

**Колледж космического машиностроения и технологий**

**Сопровождающий документ**

**По МДК.01.02 «Прикладное программирование»**

**Тема: «Программное обеспечение «WindowsFormsApp1»»**

Выполнил: Соколов Ярослав Алексеевич

Королёв 2020

[**Введение**](#_7j6uleqi2dt7) **2**

[**Серверная часть**](#_mm5d09897gy8) **3**

[Структура БД](#_9dtmkiod7pmf) 3

[Процедуры сервера](#_ag888qvjiyn3) 3

[**Клиентская часть**](#_c40iwixgsx0y) **6**

# 

# **Введение**

WindowsFormsApp1 - это программное обеспечения под платформу Windows для наблюдения и управления Bitcoin кошельками.

Программа состоит из нескольких модулей:

* Серверная часть
* Клиентская часть

Целью данного программное обеспечения является облегчение работы, повышение удобства и возможность одновременной работы с несколькими кошельками сразу.

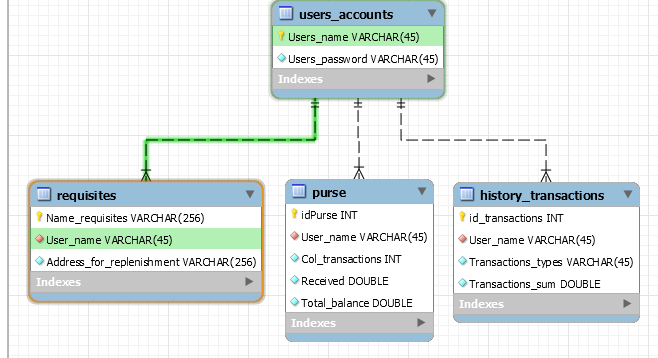
В данном документе будут рассмотрены:

* Серверная часть:
  + Структура БД
  + Процедуры сервера
* Клиентская часть:
  + Диаграмма прецедентов;
  + Сценарий использования;
  + Диаграмма классов;
  + Окно авторизации;
  + Окно главного меню;
  + Окно истории транзакций;
  + Окно кошелька;
  + Окно реквизитов;

# **Серверная часть**

## Структура БД

База данных состоит из 4 таблиц: таблица пользователей, таблица историй транзакций, таблица кошельков, таблица реквизитов.



## **Процедуры сервера**

На сервер находятся процедуры: удаления, получения новых данных, добавления новой строки, процедуры обновления данных.

**Процедура удаления**: удаляет реквизиты по имени которое получает.

Листинг 1: процедура удаления.

CREATE DEFINER=`root`@`localhost` PROCEDURE `delete\_row`(IN IN\_Name\_requisites VARCHAR(256))

BEGIN

DELETE FROM `bts\_project`.`requisites` WHERE (`Name\_requisites` = IN\_Name\_requisites);

END

**Процедура получения новых данных:** процедура для получения данных о транзакциях и изменению баланса.

Листинг 2: процедура получения новых данных.

CREATE DEFINER=`root`@`localhost` PROCEDURE `in\_data`(IN in\_idPurse INT, in transactions\_type VARCHAR(45), in transactions\_sum DOUBLE)

BEGIN

DECLARE transact INT; DECLARE balance double; DECLARE User\_n VARCHAR(45);

Set transact = (select Col\_transactions from bts\_project.purse where idPurse = in\_idPurse);

Set balance = (select Total\_balance from bts\_project.purse where idPurse = in\_idPurse) + transactions\_sum;

Set User\_n = (select User\_name from bts\_project.purse where idPurse = in\_idPurse);

INSERT INTO `bts\_project`.`history\_transactions` (`User\_name`, `Transactions\_types`, `Transactions\_sum`) VALUES (User\_n, transactions\_type, transactions\_sum);

UPDATE `bts\_project`.`purse` SET `Col\_transactions` =@transact+1 , `Total\_balance` = balance WHERE (`idPurse` = in\_idPurse);

END

**Процедура добавления новой строки**: добавляет новую строку с параметрами которые получила.

Листинг 3: процедура добавления новой строки.

