

Министерство образования и науки Российской Федерации Федеральное
государственное бюджетное образовательное учреждение высшего образования
Петрозаводский государственный университет Институт математики и
информационных технологий Кафедра информатики и математического
обеспечения

Отчет по дисциплине «Верификация ПО»

Выполнил: студент группы 22407, Угрюмов М.А.

Преподаватель: к.ф-м.н., доцент К. А. Кулаков

Петрозаводск, 2022

Объект тестирования	3
Описание приложения	3
Функциональные требования	3
Архитектура	3
Функции приложения	4
Стратегия тестирования	5
Блочное тестирование	5
Интеграционное тестирование	5
Аттестационное тестирование	5
Критерии прохождения тестирования	5
Критерии остановки тестирования	5
Критерии возобновления тестирования	5
Перечень методов для тестирования	6
Тестирование	7
Блочное тестирование	7
Модуль записи	7
Модуль вычислений	9
Модуль вычислений	10
Интеграционное тестирование	13
Аттестационное тестирование	13
Пример тестов	15
Журнал тестирования	16
Блочное тестирование	16
Интеграционное тестирование	17
Аттестационное тестирование	17
Журнал найденных ошибок Отчет об ошибке №1	18
Отчет об ошибке №2	18
Отчет об ошибке №3	18
Отчет об ошибке №4	19
Отчет об ошибке №5	19
Отчет об ошибке №6	20
Отчет об ошибке №7	20
Отчет об ошибке №8	20
Результат	21

