МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ

по учебной вычислительной (ознакомительной) практике

Головача Алексея Викторовича

студента 1 курса, 7 группы

специальность

«Прикладная Математика»

Руководитель практики:

старший преподаватель

О.О. Колб

Минск, 2024

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc170484355)

[Введение 3](#_Toc170484356)

[Глава 1 Теоретические основы 4](#_Toc170484357)

[**1.1** **Знакомство с SoftClub** 4](#_Toc170484358)

[1.2 Общая характеристика курса «Контроль версий с помощью Git» 4](#_Toc170484359)

[1.2.1 Установка платформы Git 5](#_Toc170484360)

[1.2.2 Изучение основных концепций системы контроля версий Git 5](#_Toc170484361)

[1.2.3 Работа с удаленными репозиториями 5](#_Toc170484362)

[1.5 Разработка приложения, отражающего табличные данные с возможностью корректировки 6](#_Toc170484363)

[1.6 Разработка простейшего SDI-приложения 7](#_Toc170484364)

[1.7 Выводы к главе 1 7](#_Toc170484365)

[Текст выводов. 7](#_Toc170484366)

[Глава 2 Практическая часть курса 9](#_Toc170484367)

[2.1 Unit-тесты 9](#_Toc170484368)

[2.2 Разработка GUI-приложения в QT 10](#_Toc170484369)

[2.3 Разработка приложения, отражающего табличные данные с возможностью корректировки 11](#_Toc170484370)

[2.4 Разработка простейшего SDI-приложения 13](#_Toc170484371)

[2.5 13](#_Toc170484372)

[2.6 13](#_Toc170484373)

[2.7 Разработка простейшего SDI-приложения 13](#_Toc170484374)

[Глава 3 Индивидуальный проект 14](#_Toc170484375)

[3.1 Функциональное назначение 14](#_Toc170484376)

[3.2 Структура проекта 14](#_Toc170484377)

[3.3 Интерфейс приложения 15](#_Toc170484378)

[3.4 Описание проекта 15](#_Toc170484379)

[3.5 Выводы к главе 3 15](#_Toc170484380)

[Заключение 16](#_Toc170484381)

[Список использованных источников 17](#_Toc170484382)

# Введение

Введение обычно имеет объем 1-2 страницы. В данном отчете введение должно включать:

* цель практики, задачи практики;
* выбранные направления работы, их актуальность;
* краткое представление (наличие) Индивидуального проекта.

*Используйте правильно данный документ*:

* Не изменяйте его форматирование, используйте как шаблон! Данный документ соответствует «Положению об организации подготовки и защиты курсовой работы, итоговой аттестации при освоении содержания образовательных программ высшего образования I ступени в Белорусском государственном университете».
* Заголовки, которые «пишутся» большими буквами, вводятся обычным способом «Первая буква прописная, остальные – строчные». А затем из списка Регистр (группа Шрифт вкладки Главная) выбрать опцию «ВСЕ ПРОПИСНЫЕ».
* Используйте «Формат по образцу».
* При добавлении новых глав, разделов и так далее копируйте части документа, которые уже имеют корректный формат.
* Оглавление следует обновлять автоматически (права кнопка мыши на оглавлении и button «Обновить таблицу»)

Фрагмент введения:

Целью учебной вычислительной (ознакомительной) практики является закрепление полученных знаний за первый год обучения посредством выполнения специальных учебных заданий, участия в работе над командным проектом.

Задачи практики:

* Знакомство с IT компанией, возможностями обучения и повышения квалификации, знакомство с разными направлениями профессиональной подготовки IT специалистов.
* Изучение инструментов для совместной разработки проектов и контроля версий.
* Подготовка и настройка программных сред и средств тестирования для прохождения тренинга по выбранному направлению.
* Выполнение стандартных заданий.
* Выполнение заданий повышенного уровня.
* Разработка приложение «НАЗВАНИЕ ВАШЕГО ИНДИВИДУАЛЬНОГО ПРОЕКТА» - если есть.
* Оформление отчета.

