

Колледж космического машиностроения и технологий

КУРСОВОЙ ПРОЕКТ

По МДК.01.02 «Прикладное программирование»

Тема: «Разработка приложения "Пункт обмена валюты"»

Выполнил студент

Овчинников П.А.

Группа П1-17

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Подпись)

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

Проверил преподаватель

Гусятинер Л.Б.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Оценка)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(Подпись)

Королёв 2020 г.

Оглавление

[Введение 3](#_Toc43455764)

[Глава 1. Теоретическая часть 4](#_Toc43455765)

[1.1 Изучение предметной области 4](#_Toc43455766)

[1.2 Изучение существующих разработок 6](#_Toc43455767)

[Глава 2. Проектная часть 10](#_Toc43455768)

[2.1 Построение диаграммы прецедентов 10](#_Toc43455769)

[2.2 Выбор инструментов 12](#_Toc43455770)

[2.3 Проектирование сценария 13](#_Toc43455771)

[2.4 Построение диаграммы классов 14](#_Toc43455772)

[2.5 Описание главного модуля 15](#_Toc43455773)

[2.6 Описание спецификаций к модулям 17](#_Toc43455774)

[2.7 Описание модулей 18](#_Toc43455775)

[2.8 Описание тестовых наборов модулей 19](#_Toc43455776)

[2.9 Описание применения средств отладки 20](#_Toc43455777)

[2.10 Анализ оптимальности использования памяти и быстродействия 21](#_Toc43455778)

[Глава 3. Эксплуатационная часть 22](#_Toc43455779)

[3.1 Руководство оператора 22](#_Toc43455780)

[Заключение 23](#_Toc43455781)

[Список литературы и интернет-источников 24](#_Toc43455782)

# Введение

Целью данного курсового проекта является написание программы «Пункт обмена валюты», такие пункты являются актуальными уже довольно долгое время и не прекратят таковыми быть, пока не останется на Земле одна единственная валюта. Моей задачей было упростить работу для работающих в таких пунктах людей, путём создания программы с базой данных и дополнительными вычисляющими функциями.

В данной записке будут описаны цели программы, ее применение и сама программа. Записка разделена на три главы.

В первой главе будет рассмотрена предметная область данной темы, а также приведены примеры уже существующих разработок в данной области, их описание, характеристики и отзывы пользователей.

Во второй части будут рассмотрены инструменты и модули, которые были разработаны, структура программной части и листинги ключевых частей программных модулей.

В третьей части будет рассмотрено руководство для пользователей, а также изображение графического интерфейса программы.

В результате этих трех главах будет полностью описана разработка, ее суть и назначения, а также сама область применения, что полностью опишет весь процесс разработки.

# Глава 1. Теоретическая часть

## Изучение предметной области

Валютно-денежная составляющая экономики – необходимая часть экономики любой страны. Ежеминутно меняются курсы тысяч разных валют и совершаются тысячи разных сделок от разных людей.

В независимости от стран, правительство использует различные способы отслеживания информации о сделках, и те, достаточно продвинутые из них, в плане технологий, используют компьютерные программы.

Интерфейс таких программ чаще всего простой и понятный для обычных пользователей.

Цели использования системы баз данных обменного пункта:

1. Упрощённое занесение информации в базу данных.
2. Отслеживание любой записи по коду.
3. Расчет различных статистических показателей: доля сделок по каждой валюте (в рублях) к общему объёму сделок, максимальный размер сделки (в рублях).

Основные требования:

1. Отслеживание данных клиентов, сделок, валют.
2. Интуитивный и понятный интерфейс

Современные пункты обмена валют используют программы и цифровые базы данных для быстрого и удобного доступа к и истории сделок и другой необходимой информации для работы.

## Изучение существующих разработок

* + 1. The World’s Trusted Currency Authority

TWTCA (xe.com) - продукт компании Euronet Worldwide, это сайт предназначенный для отслеживания данных по стоимости валют и использования этой информации для коммерции, так же через сайт возможен перевод финансов и конвертация из одной валюты в другую.

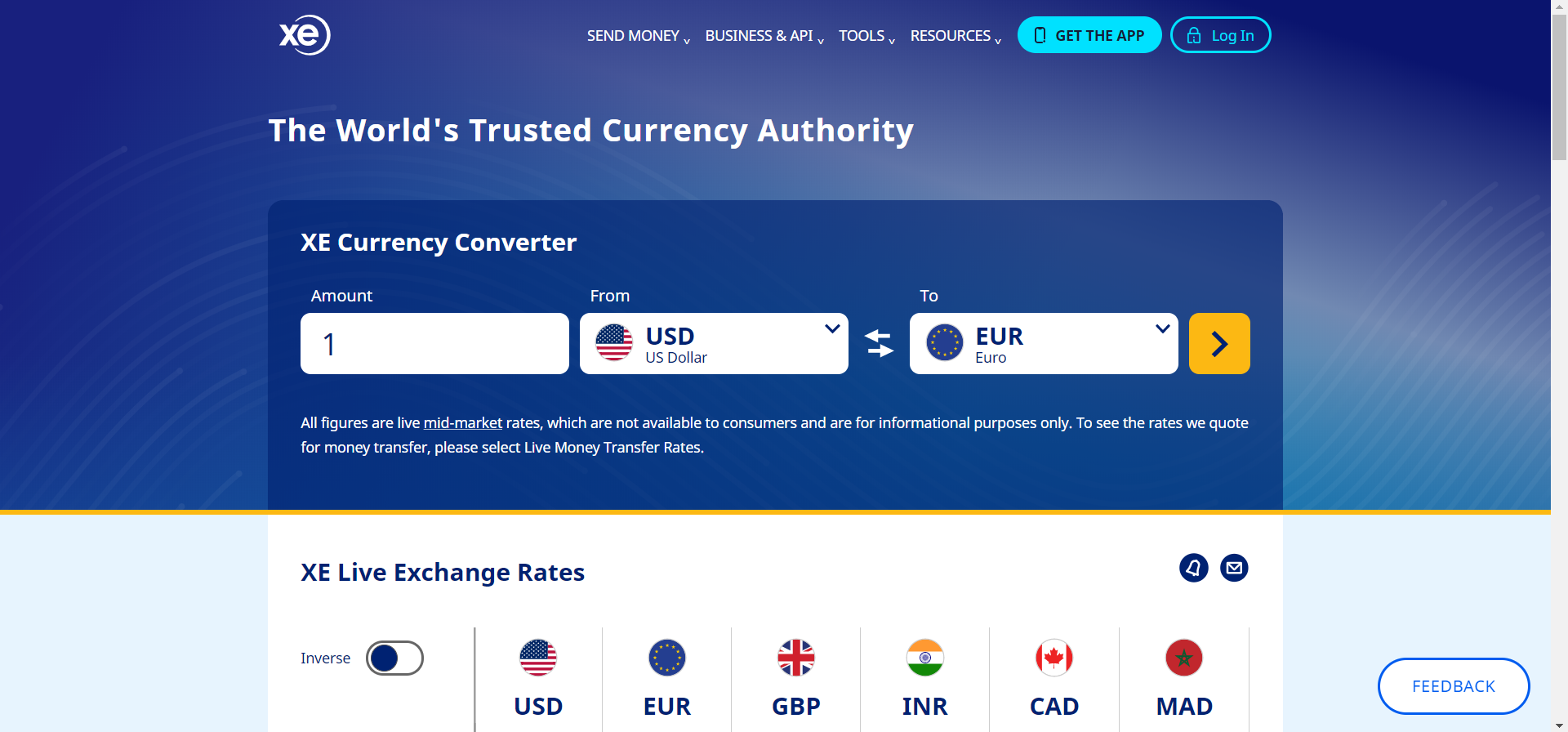


Рисунок 1. Главная страница.



Рисунок 2. Пример конвертации.

Как мы можем видеть из рисунка 2, конвертация на сайте происходит просто и быстро. В окно под надписью “Amount” пишется необходимая сумма переводимой валюты, окно правее при нажатии открывает меню для выбора переводимой валюты, а крайнее правое окно открывает меню для выбора валюты, в которую необходимо перевести. При нажатии на оранжевую кнопку происходит перевод и на экран выводится искомая информация.

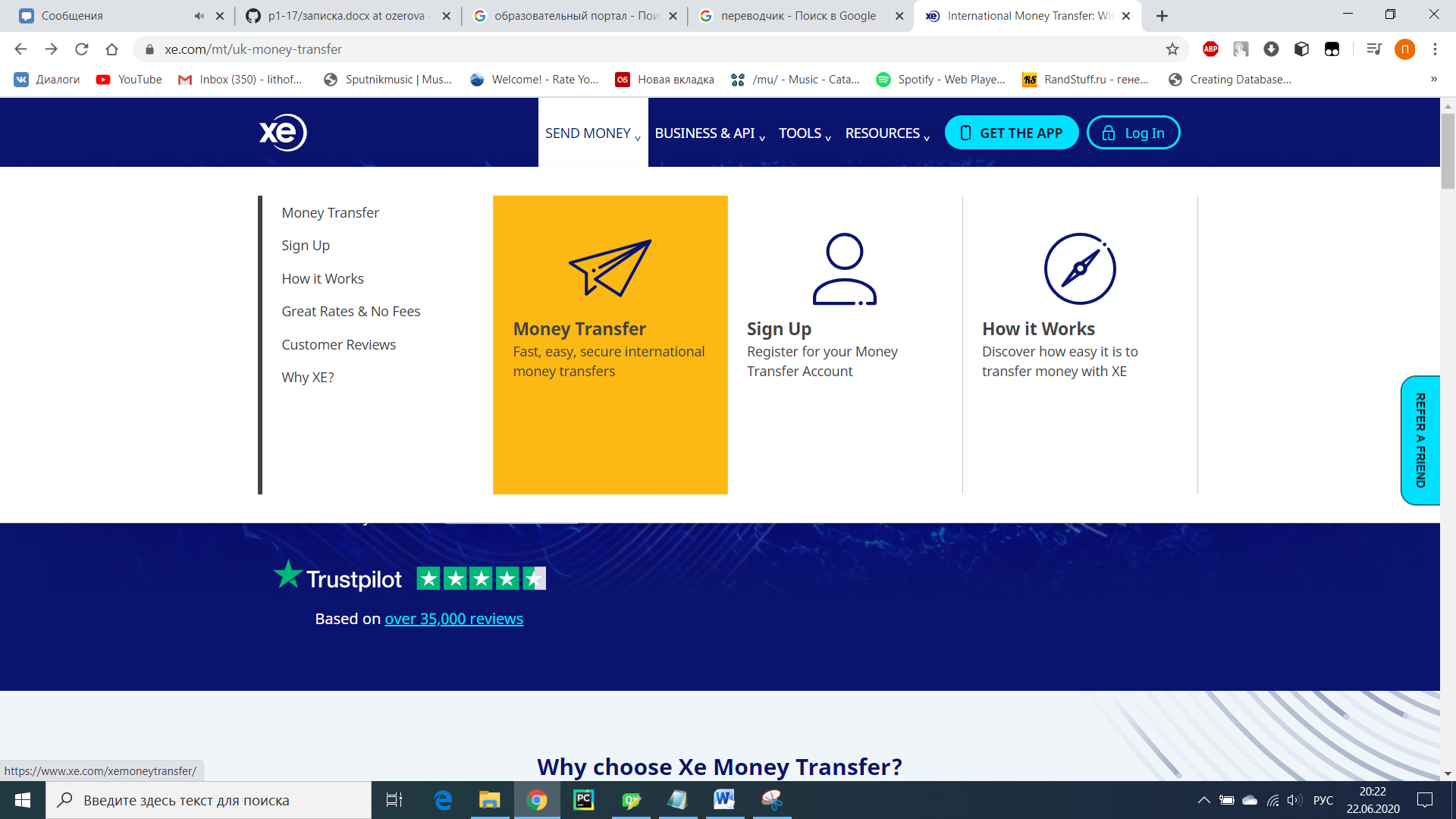


Рисунок 3. Перевод денег.

