**

Колледж космического машиностроения и технологии

**КУРСОВОЙ ПРОЕКТ**

**По дисциплине «Прикладное программирование»**

**Тема: «Разработка симулятора обменного пункта валюты»**

Выполнил студент

Трифонов Кирилл Алексеевич

Группа П2-17

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_ (Дата сдачи работы)

Королев, 2020

Оглавление

[Введение 3](#_Toc43825803)

[Глава 1. Теоретическая часть 4](#_Toc43825804)

[1.1. Описание предметной области 4](#_Toc43825805)

[1.2. Описание языка программирования С++ 4](#_Toc43825806)

[1.3. Описание существующих разработок 6](#_Toc43825807)

[Глава 2. Проектная часть 9](#_Toc43825808)

[2.1. Диаграмма прецедентов 9](#_Toc43825809)

[2.2. Выбор инструментов 10](#_Toc43825810)

[2.3. Проектирование сценария 11](#_Toc43825811)

[2.4. Диаграмма классов 12](#_Toc43825812)

[2.5. Описание главного модуля 13](#_Toc43825813)

[2.6. Описания спецификаций к модулям 14](#_Toc43825814)

[2.7. Описание модулей 16](#_Toc43825815)

[2.8. Описание тестовых наборов модулей 17](#_Toc43825816)

[2.9. Описание применения средств отладки 22](#_Toc43825817)

[2.10. Анализ оптимальности использования памяти и быстродействия 24](#_Toc43825818)

[Глава 3. Эксплуатационная часть 25](#_Toc43825819)

[3.1. Руководство оператора 25](#_Toc43825820)

[Заключение 25](#_Toc43825821)

[Список литературы и интернет-источников 25](#_Toc43825822)

[Приложения 25](#_Toc43825823)

# Введение

Уже несложно в наше время представить автоматизированную систему практически в любой сфере деятельности человека. В современных условиях банки работают в целях получения максимальных доходов. Одним из шагов для достижения максимальной прибыли является автоматизация. Согласитесь, что для компьютера, который выполняет обязанности кассира, нужно куда меньше затрат, да и удобство для пользователя никто не отменял. Правильно настроенный автомат быстрее кассира, делает гораздо меньше ошибок, не устаёт и требует меньше рабочего пространства.

Цель курсового проекта заключалась в создании приложения, которое позволит автоматизировать процесс обмена валют.

Для осуществления поставленной цели служат следующие задачи:

1. Изучение документации по C++
2. Написание кода
3. Описание руководства пользователя

# Глава 1. Теоретическая часть

* 1. Описание предметной области

В наше время существует огромное количество стран. У каждой страны существует своя валюта. С развитием отношений между этими странами участились случаи по купли/продажи различных услуг и товаров. Но так как у каждой страны своя валюта, то появилась необходимость конвертирования валюты.

Чтобы осуществлять торговлю между странами возникает потребность конверсии валют. Для работы туристических агентств так же возникает такая потребность. При поездке в другую страну человек не может оплачивать товары и услуги этой страны валютой из другой страны, поэтому ему нужно обменять валюту своей страны на валюту страны в которой он находится. Данную возможность стали предоставлять банки. Они обменивали валюту разных стран. В дальнейшем стали развиваться компьютерные технологии и появились электронные деньги. Появилась потребность конвертировать и их тоже. Для этого появились виртуальные пункты обмена валюты. Эти программы позволяли производить транзакцию денежных средств и просмотр курсов валют.

Предметная область «Обменного пункта валют» представляет собой деятельность работы обменного пункта, предназначенная, в первую очередь, для автоматизации работы кассиров, находящихся как в самом банке, так и вне его территории. Обмен производится как покупки, так и продажи денежных единиц в валюте данной страны. Существуют формы, которые предоставляют возможность изменения курса валюты продажи и покупки. Все данные заносятся в таблицу. Можно просматривать операции, проведенные кассиром за определённый срок.

* 1. Описание языка программирования С++

Язык программирования С++ это высокоуровневый компилируемый язык программирования со статической типизацией, который подходит для создания самых различных приложений. На сегодняшний день С++ является одним из самых популярных и распространенных языков.

