

**КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 01.01 Разработка программных модулей**

**Тема: Разработка системы классов для приложения обменного пункта.**

**Специальность 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
| **Выполнил студент(ка) группы 309ИС-22** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **П.Д. Донских** |
| **Руководитель** | **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** | **Л.Б. Гусятинер** |

**Москва 2024**



**УТВЕРЖДАЮ**

**Зам. директора КМПО**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Ф. Гасанов**

**«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.**

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ**

**по дисциплине: МДК 01.01 Разработка программных модулей**

**Специальность 09.02.07 «Информационные системы и программирование»**

**Студент группы 309ИС-22 Павел Донских**

**ТЕМА: «Разработка системы классов для приложения обменного пункта»**

Дата выдачи задания «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г.

Срок сдачи проекта «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2024 г.

**Москва 2024**

**Перечень вопросов, подлежащих разработке:**

ВВЕДЕНИЕ

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Информационное обеспечение задачи

1.2. Обзор и анализ существующих программных решений

1.3. Постановка задачи. Структура входной и выходной информации

2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА СИСТЕМЫ КЛАССОВ

2.1. Построение модели системы

2.2. Описание главного модуля

2.3. Описание спецификаций к модулям

2.4. Описание модулей

2.5. Расчет сложности алгоритма

3. ОТЛАДКА И ТЕСТИРОВАНИЕ СИСТЕМЫ КЛАССОВ

3.1. Описание тестовых наборов модулей

3.4. Описание применения средств отладки

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЯ

**Исходные данные:** 27.02.2006 №152-ФЗ, 27.02.2006 №149-ФЗ, 26.07.2017 №187-ФЗ, ГОСТ Р ИСО/МЭК 12207, ГОСТ 19.005–85, ГОСТ 19.105–78

**Перечень разрабатываемых материалов для визуализации:** диаграмма классов, сценарий работы приложения, блок-схемы (не менее двух), дизайн приложения.

Задание выдал:

Руководитель курсового проекта Л. Б. Гусятинер

Задание принял к исполнению П. Д. Донских

*Рассмотрено*

на заседании предметно-цикловой комиссии

информационных технологий и системного

администрирования

Протокол № \_\_\_от «\_\_» \_\_\_\_\_\_\_\_\_ 2024 г.

Председатель ПЦК \_\_\_\_\_\_\_\_\_\_\_\_\_

**ОТЗЫВ**

**научного руководителя на курсовой проект**

**Студента(ки**)

(фамилия, имя и отчество)

**группа**  **курс**  3

**Дисциплина**  МДК 01.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

**Тема работы**

Отмечаются следующие моменты: актуальность темы исследования; степень разработанности проблемы, наиболее интересно исследованные вопросы. Оценивается степень самостоятельности и творчества студента; уровень его теоретической подготовки; умение анализировать научные материалы, делать практические выводы. Знание основных концепций, научной и специальной литературы по избранной теме.

**Научный руководитель**

(Уч. степень, уч. звание, должность, ФИО)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 6](#_Toc185422860)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 8](#_Toc185422861)

[1.1 Информационное обеспечение задачи 8](#_Toc185422862)

[1.2 Обзор и анализ существующих программных решений 9](#_Toc185422863)

[1.3 Постановка задачи 10](#_Toc185422864)

[1.3.1 Основные задачи и функции. 10](#_Toc185422865)

[1.3.2 Структура входной и выходной информаций 10](#_Toc185422866)

[2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ 11](#_Toc185422867)

[2.1 Проектирование сценария 13](#_Toc185422868)

[2.2 Диаграмма прецедентов 15](#_Toc185422869)

[2.3 Диаграммы классов 15](#_Toc185422870)

[2.4 Описание главного модуля 16](#_Toc185422871)

[2.5 Описание модулей и их спецификаций 19](#_Toc185422872)

[2.6 Сложность алгоритмов 21](#_Toc185422873)

[3. ТЕСТИРОВАНИЕ И ОТЛАДКА 24](#_Toc185422874)

[1.1 Описание тестовых наборов модулей 24](#_Toc185422875)

[3.2 Описание применения средств тестирования и отладки 27](#_Toc185422876)

[ЗАКЛЮЧЕНИЕ 28](#_Toc185422877)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 29](#_Toc185422878)

[Приложение 1. Скрины работы программы для сотрудника 31](#_Toc185422879)

[Приложение 2. Скрины работы программы для администратора 34](#_Toc185422880)

[Приложение 3. Таблица план тестирования 36](#_Toc185422881)

# ВВЕДЕНИЕ

История обмена валют началась с появлением необходимости нахождения эквивалентов денежным единицам разных стран. Первой международно признанной системой обмена валют стал так называемый «Золотой стандарт». Эта система предполагала установление определенной стоимости денежной единицы страны по отношению к фиксированному количеству золота. (см. список используемых источников (15))

Использование автоматизированных систем повышает общую эффективность и обеспечивает высокую точность проводимых обменов. Внедрение таких систем позволяет обменным пунктам быстрее реагировать на изменения на рынке и предлагать клиентам безопасные способы проведения операций.

Объектом исследования является система, которая включает в себя процессы, связанные с обменом валют, его анализом и отчетностью.

Предметом исследования является разработка программного модуля, который будет автоматизировать процессы учета обменов и формирования отчетов.

Обмен валют играет важную роль в финансовой системе каждой страны, обеспечивая возможность конвертации фиатной валюты или криптовалюты как для физических лиц, так и для юридических лиц. С каждым днем интерес к операциям конвертации только растет. Это связано с увеличением международной торговли и развитием туризма. Кроме того, с ростом интернет-транзакций значение онлайн-обменников становится все более актуальным.

Цель работы: создание функционального программного модуля, который будет обеспечивать:

* Упрощение процесса записи истории обменов в файл
* Представление инструментов для генерации отчетов и анализа обменов за текущий год.

Для достижения цели необходимо:

* Проанализировать предметную область;
* Проанализировать готовые решения;
* Подготовить техническое задание;
* Обосновать выбор инструментов и средств разработки;
* Описать реализацию технического задания;
* Подготовить план тестирования;
* Выполнить тестирование.

