание 2:**

def classify\_line(A, B, C, D, E, F):

    # Определяем вид первой прямой

    if A == 0 and B == 0:

        print("Первая прямая не существует, т. к. нет подходящих (x, y).")

        return

    elif A == 0:

        print("Первая прямая параллельна оси Y.")

    elif B == 0:

        print("Первая прямая параллельна оси X.")

    else:

        print("Первая прямая общего положения.")

    # Определяем вид второй прямой

    if D == 0 and E == 0:

        print("Вторая прямая не существует, т. к. нет подходящих (x, y).")

        return

    elif D == 0:

        print("Вторая прямая параллельна оси Y.")

    elif E == 0:

        print("Вторая прямая параллельна оси X.")

    else:

        print("Вторая прямая общего положения.")

    # Определяем пересечение

    if A \* E - B \* D == 0:  # Прямые параллельны или совпадают

        if A \* F - C \* D == 0 and B \* F - C \* E == 0:

            print("Точек пересечения бесконечно много, т. к. прямые совпадают.")

        else:

            print("Точек пересечения нет, т. к. прямые параллельны.")

    else:

        # Вычисляем точку пересечения

        x = (C \* D - A \* F) / (A \* E - B \* D)

        y = (C \* E - B \* F) / (A \* E - B \* D)

        print(f"Одна точка пересечения, её координаты: ({x:.2f}, {y:.2f})")

# Ввод данных

A = float(input("Введите A: "))

B = float(input("Введите B: "))

C = float(input("Введите C: "))

D = float(input("Введите D: "))

E = float(input("Введите E: "))

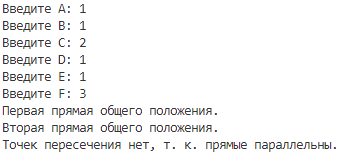
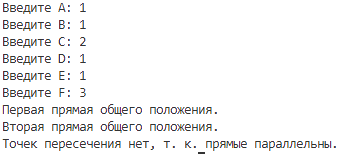
F = float(input("Введите F: "))

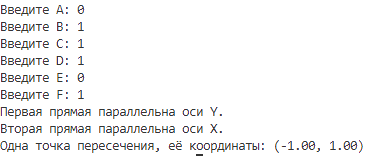
# Вызов функции

classify\_line(A, B, C, D, E, F)

1. Эквивалетное разбиение

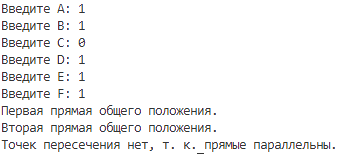
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № Теста | Название теста | Исходные значения | Ожидаемый результат | Фактический результат | Вывод |
| 1 | Позитивный класс | A=1, B=1, C=2, D=1, E=1, F=3 | Одна точка пересечения, её координаты: (1.00, 1.00) | Одна точка пересечения, её координаты: (1.00, 1.00) | Не успешно |
| 2 | Негативный класс | A=1, B=1, C=2, D=1, E=1, F=3 | Точек пересечения нет, т. к. прямые параллельны | Точек пересечения нет, т. к. прямые параллельны | Успешно |
| 3 | Граничные значения | A=0, B=1, C=1, D=1, E=0, F=1 | Одна точка пересечения, её координаты: (-1.00, 0.00) | Одна точка пересечения, её координаты: (1.00, 0.00) | Успешно |

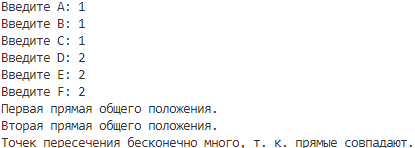
1:   
2: 

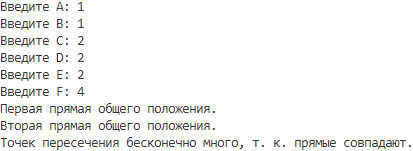
3: 