Для осуществления денежного перевода на сайте необходимо выполнить регистрацию с подтверждением электронной почты. Далее интуитивный интерфейс и подсказки позволят выполнить быстрый лёгкий и защищённый межнациональный перевод.

Данный сайт предоставляет крайне удобный доступ к валютным статистическим данным и позволяет производить защищённые транзакции.

1.2.2.CurrencyXchanger.

CurrencyXchanger (CXR) – это приложение для ведения бизнеса. Оно предоставляет возможность автоматизированных переводов, ведения бухгалтерских учётов, отчетностей и других элементов работы бухгалтера. Приложение может быть использовано как на отдельных компьютерах, так и на целых компьютерных сетях, как локальных, так и сетевых, возможна функция передачи информации через облачную технологию.

Главный офис разработчиков – Clear View Systems, расположен в Канаде, и благодаря их труду, CXR набирает популярность. На данный момент программа используется более чем 350-тью компаниями ретейлеров и оптовиков. Разработчики CXR получили оценку A+ (высший балл), от ассоциации Better Business Bureau (BBB) за функционал.

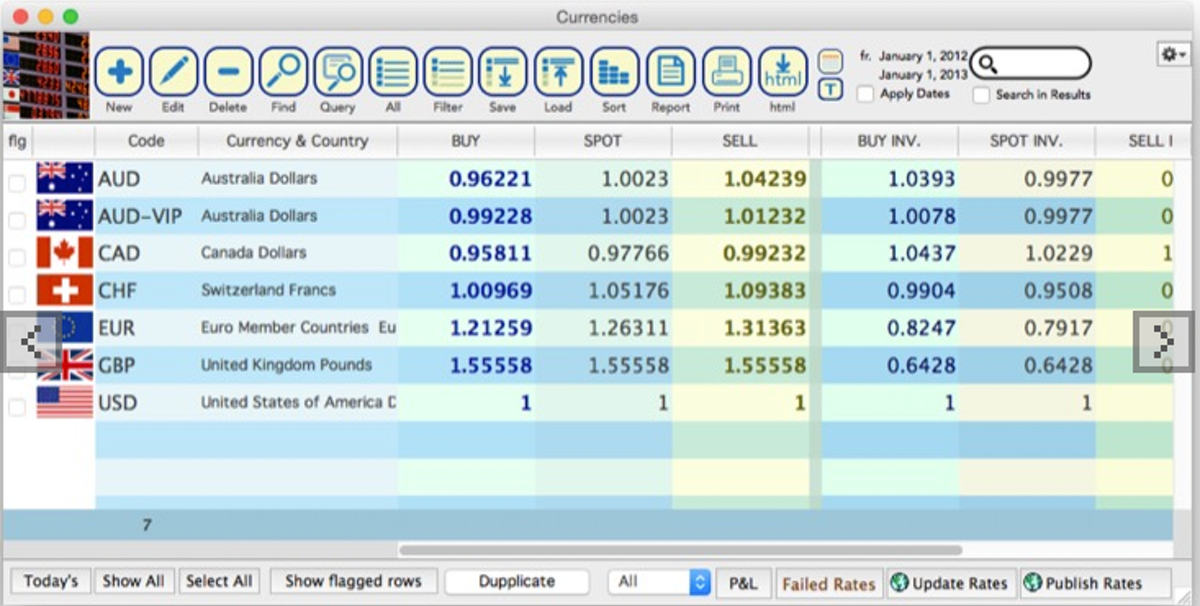


Рисунок 4 Интерфейс программы CXR на MacOS

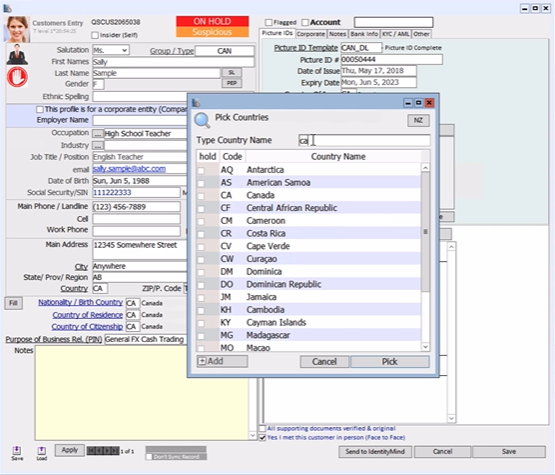


Рисунок 5 Пример регистрации клиента

Существует несколько версий CXR, находящиеся в разных ценовых категориях и имеющих соответствующий функционал.

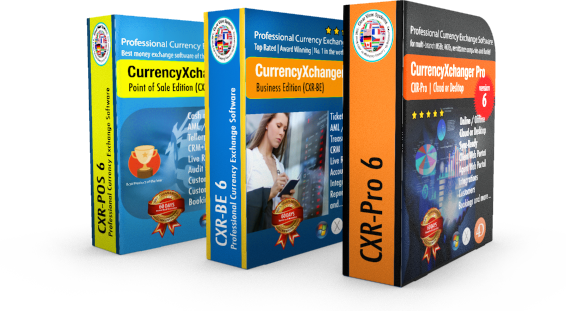


Рисунок 6 Варианты пакетов установки CXR

# **Глава 2. Проектная часть**

## Построение диаграммы прецедентов

Были построены диаграммыпрецедентов для «Клиента» и «Рабочего»

На данных диаграммах мы можем сопоставить функционал для каждого пользователя

Диаграмма прецедентов (рис. 7). Показывает, что рабочий имеет множество выборов действия: просмотр данных по клиентам, сделкам, валюте, получение статистических данных, внесение данных в любую из предоставленных таблиц. Трейдер (покупатель), будучи клиентом компании будет иметь доступ только к возможности создания новой сделки и просмотра данных о валюте.

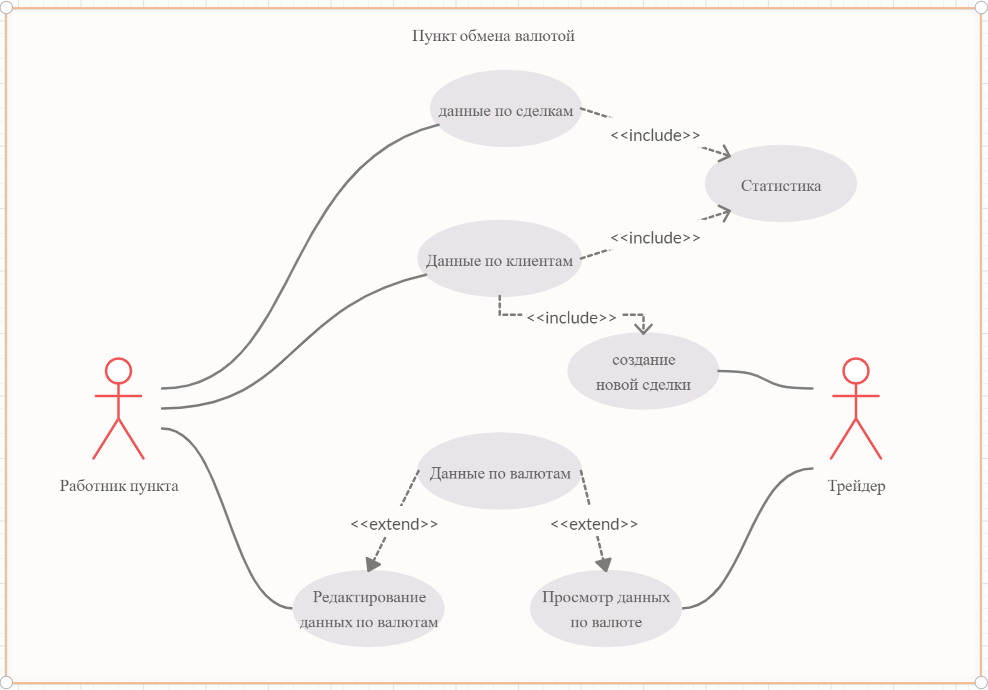


Рисунок 7 Диаграмма прецедентов.

## Выбор инструментов

При выборе инструментов я исходил из некоторых критериев, который я считал самыми главными в разработке.

Важность критерия я выбирал из: низкая, средняя, высокая.

Таблица №1 важность критерия:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Критерий | Участие в корпоративном проекте | Простота сопровождения | Наличие библиотек связанных с выбранной Базой данных | Наличие документации на русском языке | Скорость разработки |
| Важность критерия | Низкая | Средняя | Низкая | Низкая | Высокая |
| Критерий | Простота разработки графического интерфейса | Скорость обучения |  |  |  |
| Важность критерия | Высокая | Высокая |  |  |  |

Исходя из этих критериев, я сравнил 3 языка программирования от 0 до 10 баллов за критерий исходя из своих предпочтений и знаний.

Таблица №2 сравнение языков программирования по критериям:

|  |  |  |  |
| --- | --- | --- | --- |
|  | C++ | C# | Python |
| Участие в корпоративном проекте | 7 | 10 | 7 |
| Простота сопровождения | 7 | 6 | 8 |
| Наличие библиотек | 3 | 4 | 10 |
| Наличие документации на русском языке | 8 | 7 | 8 |
| Скорость разработки | 8 | 6 | 10 |
| Простота разработки графического интерфейса | 7 | 8 | 10 |
| Скорость обучения | 10 | 5 | 10 |
| Итого баллов | 50 | 46 | 63 |

По результатам сравнения мной был выбран язык программирования Python.

## Проектирование сценария

Данная программа ориентирована на такой сценарий (рис. 8).