# Глава 1 Теоретические основы

* 1. **Знакомство с SoftClub**

SoftClub — один из крупнейших разработчиков программных решений для банков, финансовых компаний, электронной торговли, биржевой отрасли и цифровой экономики в Восточной Европе и странах СНГ.

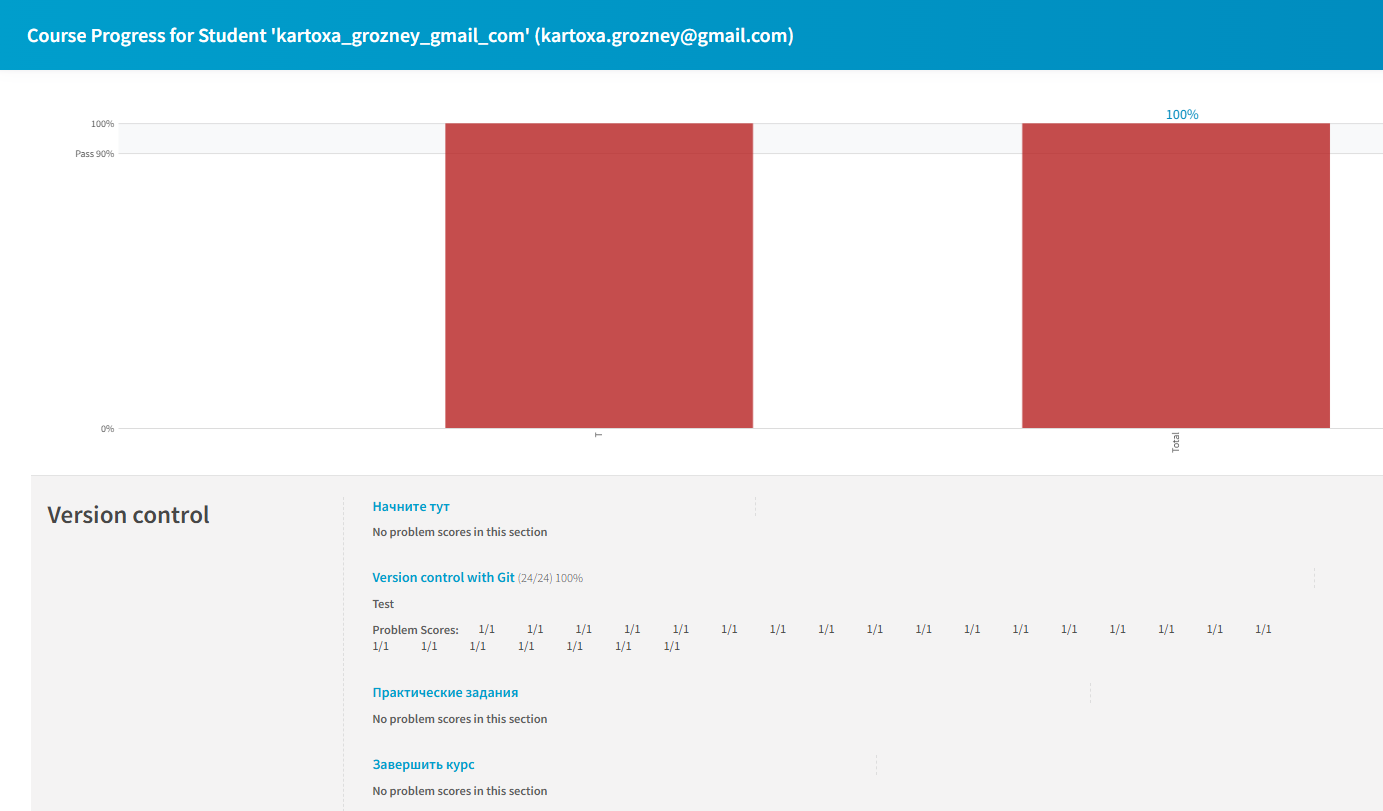
Представитель данной организации ознакомил студентов с особенностями работы в данной сфере, а также подробно рассказал про стажировку у них в компании.