Своими корнями он уходит в язык Си, который был разработан в 1969—1973 годах в компании Bell Labs программистом Деннисом Ритчи (Dennis Ritchie). В начале 1980-х годов датский программист Бьерн Страуструп (Bjarne Stroustrup), разработал С++ как расширение к языку Си. Фактически вначале C++ просто дополнял язык Си некоторыми возможностями объектно-ориентированного программирования. И поэтому сам Страуструп вначале называл его как "C with classes" ("Си с классами").

* 1. Описание существующих разработок

На рынке программного обеспечения существуют множество банковских систем и отдельных программных продуктов автоматизирующих валютно-обменные операции.

Первое рассмотренное решение - это "Union Business System" компании «ЮниСАБ». В данной системе разработан бизнес «обменный пункт». Программа «Обменный пункт» обеспечивает полную автоматизацию работы кассира "выносного" обменного пункта и опосредованную связь с банком. Его широкие возможности позволяют настраивать программный модуль в соответствии с регламентом работы обменного пункта. Все выполняемые операции сопровождаются формированием необходимых печатных документов.

Параметры каждой операции имеют гибкие настройки в соответствии с внутрибанковской инструкцией работы обменных пунктов. Программа готовит полный набор документов авансовая заявка и заявка на подкрепление, реестры по всем выполняемым операция, справка об остатках, препроводительные ведомости (валютная и рублевая), акт передачи другому кассиру.

[Сайт компании «ЮниСАБ»](http://www.unisab.ru)

Плюсами данного программного средства являются:

1. Возможность ведения электронного реестра валютно-обменных операций;
2. Формирование необходимых печатных документов и бланков строгой отчетности;
3. Простота в обучении персонала.

К минусам можно отнести:

1. Высокие требования к аппаратным средствам.

Второе рассмотренное решение - "Центавр-Дельта". Это современное решение, основанное на технологии клиент-сервер, которое обеспечивает высокую производительность при автоматизации банковской деятельности и исключает затраты, связанные с традиционными SQL системами управления базами данных.

В данном продукте присутствует модуль валютно-обменных операций. Среди основных возможностей можно отметить полную автоматизацию валютно-обменных операций, ведение операций по покупке-продаже иностранной валюты за рубли, ведение конверсионных операций с получением соответствующих отчетов, контроль правильности ввода информации, наличие девяти уровней доступа к информации и выполнению банковских операций, и другое.

Плюсы данного программного средства:

1. Возможность ведения электронного реестра валютно-обменных операций;
2. Формирование необходимых печатных документов и бланков строгой отчетности;
3. Простота в обучении персонала.

К минусам можно отнести:

1. Высокие требования к аппаратным средствам.

[Сайт "Центавр-Дельта"](http://www.absonline.ru/software/native/centavr_d/)

В разрабатываемом программном продукте достоинствами являются:

1. Простота в обучении персонала;
2. Низкие требования к аппаратным средствам.

Входными данными являются:

1. Информация о клиенте (паспортные данные);
2. Информация о курсах валют, о самой валюте и комиссии;
3. К постоянным данным можно отнести базу данных.

# Глава 2. Проектная часть

* 1. Диаграмма прецедентов

Для определения вариантов использования к проекту была построена диаграмма прецедентов

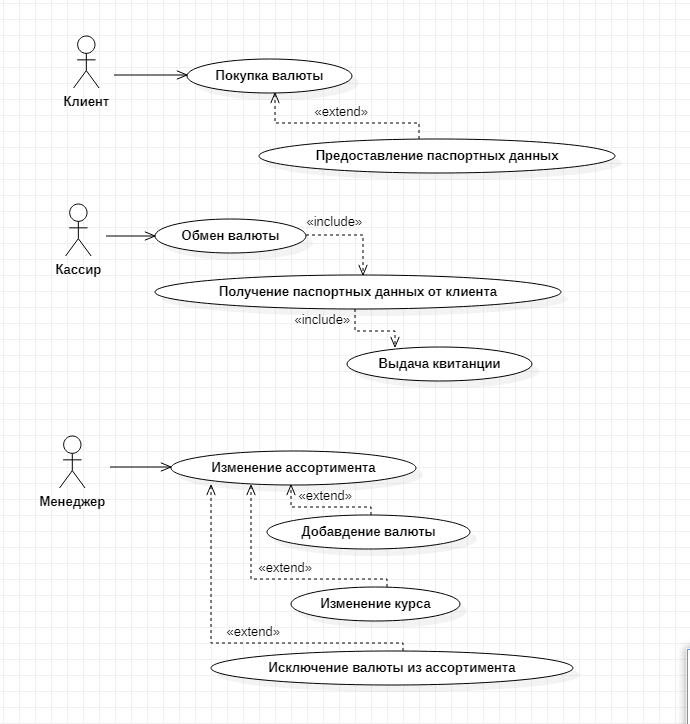


Рисунок 1. Диаграмма прецедентов для проекта

* 1. Выбор инструментов

При написании курсовой работы были использованы: интегрированная среда разработки "Microsoft Visual Studio" и язык C++.