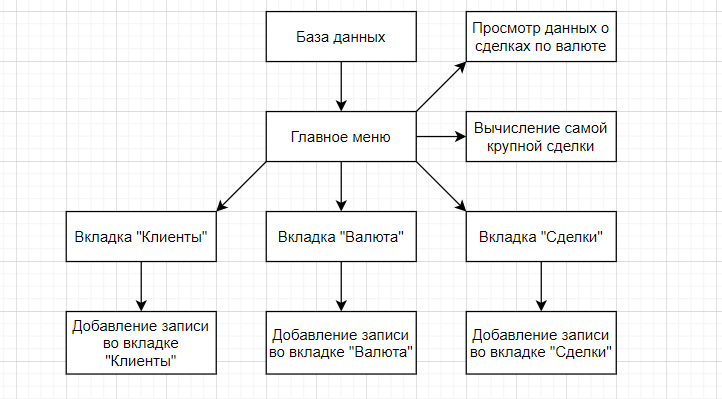


Рисунок 8. Сценарий программы

После запуска программы пользователь имеет несколько вариантов действий

1. Зайти во вкладку «Клиенты»
2. Зайти во вкладку «Валюта»
3. Зайти во вкладку «Сделки»
4. Посмотреть данные о доле сделок повалютно
5. Вычислить размер крупнейшей сделки
6. Добавить запись в любую из перечисленных выше вкладок

При добавлении записи, она автоматически помещается в созданный файл типа txt.

При выборе пунктов 4 и 5, информация будет выведена на консоль.

## Построение диаграммы классов

В данном параграфе представлены все классы, которые используются в программе, а также их взаимосвязи (рис. 9).

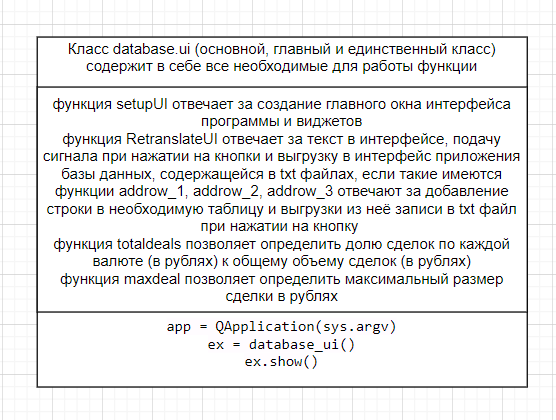


Рисунок 9. Диаграмма классов

## Описание главного модуля

В главный модуль разработки входит один класс, который отвечает за взаимосвязь со всеми компонентами базы данных, а также за их включение.

Данный модуль состоит из одного кода, который приведен в листинге 1.

Листинг 1:

class database\_ui(QtWidgets.QWidget):  
  
 def \_\_init\_\_(self):  
 QtWidgets.QWidget.\_\_init\_\_(self)  
 self.setupUi(self)  
  
 def setupUi(self, UI\_Tool):  
 UI\_Tool.setObjectName("TPayne\_MySQL\_Tool")  
 UI\_Tool.resize(610, 431)  
 self.gridLayout = QtWidgets.QGridLayout(UI\_Tool)  
 self.gridLayout.setObjectName("gridLayout")  
 self.verticalLayout = QtWidgets.QVBoxLayout()  
 self.verticalLayout.setObjectName("verticalLayout")  
 self.label = QtWidgets.QLabel(UI\_Tool)  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.label.setFont(font)  
 self.label.setLayoutDirection(QtCore.Qt.LeftToRight)  
 self.label.setScaledContents(True)  
 self.label.setAlignment(QtCore.Qt.AlignCenter)  
 self.label.setObjectName("label")  
 self.verticalLayout.addWidget(self.label)  
 self.horizontalLayout = QtWidgets.QHBoxLayout()  
 self.horizontalLayout.setObjectName("horizontalLayout")  
 self.tabWidget = QtWidgets.QTabWidget(UI\_Tool)  
 self.tabWidget.setObjectName("tabWidget")  
 self.tab = QtWidgets.QWidget()  
 self.tab.setObjectName("tab")  
 self.verticalLayout\_5 = QtWidgets.QVBoxLayout(self.tab)  
 self.verticalLayout\_5.setObjectName("verticalLayout\_5")  
 self.tableWidget\_1 = QtWidgets.QTableWidget(self.tab)  
 self.tableWidget\_1.setObjectName("tableWidget\_1")  
 self.tableWidget\_1.setColumnCount(3)  
 self.tableWidget\_1.setRowCount(1)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_1.setVerticalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_1.setHorizontalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_1.setHorizontalHeaderItem(1, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_1.setHorizontalHeaderItem(2, item)  
 self.tableWidget\_1.horizontalHeader().setCascadingSectionResizes(True)  
 self.tableWidget\_1.horizontalHeader().setSortIndicatorShown(False)  
 self.tableWidget\_1.horizontalHeader().setStretchLastSection(False)  
 self.tableWidget\_1.verticalHeader().setVisible(False)  
 self.verticalLayout\_5.addWidget(self.tableWidget\_1)  
 self.pushButton\_3 = QtWidgets.QPushButton(self.tab)  
 self.pushButton\_3.setObjectName("pushButton\_3")  
 self.verticalLayout\_5.addWidget(self.pushButton\_3)  
 self.tabWidget.addTab(self.tab, "")  
 self.tab\_2 = QtWidgets.QWidget()  
 self.tab\_2.setObjectName("tab\_2")  
 self.verticalLayout\_4 = QtWidgets.QVBoxLayout(self.tab\_2)  
 self.verticalLayout\_4.setObjectName("verticalLayout\_4")  
 self.tableWidget\_2 = QtWidgets.QTableWidget(self.tab\_2)  
 self.tableWidget\_2.setObjectName("tableWidget\_2")  
 self.tableWidget\_2.setColumnCount(3)  
 self.tableWidget\_2.setRowCount(1)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_2.setVerticalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_2.setHorizontalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_2.setHorizontalHeaderItem(1, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_2.setHorizontalHeaderItem(2, item)  
 self.tableWidget\_2.verticalHeader().setVisible(False)  
 self.verticalLayout\_4.addWidget(self.tableWidget\_2)  
 self.pushButton\_4 = QtWidgets.QPushButton(self.tab\_2)  
 self.pushButton\_4.setObjectName("pushButton\_4")  
 self.verticalLayout\_4.addWidget(self.pushButton\_4)  
 self.tabWidget.addTab(self.tab\_2, "")  
 self.tab\_3 = QtWidgets.QWidget()  
 self.tab\_3.setObjectName("tab\_3")  
 self.verticalLayout\_3 = QtWidgets.QVBoxLayout(self.tab\_3)  
 self.verticalLayout\_3.setObjectName("verticalLayout\_3")  
 self.tableWidget\_3 = QtWidgets.QTableWidget(self.tab\_3)  
 self.tableWidget\_3.setObjectName("tableWidget\_3")  
 self.tableWidget\_3.setColumnCount(4)  
 self.tableWidget\_3.setRowCount(1)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_3.setVerticalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_3.setHorizontalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_3.setHorizontalHeaderItem(1, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_3.setHorizontalHeaderItem(2, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_3.setHorizontalHeaderItem(3, item)  
 self.tableWidget\_3.verticalHeader().setVisible(False)  
 self.verticalLayout\_3.addWidget(self.tableWidget\_3)  
 self.pushButton\_5 = QtWidgets.QPushButton(self.tab\_3)  
 self.pushButton\_5.setObjectName("pushButton\_5")  
 self.verticalLayout\_3.addWidget(self.pushButton\_5)  
 self.tabWidget.addTab(self.tab\_3, "")  
 self.horizontalLayout.addWidget(self.tabWidget)  
 self.verticalLayout.addLayout(self.horizontalLayout)  
 self.gridLayout.addLayout(self.verticalLayout, 0, 0, 1, 1)  
 self.pushButton = QtWidgets.QPushButton(UI\_Tool)  
 self.pushButton.setAutoDefault(False)  
 self.pushButton.setDefault(False)  
 self.pushButton.setFlat(False)  
 self.pushButton.setObjectName("pushButton")  
 self.gridLayout.addWidget(self.pushButton, 1, 0, 1, 1)  
 self.pushButton\_2 = QtWidgets.QPushButton(UI\_Tool)  
 self.pushButton\_2.setObjectName("pushButton\_2")  
 self.gridLayout.addWidget(self.pushButton\_2, 2, 0, 1, 1)  
  
 self.retranslateUi(UI\_Tool)  
 self.tabWidget.setCurrentIndex(0)  
 QtCore.QMetaObject.connectSlotsByName(UI\_Tool)  
  
 def retranslateUi(self, TPayne\_MySQL\_Tool):  
 \_translate = QtCore.QCoreApplication.translate  
 TPayne\_MySQL\_Tool.setWindowTitle(\_translate("TPayne\_MySQL\_Tool", "CEPDB Manager"))  
 self.label.setText(\_translate("TPayne\_MySQL\_Tool", "База данных пункта обмена валютой"))  
 item = self.tableWidget\_1.verticalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "New Row"))  
 item = self.tableWidget\_1.horizontalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Код клиента"))  
 item = self.tableWidget\_1.horizontalHeaderItem(1)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "ФИО"))  
 item = self.tableWidget\_1.horizontalHeaderItem(2)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Паспорт"))  
 self.pushButton\_3.setText(\_translate("TPayne\_MySQL\_Tool", "Добавить ряд для записи"))  
 self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab), \_translate("TPayne\_MySQL\_Tool", "Клиенты"))  
 item = self.tableWidget\_2.verticalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "1"))  
 item = self.tableWidget\_2.horizontalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Код валюты"))  
 item = self.tableWidget\_2.horizontalHeaderItem(1)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Название"))  
 item = self.tableWidget\_2.horizontalHeaderItem(2)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Курс продажи"))  
 self.pushButton\_4.setText(\_translate("TPayne\_MySQL\_Tool", "Добавить ряд для записи"))  
 self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab\_2), \_translate("TPayne\_MySQL\_Tool", "Валюта"))  
 item = self.tableWidget\_3.verticalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "New Row"))  
 item = self.tableWidget\_3.horizontalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Код сделки"))  
 item = self.tableWidget\_3.horizontalHeaderItem(1)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Код клиента"))  
 item = self.tableWidget\_3.horizontalHeaderItem(2)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Код валюты"))  
 item = self.tableWidget\_3.horizontalHeaderItem(3)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Сумма"))  
 self.pushButton\_5.setText(\_translate("TPayne\_MySQL\_Tool", "Добавить ряд для записи"))  
 self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab\_3), \_translate("TPayne\_MySQL\_Tool", "Сделки"))  
 self.pushButton.setText(\_translate("TPayne\_MySQL\_Tool", "Определить долю сделок по каждой валюте"))  
 self.pushButton\_2.setText(\_translate("TPayne\_MySQL\_Tool", "Определить максимальный размер сделки в рублях"))  
  
 db\_file\_clients = open('database\_clients.txt', 'r')  
 db\_file\_curr = open('database\_curr.txt', 'r')  
 db\_file\_deals = open('database\_deals.txt', 'r')  
  
