



**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

Тестирование безопасности

(наименование темы)

по дисциплине

Тестирование информационных систем

(наименование дисциплины)

Обучающийся

Грачев Дмитрий Александрович

(ФИО)

(подпись)

Группа

ДКИП-312

Преподаватель

Авдеенков Владимир Александрович

(ФИО)

(подпись)

Лабораторная работа № 5: «Тестирование безопасности»:

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. Например, если система принимает возраст от 0 до 100 лет, можно выделить классы: отрицательные значения, значения от 0 до 100 и значения выше 100.
Анализ граничных значений	Метод анализа граничных значений основан на тестировании значений, находящихся на границах эквивалентных классов. Это связано с тем, что ошибки часто возникают именно на границах. Например, если допустимый диапазон значений составляет от 1 до 10, тесты должны включать значения 0, 1, 10 и 11.
Анализ причинно-следственных связей	Этот метод включает в себя выявление и анализ взаимосвязей между входными данными и ожидаемыми результатами. Он помогает определить, какие комбинации входных данных могут привести к

	<p>определенным результатам.</p> <p>Например, если система должна выдавать разные сообщения в зависимости от статуса пользователя и его действий, важно протестировать все возможные комбинации этих факторов.</p>
Предположение об ошибке	<p>Метод предположения об ошибке основывается на предположении, что ошибки могут возникать в определенных местах системы.</p> <p>Тестировщики могут использовать опыт и знания о предыдущих ошибках для создания тестов, которые направлены на наиболее уязвимые участки системы. Это позволяет более эффективно находить ошибки, основываясь на вероятности их возникновения.</p>

Задание №2:

```
def classify_line(A, B):
    if A == 0 and B == 0:
        return "не существует"
    elif A == 0:
        return "параллельно оси Y"
    elif B == 0:
        return "параллельно оси X"
    else:
        return "общего положения"

def intersection(A, B, C, D, E, F):
    # Проверка на существование прямых
    if A == 0 and B == 0:
        return "Система неразрешима, т. к. первая прямая не существует."
    if D == 0 and E == 0:
        return "Система неразрешима, т. к. вторая прямая не существует."

    # Определение детерминанта
    det = A * E - B * D

    if det == 0: # Прямые параллельны или совпадают
```

```

# Проверка на совпадение
if C * E == B * F and A * F == D * C:
    return "Точек пересечения бесконечно много, т. к. прямые совпадают."
else:
    return "Точек пересечения нет, т. к. прямые параллельны."
else: # Прямые пересекаются в одной точке
    x = (C * E - B * F) / det
    y = (A * F - C * D) / det
    return f"Одна точка пересечения: ({x}, {y})"

# Ввод коэффициентов
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: "))

# Определение вида каждой из прямых
line1_type = classify_line(A, B)
line2_type = classify_line(D, E)

print(f"Первая прямая: {line1_type}")
print(f"Вторая прямая: {line2_type}")

# Определение пересечения
result = intersection(A, B, C, D, E, F)
print(result)

```

1. Эквивалентные разбиения

№ Теста	Назначение теста	Исходное значение	Ожидаемый результат	Фактический результат	Вывод
1	Проверка существования прямой	A=0; B=0;	"не существует"	"не существует"	Успешно
2	Проверка параллельности с осью Y	A=0; B=1; C=5;	"параллельно оси Y"	"параллельно оси Y"	Успешно
3	Проверка параллельности с осью X	A=1; B=0; C=5;	"параллельно оси X"	"параллельно оси X"	Успешно
4	Проверка общего положения	A=1; B=1; C=5;	"общего положения"	"общего положения"	Успешно

Тест №1:

```
C:\Users\edu-msk22-813s\PycharmProjects\pythonProject1\venv>
Введите A: 0
Введите B: 0
Введите C: 5
Введите D: 4
Введите E: 5
Введите F: 8
Первая прямая: не существует
Вторая прямая: общего положения
Система неразрешима, т. к. первая прямая не существует.

Process finished with exit code 0
```

Тест №2:

```
C:\Users\edu-msk22-813s\PycharmProjects\pythonProject1\venv>
Введите A: 0
Введите B: 1
Введите C: 5
Введите D: 1
Введите E: 1
Введите F: 1
Первая прямая: параллельно оси Y
Вторая прямая: общего положения
Одна точка пересечения: (-4.0, 5.0)

Process finished with exit code 0
```

Тест №3:

```
C:\Users\edu-msk22-813s\PycharmProjects\pythonProject1\venv>
Введите A: 1
Введите B: 0
Введите C: 5
Введите D: 1
Введите E: 1
Введите F: 1
Первая прямая: параллельно оси X
Вторая прямая: общего положения
Одна точка пересечения: (5.0, -4.0)

Process finished with exit code 0
```