Рассмотрев основные направления деятельности SoftClub, студенты были удивлены масштабом проектов, над которыми работают специалисты компании. Они также узнали о том, что стажировка в SoftClub представляет собой отличную возможность для приобретения практических навыков в разработке программного обеспечения для финансовой сферы.

## Общая характеристика курса «Контроль версий с помощью Git»

В рамках практики был изучен курс «Контроль версий с помощью Git» на платформе EPAM Learning включающий в себя теоретический и практический этапы, а также промежуточные тесты, закрепляющие теоретические знания. Были выполнены практические занятия с целью закрепления знаний по работе с Git, а также изучены основы работы с платформой GitHub.

Практические занятия позволили закрепить теоретические знания и получить практический опыт работы с системой контроля версий Git и платформой GitHub. В результате прохождения курса была представлена информация об этапах обучения, отображающая на каких шагах курса возникли сложности. Это позволяет проанализировать свою успеваемость и сделать выводы для дальнейшего совершенствования навыков работы с Git.



## Установка платформы Git

Была проведена установка актуальной версии платформы Git и ее настройка для дальнейшей работы, а также выполнена авторизация на платформе GitHub.

## Изучение основных концепций системы контроля версий Git

Были изучены основные концепции системы контроля версий Git, такие как: commit, push, ветки, merge, stash, tags, remotes:

1.Концепция commit предполагает сохранение изменений в локальном репозитории с комментарием, описывающим произведенные изменения.

2.Концепция push используется для отправки локальных коммитов в удалённый репозиторий, позволяя синхронизировать изменения с другими участниками проекта.

3.Концепция ветки (branch) в Git позволяет разделять разработку на независимые направления, облегчая экспериментирование и параллельную работу над проектом.

4.Концепция merge используется для объединения изменений из разных веток, позволяя интегрировать наработки в единую кодовую базу.

5.Концепция stash даёт возможность временно сохранить незакоммиченные изменения, чтобы переключиться на другую задачу, а затем восстановить их для продолжения работы.

6.Концепция тегов (tags) позволяет помечать важные моменты в истории проекта, например, релизные версии.

7.Концепция удалённых репозиториев (remotes) даёт возможность синхронизировать локальный репозиторий с удалёнными хранилищами, обеспечивая совместную работу над проектом.

## Работа с удаленными репозиториями

Знания, полученные о работе с удалёнными репозиториями в GitHub, включают в себя создание и связывание локального репозитория с удалённым на платформе GitHub, а также использование команд push и pull для синхронизации изменений между локальной и удалённой версиями проекта. Это позволяет эффективно организовывать совместную работу над проектом, обмениваться наработками и отслеживать историю развития кодовой базы.

* 1. **Дополнение ранее разработанных классов системой Unit-тестов**

Unit-тесты – это небольшие тесты, которые проверяют отдельные части программного кода на корректность и соответствие ожидаемым результатам. Они используются программистами для проверки правильности работы отдельных модулей или функций программы.

Unit-тесты позволяют убедиться, что каждая часть кода работает правильно в изоляции от других частей программы и помогают выявить ошибки на ранних этапах разработки. В общем, unit-тесты являются важным инструментом в разработке программного обеспечения, который помогает обеспечивать стабильность и надежность кода.

* 1. **Разработка простейшего GUI-приложения в QT**

Разработка простейшего GUI-приложения в Qt – это создание графического пользовательского интерфейса с помощью библиотеки Qt, которая позволяет создавать мощные и красивые приложения для различных платформ. Для этого необходимо использовать инструменты Qt Designer для создания интерфейса и язык программирования C++ для добавления функциональности к приложению. В результате получится удобное и интуитивно понятное приложение для пользователя.