**1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

* 1. Информационное обеспечение задачи

Обменный пункт, «Обменник» — киоск продажи и скупки валюты населению. Часто вне стационарного отделения банка или не относящийся к банку вообще (см. список используемых источников (14))

Основные функции обменного пункты:

* Установление курсов валют: курсы должны обновляться в реальном времени на основе рыночных данных;
* Обработка заявок на обмен: система должна обеспечивать возможность ввода данных о сделках;
* Учет проведенных операций: все операции должны фиксироваться для дальнейшего анализа;
* Генерация отчетов: система должна формировать отчеты по сделкам за определенные периоды.

Основные термины:

1. Валюта — денежные знаки, используемые в разных странах (например, $ — доллар, ₽ — рубль);
2. Валютная пара – это финансовый инструмент, состоящий из 2-х разных валют. Обозначается краткими кодами валют через дробь, например EUR/USD – это валютная пара евро/американский доллар. Или USD/JPY – валютная пара американский доллар/японская йена; (см. список используемых источников (8))
3. Курс валюты — цена одной валюты, выраженная в другой валюте, устанавливаемая на основе рыночного спроса и предложения;
4. Обменный курс — курс, по которому происходит обмен валют (включает комиссию обменного пункта);
5. Базовая валюта — валюта, которую клиент хочет получить; (см. список используемых источников (8))
6. Котируемая валюта — валюта, за которую клиент получает базовую валюту; (см. список используемых источников (8))
7. Спред — разница цены покупки и продажи; (см. список используемых источников (8))
8. ASK — цена покупки; (см. список используемых источников (8))
9. BID — цена продажи; (см. список используемых источников (8))
10. Транзакция — любая сделка или операция, для совершения которой используется банковский счет, при этом баланс на нем меняется в меньшую или большую сторону.
    1. Обзор и анализ существующих программных решений

"Currency Tracker" — система предоставляет пользователям возможность отслеживать количество операций обмена валюты за заданный период. Функционал включает в себя создание отчетов, которые отображают объем сделок по каждой валютной.

"Forex Report Generator" — программное обеспечение предлагает инструменты для создания детализированных отчетов о сделках на рынке Forex. Пользователи могут анализировать количество сделок и прибыль по каждой валютной паре за выбранные временные интервалы.

"Exchange Statistics Generator" — система предназначена для автоматической генерации отчетов о статистике обменных операций, включая объемы обмена, популярные валюты и временные интервалы.

Ограниченная функциональность и интеграция: многие системы, такие как "Currency Tracker" и "Exchange Statistics Generator", имеют проблемы с интеграцией с другими платформами. Это может затруднить обмен данными и снизить общую эффективность работы обменного пункта.

Высокая стоимость: программное обеспечение, например, "Forex Report Generator", имеет высокую стоимость лицензии, что делает его недоступным для небольших обменных пунктов и индивидуальных инвесторов. Это ограничивает выбор доступных решений для малых бизнесов.

Необходимость ручного обновления данных: в системах, таких как "Exchange Statistics Generator", пользователи могут сталкиваться с необходимостью вручную обновлять данные. Это увеличивает риск ошибок и затрудняет процесс генерации отчетов.

* 1. Постановка задачи

Разработка программного модуля информационной системы направлена на автоматизацию ключевых бизнес-процессов, улучшения качества отчетности обменного пункта.

* + 1. Основные задачи и функции.

Ввод, удаление, редактирование данных о проведенных операциях обмена и менеджерах;

Учет количества сделок по каждой валютной паре;

Формирование отчетов о количестве операций и суммах обмененных денег за определенные периоды;

Возникает необходимость в компьютерной обработке вводимых данных о проведенных операциях с целью хранения этой информации на носителях и анализа эффективности работы обменного пункта.

* + 1. Структура входной и выходной информаций

Структура входной информации:

* Данные о менеджерах: номер менеджера, ФИО менеджера;
* Данные об обменах: номер обмена, дата обмена, номер менеджера, валюта, сумма в рублях.

Структура выходной информации:

* Готовый документ: текущая дата, количество обменов за каждый месяц, сумма обмененных денег по месяцам за текущий год, место для подписи.

# 2. ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ

При выборе инструмента и языка программирования было проведено сравнение по критериям, которые представлены в таблицах 1 и 3.

Степень важности критерия выбиралась из: низкая, ниже средней, средняя, ниже высокой, высокая.

|  |  |
| --- | --- |
| Таблица 1.  Критерии выбора инструмента | |
| Критерий | Важность критерия |
| Функционал | Ниже высокой |
| Удобство использования | Высокая |
| Скорость разработки | Ниже высокой |

Исходя из данных критериев, было проведено сравнение 3-х языков программирования от 0 до 10 баллов за критерий

|  |  |  |  |
| --- | --- | --- | --- |
| Таблица 2.  Оценка языков программирования | | | |
| Критерий/Язык программирования | Python | С++ | C |
| Функционал | 10 | 10 | 8 |
| Удобство использования | 7 | 9 | 7 |
| Скорость разработки | 7 | 7 | 7 |
| Итого | 24 | 26 | 22 |

|  |  |
| --- | --- |
| Таблица 3.  Критерии выбора среды разработки | |
| Критерий | Важность критерия |
| Простота | Средняя |
| Функционал | Высокая |
| Удобство использования | Высокая |
| Документация на русском языке | Ниже средней |

Исходя из данных критериев, было проведено сравнение 2-х среды разработки от 0 до 10 баллов за критерий.

|  |  |  |
| --- | --- | --- |
| Таблица 4.  Оценка сред разработки | | |
| Критерий/среда разработки | CodeBlocks | Microsoft Visual Studio |
| Простота | 7 | 5 |
| Функционал | 10 | 10 |
| Удобство использования | 8 | 6 |
| Документация на русском языке | 6 | 9 |
| Итого | 31 | 30 |

По результатам сравнения была выбрана среда разработки CodeBlocks

CodeBlocks — это интегрированная среда разработки (IDE) с открытым исходным кодом, предназначенная для программирования на языках C, C++ и Fortran. Она предоставляет пользователям удобный интерфейс для написания, компиляции и отладки кода. CodeBlocks поддерживает различные компиляторы и может быть расширена с помощью плагинов, что делает её гибким инструментом для разработчиков.