 for line\_clients in db\_file\_clients.readlines():  
 splited = line\_clients.split()  
 for i in range(0, 2):  
 self.tableWidget\_1.setItem(0, i, QTableWidgetItem(splited[i]))  
 self.tableWidget\_1.setItem(0, 2, QTableWidgetItem(splited[2]))  
 self.tableWidget\_1.insertRow(0)  
  
 for line\_curr in db\_file\_curr.readlines():  
 splited = line\_curr.split()  
 for i in range(0, 2):  
 self.tableWidget\_2.setItem(0, i, QTableWidgetItem(splited[i]))  
 self.tableWidget\_2.setItem(0, 2, QTableWidgetItem(splited[2]))  
 self.tableWidget\_2.insertRow(0)  
  
 for line\_deals in db\_file\_deals.readlines():  
 splited = line\_deals.split()  
 for i in range(0, 2):  
 self.tableWidget\_3.setItem(0, i, QTableWidgetItem(splited[i]))  
 self.tableWidget\_3.setItem(0, 2, QTableWidgetItem(splited[2]))  
 self.tableWidget\_3.setItem(0, 3, QTableWidgetItem(splited[3]))  
 self.tableWidget\_3.insertRow(0)  
  
 db\_file\_clients.close()  
 db\_file\_curr.close()  
 db\_file\_deals.close()  
  
 self.pushButton.pressed.connect(self.totaldeals)  
 self.pushButton\_2.pressed.connect(self.maxdeal)  
 self.pushButton\_3.pressed.connect(self.addrow\_1)  
 self.pushButton\_4.pressed.connect(self.addrow\_2)  
 self.pushButton\_5.pressed.connect(self.addrow\_3)  
  
 def addrow\_1(self):  
 db\_file\_clients = open('database\_clients.txt', 'a')  
 for i in range(0, 3):  
 temp = self.tableWidget\_1.item(0, i).text()  
 db\_file\_clients.write(temp)  
 db\_file\_clients.write(' ')  
 db\_file\_clients.write('\n')  
 db\_file\_clients.close()  
 self.tableWidget\_1.insertRow(0)  
  
 def addrow\_2(self):  
 db\_file\_curr = open('database\_curr.txt', 'a')  
 for i in range(0, 3):  
 temp = self.tableWidget\_2.item(0, i).text()  
 db\_file\_curr.write(temp)  
 db\_file\_curr.write(' ')  
 db\_file\_curr.write('\n')  
 db\_file\_curr.close()  
 self.tableWidget\_2.insertRow(0)  
  
 def addrow\_3(self):  
 db\_file\_deals = open('database\_deals.txt', 'a')  
 for i in range(0, 4):  
 temp = self.tableWidget\_3.item(0, i).text()  
 db\_file\_deals.write(temp)  
 db\_file\_deals.write(' ')  
 db\_file\_deals.write('\n')  
 db\_file\_deals.close()  
 self.tableWidget\_3.insertRow(0)  
  
 def totaldeals(self):  
  
 db\_file\_curr = open('database\_curr.txt', 'r')  
 db\_file\_deals = open('database\_deals.txt', 'r')  
 exit\_list = []  
 deals\_lines = []  
 curr\_lines = []  
 overall = 0  
  
 for line\_deals in db\_file\_deals.readlines():  
 splited\_deals = line\_deals.split()  
 deals\_lines.append(splited\_deals)  
  
 for line\_curr in db\_file\_curr.readlines():  
 splited\_curr = line\_curr.split()  
 curr\_lines.append(splited\_curr)  
 curr\_code = splited\_curr[0]  
 curr\_price = splited\_curr[2]  
 for i in range(len(deals\_lines)):  
 if curr\_code == splited\_deals[2]:  
 val = float(curr\_price) \* float(splited\_deals[3])  
 overall = overall + val  
  
 for i in range(len(curr\_lines)):  
 for j in range(len(deals\_lines)):  
 if curr\_lines[i][0] == deals\_lines[j][2]:  
 exit\_list.append(curr\_lines[i][1])  
 exit\_list.append(float(curr\_lines[i][2])\*float(deals\_lines[j][3])/float(overall))  
 print(\*exit\_list)  
  
 db\_file\_curr.close()  
 db\_file\_deals.close()  
  
 def maxdeal(self):  
  
 db\_file\_curr = open('database\_curr.txt', 'r')  
 db\_file\_deals = open('database\_deals.txt', 'r')  
 exit\_list = []  
 deals\_lines = []  
 curr\_lines = []  
  
 for line\_deals in db\_file\_deals.readlines():  
 splited\_deals = line\_deals.split()  
 deals\_lines.append(splited\_deals)  
  
 for line\_curr in db\_file\_curr.readlines():  
 splited\_curr = line\_curr.split()  
 curr\_lines.append(splited\_curr)  
  
 for i in range(len(curr\_lines)):  
 for j in range(len(deals\_lines)):  
 if curr\_lines[i][0] == deals\_lines[j][2]:  
 exit\_list.append(float(curr\_lines[i][2])\*float(deals\_lines[j][3]))  
 print('Размер максимальной сделки:', max(exit\_list), 'рублей')

## Описание спецификаций к модулям

В данной главе будут описаны публичные члены модулей курсового проекта.

В главном модуле публичным объектом являются setupUI, который представляет из себя главную форму модуля, на который закреплены объекты, предназначенные для взаимосвязи с базой данных.

В модуле отвечающим за заполнение списков, и сортировку соответствующих таблиц так же присутствуют функции, которые будут описаны в следующем параграфе.

## Описание модулей

В данной главе мы разберем три главных модуля.

* Центральный модуль

Данный модуль представляет из себя набор обязательных элементов базы данных таких как таблицы, запросы и транзакции. И центральную функцию, которая представляет из себя настройку базы данных.

* Модуль работы с базами данных.

Модуль представляет из себя несколько графических форм каждая из которых обращается в центральный модуль за своей таблицей из базы данных для дальнейшей работы с ней и подключает свою навигационную систему этой таблице что обеспечивает удобную работу. Так же этот модуль для удобства ввода имеет скрипт для подстановки соответствующих значений в выпадающие списки.

* Модуль просмотра таблиц

Модуль представляет из себя несколько графических форм каждая из которых обращается в центральный модуль за своим запросом из базы данных для дальнейшей его вывода, так же подключает навигацию к полученному запросу. Так же этот модуль имеет два скрипта которые представляют из себя

* 1. Подстановку в выпадающий список соответствующе значения
  2. При выборе значения из списка производит изменение скрипта запроса и выводит новую отсортированную таблицу.

Листинг модуля просмотра таблицы:

def setupUi(self, UI\_Tool):  
 UI\_Tool.setObjectName("TPayne\_MySQL\_Tool")  
 UI\_Tool.resize(610, 431)  
 self.gridLayout = QtWidgets.QGridLayout(UI\_Tool)  
 self.gridLayout.setObjectName("gridLayout")  
 self.verticalLayout = QtWidgets.QVBoxLayout()  
 self.verticalLayout.setObjectName("verticalLayout")  
 self.label = QtWidgets.QLabel(UI\_Tool)  
 font = QtGui.QFont()  
 font.setPointSize(11)  
 self.label.setFont(font)  
 self.label.setLayoutDirection(QtCore.Qt.LeftToRight)  
 self.label.setScaledContents(True)  
 self.label.setAlignment(QtCore.Qt.AlignCenter)  
 self.label.setObjectName("label")  
 self.verticalLayout.addWidget(self.label)  
 self.horizontalLayout = QtWidgets.QHBoxLayout()  
 self.horizontalLayout.setObjectName("horizontalLayout")  
 self.tabWidget = QtWidgets.QTabWidget(UI\_Tool)  
 self.tabWidget.setObjectName("tabWidget")  
 self.tab = QtWidgets.QWidget()  
 self.tab.setObjectName("tab")  
 self.verticalLayout\_5 = QtWidgets.QVBoxLayout(self.tab)  
 self.verticalLayout\_5.setObjectName("verticalLayout\_5")  
 self.tableWidget\_1 = QtWidgets.QTableWidget(self.tab)  
 self.tableWidget\_1.setObjectName("tableWidget\_1")  
 self.tableWidget\_1.setColumnCount(3)  
 self.tableWidget\_1.setRowCount(1)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_1.setVerticalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_1.setHorizontalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_1.setHorizontalHeaderItem(1, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_1.setHorizontalHeaderItem(2, item)  
 self.tableWidget\_1.horizontalHeader().setCascadingSectionResizes(True)  
 self.tableWidget\_1.horizontalHeader().setSortIndicatorShown(False)  
 self.tableWidget\_1.horizontalHeader().setStretchLastSection(False)  
 self.tableWidget\_1.verticalHeader().setVisible(False)  
 self.verticalLayout\_5.addWidget(self.tableWidget\_1)  
 self.pushButton\_3 = QtWidgets.QPushButton(self.tab)  
 self.pushButton\_3.setObjectName("pushButton\_3")  
 self.verticalLayout\_5.addWidget(self.pushButton\_3)  
 self.tabWidget.addTab(self.tab, "")  
 self.tab\_2 = QtWidgets.QWidget()  
 self.tab\_2.setObjectName("tab\_2")  
 self.verticalLayout\_4 = QtWidgets.QVBoxLayout(self.tab\_2)  
 self.verticalLayout\_4.setObjectName("verticalLayout\_4")  
 self.tableWidget\_2 = QtWidgets.QTableWidget(self.tab\_2)  
 self.tableWidget\_2.setObjectName("tableWidget\_2")  
 self.tableWidget\_2.setColumnCount(3)  
 self.tableWidget\_2.setRowCount(1)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_2.setVerticalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_2.setHorizontalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_2.setHorizontalHeaderItem(1, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_2.setHorizontalHeaderItem(2, item)  
 self.tableWidget\_2.verticalHeader().setVisible(False)  
 self.verticalLayout\_4.addWidget(self.tableWidget\_2)  
 self.pushButton\_4 = QtWidgets.QPushButton(self.tab\_2)  
 self.pushButton\_4.setObjectName("pushButton\_4")  
 self.verticalLayout\_4.addWidget(self.pushButton\_4)  
 self.tabWidget.addTab(self.tab\_2, "")  
 self.tab\_3 = QtWidgets.QWidget()  
 self.tab\_3.setObjectName("tab\_3")  
 self.verticalLayout\_3 = QtWidgets.QVBoxLayout(self.tab\_3)  
 self.verticalLayout\_3.setObjectName("verticalLayout\_3")  
 self.tableWidget\_3 = QtWidgets.QTableWidget(self.tab\_3)  
 self.tableWidget\_3.setObjectName("tableWidget\_3")  
 self.tableWidget\_3.setColumnCount(4)  
 self.tableWidget\_3.setRowCount(1)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_3.setVerticalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_3.setHorizontalHeaderItem(0, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_3.setHorizontalHeaderItem(1, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_3.setHorizontalHeaderItem(2, item)  
 item = QtWidgets.QTableWidgetItem()  
 self.tableWidget\_3.setHorizontalHeaderItem(3, item)  
 self.tableWidget\_3.verticalHeader().setVisible(False)  
 self.verticalLayout\_3.addWidget(self.tableWidget\_3)  
 self.pushButton\_5 = QtWidgets.QPushButton(self.tab\_3)  
 self.pushButton\_5.setObjectName("pushButton\_5")  
 self.verticalLayout\_3.addWidget(self.pushButton\_5)  
 self.tabWidget.addTab(self.tab\_3, "")  
 self.horizontalLayout.addWidget(self.tabWidget)  
 self.verticalLayout.addLayout(self.horizontalLayout)  
 self.gridLayout.addLayout(self.verticalLayout, 0, 0, 1, 1)  
 self.pushButton = QtWidgets.QPushButton(UI\_Tool)  
 self.pushButton.setAutoDefault(False)  
 self.pushButton.setDefault(False)  
 self.pushButton.setFlat(False)  
 self.pushButton.setObjectName("pushButton")  
 self.gridLayout.addWidget(self.pushButton, 1, 0, 1, 1)  
 self.pushButton\_2 = QtWidgets.QPushButton(UI\_Tool)  
 self.pushButton\_2.setObjectName("pushButton\_2")  
 self.gridLayout.addWidget(self.pushButton\_2, 2, 0, 1, 1)  
  