Объект тестирования

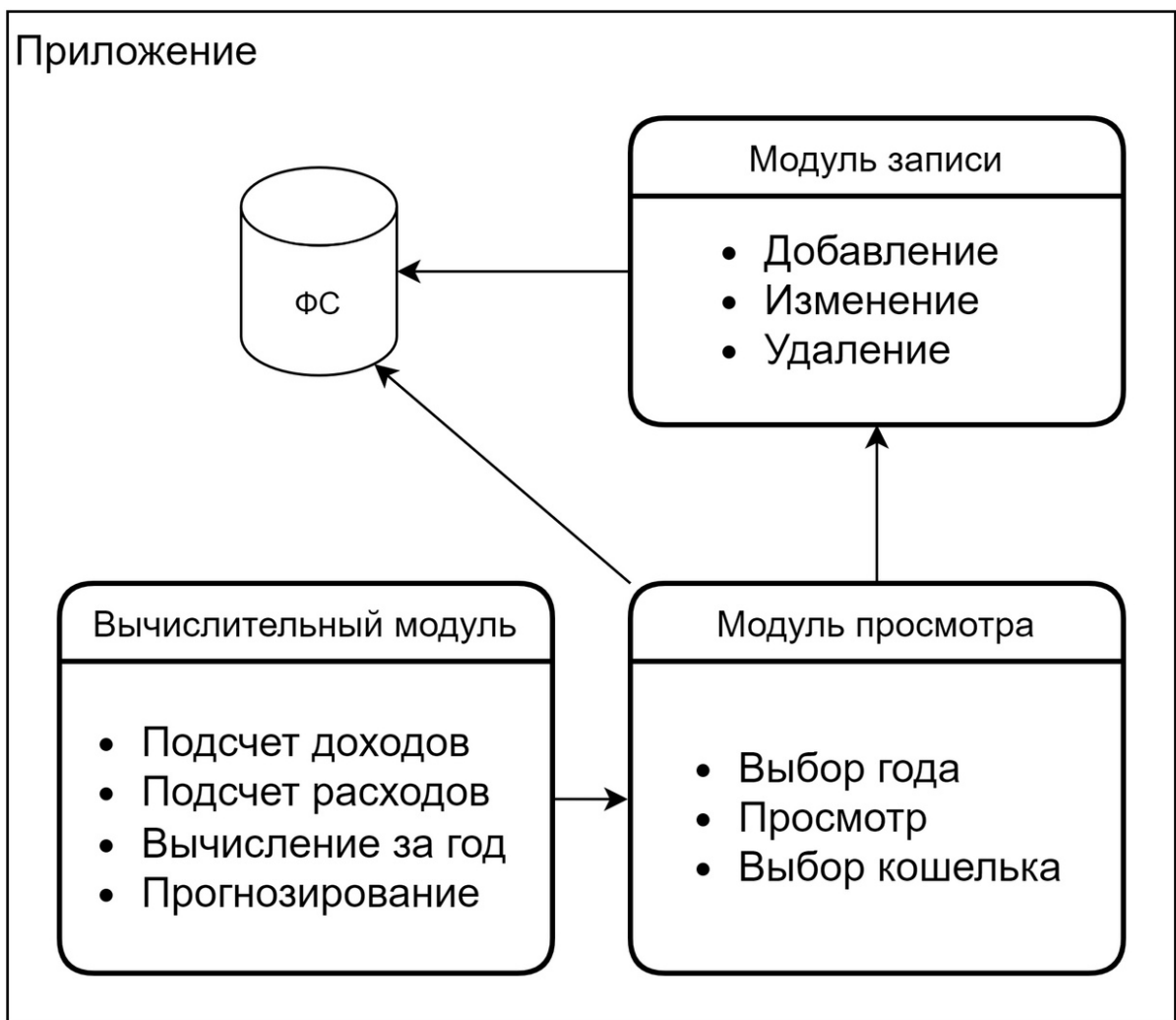
Описание приложения

Объектом тестирования является приложение-помощник для ведения учетов и расходов в повседневной жизни.

Функциональные требования

1. Сохранение в файл
2. Открытие из файла
3. Редактирование
4. Наличие графического интерфейса

Архитектура



Функции приложения

1. Модуль просмотра

- `int getWalletNum()` - получить количество кошельков
- `QVector<QString> wlist()` - получить список кошельков
- `QStringList getData(QString wal)` - получить данные о записях в кошельке
- `QStringList parseAll (QString log)` - преобразовать запись для отображения
- `QStringList parseYears(QString wal)` - получить года записей
- `QStringList getYearData(QString wal, QString year)` - получить записи за определенный год
- `QStringList sortByDate (QStringList data)` - отсортировать записи по дате
- `double getYearSum(QString wal, QString year)` - получить общую сумму за год

2. Модуль записи

- `bool deleteEntry(QString log, QString wal)` - удалить запись
- `bool createWallet(QString wal)` - создать кошелек
- `bool deleteWallet(QString wal)` - удалить кошелек
- `bool newEntry(QString log, QString wal)` - создать новую запись
- `bool editEntry(QString oldlog, QString wal, QString newlog)` - изменить запись в кошельке

3. Модуль вычислений

- `double getYearRashod(QString wal, QString year)` - получить расход за год
- `double getYearDohod(QString wal, QString year)` - получить доход за год
- `int getOpNum(QString wal)` - получить количество операций в кошельке
- `double getBal(QString wal)` - получить баланс кошелька
- `double getPrognoz(QString wal)` - получить прогноз баланса за месяц

Стратегия тестирования

Блочное тестирование

Блочные тесты проверяют работоспособность отдельных функций программы. Их поведение не зависит от результатов работы других тестируемых функций. Для их проверки в качестве аргументов при вызове функции передаются различные значения, после чего возвращаемый результат функции проверяется на наличие ошибок.

Будут проверяться функции, написанные на языке C++ с помощью googletest

Интеграционное тестирование

При интеграционном тестировании проверяется взаимодействие существующих модулей. Описанные выше, в блочном тестировании, функции, принимают ввод и отдают вывод другим. Таким образом проверяется отсутствие ошибок во взаимодействии их между друг другом.

Будут проверяться функции, написанные на языке C++ с помощью googletest

Аттестационное тестирование

Тестирование работы программы в целом. Производится путем запуска тестировщиком скомпилированной программы и созданием необходимых конфигурационных файлов. Таким образом проверяется работоспособность программы в виде приближенном к реальным условиям эксплуатации.

Проверка проводится вручную.

Критерии прохождения тестирования

Тест считается пройденным если полученный и ожидаемый результат совпадают. Тестирование считается пройденным если во время его прохождения не выявлено критических ошибок и процент пройденных тестов не меньше 80%.

Критерии остановки тестирования

Тестирование должно быть остановлено если количество не пройденных тестов больше 30% от общего количества, а также при обнаружении критических ошибок сильно влияющих на функциональность приложения.

Критерии возобновления тестирования

Тестирование возобновляется при исправлении ошибок на предыдущем этапе тестирования.