* 1. Проектирование сценария

В данном разделе приведён пример сценария использования программы сотрудником и администратором (см. рисунок 1).

После запуска программы менеджер должен загрузить файлы менеджеров, операций обменов. Затем он может редактировать данные в файлах или записать новые данные. После этих действий программа выдаст отчет об обменах (см. приложение 1).

Также, для администратора предусмотрен режим редактирования данных, в который можно получить доступ, используя пароль. В нем администратору доступны: добавление и удаление данных о менеджерах (см приложение 2).

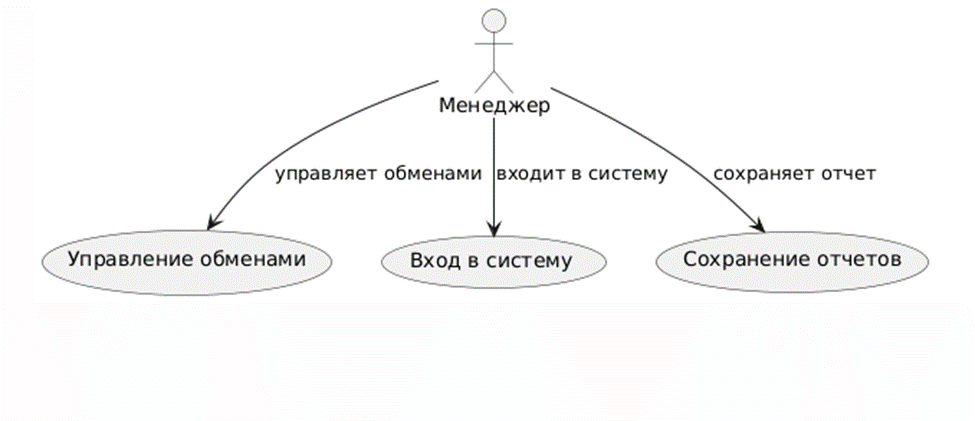
Рисунок сценария находится на следующей странице



Рисунок 1. Сценарий использования

* 1. Диаграмма прецедентов

Для определения вариантов использования к проекту была построена диаграмма прецедентов

Рисунок 2. Диаграмма прецедентов для менеджера

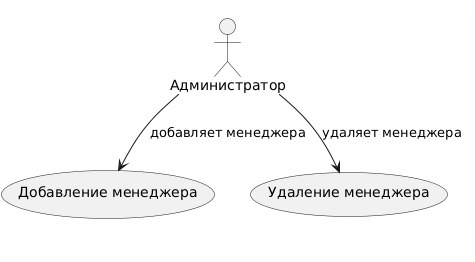


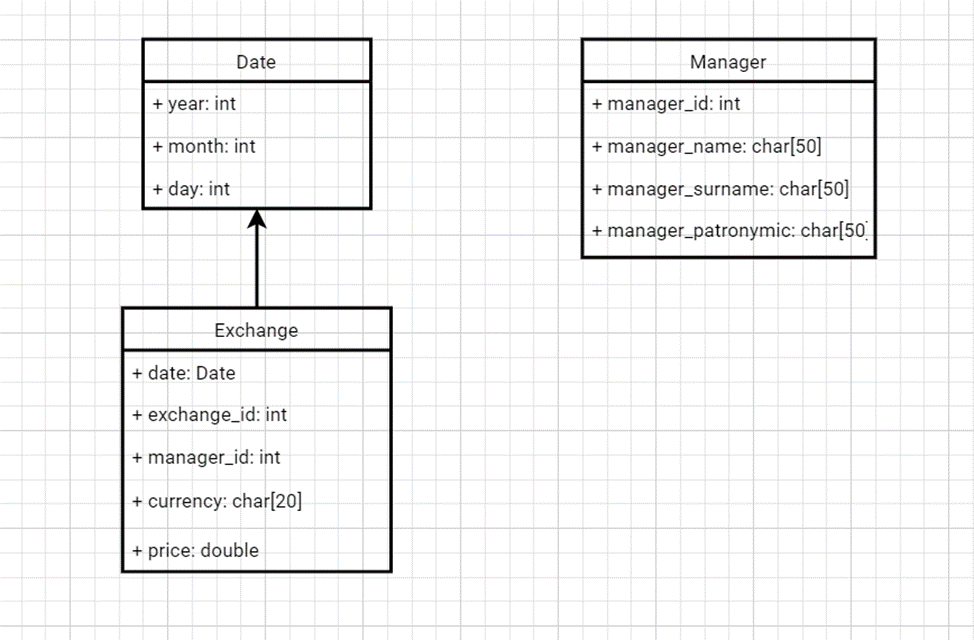
Рисунок 3. Диаграмма прецедентов для администратора

* 1. Диаграммы классов

В диаграмме классов представлены классы: manager, exchange, date.

В классах используется спецификатор доступа public. Public говорит нам о том, что доступ открыт всем, кто видит определение данных классов.

Блок, стоящий после названия класса, показывает нам объявление переменных, а знак “+” указывает на спецификацию доступа, в данном случае только public. Диаграмма классов представлена на следующей странице.

Рисунок 4. Диаграмма классов

* 1. Описание главного модуля

Главный модуль представляет собой main.cpp

К главному модулю подключаются остальные модули, содержащие в себе классы и функции.