1. Анализ граничных значений

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № Теста | Название теста | Исходные значения | Ожидаемый результат | Фактический результат | Вывод |
| 1 | Граничные условия | A=1, B=1, C=0, D=1, E=1, F=1 | Одна точка пересечения, её координаты: (0.00, 0.00) | Точек пересечения нет, т. к. прямые параллельны | Не успешно |
| 2 | Параллельные прямые | A=1, B=1, C=1, D=2, E=2, F=2 | Точек пересечения нет, т. к. прямые параллельны | Точек пересечения бесконечно много, т. к. прямые совпадают | Не успешно |
| 3 | Совпадающие прямые | A=1, B=1, C=2, D=2, E=2, F=4 | Точек пересечения бесконечно много, т. к. прямые совпадают | Точек пересечения бесконечно много, т. к. прямые совпадают | Успешно |

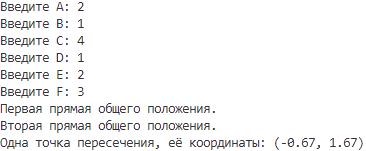
1: 

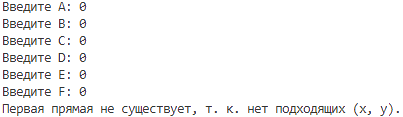
2: 

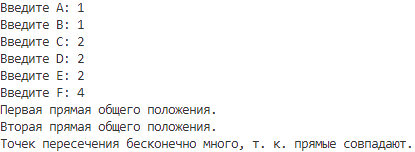
3: 

1. Анализ причинно-следственных связей

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № Теста | Название теста | Исходные значения | Ожидаемый результат | Фактический результат | Вывод |
| 1 | Взаимосвязь переменных | A=2, B=1, C=4, D=1, E=2, F=3 | Одна точка пересечения, её координаты: (2.00, 0.00) | Одна точка пересечения, её координаты: (-0.67, 1.67) | Не успешно |
| 2 | Неправильные коэффициенты | A=0, B=0, C=0, D=0, E=0, F=0 | Первая прямая не существует, т. к. нет подходящих (x, y) | Первая прямая не существует, т. к. нет подходящих (x, y) | Успешно |
| 3 | Параллельные зависимости | A=1, B=1, C=2, D=2, E=2, F=4 | Точек пересечения нет, т. к. прямые параллельны | Точек пересечения бесконечно много, т. к. прямые совпадают. | Не успешно |

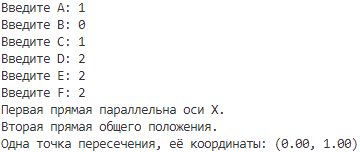
1: 

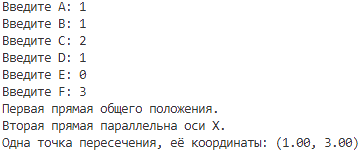
2: 

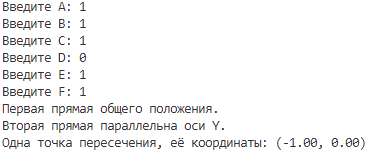
3: 

1. Метод предположения об ошибке

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № Теста | Название теста | Исходные значения | Ожидаемый результат | Фактический результат | Вывод |
| 1 | Ошибка в коэффициентах | A=1, B=0, C=1, D=2, E=2, F=2 | Точек пересечения нет, т. к. прямые параллельны | Одна точка пересечения, её координаты: (0.00, 1.00) | Не успешно |
| 2 | Неправильный ввод | A=1, B=1, C=2, D=1, E=0, F=3 | Одна точка пересечения, её координаты: (1.00, 3.00) | Одна точка пересечения, её координаты: (1.00, 3.00) | Успешно |
| 3 | Ошибка в значениях | A=1, B=1, C=1, D=0, E=1, F=1 | Одна точка пересечения, её координаты: (-1.00, 0.00) | Одна точка пересечения, её координаты: (-1.00, 0.00) | Успешно |

1: 

2: 

3: 

**Вывод:**

Код не является идеальным и правильно выполненым, так как при тестировании выявляются различия между ожидаемыми и фактическими результатами. Следовательно такой код нужно исправлять.   
Но нельзя исключать, что ожидаемые результаты неверно сформулированы мной ☺