Перечень методов для тестирования

Тестирование не будет проводиться на следующих методах:

`int getWalletNum()` - получить количество кошельков - является более упрощенной реализацией метода `wlist()`

`int getOpNum(QString wal)` - получить количество операций в кошельке - является более упрощенной реализацией метода `getData()`

Остальные методы будут тестироваться

Модуль просмотра

- `QVector<QString> wlist()` - получить список кошельков
- `QStringList getData(QString wal)` - получить данные о записях в кошельке
- `QStringList parseAll (QString log)` - преобразовать запись для отображения
- `QStringList parseYears(QString wal)` - получить года записей
- `QStringList getYearData(QString wal, QString year)` - получить записи за определенный год
- `QStringList sortByDate (QStringList data)` - отсортировать записи по дате
- `double getYearSum(QString wal, QString year)` - получить общую сумму за год

Модуль записи

- `bool deleteEntry(QString log, QString wal)` - удалить запись
- `bool createWallet(QString wal)` - создать кошелек
- `bool deleteWallet(QString wal)` - удалить кошелек
- `bool newEntry(QString log, QString wal)` - создать новую запись
- `bool editEntry(QString oldlog, QString wal, QString newlog)` - изменить запись в кошельке

Модуль вычислений

- `double getYearRashod(QString wal, QString year)` - получить расход за год
- `double getYearDohod(QString wal, QString year)` - получить доход за год
- `double getBal(QString wal)` - получить баланс кошелька
- `double getPrognoz(QString wal)` - получить прогноз баланса за месяц

Тестирование

Блочное тестирование

Модуль записи

Тест № 1

Функция: createWallet(string)

Цель : Проверка создания кошелька при правильном вводе

Тип : Позитивный

Входные данные : string - "test123абв"

Ожидаемый результат : создается файл с таким именем

Тест № 2

Функция: createWallet(string)

Цель : Проверка создания кошелька при уже существующем кошельке, уже существующий файл с таким названием

Тип : Позитивный

Входные данные : string - "test123абв"

Ожидаемый результат : файл с таким именем не изменяется

Тест № 3

Функция: createWallet(string)

Цель : Проверка создания кошелька при пустом вводе

Тип : Негативный

Входные данные : string - ""

Ожидаемый результат : новых файлов не создавалось

Тест № 4

Функция: deleteWallet(string)

Цель : Проверка удаления совпадающего с таким названием кошелька

Тип : Позитивный

Входные данные : string - "test123абв", файл с таким именем

Ожидаемый результат : файл с таким именем не существует

Тест № 5

Функция: deleteWallet(string)

Цель : Проверка удаления кошелька при пустом вводе

Тип : Негативный

Входные данные : string - ""

Ожидаемый результат : отсутствует

Тест № 6

Функция: newEntry(string, string)

Цель : Проверка создания записи в кошельке

Тип : Позитивный

Входные данные : string - “aaa”, string - “test”
Ожидаемый результат : в файле появилась строка “aaa”

Тест № 7

Функция: newEntry(string, string)
Цель : Проверка отсутствия записи при пустой строке
Тип : Негативный
Входные данные : string - “”, string - “test”
Ожидаемый результат : отсутствует

Тест № 8

Функция: delEntry(string, string)
Цель : Проверка удаления совпадающей с содержанием записи
Тип : Позитивный
Входные данные : string - “aaa”, string - “test”
Ожидаемый результат : запись “aaa” отсутствует

Тест № 9

Функция: delEntry(string, string)
Цель : Проверка отсутствия удаления записей при пустом вводе
Тип : Позитивный
Входные данные : string - “”, string - “test”
Ожидаемый результат : отсутствует

Тест № 10

Функция: delEntry(string, string)
Цель : Проверка удаления только одной записи при двух одинаковых в кошельке
Тип : Негативный
Входные данные : string - “aaa”, string - “test”
Ожидаемый результат : из 2 записей “aaa” остается только одна

Тест № 11

Функция: delEntry(string, string)
Цель : Проверка удаления записи при отсутствии совпадений
Тип : Негативный
Входные данные : string - “bbb”, string - “test”
Ожидаемый результат : изменения отсутствуют

Тест № 12

Функция: editEntry(string, string, string)
Цель : Проверка изменения записи
Тип : Позитивный
Входные данные : string - “aaa”, string - “test”, string - “bbb”, структура с данными кошелька
Ожидаемый результат : запись “aaa” изменена на “bbb”

Тест № 13

Функция: editEntry(string, string, string)
Цель : Проверка отсутствия изменения записи при пустом вводе