## Разработка приложения, отражающего табличные данные с возможностью корректировки

В ходе подготовки к практической части летней практики было выяснено, что разработка приложения, отражающего табличные данные с возможностью корректировки, является интересной и полезной задачей. Ниже описаны основные шаги и компоненты такого проекта:

1. Используя библиотеку Qt's QAxObject, можно взаимодействовать с Excel напрямую из вашего Qt-приложения. Это позволяет открывать, читать, редактировать и сохранять Excel-файлы.

2. Для отображения табличных данных в вашем Qt-приложении, вы можете использовать виджет QTableView. Он позволяет отображать данные из QStandardItemModel или другой модели, реализующей интерфейс QAbstractItemModel.

3. При открытии Excel-файла, можно загружать данные из таблицы в QStandardItemModel, а затем связывать эту модель с QTableView для отображения. Любые изменения, внесенные пользователем в таблицу, должны обновляться в QStandardItemModel.

4. Предоставьте пользователю возможность редактировать данные прямо в QTableView. Изменения должны отражаться как в QStandardItemModel, так и в открытом Excel-файле.

5. Разработайте интуитивно понятный и эффективный интерфейс пользователя, который позволит легко взаимодействовать с таблицами и Excel-файлами.

Это лишь общие рекомендации, и реальная реализация проекта может отличаться в зависимости от требований и целей проекта. Важно тщательно спроектировать и протестировать приложение, чтобы обеспечить надежную и эффективную работу с Excel-файлами.

## Разработка простейшего SDI-приложения

SDI (Single Document Interface) - это тип архитектуры приложений, при которой каждый открытый документ отображается в отдельном главном окне приложения.

Основные особенности SDI-приложений:

Каждый документ открывается в своем отдельном главном окне. Это означает, что для работы с несколькими документами одновременно пользователь может открыть несколько экземпляров основного окна приложения.

Каждое главное окно имеет собственный набор меню, панелей инструментов и других элементов управления, относящихся только к текущему документу.

Действия, выполняемые пользователем, применяются только к активному документу, отображаемому в текущем главном окне.

Закрытие одного из главных окон приводит к закрытию только связанного с ним документа, остальные открытые документы продолжают работать.

SDI-архитектура обеспечивает простоту и интуитивность использования приложения, особенно для работы с множеством независимых документов. Она подходит для приложений, в которых пользователю требуется одновременно работать с несколькими документами.

## Выводы к главе 1

## Текст выводов.

Выводы и предложения должны быть конкретными, реальными и обоснованными, вытекать из полученных результатов. Выводы пишутся тезисно (по пунктам).

# Глава 2 Практическая часть курса

## 2.1 Unit-тесты

1. Необходимо составить UNIT-тест для конструктора класса Calculator в рамках модульного тестирования.

// -- tests --

void TestCalculator::testConstructor()

{

// default values

Calculator c1;

QVERIFY(c1.GetA() == 0);

QVERIFY(c1.GetB() == 0);

// full constructor

const int A = 10;

const int B = 2;

Calculator c2(A, B);

QVERIFY2(c2.GetA() == A, "first operand doesn't match");

QVERIFY2(c2.GetB() == B, "second operand doesn't match");

}

Этот тест проверяет, что конструктор класса Calculator корректно инициализирует свойства объекта (a и b) при создании объекта с использованием как конструктора по умолчанию, так и конструктора с аргументами.