Листинг 1. Код главного модуля

// main.cpp

#include <stdio.h>

#include <locale.h>

#include <stdlib.h>

#include <Windows.h>

#include "func.h"

int main() {

system("color F0"); // Устанавливает цвет консоли: белый фон и черный текст

setlocale(LC\_ALL, ""); // Устанавливает локаль для корректного отображения символов

int man\_count = 0; // Количество менеджеров

int exchange\_count = 0; // Количество обменов

Exchange exchanges[AR\_EXCHANGE\_LEN]; // Массив для хранения обменов

Manager managers[AR\_MAN\_LEN]; // Массив для хранения менеджеров

// Загрузка данных о менеджерах из файла

man\_count = load\_managers("managers.txt", managers, AR\_MAN\_LEN);

printf("Файл менеджеров загружен в память\n\n");

int item = -1; // Переменная для выбора пункта меню

while (item != 0) {

// Отображение главного меню

printf("Главное меню:\n");

printf("0. Выход\n");

printf("1. О программе\n");

printf("2. Я менеджер\n");

printf("3. Я администратор\n");

printf("Выберите действие: ");

scanf("%d", &item);

system("cls"); // Очистка экрана

switch (item) {

case 0:

// Завершение программы

printf("Спасибо за использование нашей разработки\n\n");

break;

case 1:

// Информация о программе

printf("309ИС-22 Донских П. Д.\n");

printf("Генерация отчетов для обменного пункта валют\n\n");

break;

case 2:

// Вход для менеджера

manager\_login(managers, man\_count, exchanges, &exchange\_count);

break;

case 3:

// Вход для администратора

admin\_login(managers, &man\_count);

break;

default:

// Обработка неверного выбора

printf("Ошибка. Пункты от 0 до 3\n\n");

break;

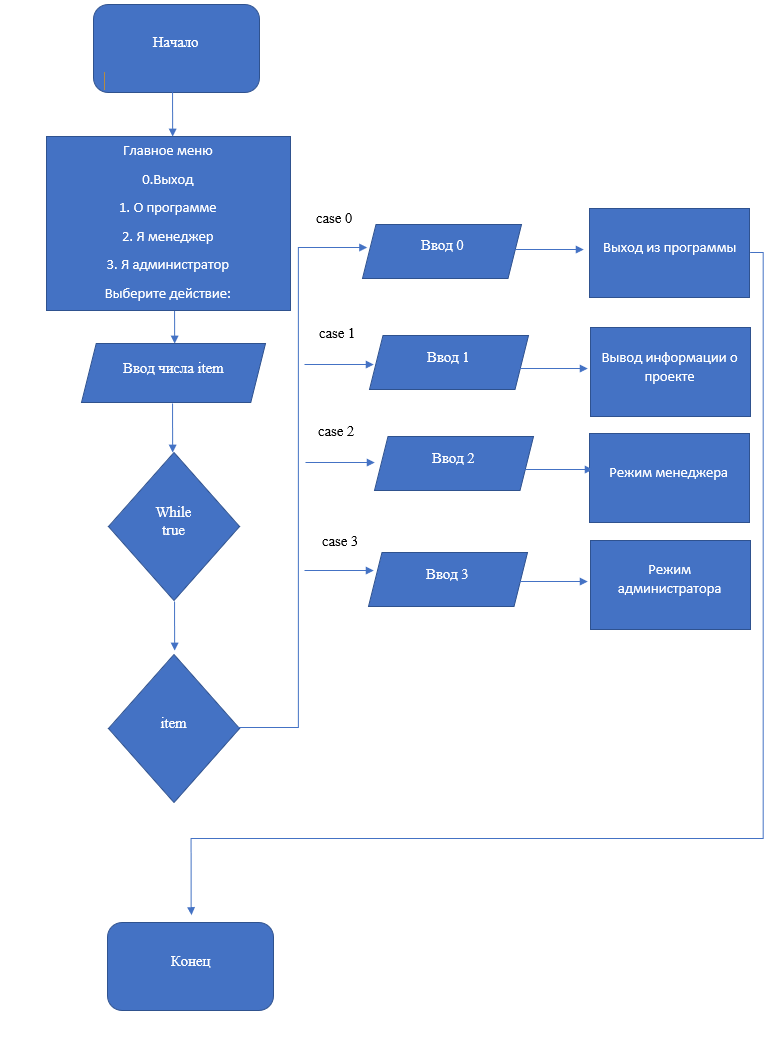
}

}

return 0;

}

На другой странице представлена блок-схема главного модуля.

Рисунок 5. Блок-схема главного модуля

* 1. Описание модулей и их спецификаций

В данной главе описаны используемые модули.

1. Главный модуль (см. 2 ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОДУЛЯ пункт 2.4)
2. Модуль функций

В модуле функций находятся функции, используемые в проекте:

* int get\_current\_year(). Получает текущий год, используя системное время и локальные настройки.
* void get\_current\_date(). Форматирует текущую дату в строковом формате и сохраняет её в переданном буфере.
* void get\_current\_date1(). Форматирует текущую дату в строковом формате с использованием английских названий месяцев и сохраняет её в переданном буфере.
* void manager\_to\_s(). Преобразует информацию о менеджере в строку формата "ID. Фамилия Имя Отчество".
* void exchange\_to\_s(). Преобразует информацию об обмене в строку, включая ID, дату, ID менеджера, валюту и цену.
* void admin\_menu(). Отображает меню администратора и обрабатывает выбор действий (добавление или удаление менеджера).
* void manager\_login(). Позволяет менеджеру войти в систему, проверяя его ID и переходя в меню менеджера.
* void show\_menu(). Отображает основное меню для менеджера с доступными действиями.
* void manager\_menu(). Отображает меню для менеджера и обрабатывает выбор действий (сохранение обмена, редактирование и загрузка данных).
* void admin\_login(). Позволяет администратору войти в систему, проверяя введённый пароль.
* bool check\_exchange\_id\_exists(). Проверяет, существует ли обмен с заданным ID в файле.
* void save\_new\_exchange(). Сохраняет новый обмен в файл после ввода всех необходимых данных.
* void edit\_exchange(). Позволяет редактировать существующий обмен по его ID.
* bool validate\_date(). Проверяет корректность даты (месяц, день) с учётом високосных годов.
* int load\_exchanges(). Загружает данные об обменах из файла в массив структур Exchange.
* int load\_managers(). Загружает данные о менеджерах из файла в массив структур Manager.
* void add\_manager(). Добавляет нового менеджера в массив и сохраняет его данные в файл.
* void remove\_manager(). Удаляет менеджера из массива по его ID и обновляет файл с данными.
* void do\_query(). Формирует текстовый отчет по обменам за текущий год и сохраняет его в файл.
* void do\_query1(). Формирует HTML отчет по обменам за текущий год и сохраняет его в файл.