|  |  |  |
| --- | --- | --- |
|  | CodeBlocks | Microsoft Visual Studio |
| Простота | 6 | 5 |
| Быстрота | 10 | 9 |
| Функционал | 7 | 10 |
| Удобство использования | 6 | 8 |
| Документация на русском языке | 5 | 9 |
| Итого | 34 | 41 |

Microsoft Visual Studio — инструментальная среда разработки, включающая в себя интегрированную среду разработки, редактор исходного кода, встроенный отладчик. Многие другие инструменты возможно получить благодаря подключению плагинов – сторонних расширений. Был выбран по причине наличия интеграции с Unity 3d, поддержки языка программирования С#, наличие возможности написание собственных Unit-тестов.

* 1. Проектирование сценария

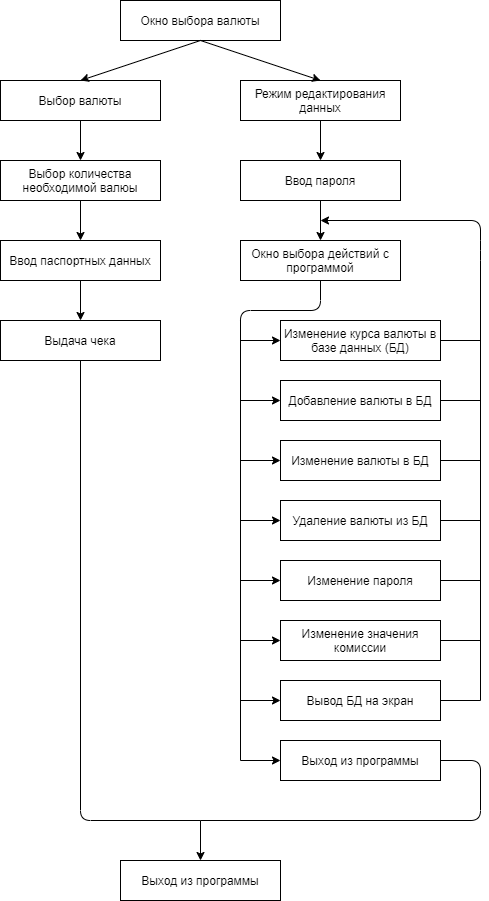


Рисунок 2. Сценарий использования

* 1. Диаграмма классов

В диаграмме классов представлены два класса Money (для валюты) и Person (для клиента).

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

Блок стоящий после названия класса показывает нам объявленные переменные, а знак “+” указывает на спецификацию доступа, в данному случае только public.

После блока с атрибутами, расположен блок с операциями (функциями класса).

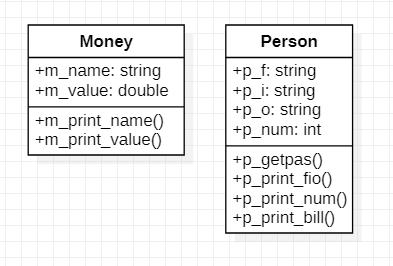


Рисунок 3. Диаграмма классов

* 1. **Описание главного модуля**

Главный модуль представляет собой файл bank.cpp.

К главному модулю подключаются остальные модули, содержащие в себе классы и функции. Так же в нём проверяется есть ли данные в БД.

Листинг 1.

#include "functions.h"

#include "Person.h"

extern vector <Money> dollar;

int main()

{

system("chcp 1251");

system("cls");

cout << "Для доступа к режиму редактирования нажмите '0', пароль - '0'" << endl;

double commission = get\_commission();

int val;

Person jd;

getff();

print();

if (dollar.size() < 1) //проверка на отсутствие данных в бд

{

cout << "Ассортимент пуст, обратитесь к менеджеру" << endl;

cin >> val;

if (val == 0)

worker();

}

else

{

val = choice\_your\_destiny();

if (val == 0)

{

worker();

}

else

{

double amount;

cout << "Введите количество необходимой валюты: ";

cin >> amount;

jd.p\_getpas();

jd.p\_print\_bill(calculate\_sum(amount, dollar[val-1].m\_value, commission), commission, amount, val);

}

}

return 0;

}

* 1. **Описания спецификаций к модулям**

Спецификация к модулю Money (Листинг 2).