 self.retranslateUi(UI\_Tool)  
 self.tabWidget.setCurrentIndex(0)  
 QtCore.QMetaObject.connectSlotsByName(UI\_Tool)  
  
def retranslateUi(self, TPayne\_MySQL\_Tool):  
 \_translate = QtCore.QCoreApplication.translate  
 TPayne\_MySQL\_Tool.setWindowTitle(\_translate("TPayne\_MySQL\_Tool", "CEPDB Manager"))  
 self.label.setText(\_translate("TPayne\_MySQL\_Tool", "База данных пункта обмена валютой"))  
 item = self.tableWidget\_1.verticalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "New Row"))  
 item = self.tableWidget\_1.horizontalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Код клиента"))  
 item = self.tableWidget\_1.horizontalHeaderItem(1)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "ФИО"))  
 item = self.tableWidget\_1.horizontalHeaderItem(2)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Паспорт"))  
 self.pushButton\_3.setText(\_translate("TPayne\_MySQL\_Tool", "Добавить ряд для записи"))  
 self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab), \_translate("TPayne\_MySQL\_Tool", "Клиенты"))  
 item = self.tableWidget\_2.verticalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "1"))  
 item = self.tableWidget\_2.horizontalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Код валюты"))  
 item = self.tableWidget\_2.horizontalHeaderItem(1)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Название"))  
 item = self.tableWidget\_2.horizontalHeaderItem(2)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Курс продажи"))  
 self.pushButton\_4.setText(\_translate("TPayne\_MySQL\_Tool", "Добавить ряд для записи"))  
 self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab\_2), \_translate("TPayne\_MySQL\_Tool", "Валюта"))  
 item = self.tableWidget\_3.verticalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "New Row"))  
 item = self.tableWidget\_3.horizontalHeaderItem(0)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Код сделки"))  
 item = self.tableWidget\_3.horizontalHeaderItem(1)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Код клиента"))  
 item = self.tableWidget\_3.horizontalHeaderItem(2)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Код валюты"))  
 item = self.tableWidget\_3.horizontalHeaderItem(3)  
 item.setText(\_translate("TPayne\_MySQL\_Tool", "Сумма"))  
 self.pushButton\_5.setText(\_translate("TPayne\_MySQL\_Tool", "Добавить ряд для записи"))  
 self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab\_3), \_translate("TPayne\_MySQL\_Tool", "Сделки"))  
 self.pushButton.setText(\_translate("TPayne\_MySQL\_Tool", "Определить долю сделок по каждой валюте"))  
 self.pushButton\_2.setText(\_translate("TPayne\_MySQL\_Tool", "Определить максимальный размер сделки в рублях"))

## Описание тестовых наборов модулей

В этом разделе показано умение применять средства отладки.

В ходе написания курсового проекта при попытке запустить скрипт, окно программы было принуждённо закрыто и было получено данное сообщение:

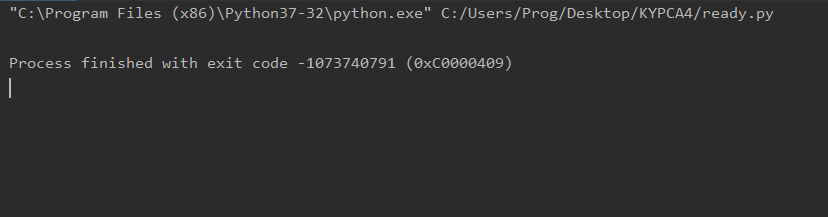


Рисунок 10 До применения средств отладки

После получения данного сообщения были просмотрены 275 строка модуля ready.py и была обнаружена ошибка, которая впоследствии была устранена, а после попытки запуска скрипта получено данное сообщение:

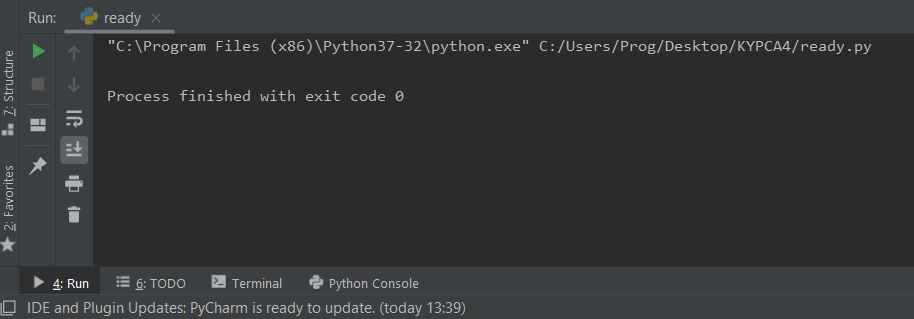


Рисунок 11. После применения средств отладки

Это означает что ошибка была устранена и скрипт запустился.

## **Анализ оптимальности использования памяти и быстродействия**

В данном разделе будет проведен анализ оптимальности использования памяти и быстродействия программы.

Список принятых оптимальных решений:

1. Подключение некоторых модулей внутри функций/методов.

В данном проекте некоторые модули были подключены не в весь модуль, а только в функции/методы, которые его используют. Сделано это потому что работа с локальными объектами быстрее работы с глобальными объектами, к тому же импортироваться эти модули будут только при срабатывании этих функций что ускорит запуск программы.

# Глава 3. Эксплуатационная часть

## 3.1 Руководство оператора

АННОТАЦИЯ

В данном программном документе приведено руководство оператора по применению и эксплуатации программы «Currency Exchange Point», предназначенной для облегчения работы пунктами обменом валюты.

В данном программном документе, в разделе «Назначение программы» указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе «Условия выполнения программы» указаны условия, необходимые для выполнения программы (минимальный состав аппаратных и программных средств и т.п.).

В данном программном документе, в разделе «Выполнение программы» указана последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузку и управляет выполнением программы, а также ответы программы на эти команды.

Оформление программного документа «Руководство оператора» произведено по требованиям ЕСПД (ГОСТ 19.101-77 [[1]](#footnote-1)1), ГОСТ 19.103-77 [[2]](#footnote-2)2), ГОСТ 19.104-78\* [[3]](#footnote-3)3), ГОСТ 19.105-78\* [[4]](#footnote-4)4), ГОСТ 19.106-78\* [[5]](#footnote-5)5), ГОСТ 19.505-79\* [[6]](#footnote-6)6), ГОСТ 19.604-78\* [[7]](#footnote-7)7)).

1. Назначение программы
   1. Функциональное назначение программы

Специальное программное обеспечение «Currency Exchange Point» используется для управления записями, информацией о клиентах, валюте и сделках.

* 1. Эксплуатационное назначение программы

Специальное программное обеспечение «Currency Exchange Point» может эксплуатироваться на объектах любого масштаба в сфере финансового бизнеса для облегчения работы персонала.

* 1. Состав функций
     1. Функция смены таблицы

Эта функция позволяет менять таблицу в зависимости от необходимости.

* + 1. Функция добавления элемента в таблицу.

Эта функция позволяет добавлять нужную информацию в таблицу.

* + 1. Функция высчитывания наибольшей сделки.
    2. Функция вывода доли по сделкам.

1. Условия выполнения программы
   1. Минимальный состав аппаратных средств

ОС: Windows 10

Процессор: Как минимум 1 ГГц или SoC.

ОЗУ: 1 ГБ (для 32-разрядных систем) или 2 ГБ (для 64-разрядных систем).

Место на жестком диске: 20 мб.

Видеоадаптер: DirectX версии не ниже 9 с драйвером WDDM 1.0.

Дисплей: 800 x 600.

* 1. Минимальный состав программных средств

Дополнительные программные средства не требуются.

* 1. Требование к персоналу (пользователю)

Конечный пользователь программы должен обладать практическими навыками работы с графическим пользовательским интерфейсом операционной системы.

1. Выполнение программы
   1. Загрузка и запуск программы

При запуске программы пользователь сразу же может наблюдать на экране окно программы. В окне таблицы уже может быть занесена информация, если заранее уже были созданы файлы database\_clients.txt, database\_curr.txt, database\_deals.txt и в них уже была занесена какая-либо информация. В противном случае эти файлы автоматически создаются программой при первом открытии и не имеют никаких записей при открытии. В случае, представленном на рисунке 12, данные уже были занесены в текстовые файлы.

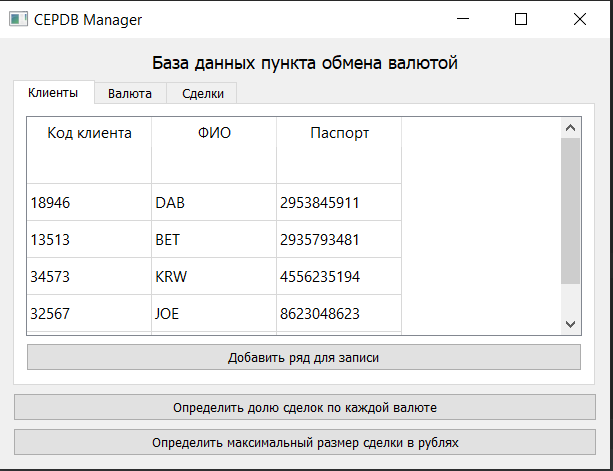


Рисунок 12 Начальный экран программы

* 1. Выполнение программы

Для перемещения по вкладкам базы данных необходимо навести курсор на одну из надписей «Валюта», «Сделки», или «Клиенты» и нажать левую кнопку мыши. По умолчанию программа открывается на вкладке «Клиенты», поэтому чтобы посмотреть базу данных по клиентам при запуске, на эту вкладку нажимать не требуется.