1. Модуль, содержащий класс manager (manager.h)

Класс manager предназначен для представления менеджера, который работает с обменами. (см. рисунок 3)

1. Модуль, содержащий класс date (date.h)

Класс date предназначен для представления даты. (см. рисунок 3)

1. Модуль, содержащий класс exchange (exchange.h)

Класс exchange предназначен для представления информации об обмене. (см. рисунок 3)

* 1. Сложность алгоритмов

1. int get\_current\_year()
2. Сложность: O(1)
3. Возвращает текущий год, используя системное время.
4. void get\_current\_date(char\* date\_str)
5. Сложность: O(1)
6. Форматирует текущую дату в строковом формате.
7. void get\_current\_date1(char\* date\_str)
8. Сложность: O(1)
9. Возвращает текущую дату с английскими названиями месяцев.
10. void manager\_to\_s(char s[150], const Manager& manager)
11. Сложность: O(1)
12. Преобразует информацию о менеджере в строку.
13. void exchange\_to\_s(char s[100], const Exchange& exchange)
14. Сложность: O(1)
15. Преобразует информацию об обмене в строку.
16. void admin\_menu(Manager managers[], int\* man\_count)
17. Сложность: O(1)
18. Отображает меню администратора и обрабатывает выбор действий.
19. void manager\_login(Manager managers[], int man\_count, Exchange exchanges[], int\* exchange\_count)
20. Сложность: O(n)
21. Ищет менеджера по ID в массиве.
22. void show\_menu()
23. Сложность: O(1)
24. Отображает меню для менеджера.
25. void manager\_menu(Manager managers[], int man\_count, Exchange exchanges[], int\* exchange\_count, int current\_manager\_id)
26. Сложность: O(n)
27. Обрабатывает действия менеджера и может вызывать функции с линейной сложностью.
28. void admin\_login(Manager managers[], int\* man\_count)
29. Сложность: O(1)
30. Проверяет введённый пароль администратора.
31. bool check\_exchange\_id\_exists(const char\* fname, int exchange\_id)
32. Сложность: O(n)
33. Проверяет существование ID обмена в файле.
34. void save\_new\_exchange(const char\* fname, int current\_manager\_id)
35. Сложность: O(n)
36. Сохраняет новый обмен в файл после проверки уникальности ID.
37. void edit\_exchange(const char\* fname, int current\_manager\_id)
38. Сложность: O(n)
39. Загружает все обмены из файла и редактирует нужный ID.
40. bool validate\_date(const Date& date)
41. Сложность: O(1)
42. Проверяет корректность даты с фиксированным числом операций.
43. int load\_exchanges(const char\* fname, Exchange exchanges[], int limit)
44. Сложность: O(n)
45. Загружает обмены из файла до заданного лимита.
46. int load\_managers(const char\* fname, Manager managers[], int limit)
47. Сложность: O(n)
48. Загружает менеджеров из файла аналогично предыдущей функции.
49. void add\_manager(Manager managers[], int\* count)
50. Сложность: O(n)
51. Проверяет уникальность ID нового менеджера перед добавлением.
52. void remove\_manager(Manager managers[], int\* count)
53. Сложность: O(n)
54. Находит менеджера по ID и сдвигает массив для удаления.
55. void do\_query(const Exchange exchanges[], int limit, const char\* fname)
56. Сложность: O(n)
57. Подсчитывает статистику по месяцам на основе загруженных обменов.
58. void do\_query1(const Exchange exchanges[], int limit, const char\* fname)
59. Сложность аналогична предыдущей функции и зависит от реализации (O(n)).

# 3. ТЕСТИРОВАНИЕ И ОТЛАДКА

## Описание тестовых наборов модулей

Во время разработки информационной системы проводились тестирование и отладка. Тестирование позволяет выявить ошибки и несоответствия в работе системы на ранних стадиях разработки, что значительно снижает риски и затраты на исправление проблем в будущем, Отладка же, в твою очередь, включает в себя анализ и исправление обнаруженных ошибок.

Для тестирования использовались тест-кейсы (позитивные и негативные), которые проводились по тест плану (см приложение 3).

Позитивные тест-кейсы:

1. Тест успешного добавления нового обмена:

* Описание: в меню менеджера выбрать действие 1. Ввести id обмена «7», даты обмена «12.12.2024», id менеджера «1», валюта «USD», сумма обмена «100555».
* Ожидаемый результат: данные успешно сохранены в файл.
* Фактический результат: данные успешно сохранены в файл.

1. Тест успешной загрузки из файла обмена:

* Описание: в меню менеджера выбрать действие 4.
* Ожидаемый результат: файл обмена загружен в память.
* Фактический результат: файл обмена загружен в память.

1. Тест успешной загрузки из файла менеджеров:

* Описание: в меню менеджера выбрать действие 3.
* Ожидаемый результат: файл обмена загружен в память.
* Фактический результат: файл обмена загружен в память.

1. Тест успешного добавления нового менеджера:

* Описание: в меню администратора выбрать действие 1. Ввести id нового менеджера «3», ФИО нового менеджера «Халатян Артем Макарович».
* Ожидаемый результат: менеджер успешно добавлен.
* Фактический результат: менеджер успешно добавлен, но из-за кодировки файла данные выводятся некорректно.

1. Тест успешного запроса по обменам:

* Описание: в меню менеджера выбрать действие 5.
* Ожидаемый результат: Сохранение результата в файле query1.html.
* Фактический результат: Сохранение результата в файле query1.html.

1. Тест успешного запроса по обменам:

* Описание: в меню менеджера выбрать действие 7.
* Ожидаемый результат: Сохранение результата в файле query1.txt.
* Фактический результат: Сохранение результата в файле query1.txt.

Негативные тест-кейсы:

1. Тест неуспешного добавления нового обмена:

* Описание: в меню менеджера выбрать действие 1. Ввести id обмена «8», даты обмена «HELLO», id менеджера «2», валюта «EURO», сумма обмена «11111».
* Ожидаемый результат: ошибка ввода, ввод даты заново.
* Фактический результат: ошибка ввода, ввод даты заново.