Листинг 2.

#pragma once

#include <iostream>

using namespace std;

class Money // класс для валюты

{

public:

string m\_name; // название валюты

double m\_value; // курс валюты к рублю

Money(string name, double value)

{

this->m\_name = name;

this->m\_value = value;

}

void m\_print\_name() // вывод названия с учётом '\_'

{

for (int i = 0; i < m\_name.length(); i++)

{

if (m\_name[i] == '\_')

cout << " ";

else

cout << m\_name[i];

}

}

void m\_print\_value() // вывод курса

{

cout << m\_value;

}

};

Спецификация к модулю Person (Листинг 3).

Листинг 3.

#pragma once

#include <iomanip>

#include <vector>

#include "Money.h"

using namespace std;

extern vector <Money> dollar;

class Person // класс для паспортных данных пользователя

{

public:

string p\_f; // фамилия

string p\_i; // имя

string p\_o; // отчество

int p\_num; // номер паспорта

void p\_getpas() // метод получения пасортных данных клиента

{

cout << "Введите свои паспортные данные (ФИО, серия и №) через пробел: ";

cin >> p\_f >> p\_i >> p\_o >> p\_num;

}

void p\_print\_fio() // вывод ФИО

{

cout << p\_f << " " << p\_i << " " << p\_o;

}

void p\_print\_num() // вывод номера паспорта

{

cout << p\_num;

}

void p\_print\_bill(long double total, double commission, double amount, int val) // печать квитанции

{

cout << "\nЧек";

cout << "\nФИО клиента: ";

p\_print\_fio();

cout << "\nНомер паспорта: ";

p\_print\_num();

cout << "\nВалюта: ";

dollar[val - 1].m\_print\_name();

cout << "\nКоличество валюты: " << amount;

cout << "\nКурс: ";

dollar[val - 1].m\_print\_value();

cout << "р за единицу (";

dollar[val - 1].m\_print\_name();

cout << ")";

cout << "\nКомиссия: " << commission << "%";

cout << fixed << setprecision(2) << "\nИтого к оплате: " << total << " р";

cout << endl;

}

};

* 1. **Описание модулей**

В данной главе описаны используемые модули.

1. Главный модуль

Главный модуль представляет собой файл bank.cpp. К главному модулю подключаются остальные модули, содержащие в себе классы и функции. Так же в нём проверяется есть ли данные в БД.

1. Модуль функций

В модуле функций находятся функции, используемые в проекте:

* getff() - заполнение БД данными из входного файла
* print() - вывод БД на экран
* vector\_to\_file() - заполнение входного файла данными из БД
* get\_commission() - получение комиссии из входного файла
* choice\_your\_destiny() - функция выбора валюты пользователем
* add\_field() - функция добавления поля в БД
* edit\_field() - функция изменения курса валюты в поле БД
* delete\_field() - функция удаление поля БД
* change\_field() - функция изменения поля БД
* change\_password() - функция изменения пароля входа в режим редактирования
* change\_commission() - функция изменения комиссии
* worker\_menu() - функция выбора действия с БД
* worker() - функция проверки пароля для доступа к режиму редактирования БД
* calculate\_sum() - функция вычисления итогового счёта

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

Класс Money предназначен для корректной работы базы данных. В него входят следующие поля и методы:

* поле названия валюты
* поле курса валюты (относительно рубля)
* метод, предназначенный для вывода названия валюты
* метод, предназначенный для вывода курса валюты

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

Класс Person предназначен для работы с данными клиента (фамилия, имя, отчество, серия и номер паспорта)

* 1. **Описание тестовых наборов модулей**

В этом разделе будут продемонстрированы результаты тестирования "черного ящика"

Тест 1. Изменение курса валют

Действия: Зайти в режим редактирования, выбрать действие "изменение курса валют", изменить курс выбранной валюты, в данном случае валюта - "Доллар США".