Тест №4:

```

C:\Users\edu-msk22-813s\PycharmProjects\pythonPr
Введите A: 1
Введите B: 1
Введите C: 5
Введите D: 1
Введите E: 1
Введите F: 1
Первая прямая: общего положения
Вторая прямая: общего положения
Точек пересечения нет, т. к. прямые параллельны.

Process finished with exit code 0

```

2. Анализ граничных условий

№ Теста	Назначение теста	Исходное значение	Ожидаемый результат	Фактический результат	Вывод
1	Проверка на совпадение прямых	A=1; B=2; C=3; D=2; E=4; F=6;	"Точек пересечения бесконечно много, т. к. прямые совпадают."	"Точек пересечения бесконечно много, т. к. прямые совпадают."	Успешно
2	Проверка на параллельность	A=1; B=2; C=3; D=2; E=4; F=5;	"Точек пересечения нет, т. к. прямые параллельны."	"Точек пересечения нет, т. к. прямые параллельны."	Успешно
3	Проверка на пересечение	A=1; B=1; C=2; D=1; E=-1; F=0;	"Одна точка пересечения: (1, 1)"	"Одна точка пересечения: (1, 1)"	Успешно

Тест №1:

```
C:\Users\edu-msk22-813s\PycharmProjects\pythonProject1\venv\
Введите A: 1
Введите B: 2
Введите C: 3
Введите D: 2
Введите E: 4
Введите F: 6
Первая прямая: общего положения
Вторая прямая: общего положения
Точек пересечения бесконечно много, т. к. прямые совпадают.

Process finished with exit code 0
```

Тест №2:

```
C:\Users\edu-msk22-813s\PycharmProjects\pythonPro
Введите A: 1
Введите B: 2
Введите C: 3
Введите D: 2
Введите E: 4
Введите F: 5
Первая прямая: общего положения
Вторая прямая: общего положения
Точек пересечения нет, т. к. прямые параллельны.

Process finished with exit code 0
```

Тест №3:

```
C:\Users\edu-msk22-813s\PycharmProje
Введите A: 1
Введите B: 1
Введите C: 2
Введите D: 1
Введите E: -1
Введите F: 0
Первая прямая: общего положения
Вторая прямая: общего положения
Одна точка пересечения: (1.0, 1.0)

Process finished with exit code 0
```

3. Анализ причинно-следственных связей

№ Теста	Назначение теста	Исходное значение	Ожидаемый результат	Фактический результат	Вывод
1	Проверка на не существование первой прямой	A=0; B=0;	"Система неразрешима, т. к. первая прямая не существует."	"Система неразрешима, т. к. первая прямая не существует."	Успешно
2	Проверка на не существование второй прямой	C=0; D=0;	"Система неразрешима, т. к. вторая прямая не существует."	"Система неразрешима, т. к. вторая прямая не существует."	Успешно

Тест №1:

```

C:\Users\edu-msk22-813s\PycharmProjects\pythonProject1\venv\Scripts>python test1.py
Введите A: 0
Введите B: 0
Введите C: 1
Введите D: 1
Введите E: 1
Введите F: 1
Первая прямая: не существует
Вторая прямая: общего положения
Система неразрешима, т. к. первая прямая не существует.

Process finished with exit code 0

```

Тест №2:


```

C:\Users\edu-msk22-813s\PycharmProj
Введите A: 1
Введите B: 1
Введите C: 0
Введите D: 0
Введите E: 1
Введите F: 1
Первая прямая: общего положения
Вторая прямая: параллельно оси Y
Одна точка пересечения: (-1.0, 1.0)

Process finished with exit code 0

```

4. Метод предположения об ошибке

№ Теста	Назначение теста	Исходное значение	Ожидаемый результат	Фактический результат	Вывод
1	Проверка на случай, когда обе прямые совпадают	A=1; B=2; C=3; D=2; E=4; F=6;	"Точек пересечения бесконечно много, т. к. прямые совпадают."	"Точек пересечения бесконечно много, т. к. прямые совпадают."	Успешно
2	Проверка на случай, когда прямые пересекаются	A=1; B=1; C=2; D=1; E=-1; F=0;	"Одна точка пересечения: (1, 1)"	"Одна точка пересечения: (1, 1)"	Успешно

Тест №1:

```
C:\Users\edu-msk22-813s\PycharmProjects\pythonProject1\venv
Введите A: 1
Введите B: 2
Введите C: 3
Введите D: 2
Введите E: 4
Введите F: 6
Первая прямая: общего положения
Вторая прямая: общего положения
Точек пересечения бесконечно много, т. к. прямые совпадают.

Process finished with exit code 0
```

Тест №2:

```
C:\Users\edu-msk22-813s\PycharmProj
Введите A: 1
Введите B: 1
Введите C: 2
Введите D: 1
Введите E: -1
Введите F: 0
Первая прямая: общего положения
Вторая прямая: общего положения
Одна точка пересечения: (1.0, 1.0)

Process finished with exit code 0
```