1. Тест неуспешного добавления нового обмена:

* Описание: в меню менеджера выбрать действие 1. Ввести id обмена «8», даты обмена «13.13.2024», id менеджера «1», валюта «EURO», сумма обмена «сто тысяч».
* Ожидаемый результат: ошибка ввода, ввод суммы обмена заново.
* Фактический результат: ошибка ввода, ввод суммы обмена заново.

1. Тест неуспешного редактирования обмена:

* Описание: в меню менеджера выбрать действие 2. Ввести id обмена «7», даты обмена «BYE», id менеджера «1», валюта «USD», сумма обмена «100555».
* Ожидаемый результат: ошибка ввода, ввод даты заново.
* Фактический результат: ошибка ввода, ввод даты заново.

1. Тест неуспешного удаления менеджера:

* Описание: в меню администратора выбрать действие 2. Ввести id существующего менеджера «4».
* Ожидаемый результат: менеджер не найден.
* Фактический результат: менеджер не найден.

В результате тестирования было обнаружено, что из-за проблем с локализацией при добавлении нового менеджера и формировании отчета в файл с расширением html текст неправильно загружается в файл.

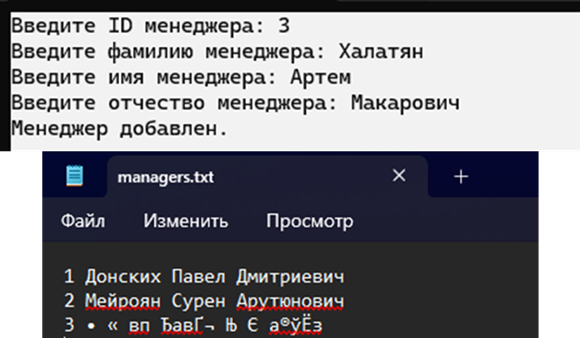


Рисунок 6. Проблема с локализацией при добавлении менеджера

Рисунок с проблемой с локализацией в файле .html находится на следующей странице.

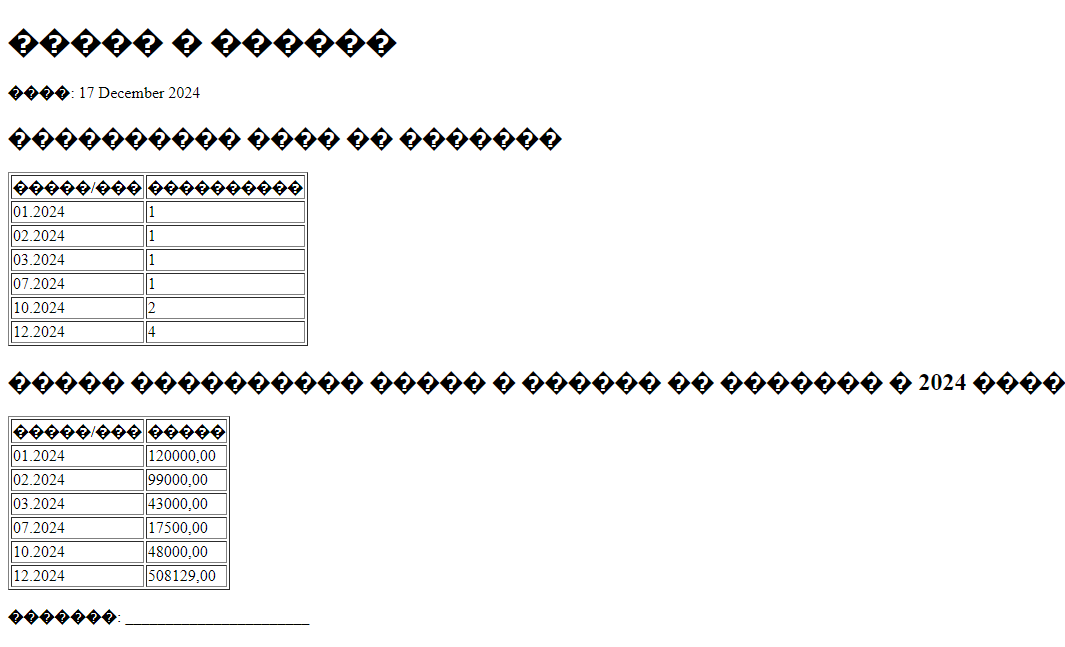


Рисунок 7. Проблема с локализацией при составлении отчета с расширением html

Решение проблем: переход на другую операционную систему (LINUX).

3.2 Описание применения средств тестирования и отладки

Для тестирования программного обеспечения был применен метод белого ящика. Он позволил проверить поведение программы при различных сценариях ввода, охватить ключевые функциональные блоки приложения, и обеспечить качество их реализации.

Каждый функциональный блок тестировался отдельно, что позволило изолировать возможные ошибки.

# ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта была написана программа "CurrencyReport" для автоматизации процесса отчетности обменного пункта.

В ходе работы были проанализированы предметная область, существующие разработки, посвященные данному направлению, получены практические навыки с работой в CodeBlocks.

Так же планируется продолжить работу над данным проектом с целью расширения возможностей и удобства для пользователей. Планы по доработке представлены ниже.

1. Переход на новый стек;
2. Реализация методов и функций, учитывающих количество оставшихся в наличии валют;
3. Добавить реализацию функций для управления обменами;
4. Обновление курса валют в режиме реального времени;
5. Создание и последующая доработка интерфейса;
6. Создание и последующая доработка баз данных.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Нормативно-правовые источники:

1. Федеральный закон "Об информации, информационных технологиях и о защите информации" от 27.07.2006 N 149-ФЗ —  <https://www.consultant.ru/document/cons_doc_LAW_61798/>
2. Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ —  <https://www.consultant.ru/document/cons_doc_LAW_61801/>
3. Федеральный закон "О безопасности критической информационной инфраструктуры Российской Федерации" от 26.07.2017 N 187-ФЗ — <https://www.consultant.ru/document/cons_doc_LAW_220885/>
4. РОССТАНДАРТ — <https://www.gost.ru/portal/gost/home/standarts/catalognational>
5. ГОСТ Р ИСО/МЭК 12207 — <https://docs.cntd.ru/document/1200082859>
6. ЕСПД —  <https://www.swrit.ru/gost-espd.html>

Учебники, учебные пособия, статьи:

1. Кузнецов, А. П. (2019). Обменные пункты: организация и управление. Санкт-Петербург: Издательство "Экономика".
2. Сидоров, В. Н. (2021). Анализ валютного рынка. Москва: Издательство "Наука".