Ожидаемый результат: Изменение предыдущего курса валюты "Доллар США" (65.5р) на новое значение - 70.5р.

Результат теста:

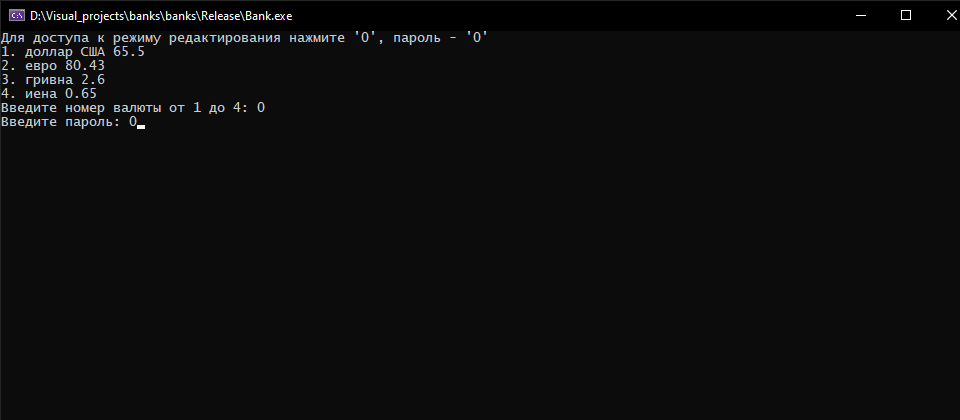


Рисунок 4. Вход в режим редактирования данных

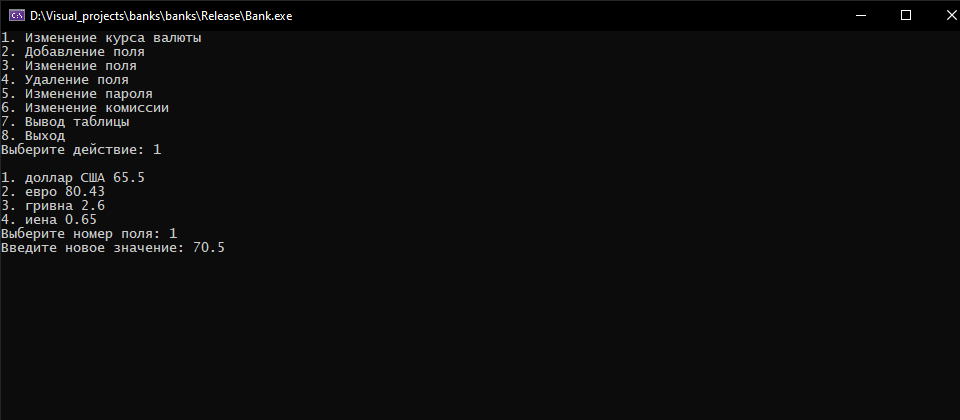


Рисунок 5. Изменение курса валюты "доллар США"

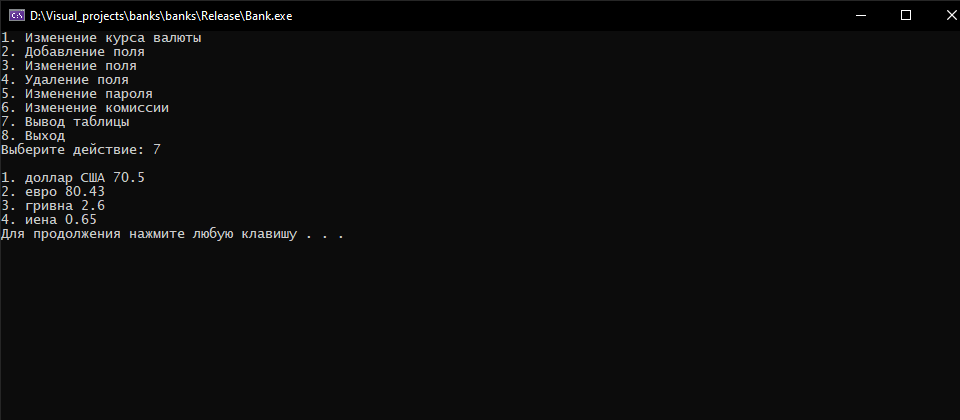


Рисунок 6. Новое значение валюты "доллар США"

Тест 2. Добавление поля в таблицу

Действия: Зайти в режим редактирования, выбрать действие "добавление поля", добавить поле "тенге" с курсом 0.17р.