2. Необходимо составить UNIT-тест который проверяет работу метода Sum() класса Calculator в рамках модульного тестирования.

void TestCalculator::testSum()

{

// sum default

QCOMPARE(mCalc.Sum(), A0 + B0);

// sum after setting a and b

const int A = 10;

const int B = 2;

mCalc.SetA(A);

mCalc.SetB(B);

QCOMPARE(mCalc.Sum(), A + B);

}

Этот тест проверяет, что метод Sum() класса Calculator корректно

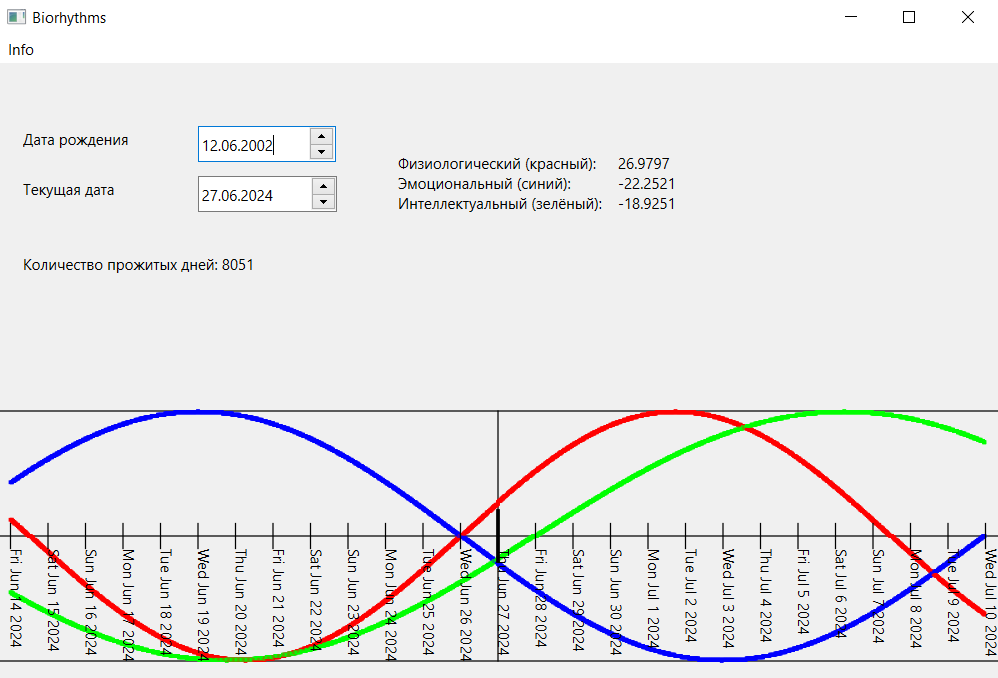
возвращает сумму текущих значений a и b, как в случае значений по умолчанию, так и после их изменения.

Перечислить пройденные решенные задачи, тесты, полученные оценки. Для каждой задачи в краткой форме – постановка задачи на русском языке, фрагмент кода решения, результат тестирования, дополнительное описание (при необходимости). В заключительном отчете допускается отсутствие скриншотов, подтверждающих полученную оценку тестирования, если таковые были в предварительном отчете, размещенном на edufpmi.

## 2.2 Разработка GUI-приложения в QT

Необходимо разработать простейшее приложения в QT (Диалоговое окно с виджетами, обеспечивающими ввод информации, вызов расчетной части приложения и вывод результатов)

За основу приложения взято построение графика биоритма. Для выполнения данной задачи был создан интуитивно понятный интерфейс:



Программа принимает от пользователя две даты:

1. Дату его рождения;
2. Текущую дату;

На основании полученной информации программа строит график, который отображает синхронизацию организма с ходом времени.

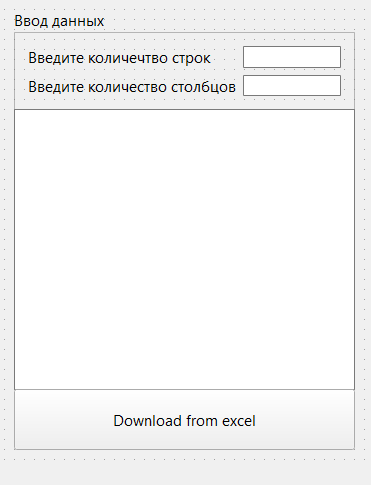
Благодаря данной работе был получен опыт работы с графическим редактором в QT, а также изучены новые библиотеки, такие как:

1.QDate:  
QDate - это класс в Qt, который представляет дату в григорианском календаре. Он предоставляет множество полезных методов для работы с датами, включая функции для получения дня недели, недели года, месяца и года, а также для вычисления разницы между двумя датами. QDate также поддерживает преобразование между разными форматами представления дат.