Тип : Позитивный

Входные данные : string - "aaa", string - "test", string - "", структура с данными кошелька

Ожидаемый результат : изменения отсутствуют

Тест № 14

Функция: editEntry(string, string, string)

Цель : Проверка удаления только одной записи при двух одинаковых в кошельке

Тип : Негативный

Входные данные : string - "", string - "test", структура с данными кошелька

Ожидаемый результат : отсутствует

Модуль вычислений

Тест № 15

Функция: getYearDohod(string, string)

Цель : Проверка правильного вычисления суммы

Тип : Позитивный

Входные данные : string - "test", string - "2022", структура с данными кошелька

Ожидаемый результат : 500.50

Тест № 16

Функция: getYearDohod(string, string)

Цель : Проверка правильного вычисления суммы при неверном запросе года

Тип : Позитивный

Входные данные : string - "test", string - "2030", структура с данными кошелька

Ожидаемый результат : NULL

Тест № 17

Функция: getYearDohod(string, string)

Цель : Проверка попытки вычисления суммы в несуществующем кошельке

Тип : Негативный

Входные данные : string - "testtest", string - "2022"

Ожидаемый результат : NULL

Тест № 18

Функция: getYearRashod(string, string)

Цель : Проверка правильного вычисления суммы расходов

Тип : Позитивный

Входные данные : string - "test", string - "2022", структура с данными кошелька

Ожидаемый результат : -200.20

Тест № 19

Функция: getYearRashod(string, string)

Цель : Проверка правильного вычисления суммы расходов при неверном запросе года

Тип : Позитивный

Входные данные : string - “test”, string - “2030”, структура с данными кошелька
Ожидаемый результат : NULL

Тест № 20

Функция: getYearRashod(string, string)

Цель : Проверка попытки вычисления суммы расходов в несуществующем кошельке

Тип : Негативный

Входные данные : string - “testtest”, string - “2022”

Ожидаемый результат : NULL

Тест № 21

Функция: getBal(string)

Цель : Проверка правильного вычисления баланса

Тип : Позитивный

Входные данные : string - “test”, структура с данными кошелька

Ожидаемый результат : 300.30

Тест № 22

Функция: getBal(string)

Цель : Проверка работоспособности вычисления баланса пустого кошелька

Тип : Позитивный

Входные данные : string - “test”, пустая структура с данными кошелька

Ожидаемый результат : 0

Тест № 23

Функция: getBal(string)

Цель : Проверка попытки вычисления баланса в несуществующем кошельке

Тип : Негативный

Входные данные : string - “testtest”

Ожидаемый результат : NULL

Тест № 24

Функция: getPrognoz(string)

Цель : Проверка правильного вычисления прогноза

Тип : Позитивный

Входные данные : string - “test”, структура с данными кошелька

Ожидаемый результат : 300.30

Тест № 25

Функция: getPrognoz(string)

Цель : Проверка работоспособности вычисления прогноза пустого кошелька

Тип : Позитивный

Входные данные : string - “test”, пустая структура с данными кошелька

Ожидаемый результат : 0

Тест № 26

Функция: getPrognoz(string)

Цель : Проверка попытки вычисления баланса в несуществующем кошельке

Тип : Негативный
Входные данные : string - “testtest”
Ожидаемый результат : NULL

Модуль вычислений

Тест № 27
Функция: wlist()
Цель : Проверка правильного вычисления кошельков
Тип : Позитивный
Входные данные : созданные кошельки test и test2
Ожидаемый результат : [“test”, “test2”]

Тест № 28
Функция: getWalletNum()
Цель : Проверка работоспособности вычисления при отсутствии кошельков
Тип : Позитивный
Входные данные : отсутствуют
Ожидаемый результат : []

Тест № 29
Функция: getData(string)
Цель : Проверка правильного вычисления записей кошелька
Тип : Позитивный
Входные данные : string - “test”, структура с данными кошелька
Ожидаемый результат : [“aaa&500.50\$02.12.2022”, “aa2&-200.20\$01.12.2022”]

Тест № 30
Функция: getData(string)
Цель : Проверка работоспособности вычисления при отсутствии записей в кошельке
Тип : Позитивный
Входные данные : string - “test”, пустая структура с данными кошелька
Ожидаемый результат : []

Тест № 31
Функция: getData(string)
Цель : Проверка работоспособности при вводе несуществующего кошелька
Тип : Позитивный
Входные данные : string - “testtest”
Ожидаемый результат : NULL