Интернет-ресурсы:

1. Официальный сайт Центрального банка Российской Федерации.  (2023). Курсы валют – <https://cbr.ru/>
2. Ввод/вывод и работа с файлами в C++ — <https://metanit.com/cpp/tutorial/8.3.php>
3. Классы —  <https://metanit.com/cpp/tutorial/5.1.php>
4. Функции — <https://metanit.com/cpp/tutorial/3.1.php>
5. Указатели — <https://metanit.com/cpp/tutorial/4.1.php>
6. Обменный пункт — <https://ru.wikipedia.org/wiki/Обменный_пункт>
7. История обмена валют — <https://web.archive.org/web/20121225213734/http://ru.wikipedia.org/wiki/Обмен_валют>

# Приложение 1. Скрины работы программы для сотрудника

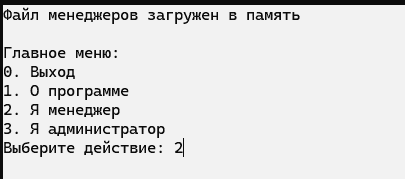


Рисунок 8. Основное меню

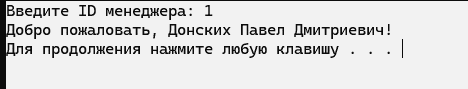


Рисунок 9. Вход в меню менеджера

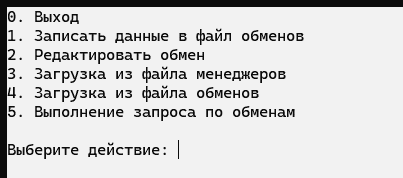


Рисунок 10. Меню менеджера

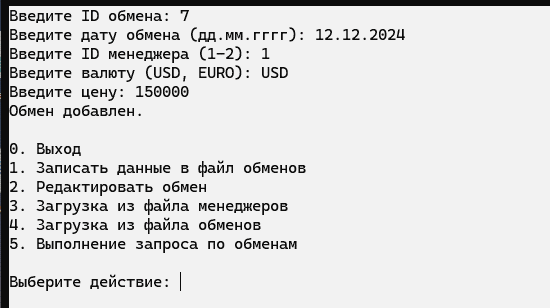
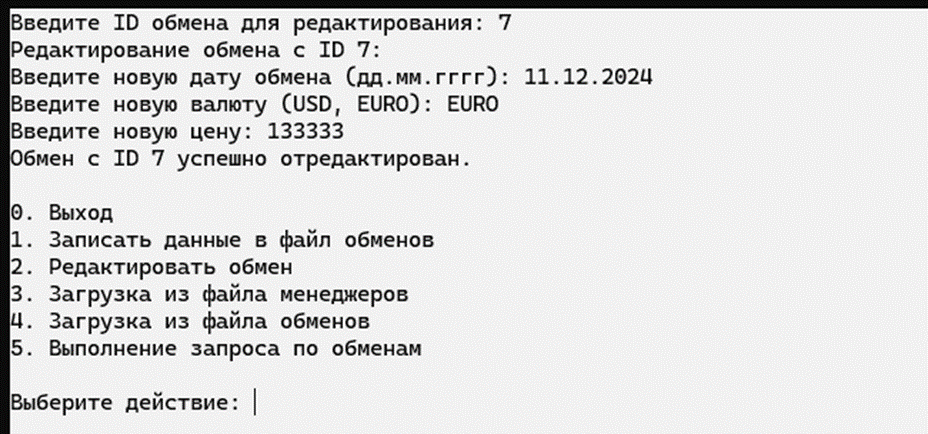


Рисунок 11. Добавление нового обмена

Рисунок 12. Редактирование обмена

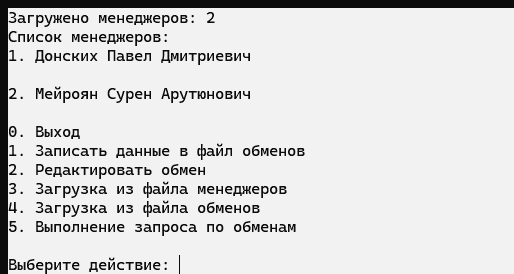
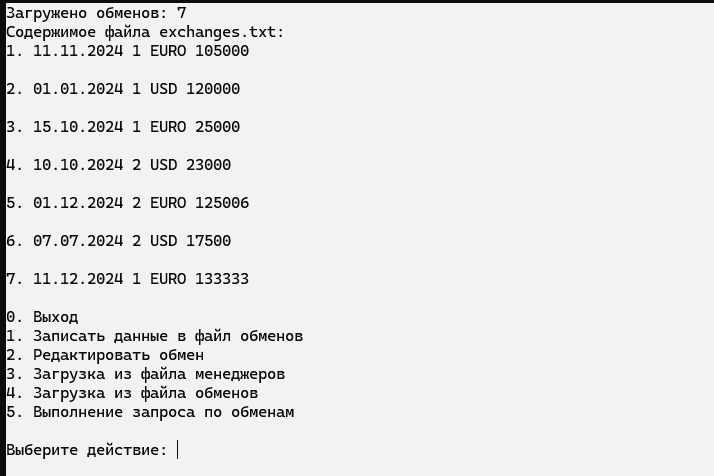