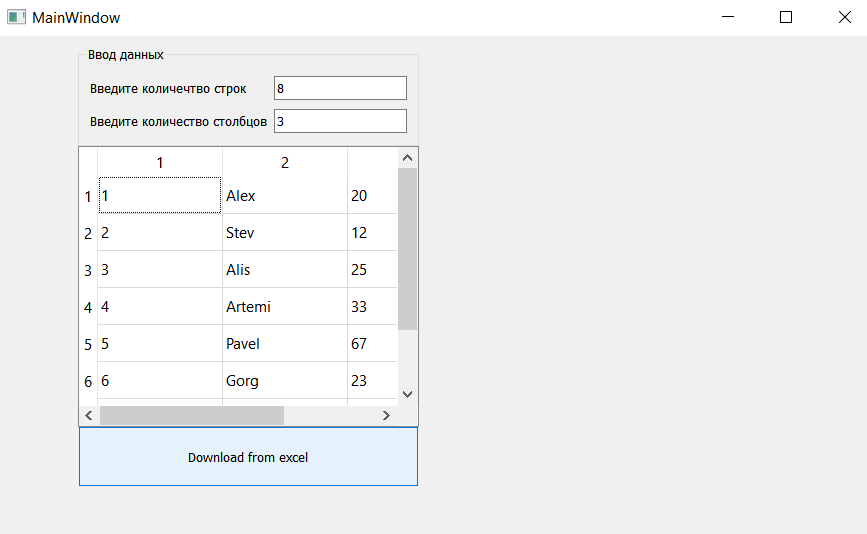
2.QPainter:  
QPainter - это класс в Qt, который предоставляет средства для рисования графических примитивов, таких как линии, многоугольники, текст и изображения, на виджетах или других устройствах вывода. QPainter работает с системой координат виджета и может использоваться для создания и манипулирования различными графическими объектами в вашем приложении. Он является ключевым инструментом для создания пользовательских визуальных эффектов и элементов интерфейса.

## 2.3 Разработка приложения, отражающего табличные данные с возможностью корректировки

Во время выполнения практических заданий было спроектировано диалоговое окно, которое запрашивает у пользователя количество столбцов и строк, и при нажатии на кнопку “Download from excel” заполняет ее данными с исходной таблицы EXEL.



После выполнения пользователем всех необходимых действий, диалоговое окно принимает следующий вид:



Программа корректно считала данный из исходной таблицы EXEL, а также корректно редактирует полученные данные и записывает их обратно.

В результате выполнения данной работы были изучены такие библиотеки как:

1.QAxObject:

QAxObject - это класс в Qt, который позволяет интегрировать ActiveX и COM-объекты в приложения Qt, обеспечивая взаимодействие между компонентами на различных платформах.

2.QTableView:

QTableView - это виджет табличного представления данных в Qt, который предоставляет гибкий и настраиваемый интерфейс для отображения и взаимодействия с табличными данными в приложениях.

3.QStandardItemModel:

QStandardItemModel - это класс модели данных в Qt, который используется для хранения и управления структурированными данными, такими как таблицы, деревья и списки, для их отображения в различных виджетах представления.

Благодаря данной работе, были получены новые знания для работы с базами данных, а также закреплены знания полученные ранее.

## Разработка простейшего SDI-приложения

Необходимо разработать простейшее SDI-приложение, иллюстрирующее использование меню, корректировку, сохранение и восстановление текстовой, числовой информации в текстовом и бинарном формате.