Ожидаемый результат: Добавление в таблицу нового поля с валютой "тенге" и курсом 0.17.

Результат теста:

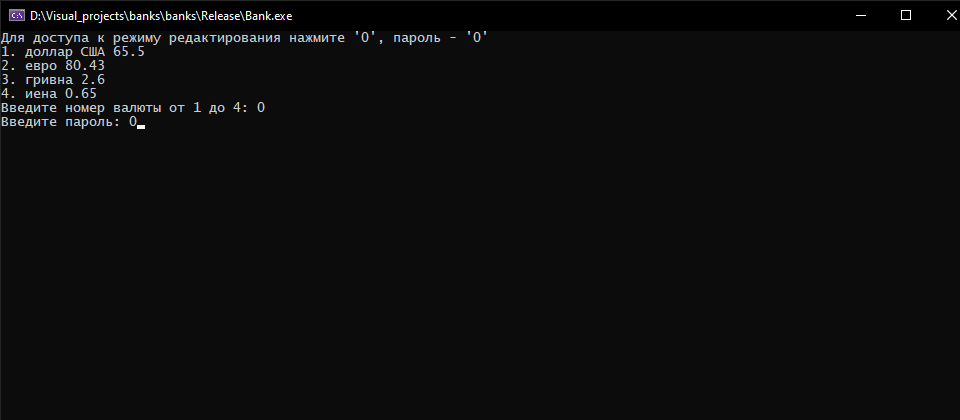


Рисунок 7. Вход в режим редактирования данных

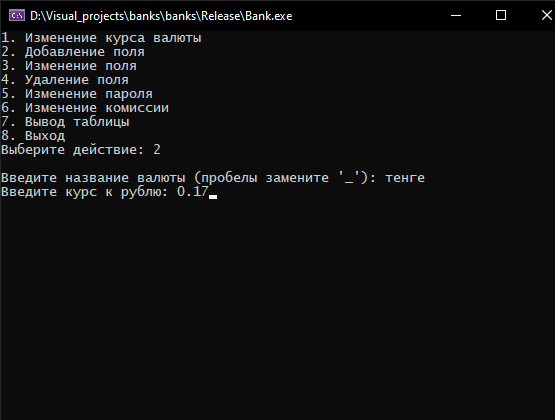


Рисунок 8. Добавление валюты

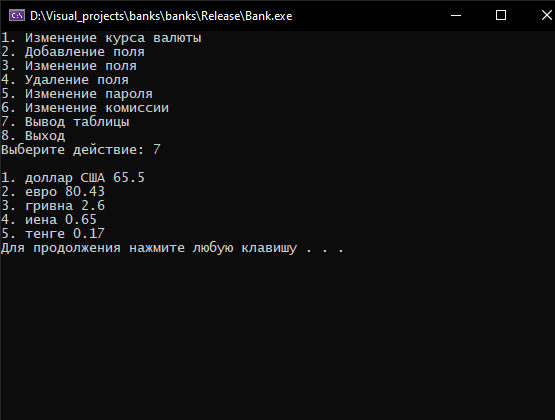


Рисунок 9. Валюта добавлена

Тест 3. Смена пароля доступа в режим редактирования данных

Действия: Зайти в режим редактирования, выбрать действие "изменение пароля", поменять пароль с "0" на "Здравствуйте, Леонид Борисович".

Ожидаемый результат: Изменение пароля на "Здравствуйте, Леонид Борисович".

Результат теста:

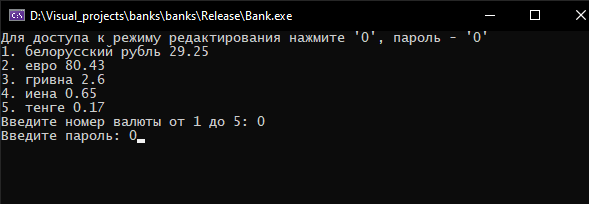


Рисунок 10. Вход в режим редактирования данных

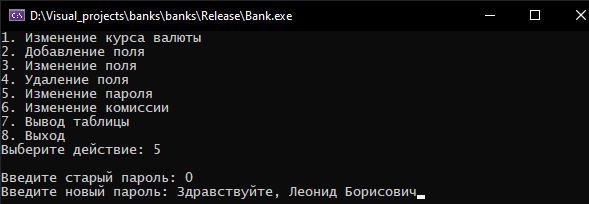


Рисунок 11. Смена пароля

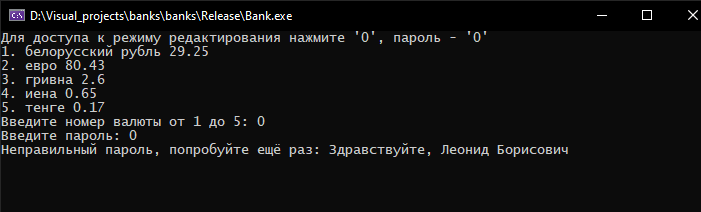


Рисунок 12. Предыдущий пароль не верен

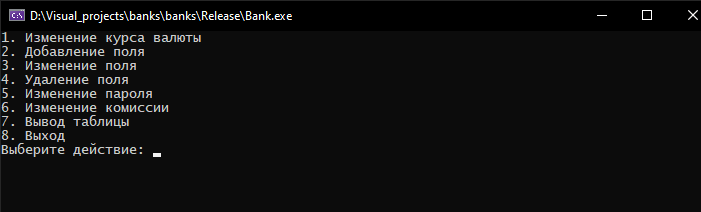


Рисунок 13. Режим редактирования, вход в который был произведён с помощью нового пароля

* 1. **Описание применения средств отладки**

В ходе написания курсового проекта было получено следующее сообщение:

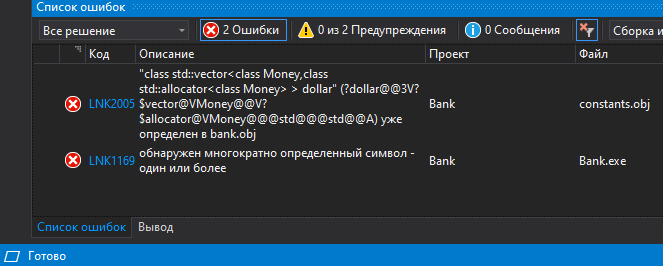


Рисунок 14. Обнаружена ошибка

Суть ошибки заключается в векторе, в котором хранится информация о валютах. У вектора отсутствует спецификатор extern, который сообщает компилятору, что следующие за ним типы и имена переменных объявляются где-то в другом месте. Другими словами, extern позволяет компилятору знать о типах и именах глобальных переменных без действительного создания этих переменных.

Для решения данной ошибки нужно перейти в файл bank.cpp и обратить внимание на 4 строку (рис 16).

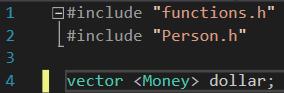


Рисунок 15. Ошибка в файле bank.cpp

Необходимо изменить строку vector <Money> dollar; на строку extern vector <Money> dollar;, тем самым указав компилятору, что вектор определён в другом месте.

Результатом отладки стало следующее сообщение:

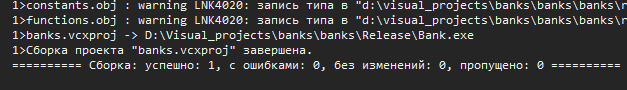


Рисунок 16. Результат отладки программы

* 1. **Анализ оптимальности использования памяти и быстродействия**

В данном разделе представлен анализ оптимальности использования памяти и быстродействия программы.

Принятые оптимальные решения:

* В целях оптимизации кода, было принято решение часто используемые блоки кода сделать функциями. Примеры таких функций представлены в Листинге 4.

Листинг 4.

void print() //вывод таблицы с данными на экран

{

for (int i = 0; i < dollar.size(); i++)

{

cout << i + 1 << ". ";

dollar[i].m\_print\_name();

cout << " ";

dollar[i].m\_print\_value();

cout << endl;

}

}

void vector\_to\_file() //запись таблицы с данными в файл "filename"

{

ofstream fin(filename);

for (int i = 0; i < dollar.size(); i++)

fin << dollar[i].m\_name << " " << dollar[i].m\_value << endl;

fin.close();

}

**Глава 3. Эксплуатационная часть**

* 1. **Руководство оператора**

1. **Назначение программы**

Основной функцией проекта "Bank\_helper" является автоматизация процесса покупки валюты.

**Заключение**

**Список литературы и интернет-источников**

**Приложения**