Тест № 32
Функция: parseYears(string)
Цель : Проверка правильного вычисления годов записей кошелька
Тип : Позитивный
Входные данные : string - “test”, структура с данными кошелька
Ожидаемый результат : [“2022”]

Тест № 33

Функция: parseYears(string)

Цель : Проверка работоспособности вычисления при отсутствии записей в кошельке

Тип : Позитивный

Входные данные : string - "test", пустая структура с данными кошелька

Ожидаемый результат : []

Тест № 34

Функция: parseYears(string)

Цель : Проверка работоспособности при вводе несуществующего кошелька

Тип : Позитивный

Входные данные : string - "testtest"

Ожидаемый результат : NULL

Тест № 35

Функция: parseAll(string)

Цель : Проверка правильного вычисления годов записей кошелька

Тип : Позитивный

Входные данные : string - "test&100\$01.12.2022"

Ожидаемый результат : ["test", "100", "01.12.2022"]

Тест № 36

Функция: parseAll(string)

Цель : Проверка работоспособности вычисления при пустой записи

Тип : Негативный

Входные данные : string - ""

Ожидаемый результат : NULL

Тест № 37

Функция: parseAll(string)

Цель : Проверка работоспособности при некорректном вводе

Тип : Позитивный

Входные данные : string - "aaa&\$&5&\$&2"

Ожидаемый результат : NULL

Тест № 38

Функция: getYearData(string, string)

Цель : Проверка правильного вычисления годов записей кошелька

Тип : Позитивный

Входные данные : string - "test", string - "2022"

Ожидаемый результат : ["aaa&500.50\$02.12.2022", "aa2&-200.20\$01.12.2022"]

Тест № 39

Функция: getYearData(string, string)

Цель : Проверка работоспособности вычисления при пустом кошельке

Тип : Позитивный

Входные данные : string - “testtest”, string - “2022”
Ожидаемый результат : NULL

Тест № 40

Функция: getYearData(string, string)

Цель : Проверка работоспособности при отсутствии в этот год

Тип : Позитивный

Входные данные : string - “test”, string - “2021”

Ожидаемый результат : NULL

Тест № 41

Функция: getYearData(string, string)

Цель : Проверка работоспособности при отсутствии кошелька

Тип : Негативный

Входные данные : string - “testtest”, string - “2022”

Ожидаемый результат : NULL

Тест № 42

Функция: sortByDate(string[])

Цель : Проверка работоспособности при корректном вводе

Тип : Позитивный

Входные данные : string[] - [“aaa&500.50\$02.12.2022”, “aa2&-200.20\$01.12.2022”]

Ожидаемый результат : [”aa2&-200.20\$01.12.2022”, “aaa&500.50\$02.12.2022”]

Тест № 43

Функция: sortByDate(string[])

Цель : Проверка работоспособности при пустом вызове

Тип : Позитивный

Входные данные : string[] - []

Ожидаемый результат : []

Тест № 44

Функция: sortByDate(string[])

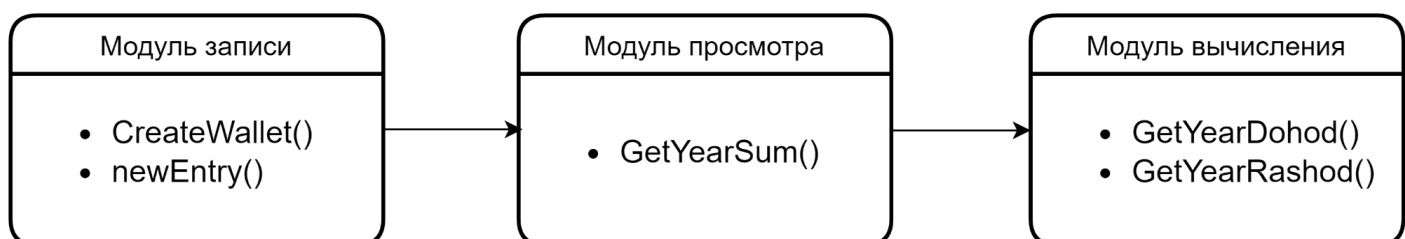
Цель : Проверка работоспособности при некорректном вводе

Тип : Негативный

Входные данные : string[] - [“02.12.2022aaa&500.50\$”, “01.12.2022aa2&-200.20\$”]