Для решения данной задачи были изучены следующие библиотеки:

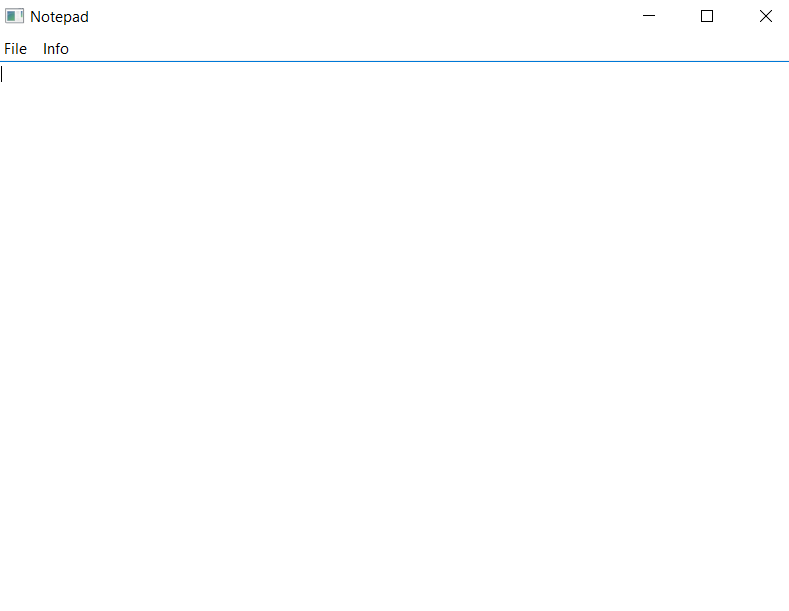
1. QPlainTextEdit:

QPlainTextEdit - это виджет в Qt для отображения и редактирования простого текстового содержимого, предоставляющий базовые функции для работы с текстовыми документами, такие как форматирование, навигация и поиск.

2. QTextStream:

QTextStream - это класс в Qt для чтения и записи текстовых данных, предоставляющий простой и эффективный интерфейс для работы с потоками ввода-вывода текста в различных кодировках, поддерживающий форматирование текста с помощью операторов потока.

После запуска кода, получили следующий вывод:



Программа корректно сохраняет, считывает и записывает данные, а также редактирует файл.

Благодаря данной работе, был получен опыт работы с такими библиотеками как:

QPlainTextEdit, QTextStream. Закреплены знания работы с текстовыми и бинарными файлами.

## 

## 

* 1. **Выводы к главе 2**

Текст выводов.

Выводы и предложения должны быть конкретными, реальными и обоснованными, вытекать из полученных результатов. Выводы пишутся тезисно (по пунктам).

# Глава 3 Индивидуальный проект

В данной главе представлен разработанный проект (при наличии).

Описать самостоятельно реализованный проект (проекты): назначение, фрагменты кода с комментариями, копии экранов.

## 3.1 Функциональное назначение

Текст.

Каждый структурный элемент (введение, заключение, главы, список использованных источников, приложения) начинать с новой страницы!

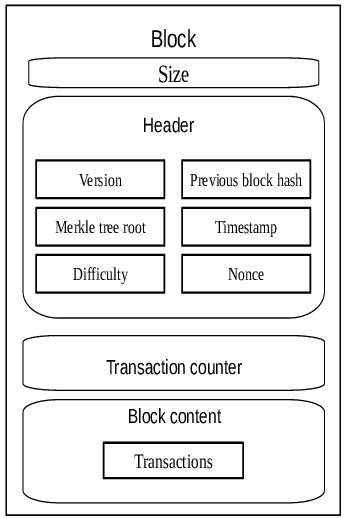
Недопустим в главе только один раздел, в разделе – только один подраздел.

## 3.2 Структура проекта

При описании структуры проекта, возможно, понадобится оформить рисунки и таблицы.

*Пример оформления рисунка, подписи к нему и ссылки на рисунок*.

На рисунке 3.1 представлена структура блока в блокчейне. Блок содержит в себе заголовок и информацию о транзакциях.