Рисунок 13. Загрузка из файла менеджеров

Рисунок 14. Загрузка из файлов обмена

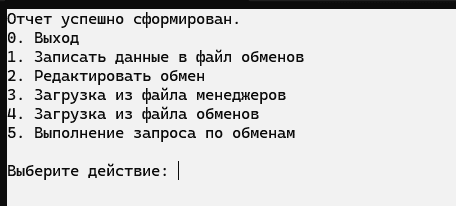


Рисунок 15. Выполнение запроса по обменам

# Приложение 2. Скрины работы программы для администратора

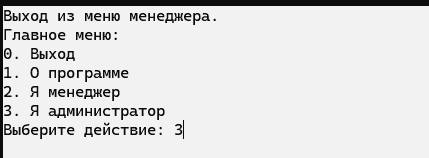


Рисунок 16. Основное меню

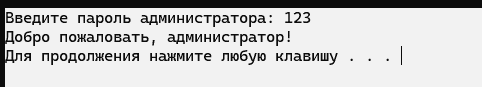


Рисунок 17. Вход в меню менеджера

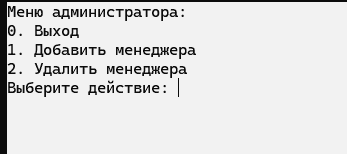


Рисунок 18. Меню администратора

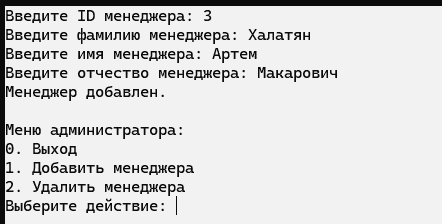


Рисунок 19. Добавление менеджера

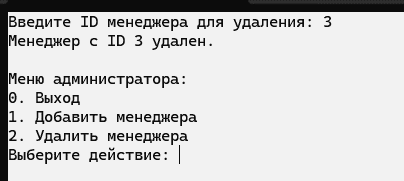


Рисунок 20. Удаление менеджера

# Приложение 3. Таблица план тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Таблица 7.  План тестирования для программного модуля обменного пункта | | | | | |
| **ID теста** | **Описание теста (тип)** | **Предусловия** | **Шаги для воспроизведения** | **Ожидаемый результат** | **Фактический результат** |
| **1** | Ввод данных об обмене (позитивный). | Пользователь находится в меню менеджера. | 1. Выбрать действие 2. 2. Ввести id менеджера. 3. Выбрать действие 1. 4. Ввести данные в поля «id обмена», «дата обмена», «валюта», «сумма обмена». 5. Нажать кнопку «Enter» на клавиатуре. | Вывод на экран:  «данные успешно сохранены в файл». | Вывод на экран:  «данные успешно сохранены в файл». |
| **2** | Ввод данных об обмене (негативный). | Пользователь находится в меню менеджера. | 1. Выбрать действие 2. 2. Ввести id менеджера. 3. Выбрать действие 1. 4. Ввести некорректное обозначение валюты. 5. Нажать кнопку «Enter» на клавиатуре. | Вывод на экран:  «неверный ввод валюты»;  «введите заново». | Вывод на экран:  «неверный ввод валюты»;  «введите заново». |
| **3** | Ввод данных об обмене (негативный). | Пользователь находится в меню менеджера. | 1. Выбрать действие 2. 2. Ввести id менеджера. 3. Выбрать действие 1. 4. Ввести некорректное значение суммы обмена 5. Нажать кнопку «Enter» на клавиатуре. | Вывод на экран:  «ошибка ввода»; «введите заново». | Вывод на экран:  «ошибка ввода»; «введите заново». |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Продолжение таблицы 7 | | | | | |
| **4** | Загрузка данных из файла обмена (позитивный). | Пользователь находится в меню менеджера. | 1. Выбрать действие 2. 2. Ввести id менеджера. 3. Выбрать действие 4. 4. Нажать кнопку «Enter» на клавиатуре. | Вывод на экран: «файл обмена загружен в память». | Вывод на экран: «файл обмена загружен в память». |
| **5** | Загрузка данных их файла менеджеров (позитивный). | Пользователь находится в меню менеджера. | 1. Выбрать действие 2. 2. Ввести id менеджера. 3. Выбрать действие 3. 4. Нажать кнопку «Enter» на клавиатуре. | Вывод на экран: «файл менеджеров загружен в память». | Вывод на экран: «файл менеджеров загружен в память». |
| **6** | Редактирование данных об обмене (негативный) | Пользователь находится в меню менеджера. | 1. Выбрать действие 2. 2. Ввести id менеджера. 3. Выбрать действие 2. 4. Ввести некорректные данные в поле «дата продажи». 5. Нажать кнопку «Enter» на клавиатуре. | Вывод на экран:  «ошибка ввода даты»; «введите дату заново». | Вывод на экран:  «ошибка ввода даты»; «введите дату заново». |
| **7** | Удаление менеджера (негативный) | Пользователь находится в меню администратора. | 1. Выбрать действие 3. 2. Ввести пароль админа. 3. Выбрать 2 (удалить менеджера) 4. Ввести некорректный id менеджера. 5. Нажать кнопку «Enter» на клавиатуре. | Вывод на экран:  «менеджер не найден». | Вывод на экран:  «менеджер не найден». |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Продолжение таблицы 7 | | | | | |
| **8** | Добавление менеджеров (позитивный) | Пользователь находится в меню редактирования данных о менеджере. | 1. Выбрать действие 3. 2. Ввести пароль админа. 3. Выбрать действие 1 4. Ввести корректное данные: id\_менеджера, ФИО. 5. Нажать кнопку «Enter» на клавиатуре. | Вывод на экран:  «менеджер успешно добавлен». | Вывод на экран:  «менеджер успешно добавлен». |
| **9** | Запрос по обменам (позитивный) | Пользователь находится в меню менеджера. | 1. Выбрать действие 2. 2. Ввести id админа 3. Выбрать действие 6. 4. Нажать кнопку «Enter» на клавиатуре. | 1. Вывод на экран:  «Результат сохранен в файле query1.txt» 2. Сохранение результата в файле query1.txt. | 1. Вывод на экран:  «Результат сохранен в файле query1.txt» 2. Сохранение результата в файле query1.txt. |
| **10** | Запрос по обменам (позитивный) | Пользователь находится в меню менеджера. | 1. Выбрать действие 2. 2. Ввести id админа 3. Выбрать действие 5. 4. Нажать кнопку «Enter» на клавиатуре. | 1. Вывод на экран:  «Результат сохранен в файле query1.html» 2. Сохранение результата в файле query1.html. | 1. Вывод на экран:  «Результат сохранен в файле query1.html» 2. Сохранение результата в файле query1.html. |