Ожидаемый результат : [“02.12.2022aaa&500.50\$”, “01.12.2022aa2&-200.20\$”]

Интеграционное тестирование



Тест № 1

Функция: `newEntry(string, string)`

Цель : Запись в кошелек после его создания

Тип : Позитивный

Входные данные : `string - "aaa"`, `string - "test"`

Ожидаемый результат : кошелек `"test"` содержит строку `"aaa"`

Тест № 2

Функция: `getYearSum(string, string)`

Цель : Проверка работоспособности модулей просмотра и модуля вычислений в паре

Тип : Позитивный

Входные данные : создание кошелька, `string - "test"`, `string - "2022"`

Ожидаемый результат : 300.30

Тест № 3

Функция: `getYearSum(string, string)`

Цель : Проверка работоспособности модулей просмотра и модуля вычислений в паре при отсутствии записей

Тип : Позитивный

Входные данные : создание кошелька, `string - "test"`, `string - "2022"`, пустой кошелек

Ожидаемый результат : 300.30

Тест № 4

Функция: `newEntry(string, string)`

Цель : Попытка создать запись в несуществующем кошельке, не вызывая до этого функцию создания

Тип : Негативный

Входные данные : `string - "aaa"`, `string - "test"`

Ожидаемый результат : отсутствие действий

Аттестационное тестирование

Тест № 1

Описание : Проверка работоспособности модуля выбора кошелька

Тип : Позитивный

Входные данные : При нажатии пользователем на пустое поле кошелька, появляется выпадающий список; пользователь нажимает на один из появившихся кошельков

Ожидаемый результат : Выводятся название кошелька и все данные о нём

Тест № 2

Описание : Проверка работоспособности модуля выбора записи

Тип : Позитивный

Входные данные : Пользователь нажимает на запись

Ожидаемый результат : Запись меняет цвет на зелёный, появляются дополнительные кнопки изменения и удаления записи

Тест № 3

Описание : Проверка работоспособности модуля добавления дохода

Тип : Позитивный

Входные данные : Пользователь нажимает на кнопку дохода, вводит данные и нажимает “сохранить”

Ожидаемый результат : Запись появляется в списке

Тест № 4

Описание : Проверка работоспособности модуля добавления дохода

Тип : Позитивный

Входные данные : Пользователь нажимает на кнопку дохода, вводит данные и нажимает “сохранить”

Ожидаемый результат : Запись появляется в списке

Тест № 5

Описание : Проверка работоспособности модуля добавления расхода

Тип : Позитивный

Входные данные : Пользователь нажимает на кнопку расхода, вводит данные и нажимает “сохранить”

Ожидаемый результат : Запись появляется в списке

Тест № 6

Описание : Проверка работоспособности модуля изменения

Тип : Позитивный

Входные данные : Пользователь нажимает на запись, нажимает кнопку “изменить”, вводит новые данные

Ожидаемый результат : Запись в списке изменяется

Тест № 7

Описание : Проверка работоспособности модуля удаления

Тип : Позитивный

Входные данные : Пользователь нажимает на запись, нажимает кнопку “удалить”, во всплывающем окне выбирает “удалить”

Ожидаемый результат : Запись удаляется из списка

Тест № 8

Описание : Проверка работоспособности модуля вычисления

Тип : Позитивный

Входные данные : Пользователь нажимает на кошелек

Ожидаемый результат : Выводится информация о текущем балансе, расходах и доходах за год, название кошелька, прогноз