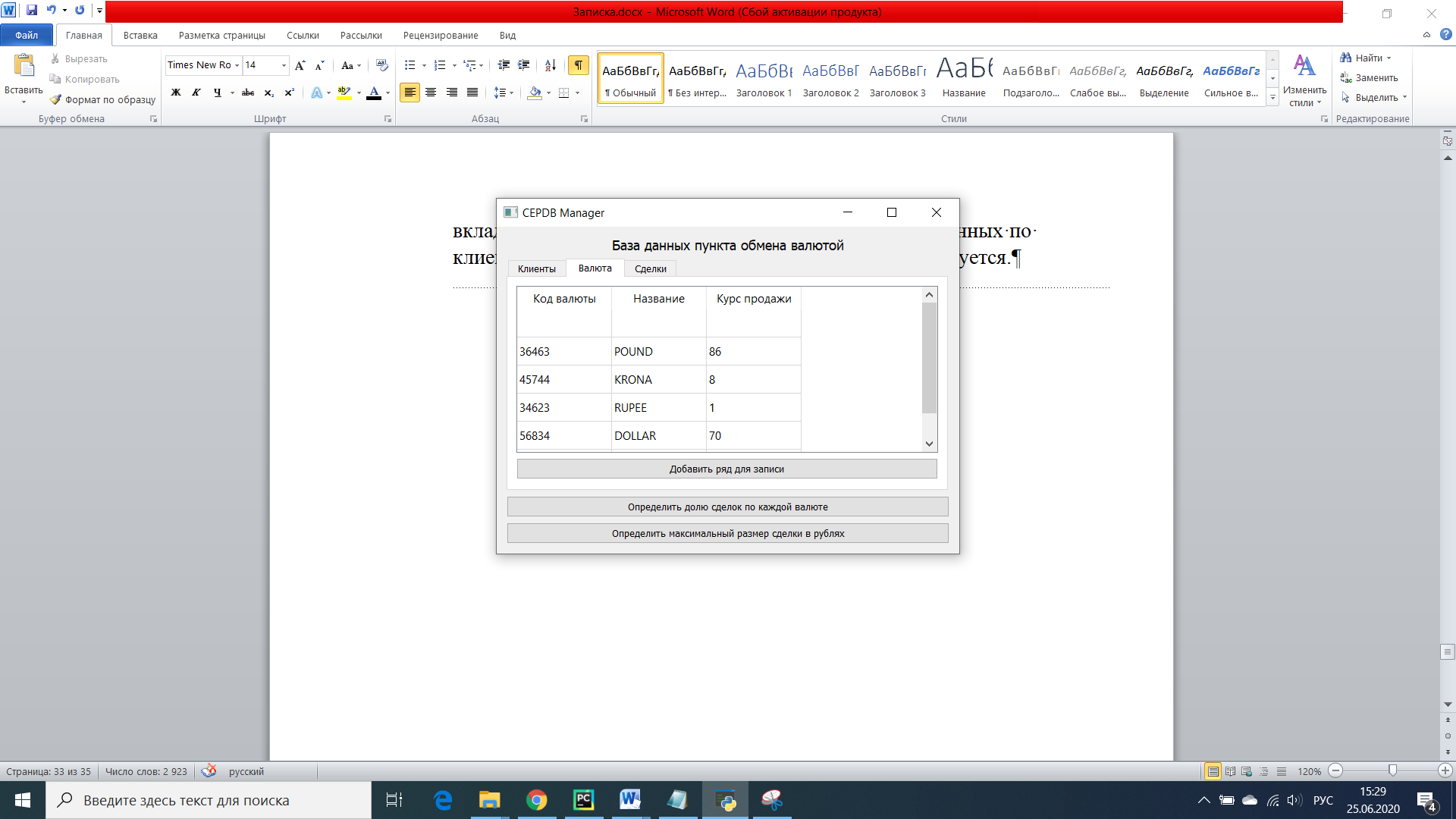


Рисунок 13 Открыта вкладка «Валюта»

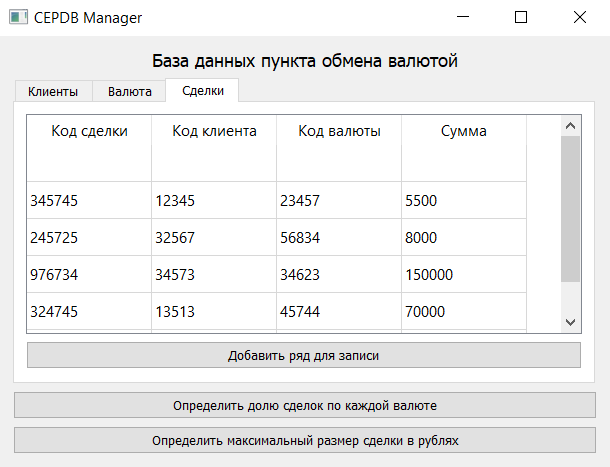


Рисунок 14 Открыта вкладка «Сделки»

* 1. Функционал интерфейса.

Интерфейс программы имеет возможность растягиваться до необходимых размеров. При наведении курсора на один из углов окна программы и появления подсказки (курсор превращается в линию со стрелками на каждом краю, направленную на 45 градусов от нижней границы экрана монитора) и зажатия левой кнопки мыши с последующим передвижения курсора в произвольном направлении, можно заметить, что элементы окна (кроме текста) будут динамически расширяться в направлении перемещения курсора.

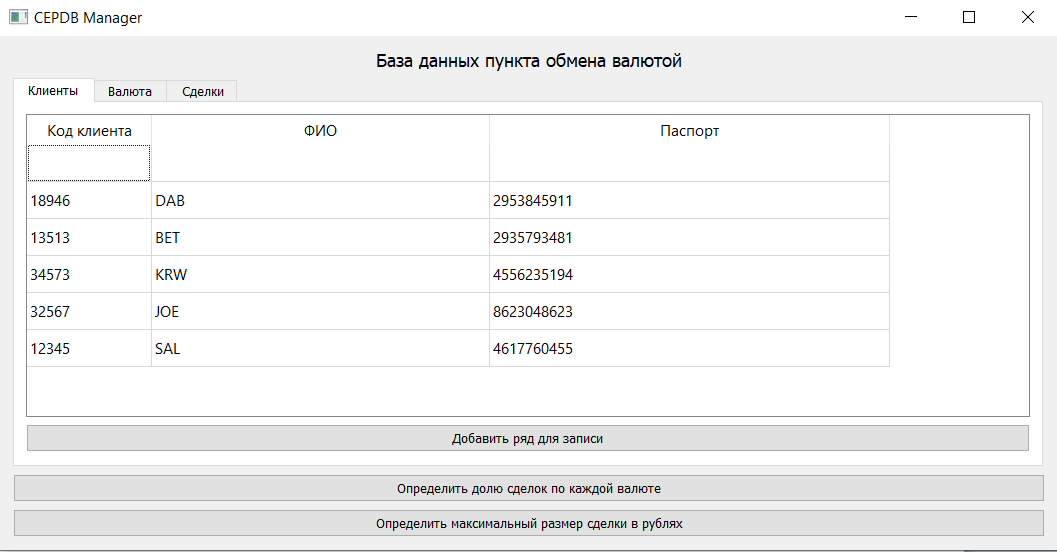


Рисунок 15 Изменённое окно программы

* 1. Функционал кнопок.
     1. Кнопка «Добавить ряд для записи».

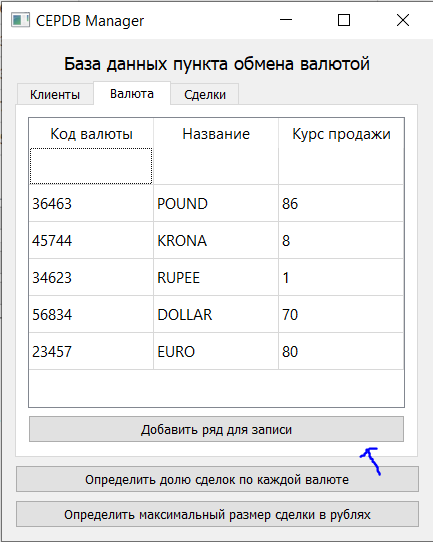


Рисунок 16 Кнопка «Добавить ряд для записи» во вкладке «Валюта»

Данная кнопка расположена в каждой из вкладок и имеет один функционал. Для правильной работы необходимо сначала вписать данные (показано на рисунке 17) в верхний ряд таблицы и только после этого произвести нажатие (показано на рисунке 18).

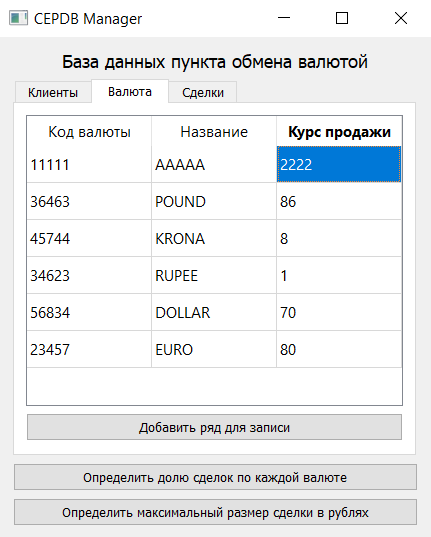


Рисунок 17 шаг 1. Ввод данных.

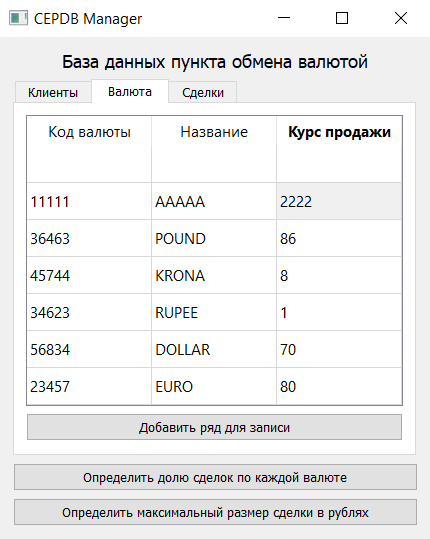


Рисунок 18 шаг 2. Нажатие на кнопку

После выполнения перечисленных действий, строка будет успешно занесена в текстовый файл (Рисунок 19)

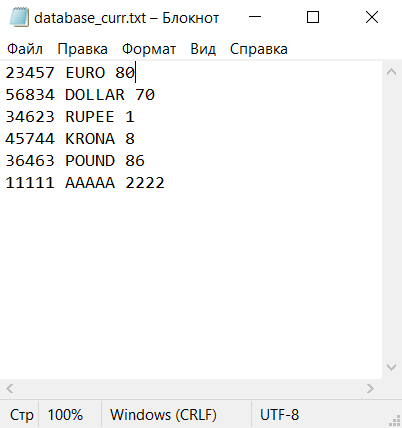


Рисунок 19 Содержимое текстового файла после выполнения операции

Приведёные выше действия имеют аналогичный результат и для других вкладок.