CREATE DEFINER=`root`@`localhost` PROCEDURE `new\_row`(IN IN\_Name\_requisites VARCHAR(256), IN IN\_User\_name VARCHAR(45), IN IN\_Address\_for\_replenishment VARCHAR(256))

BEGIN

INSERT INTO `bts\_project`.`requisites` (`Name\_requisites`, `User\_name`, `Address\_for\_replenishment`) VALUES (IN\_Name\_requisites, IN\_User\_name, IN\_Address\_for\_replenishment);

END

**Процедура обновления данных**: обновляет данные строки.

Листинг 4: процедура удаления.

CREATE DEFINER=`root`@`localhost` PROCEDURE `update\_row`(IN IN\_Name\_requisites VARCHAR(256), IN IN\_User\_name VARCHAR(45), IN IN\_Address\_for\_replenishment VARCHAR(256))

BEGIN

UPDATE `bts\_project`.`requisites` SET `User\_name` = IN\_User\_name, `Address\_for\_replenishment` = IN\_Address\_for\_replenishment WHERE (`Name\_requisites` = IN\_Name\_requisites);

END

# 

# **Клиентская часть**

**Диаграмма прецедентов**

Разработчик выполняет поддержку программы посредством обновлений и сбором информации об ошибках в программе.

Системный администратор устанавливает обновления, создает учетные записи и выявляет ошибки программы.

Пользователь программы посредством ввода множеств и выбора действия, которое выполнит программа, получает выходные данные.

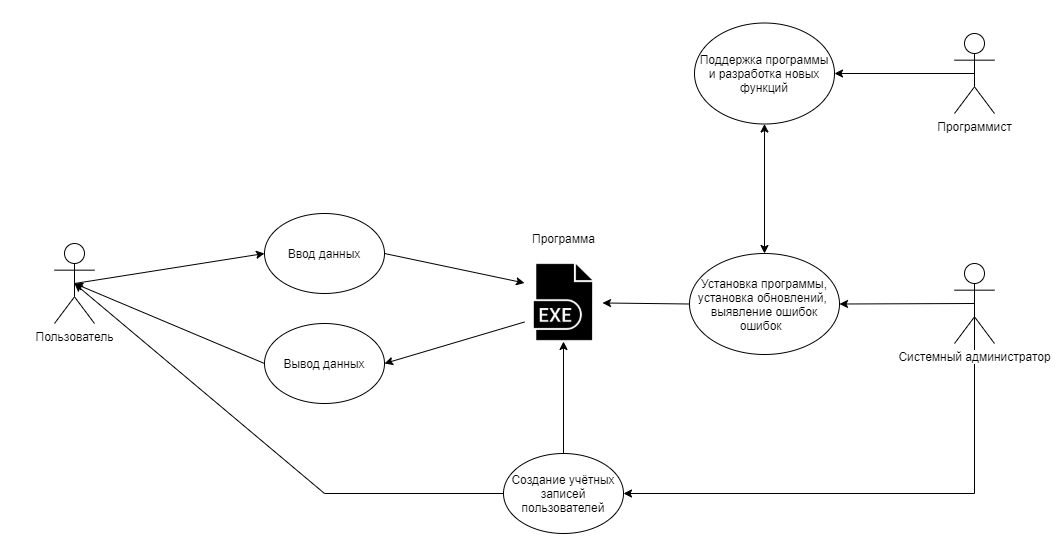


Рисунок 1. Диаграмма прецедентов.

**Сценарий использования**

Сценарий использования:

1. Пользователь запускает программу;
2. Входит в систему;
3. В главном меню выбирает действие:
   1. Окно авторизации;
   2. Кошелек;
   3. Реквизиты;
   4. Выход из системы;
4. В выбранном окне пользователь получает данные или вводит их;
5. Выходит из программы.

