**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**"Национальный исследовательский университет**

**"Высшая школа экономики"**

Московский институт электроники и математики им. А. Н. Тихонова НИУ ВШЭ

Департамент компьютерной инженерии

Курс: Алгоритмизация и программирование

**Документация по тестированию**

Студент: Попцова Дарья Алексеевна

Группа: БИВ243

Москва 2025

Оглавление

[1. План тестирования 4](#_Toc194265918)

[2. Тест-кейсы 4](#_Toc194265919)

[3. Чек-лист 5](#_Toc194265920)

[4. Баг-репорт 5](#_Toc194265921)

[5. Отчёт о тестировании 6](#_Toc194265922)

[6. Инструкция 6](#_Toc194265923)

# 1. План тестирования

**1.1 Цель**

Обеспечить корректность работы консольного C++‑приложения, рассчитывающего минимальную комиссию перевода между банками с учётом:

* комиссионных правил (Fixed Fee + Percent),
* ограничений на число посредников (≤ 1),
* равномерного разбиения суммы на k ∈ [1;5] частей.

**1.2 Объекты тестирования**

|  |  |
| --- | --- |
| **Модуль** | **Проверяемые аспекты** |
| **Парсер CSV** | корректность разбора кавычек, пропуск пустых строк, сообщения об ошибках |
| **Построение графа** | правила наличия ребра («пересечение корреспондентов»), пересчёт весов при разном *amount* |
| **Алгоритм Дейкстры (ограниченный)** | поиск кратчайшего пути ≤ 2 рёбер, восстановление маршрута |
| **Оптимизатор разбиения** | цикл по \* k\*, выбор наименьшей общей комиссии |
| **CLI‑интерфейс** | валидация ввода, обработка ошибок, вывод результатов |

**1.3 Подход**

* **Функциональное тестирование** – ручное и автоматическое (Catch2) по заранее описанным сценариям.
* **Тестирование граничных значений** – суммы ровно в точке порога, 0 ₽, очень большие суммы.
* **Модульные тесты** для splitCsvLine, loadBanks, CommissionTable::inFee/outFee.
* **Системные интеграционные** – полный цикл ввода‑обработки‑вывода.

Среда: Windows 10 / Ubuntu 22.04, g++ 9.4.0, ручное тестирование.

Ресурсы: 1 тестировщик, лог-файлы, CSV.

# 2. Тест-кейсы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Название** | **Предусловие** | **Шаги** | **Ожидаемый результат** |
| **TC‑01** | Прямой перевод ≤ порога | BankA → BankB, сумма 500 ₽ | 1. Запустить приложение.2. В поле Source ввести **BankA**.3. В поле Destination ввести **BankB**.4. Ввести сумму **500**.5. Нажать **Enter** и дождаться результата. | Split = 1; комиссия 27 ₽; маршрут A → B |
| **TC‑02** | Прямой перевод с выгодным split | BankA → BankB, 1 500 ₽ | 1‑4. Повторить шаги TC‑01, но ввести сумму **1 500**.5. Зафиксировать вывод. | Split = 2; комиссия 54 ₽; маршрут A → B |
| **TC‑03** | Прямой перевод через крупную сумму | BankM → BankE, 1 500 ₽ | 1. Запустить приложение.2. Ввести **BankM** как отправителя.3. Ввести **BankE** как получателя.4. Ввести **1 500**.5. Подтвердить. | Split = 2; комиссия 34 ₽; маршрут M → E |
| **TC‑04** | Один посредник + split | BankB → BankE, 1 500 ₽ | 1. Запустить приложение.2. Source — **BankB**.3. Destination — **BankE**.4. Сумма **1 500**.5. Подтвердить. | Split = 2; комиссия 68 ₽; маршрут B → H → E |
| **TC‑05** | Нет маршрута ≤ 1 посредника | BankJ → BankL, 1 500 ₽ | 1‑4. Аналогично TC‑04, но банки **BankJ**, **BankL**.5. Подтвердить. |  |
| **TC‑06** | Граничная сумма ровно порог | BankA → BankB, 1 000 ₽ | 1‑3. Аналогично TC‑01.4. Ввести сумму **1 000**.5. Подтвердить. | Используется правило Threshold = 1000 |
| **TC‑07** | Сумма > 1е12 | BankC → BankD, 2e12 ₽ | 1. Запустить приложение.2. Source **BankC**.3. Destination **BankD**.4. Сумма **2000000000000**.5. Подтвердить. | Применяется последний тариф (1e12) |
| **TC‑08** | Неверное имя банка | XYZ → BankA, 100 ₽ | 1. Запустить приложение.2. Source — **XYZ**.3. Destination — **BankA**.4. Сумма **100**.5. Подтвердить. | Ошибка «Bank not found» |
| **TC‑09** | Нулевая сумма | BankA → BankB, 0 ₽ | 1‑3. Аналогично TC‑01.4. Ввести сумму **0**.5. Подтвердить. | Сообщение о недопустимой сумме |
| **TC‑10** | Split = 5 максимальный | BankA → BankC, 2 500 ₽ | 1. Запустить приложение.2. Source **BankA**.3. Destination **BankC**.4. Сумма **2 500**.5. Подтвердить. | Проверить корректность 5‑кратного цикла |

# 3. Чек-лист

Программа компилируется без ошибок

CSV-файлы загружаются корректно

Прямой маршрут отображается правильно

Маршрут через посредника отображается правильно

Маршрут с разбиение на части отображается правильно

Комиссия рассчитывается

Ввод суммы работает

Граничные значения комиссий обрабатываются

Ошибка при вводе некорректного банка

Ошибка при пустой строке в CSV

Обработка нечисловых значений

# 4. Баг-репорт

ID: BUG‑2025‑001

Описание: При отсутствии файла Borders.csv программа не сообщает об ошибке, но продолжает выполнение с неверной логикой расчёта.

Важность: Высокая.

Приоритет: Средний.

Условия: Ubuntu 22.04, g++ 9.4.0, версия MVP 1.0

Шаги воспроизведения:

|  |
| --- |
| 1. Запустить программу 2. Ввести *BankA*, *BankB*, *2000* |

Ожидаемый результат: Split = 2, комиссия 54

Фактический результат: Optimal split: 2 parts of 1000 RUB each

Total commission: 54 RUB

Path: BankA -> BankB

# 5. Отчёт о тестировании

Тестировщик: Попцова Дарья Алексеевна

Покрытие: 10 тест-кейсов, охвачено 100% ключевых функций

Методы: ручное тестирование, анализ вывода

Результаты: Пройдены 10/10

# 6. Инструкция

1. Убедитесь, что файлы main.cpp, Banks.csv, Commissions.csv находятся в одной директории.

2. Скомпилируйте программу: g++ -std=c++17 main.cpp -o transfer

3. Запустите программу: ./transfer

4. Введите имена банков и сумму

5. Проверьте, корректно ли рассчитан маршрут и комиссия