* + 1. Кнопка «Определить долю сделок по каждой валюте»

При нажатии на кнопку «Определить долю сделок по каждой валюте» на консоли, возникающей при открытии программы, будет выведен список. Поочерёдно на консоли выводится название валюты из базы данных, за ней идёт доля сделок по этой валюте, относительно общей суммы сделок.

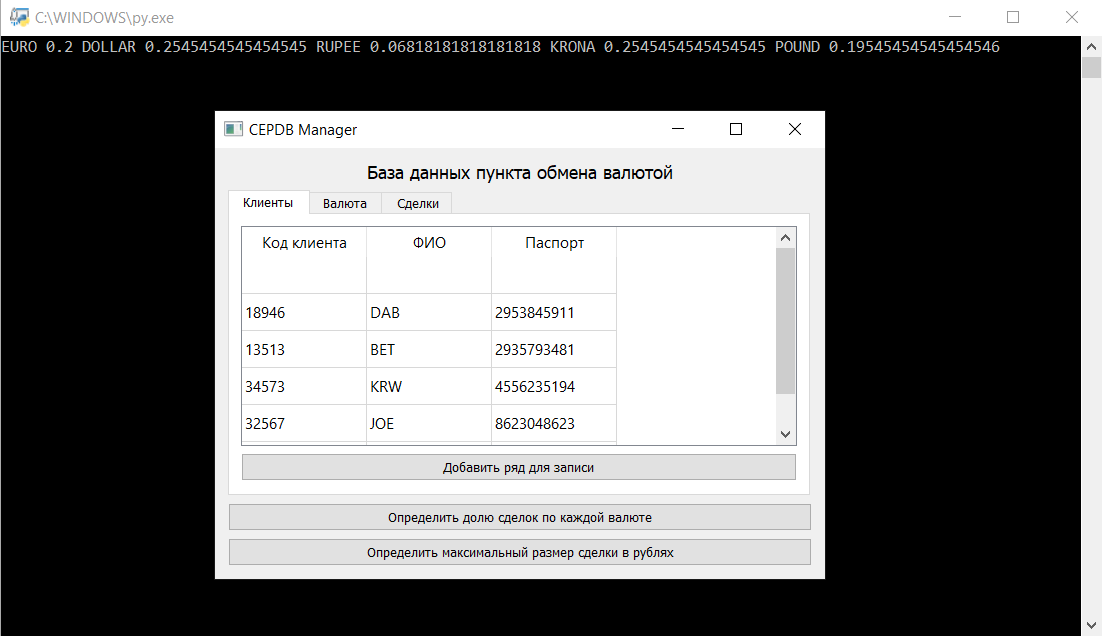


Рисунок 20 Пример работы кнопки "Определить долю сделок по каждой валюте"

* + 1. Кнопка «Определить максимальный размер сделки в рублях»

При нажатии на кнопку «Определить максимальный размер сделки в рублях» на консоли, возникающей при открытии программы, будет выведен максимальный размер сделки в рублях.

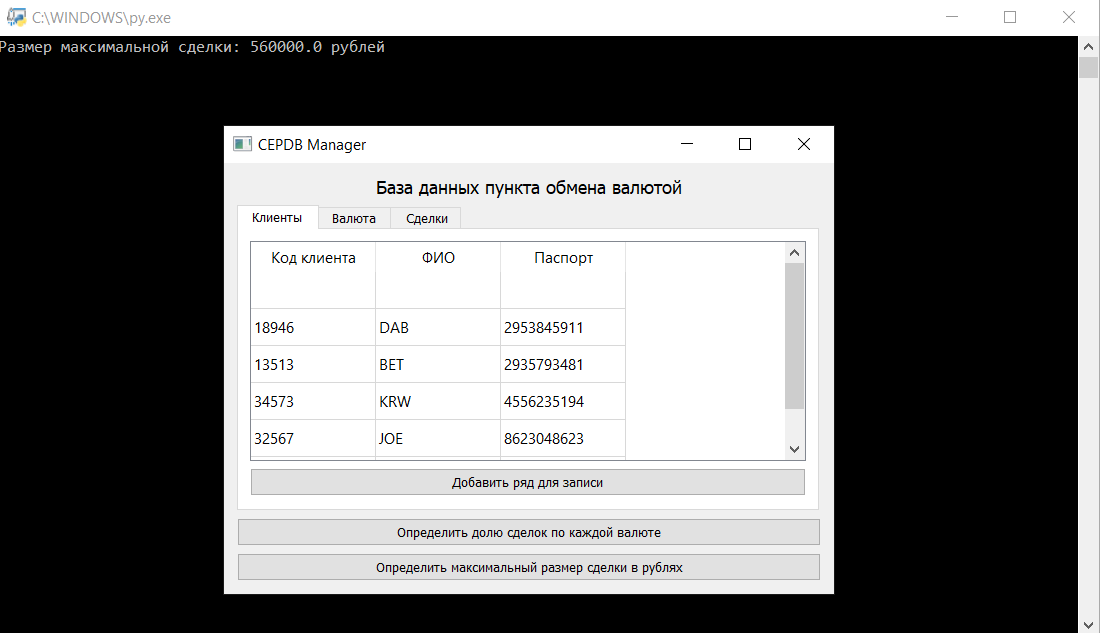


Рисунок 21 Пример работы кнопки "Определить максимальный размер сделки в рублях"

* + 1. Кнопки внешнего интерфейса.



Рисунок 22 Кнопки внешнего интерфейса

На рисунке 22 представлены кнопки внешнего интерфейса. Их функционал слева-направо: свернуть окно программы в трей, развернуть окно программы на весь экран, закрыть окно программы (при этом завершается работа программы).

# Заключение

В результате выполнения курсового проекта была написана программа «Currency Exchange Point» для упрощения ведения бизнеса в сфере валютных бирж, а именно для отслеживания информации по клиентам, валютам и сделкам.

В ходе работы были проанализированы предметная область, существующие разработки, посвященные данному направлению, получены практические навыки по созданию UI с помощью библиотеки PyQt5 и приложения QT Designer.

Также планируется продолжать работу над данным проектом с целью расширения возможностей и удобства приложения для пользователей. Планы по доработкам представлены ниже.

To-do лист:

1. Доработка интерфейса с целью упрощения работы с программой.
2. Перенос всех данный в сетевую БД.
3. Улучшение дизайна интерфейса.

# Список литературы и интернет-источников

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учеб. пособие / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Сидорова-Виснадул; под ред. Л.Г. Гагариной. — Москва: ИД «ФОРУМ»: ИНФРА-М, 2019. — 400 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-104071-3. - Текст: электронный. - URL: <https://znanium.com/catalog/product/1011120>
2. Сайт, используемый как один из примеров существующих разработок

<https://www.xe.com/>

Приложение.

Ниже представлен полный код программы.

from PyQt5 import QtCore, QtGui, QtWidgets

from PyQt5.QtWidgets import QApplication, QWidget, QTableWidgetItem

import sys

class database\_ui(QtWidgets.QWidget):

def \_\_init\_\_(self):

QtWidgets.QWidget.\_\_init\_\_(self)

self.setupUi(self)

def setupUi(self, UI\_Tool):

UI\_Tool.setObjectName("TPayne\_MySQL\_Tool")

UI\_Tool.resize(610, 431)

self.gridLayout = QtWidgets.QGridLayout(UI\_Tool)

self.gridLayout.setObjectName("gridLayout")

self.verticalLayout = QtWidgets.QVBoxLayout()

self.verticalLayout.setObjectName("verticalLayout")

self.label = QtWidgets.QLabel(UI\_Tool)

font = QtGui.QFont()

font.setPointSize(11)

self.label.setFont(font)

self.label.setLayoutDirection(QtCore.Qt.LeftToRight)

self.label.setScaledContents(True)

self.label.setAlignment(QtCore.Qt.AlignCenter)

self.label.setObjectName("label")

self.verticalLayout.addWidget(self.label)

self.horizontalLayout = QtWidgets.QHBoxLayout()

self.horizontalLayout.setObjectName("horizontalLayout")

self.tabWidget = QtWidgets.QTabWidget(UI\_Tool)

self.tabWidget.setObjectName("tabWidget")

self.tab = QtWidgets.QWidget()

self.tab.setObjectName("tab")

self.verticalLayout\_5 = QtWidgets.QVBoxLayout(self.tab)

self.verticalLayout\_5.setObjectName("verticalLayout\_5")

self.tableWidget\_1 = QtWidgets.QTableWidget(self.tab)

self.tableWidget\_1.setObjectName("tableWidget\_1")

self.tableWidget\_1.setColumnCount(3)

self.tableWidget\_1.setRowCount(1)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_1.setVerticalHeaderItem(0, item)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_1.setHorizontalHeaderItem(0, item)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_1.setHorizontalHeaderItem(1, item)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_1.setHorizontalHeaderItem(2, item)

self.tableWidget\_1.horizontalHeader().setCascadingSectionResizes(True)

self.tableWidget\_1.horizontalHeader().setSortIndicatorShown(False)

self.tableWidget\_1.horizontalHeader().setStretchLastSection(False)

self.tableWidget\_1.verticalHeader().setVisible(False)

self.verticalLayout\_5.addWidget(self.tableWidget\_1)

self.pushButton\_3 = QtWidgets.QPushButton(self.tab)

self.pushButton\_3.setObjectName("pushButton\_3")

self.verticalLayout\_5.addWidget(self.pushButton\_3)

self.tabWidget.addTab(self.tab, "")

self.tab\_2 = QtWidgets.QWidget()

self.tab\_2.setObjectName("tab\_2")

self.verticalLayout\_4 = QtWidgets.QVBoxLayout(self.tab\_2)

self.verticalLayout\_4.setObjectName("verticalLayout\_4")

self.tableWidget\_2 = QtWidgets.QTableWidget(self.tab\_2)

self.tableWidget\_2.setObjectName("tableWidget\_2")

self.tableWidget\_2.setColumnCount(3)

self.tableWidget\_2.setRowCount(1)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_2.setVerticalHeaderItem(0, item)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_2.setHorizontalHeaderItem(0, item)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_2.setHorizontalHeaderItem(1, item)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_2.setHorizontalHeaderItem(2, item)

self.tableWidget\_2.verticalHeader().setVisible(False)

self.verticalLayout\_4.addWidget(self.tableWidget\_2)

self.pushButton\_4 = QtWidgets.QPushButton(self.tab\_2)

self.pushButton\_4.setObjectName("pushButton\_4")

self.verticalLayout\_4.addWidget(self.pushButton\_4)

self.tabWidget.addTab(self.tab\_2, "")

self.tab\_3 = QtWidgets.QWidget()

self.tab\_3.setObjectName("tab\_3")

self.verticalLayout\_3 = QtWidgets.QVBoxLayout(self.tab\_3)