  
Рисунок 3.1 – Структура блока в блокчейне

*Пример оформления таблицы*.

В таблице 1 приведена сравнительная характеристика разных типов блокчейн-сетей. Выбор того или иного типа блокчейна зависит от выдвинутых для сети требований и потребностей заказчика.

Таблица 1 – Сравнение различных типов блокчейн-сетей

|  |  |  |  |
| --- | --- | --- | --- |
|  | Публичный блокчейн | Блокчейн консорциума | Частный блокчейн |
| Участники в процессе консенсуса |  |  |  |
| Чтение |  |  |  |
| Уровень обеспечения неизменяемости |  |  |  |
| Эффективность использования ресурсов |  |  |  |
| Централизация |  |  |  |
| Право на участие в процессе консенсуса |  |  |  |

Как видно из таблицы …

Рисунок, таблица или код, не должны быть последним содержимым параграфа, после них обязательно должно быть текстовое содержимое.

## 3.3 Интерфейс приложения

В этом разделе приводят копии экранов, которые оформляются как рисунки и сопровождаются пояснительным текстом. В тексте должны быть ссылки на все рисунки.

## 3.4 Описание проекта

Текст описания.

## 3.5 Выводы к главе 3

Текст выводов.

Выводы и предложения должны быть конкретными, реальными и обоснованными, вытекать из полученных результатов. Выводы пишутся тезисно (по пунктам).

# Заключение

В общем случае заключение должно содержать:

* краткие выводы по результатам **индивидуальной** работы;
* оценку количества и полноты решений, поставленных задач по каждому модулю (уровню сложности и т.п.).

Недопустимо искажение результатов или умолчание отрицательных результатов.

Помимо оценки результатов работы, заключение по работе может содержать информацию о пути и целях дальнейшей работы.

# Список использованных источников

1. Усов, В. Swift. Разработка приложений под iOS на основе фреймворка UIKit // В. Усов. – Москва, 2021. – 492 с. (Книга с одним автором)
2. Дробышевский, Н. П. Ревизия и аудит : учеб.-метод. пособие / Н. П. Дробышевский. – Минск : Амалфея : Мисанта, 2013. – 415 с. (Книга с одним автором)
3. Дьяченко, Л. С. Методические рекомендации по подготовке и сдаче государственного экзамена по педагогике / Л. С. Дьяченко, Н. К. Зинькова, Р. В. Загорулько. – Витебск : Витеб. гос. ун-т, 2013. – 53 с. (Книга с тремя авторами)
4. Национальный статистический комитет Республики Беларусь [Электронный ресурс]. – Режим доступа: http://belstat.gov.by/. – Дата доступа: 24.06.2016. (Сайт целиком, т.е. url заканчивается доменом первого уровня)
5. Полоник, С. С. Теоретико-методологические основы обеспечения внешнеэкономической безопасности Республики Беларусь в условиях усиления международной конкуренции / С. С. Полоник, Э. В. Хоробрых, А. А. Литвинчук // Вестн. Полоц. гос. ун-та. Сер. D, Экон. и юрид. науки. – 2013. – № 5. – С. 7–16. (Статья из журнала)
6. Жданухин, Д. Ю. Коллекторские агентства: основные черты деятельности, история появления и зарубежный опыт [Электронный ресурс] / Д. Ю. Жданухин // Юридическая Россия : Федер. правовой портал. – Режим доступа: http://law.edu.ru/doc/document.asp?docID=1233648. – Дата доступа: 22.06.2016. (Статья с сайта с указанием автора)
7. О демографической ситуации в январе – июне 2016 года [Электронный ресурс] // Национальный статистический комитет Республики Беларусь. – Режим доступа: http://www.belstat.gov.by/o-belstate\_2/novosti-i-meropriyatiya/novosti/o\_demograficheskoy\_situatsii\_v\_yanvare\_iyune\_2016\_g/. – Дата доступа: 29.07.2016. (Статья с сайта без автора)