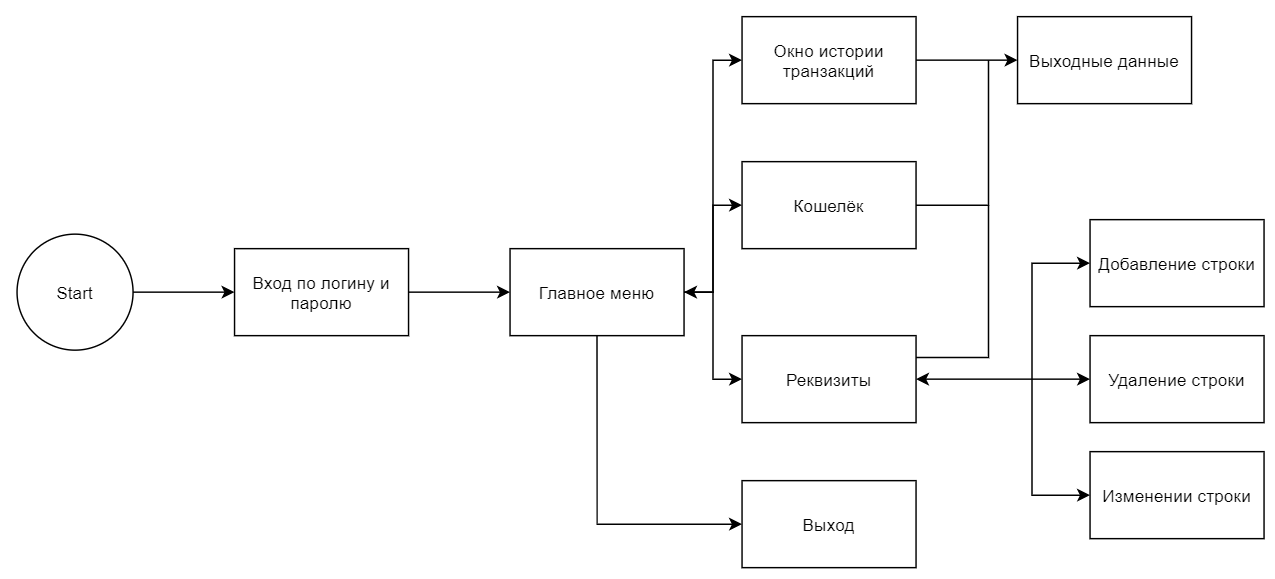


Рисунок 2. Сценарий использования

**Диаграмма классов**

Описание диаграммы классов: Пользователь вводит данные, программа обращается к классу Login\_window для того, чтобы пройти авторизацию, после которой обращается к классу MainForm, откуда получает доступ к обращению, к дочерним окнам.

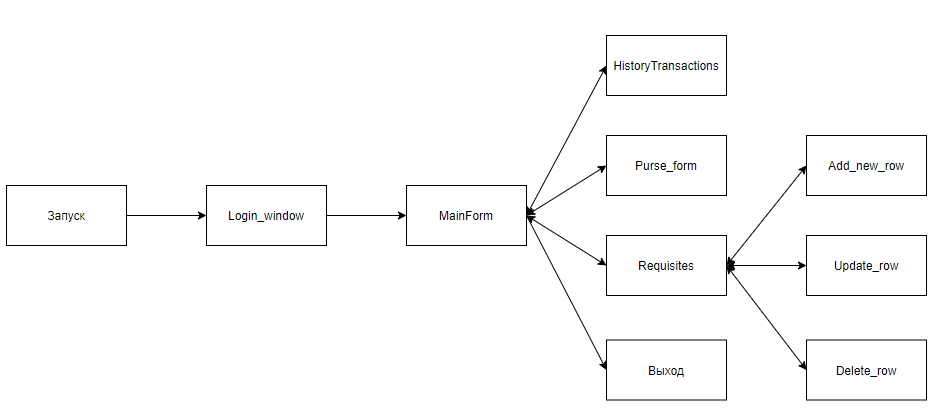


Рисунок 3. Диаграмма классов

**Состав программы:**

1. Окно авторизации;
2. Главное меню;
3. Кошелек;
4. Реквизиты;
   1. Вспомогательное окно добавления строки;
   2. Вспомогательное окно удаления строки;
   3. Вспомогательное окно изменения строки;

**Окно авторизации**

Окно авторизации состоит из полей: логина и пароля. При нажатии на кнопку происходит запрос на соответствие введенных данных с табличными.

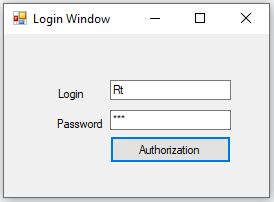


Рисунок 4. Окно авторизации

**Главное меню**

Окно главного меню состоит из кнопок: истории транзакций, кошелька, реквизитов, кнопки возврата.

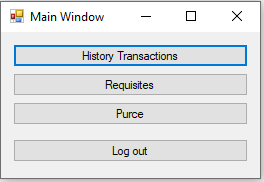


Рисунок 5. Окно главного меню.