self.verticalLayout\_3.setObjectName("verticalLayout\_3")

self.tableWidget\_3 = QtWidgets.QTableWidget(self.tab\_3)

self.tableWidget\_3.setObjectName("tableWidget\_3")

self.tableWidget\_3.setColumnCount(4)

self.tableWidget\_3.setRowCount(1)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_3.setVerticalHeaderItem(0, item)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_3.setHorizontalHeaderItem(0, item)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_3.setHorizontalHeaderItem(1, item)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_3.setHorizontalHeaderItem(2, item)

item = QtWidgets.QTableWidgetItem()

self.tableWidget\_3.setHorizontalHeaderItem(3, item)

self.tableWidget\_3.verticalHeader().setVisible(False)

self.verticalLayout\_3.addWidget(self.tableWidget\_3)

self.pushButton\_5 = QtWidgets.QPushButton(self.tab\_3)

self.pushButton\_5.setObjectName("pushButton\_5")

self.verticalLayout\_3.addWidget(self.pushButton\_5)

self.tabWidget.addTab(self.tab\_3, "")

self.horizontalLayout.addWidget(self.tabWidget)

self.verticalLayout.addLayout(self.horizontalLayout)

self.gridLayout.addLayout(self.verticalLayout, 0, 0, 1, 1)

self.pushButton = QtWidgets.QPushButton(UI\_Tool)

self.pushButton.setAutoDefault(False)

self.pushButton.setDefault(False)

self.pushButton.setFlat(False)

self.pushButton.setObjectName("pushButton")

self.gridLayout.addWidget(self.pushButton, 1, 0, 1, 1)

self.pushButton\_2 = QtWidgets.QPushButton(UI\_Tool)

self.pushButton\_2.setObjectName("pushButton\_2")

self.gridLayout.addWidget(self.pushButton\_2, 2, 0, 1, 1)

self.retranslateUi(UI\_Tool)

self.tabWidget.setCurrentIndex(0)

QtCore.QMetaObject.connectSlotsByName(UI\_Tool)

def retranslateUi(self, TPayne\_MySQL\_Tool):

\_translate = QtCore.QCoreApplication.translate

TPayne\_MySQL\_Tool.setWindowTitle(\_translate("TPayne\_MySQL\_Tool", "CEPDB Manager"))

self.label.setText(\_translate("TPayne\_MySQL\_Tool", "База данных пункта обмена валютой"))

item = self.tableWidget\_1.verticalHeaderItem(0)

item.setText(\_translate("TPayne\_MySQL\_Tool", "New Row"))

item = self.tableWidget\_1.horizontalHeaderItem(0)

item.setText(\_translate("TPayne\_MySQL\_Tool", "Код клиента"))

item = self.tableWidget\_1.horizontalHeaderItem(1)

item.setText(\_translate("TPayne\_MySQL\_Tool", "ФИО"))

item = self.tableWidget\_1.horizontalHeaderItem(2)

item.setText(\_translate("TPayne\_MySQL\_Tool", "Паспорт"))

self.pushButton\_3.setText(\_translate("TPayne\_MySQL\_Tool", "Добавить ряд для записи"))

self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab), \_translate("TPayne\_MySQL\_Tool", "Клиенты"))

item = self.tableWidget\_2.verticalHeaderItem(0)

item.setText(\_translate("TPayne\_MySQL\_Tool", "1"))

item = self.tableWidget\_2.horizontalHeaderItem(0)

item.setText(\_translate("TPayne\_MySQL\_Tool", "Код валюты"))

item = self.tableWidget\_2.horizontalHeaderItem(1)

item.setText(\_translate("TPayne\_MySQL\_Tool", "Название"))

item = self.tableWidget\_2.horizontalHeaderItem(2)

item.setText(\_translate("TPayne\_MySQL\_Tool", "Курс продажи"))

self.pushButton\_4.setText(\_translate("TPayne\_MySQL\_Tool", "Добавить ряд для записи"))

self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab\_2), \_translate("TPayne\_MySQL\_Tool", "Валюта"))

item = self.tableWidget\_3.verticalHeaderItem(0)

item.setText(\_translate("TPayne\_MySQL\_Tool", "New Row"))

item = self.tableWidget\_3.horizontalHeaderItem(0)

item.setText(\_translate("TPayne\_MySQL\_Tool", "Код сделки"))

item = self.tableWidget\_3.horizontalHeaderItem(1)

item.setText(\_translate("TPayne\_MySQL\_Tool", "Код клиента"))

item = self.tableWidget\_3.horizontalHeaderItem(2)

item.setText(\_translate("TPayne\_MySQL\_Tool", "Код валюты"))

item = self.tableWidget\_3.horizontalHeaderItem(3)

item.setText(\_translate("TPayne\_MySQL\_Tool", "Сумма"))

self.pushButton\_5.setText(\_translate("TPayne\_MySQL\_Tool", "Добавить ряд для записи"))

self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab\_3), \_translate("TPayne\_MySQL\_Tool", "Сделки"))

self.pushButton.setText(\_translate("TPayne\_MySQL\_Tool", "Определить долю сделок по каждой валюте"))

self.pushButton\_2.setText(\_translate("TPayne\_MySQL\_Tool", "Определить максимальный размер сделки в рублях"))

db\_file\_clients = open('database\_clients.txt', 'r')

db\_file\_curr = open('database\_curr.txt', 'r')

db\_file\_deals = open('database\_deals.txt', 'r')

for line\_clients in db\_file\_clients.readlines():

splited = line\_clients.split()

for i in range(0, 2):

self.tableWidget\_1.setItem(0, i, QTableWidgetItem(splited[i]))

self.tableWidget\_1.setItem(0, 2, QTableWidgetItem(splited[2]))

self.tableWidget\_1.insertRow(0)

for line\_curr in db\_file\_curr.readlines():

splited = line\_curr.split()

for i in range(0, 2):

self.tableWidget\_2.setItem(0, i, QTableWidgetItem(splited[i]))

self.tableWidget\_2.setItem(0, 2, QTableWidgetItem(splited[2]))

self.tableWidget\_2.insertRow(0)

for line\_deals in db\_file\_deals.readlines():

splited = line\_deals.split()

for i in range(0, 2):

self.tableWidget\_3.setItem(0, i, QTableWidgetItem(splited[i]))

self.tableWidget\_3.setItem(0, 2, QTableWidgetItem(splited[2]))

self.tableWidget\_3.setItem(0, 3, QTableWidgetItem(splited[3]))

self.tableWidget\_3.insertRow(0)

db\_file\_clients.close()

db\_file\_curr.close()

db\_file\_deals.close()

self.pushButton.pressed.connect(self.totaldeals)

self.pushButton\_2.pressed.connect(self.maxdeal)

self.pushButton\_3.pressed.connect(self.addrow\_1)

self.pushButton\_4.pressed.connect(self.addrow\_2)

self.pushButton\_5.pressed.connect(self.addrow\_3)

def addrow\_1(self):

db\_file\_clients = open('database\_clients.txt', 'a')

for i in range(0, 3):

temp = self.tableWidget\_1.item(0, i).text()

db\_file\_clients.write(temp)

db\_file\_clients.write(' ')

db\_file\_clients.write('\n')

db\_file\_clients.close()

self.tableWidget\_1.insertRow(0)

def addrow\_2(self):

db\_file\_curr = open('database\_curr.txt', 'a')

for i in range(0, 3):

temp = self.tableWidget\_2.item(0, i).text()

db\_file\_curr.write(temp)

db\_file\_curr.write(' ')

db\_file\_curr.write('\n')

db\_file\_curr.close()

self.tableWidget\_2.insertRow(0)

def addrow\_3(self):

db\_file\_deals = open('database\_deals.txt', 'a')

for i in range(0, 4):

temp = self.tableWidget\_3.item(0, i).text()

db\_file\_deals.write(temp)

db\_file\_deals.write(' ')

db\_file\_deals.write('\n')

db\_file\_deals.close()

self.tableWidget\_3.insertRow(0)

def totaldeals(self):

db\_file\_curr = open('database\_curr.txt', 'r')

db\_file\_deals = open('database\_deals.txt', 'r')

exit\_list = []

deals\_lines = []

curr\_lines = []

overall = 0

for line\_deals in db\_file\_deals.readlines():

splited\_deals = line\_deals.split()

deals\_lines.append(splited\_deals)

for line\_curr in db\_file\_curr.readlines():

splited\_curr = line\_curr.split()

curr\_lines.append(splited\_curr)

curr\_code = splited\_curr[0]

curr\_price = splited\_curr[2]

for i in range(len(deals\_lines)):

if curr\_code == splited\_deals[2]:

val = float(curr\_price) \* float(splited\_deals[3])

overall = overall + val

for i in range(len(curr\_lines)):

for j in range(len(deals\_lines)):

if curr\_lines[i][0] == deals\_lines[j][2]:

exit\_list.append(curr\_lines[i][1])

exit\_list.append(float(curr\_lines[i][2])\*float(deals\_lines[j][3])/float(overall))

print(\*exit\_list)

db\_file\_curr.close()

db\_file\_deals.close()

def maxdeal(self):

db\_file\_curr = open('database\_curr.txt', 'r')

db\_file\_deals = open('database\_deals.txt', 'r')

exit\_list = []

deals\_lines = []

curr\_lines = []

for line\_deals in db\_file\_deals.readlines():

splited\_deals = line\_deals.split()

deals\_lines.append(splited\_deals)

for line\_curr in db\_file\_curr.readlines():

splited\_curr = line\_curr.split()

curr\_lines.append(splited\_curr)

for i in range(len(curr\_lines)):

for j in range(len(deals\_lines)):

if curr\_lines[i][0] == deals\_lines[j][2]:

exit\_list.append(float(curr\_lines[i][2])\*float(deals\_lines[j][3]))

print('Размер максимальной сделки:', max(exit\_list), 'рублей')

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

ex = database\_ui()

ex.show()

sys.exit(app.exec\_())

1. 1) ГОСТ 19.101-77 ЕСПД. Виды программ и программных документов [↑](#footnote-ref-1)
2. 2) ГОСТ 19.103-77 ЕСПД. Обозначение программ и программных документов [↑](#footnote-ref-2)
3. 3) ГОСТ 19.104-78\* ЕСПД. Основные надписи [↑](#footnote-ref-3)
4. 4) ГОСТ 19.105-78\* ЕСПД. Общие требования к программным документам [↑](#footnote-ref-4)
5. 5) ГОСТ 19.106-78\* ЕСПД. Общие требования к программным документам, выполненным печатным способом [↑](#footnote-ref-5)
6. 6) ГОСТ 19.505-79\* ЕСПД. Руководство оператора. Требования к содержанию и оформлению [↑](#footnote-ref-6)
7. 7) ГОСТ 19.604-78\* ЕСПД. Правила внесения изменений в программные документы, выполненные печатным способом [↑](#footnote-ref-7)