Тест № 9

Описание : Проверка работоспособности модуля отображения

Тип : Негативный

Входные данные : Пользователь нажимает на кошелек

Ожидаемый результат : Не выводятся кнопки удаления и изменения записи

Нагрузочное тестирование

Тест № 1

Описание : Проверка при малом объеме входных данных

Тип : Позитивный

Входные данные : Файл кошелька с записями размером 10Кб

Ожидаемый результат : Отсутствие ошибок

Тест № 2

Описание : Проверка при среднем объеме входных данных

Тип : Позитивный

Входные данные : Файл кошелька с записями размером 10Мб

Ожидаемый результат : Отсутствие ошибок

Тест № 3

Описание : Проверка при большом объеме входных данных

Тип : Позитивный

Входные данные : Файл кошелька с записями размером 1Гб

Ожидаемый результат : Отсутствие ошибок

Пример тестов

```
TEST(createWal, pos){
    QString wal = "test";
    Func f;
    f.createWallet(wal);
    std::fstream file;
    file.open("test.mh", std::ios::in);
    int isOpen = 0;
    if(file)
        isOpen = 1;
    ASSERT_EQ(isOpen, 1);
    file.close();
    if(isOpen)
        remove("test.mh");
}
```

```
TEST(createWal, neg){
    QString wal = "";
    Func f;
    f.createWallet(wal);
    std::fstream file;
    file.open(".mh", std::ios::in);
    int isOpen = 0;
    if(file)
        isOpen = 1;
    ASSERT_EQ(isOpen, 0);
    file.close();
    if(isOpen)
        remove(".mh");
}
```


Журнал тестирования

Блочное тестирование

Дата	Тестируемый	Тест	Результат
11.12.2022	Угрюмов М.А.	1	Пройдено
13.12.2022	Угрюмов М.А.	2	Не пройдено (см. отчет 9)
11.12.2022	Угрюмов М.А.	3	Пройдено
11.12.2022	Угрюмов М.А.	4	Пройдено
11.12.2022	Угрюмов М.А.	5	Пройдено
11.12.2022	Угрюмов М.А.	6	Пройдено
11.12.2022	Угрюмов М.А.	7	Пройдено
11.12.2022	Угрюмов М.А.	8	Пройдено
11.12.2022	Угрюмов М.А.	9	Пройдено
11.12.2022	Угрюмов М.А.	10	Пройдено
13.12.2022	Угрюмов М.А.	11	Пройдено
11.12.2022	Угрюмов М.А.	12	Пройдено
11.12.2022	Угрюмов М.А.	13	Не пройдено (см. отчет 1)
11.12.2022	Угрюмов М.А.	14	Пройдено
11.12.2022	Угрюмов М.А.	15	Пройдено
11.12.2022	Угрюмов М.А.	16	Пройдено
11.12.2022	Угрюмов М.А.	17	Пройдено
11.12.2022	Угрюмов М.А.	18	Пройдено
11.12.2022	Угрюмов М.А.	19	Пройдено
11.12.2022	Угрюмов М.А.	20	Пройдено
11.12.2022	Угрюмов М.А.	21	Пройдено
11.12.2022	Угрюмов М.А.	22	Пройдено
11.12.2022	Угрюмов М.А.	23	Не пройдено (см. отчет 2)
11.12.2022	Угрюмов М.А.	24	Не пройдено (см. отчет 3)
11.12.2022	Угрюмов М.А.	25	Не пройдено (см. отчет 4)

11.12.2022	Угрюмов М.А.	26	Не пройдено (см. отчет 5)
11.12.2022	Угрюмов М.А.	27	Пройдено
11.12.2022	Угрюмов М.А.	28	Пройдено
11.12.2022	Угрюмов М.А.	29	Пройдено
11.12.2022	Угрюмов М.А.	30	Пройдено
11.12.2022	Угрюмов М.А.	31	Пройдено
11.12.2022	Угрюмов М.А.	32	Пройдено
11.12.2022	Угрюмов М.А.	33	Пройдено
11.12.2022	Угрюмов М.А.	34	Пройдено
11.12.2022	Угрюмов М.А.	35	Пройдено
11.12.2022	Угрюмов М.А.	36	Не пройдено (см. отчет 6)
11.12.2022	Угрюмов М.А.	37	Не пройдено (см. отчет 7)
11.12.2022	Угрюмов М.А.	38	Пройдено
11.12.2022	Угрюмов М.А.	39	Пройдено
11.12.2022	Угрюмов М.А.	40	Пройдено
11.12.2022	Угрюмов М.А.	41	Пройдено
11.12.2022	Угрюмов М.А.	42	Пройдено
11.12.2022	Угрюмов М.А.	43	Пройдено
11.12.2022	Угрюмов М.А.	44	Пройдено

Интеграционное тестирование

Дата	Тестирующий	Тест	Результат
11.12.2022	Угрюмов М.А.	1	Пройдено
11.12.2022	Угрюмов М.А.	2	Пройдено
13.12.2022	Угрюмов М.А.	3	Пройдено
13.12.2022	Угрюмов М.А.	4	Не пройдено (см. отчет 10)