**История транзакций**

Окно истории транзакций состоит из: таблицы, кнопки обновления, кнопки возврата в главное меню.

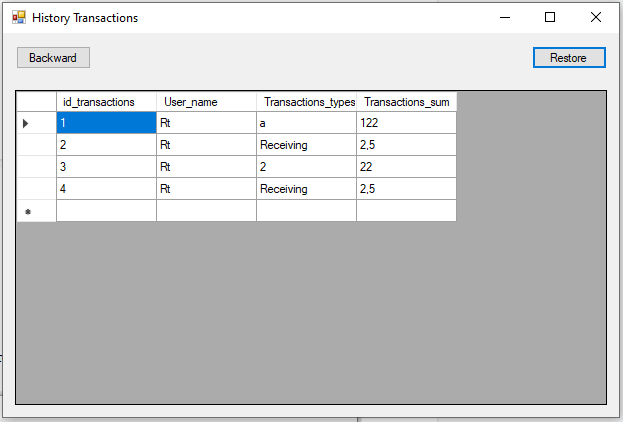
****

Рисунок 6. Окно истории транзакций

**Кошелек**

Окно кошелек состоит из: таблицы, кнопки обновления, кнопки возврата в главное меню.

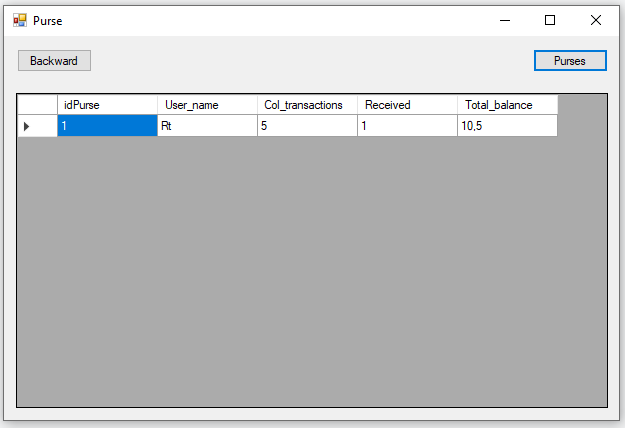


Рисунок 7. Окно кошелька

**Реквизиты**

Окно реквизиты состоит из: таблицы, кнопки обновления, кнопки добавления, кнопки удаления, кнопки изменения, кнопки возврата в главное меню.

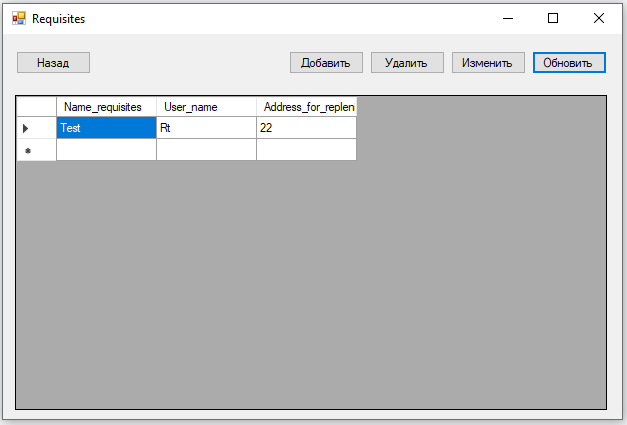


Рисунок 8. Окно реквизитов

**Вспомогательное окно добавления строки**

Окно добавления строки состоит из: полей ввода имени реквизитов и адреса, кнопка добавления.

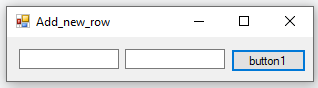


Рисунок 9. Вспомогательное окно добавления строки

**Вспомогательное окно удаления строки**

Окно удаления строки состоит из: выпадающего меню в котором содержатся наименования реквизитов, кнопка удаления.

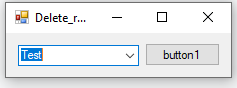


Рисунок 10. Вспомогательное окно удаления строки

**Вспомогательное окно изменения строки**

Окно изменения строки состоит из: выпадающего меню, поля ввода адреса реквизитов, кнопка добавления.

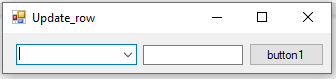


Рисунок 11. Вспомогательное окно изменения строки