Аттестационное тестирование

Дата	Тестирующий	Тест	Результат
11.12.2022	Угрюмов М.А.	1	Пройдено

11.12.2022	Угрюмов М.А.	2	Пройдено
11.12.2022	Угрюмов М.А.	3	Пройдено
12.12.2022	Угрюмов М.А.	4	Пройдено
12.12.2022	Угрюмов М.А.	5	Пройдено
12.12.2022	Угрюмов М.А.	6	Пройдено
12.12.2022	Угрюмов М.А.	7	Пройдено
12.12.2022	Угрюмов М.А.	8	Не пройдено (см. отчет 8)
12.12.2022	Угрюмов М.А.	9	Пройдено

Нагрузочное тестирование

Дата	Тестирующий	Тест	Результат
11.12.2022	Угрюмов М.А.	1	Пройдено
11.12.2022	Угрюмов М.А.	2	Не проводилось
11.12.2022	Угрюмов М.А.	3	Не проводилось

Журнал найденных ошибок Отчет об ошибке №1

Тест: Блочный 13

Объект тестирования: Функция editEntry(string, string, string)

Входные данные : string - "aaa", string - "test", string - ""

Ожидаемый результат: отсутствует

Результат: Запись "aaa" исчезла

Причина: не прописана проверка исключения в функции

Дата проведения: 11.12.22

Статус: не исправлено

Отчет об ошибке №2

Тест: Блочный 23

Объект тестирования: Функция getBal(string)

Входные данные : string - “testtest”

Ожидаемый результат: NULL

Результат: Ошибка выполнения

Причина: не прописана проверка исключения в функции

Дата проведения: 11.12.22

Статус: не исправлено

Отчет об ошибке №3

Тест: Блочный 24

Объект тестирования: Функция getPrognoz(string)

Входные данные : string - “test”

Ожидаемый результат: 300.30

Результат: Ошибка выполнения

Причина: функция не реализована

Дата проведения: 11.12.22

Статус: не исправлено

Отчет об ошибке №4

Тест: Блочный 25

Объект тестирования: Функция getPrognoz(string)

Входные данные : string - “test”

Ожидаемый результат: 0

Результат: Ошибка выполнения

Причина: функция не реализована

Дата проведения: 11.12.22

Статус: не исправлено

Отчет об ошибке №5

Тест: Блочный 26

Объект тестирования: Функция getPrognoz(string)

Входные данные : string - "testtest"

Ожидаемый результат: NULL

Результат: Ошибка выполнения

Причина: функция не реализована

Дата проведения: 11.12.22

Статус: не исправлено

Отчет об ошибке №6

Тест: Блочный 36

Объект тестирования: Функция parseAll(string)

Входные данные : string - ""

Ожидаемый результат: NULL

Результат: пустая строка

Причина: не прописана проверка исключения в функции

Дата проведения: 11.12.22

Статус: не исправлено

Отчет об ошибке №7

Тест: Блочный 37

Объект тестирования: Функция parseAll(string)

Входные данные : string - "aaa&\$\$5&\$\$2"

Ожидаемый результат: NULL

Результат: Запись "aaa" исчезла

Причина: неправильная работа функции

Дата проведения: 11.12.22

Статус: не исправлено

Отчет об ошибке №8

Тест: Аттестационный 8

Объект тестирования: Модуль вычисления, просмотра

Входные данные : действия пользователя

Ожидаемый результат: отображение полной информации о кошельке

Результат: Не отображается прогноз баланса

Причина: функция не реализована

Дата проведения: 12.12.22

Статус: не исправлено

Отчет об ошибке №9

Тест: Блочный 2

Объект тестирования: функция createWallet(string)

Входные данные : string - “test123абв”, уже существующий файл с таким названием

Ожидаемый результат: файл не изменяется

Результат: файл перезаписывается и содержимое стирается

Причина: не прописана проверка исключения в функции

Дата проведения: 13.12.22

Статус: не исправлено

Отчет об ошибке №10

Тест: Интеграционный 4

Объект тестирования: функция newEntry(string, string)

Входные данные : string - "aaa", string - "test", не создан кошелек

Ожидаемый результат: отсутствие действий

Результат: создание файла и запись в него

Причина: не прописана проверка исключения в функции

Дата проведения: 13.12.22

Статус: не исправлено

Результат

Проведенное тестирование показало, что программа нуждается в доработках и исправлении ошибок. Общая работоспособность сохраняется, но при неправильном обращении с программой могут возникнуть ошибки. Количество пройденных тестов > 80%. Покрытие кода - 94%.