МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ

по учебной вычислительной (ознакомительной) практике

Усачёнка Артемия Дмитриевича

студента 1 курса, 7 группы

специальность «Прикладная Математика»

Руководитель практики:

старший преподаватель

Е.В. Пазюра

Минск, 2024

# ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc170476869)

[Введение 3](#_Toc170476870)

[Глава 1 Теоретические основы 4](#_Toc170476871)

[1.1 Знакомство с SoftClub 4](#_Toc170476872)

[1.2 Общая характеристика курса «Контроль версий с помощью Git» 4](#_Toc170476873)

[1.2.1 Установка платформы Git 4](#_Toc170476874)

[1.2.2 Изучение основных концепций системы контроля версий Git 4](#_Toc170476875)

[1.2.3 Работа с удаленными репозиториями 5](#_Toc170476879)

[1.3 Unit тестирование 5](#_Toc170476880)

[1.4 Разработка простейшего GUI-приложения в QT 6](#_Toc170476881)

[1.5 Выводы к главе 1 6](#_Toc170476882)

[Текст выводов. 6](#_Toc170476883)

[Глава 2 Практическая часть курса 7](#_Toc170476884)

[2.1 Unit-тестирование 7](#_Toc170476885)

[2.2 Дополнительный материал 9](#_Toc170476886)

[2.3 Выводы к главе 2 9](#_Toc170476887)

[Глава 3 Индивидуальный проект 10](#_Toc170476888)

[3.1 Функциональное назначение 10](#_Toc170476889)

[3.2 Структура проекта 10](#_Toc170476890)

[3.3 Интерфейс приложения 11](#_Toc170476891)

[3.4 Описание проекта 11](#_Toc170476892)

[3.5 Выводы к главе 3 11](#_Toc170476893)

[Заключение 12](#_Toc170476894)

[Список использованных источников 13](#_Toc170476895)

# 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

## Знакомство с SoftClub

SoftClub - это ИТ-компания, специализирующаяся на разработке программного обеспечения и предоставлении консалтинговых услуг. Была прослушана лекция компании про работу в данной сфере, а так же было рассказано про стажировку в SoftClub. Кроме того, на презентации вакансий участники узнали, что стажировка в SoftClub предоставляет отличные возможности для получения практического опыта в разработке программного обеспечения, ориентированного на финансовую отрасль.

Стажировка в SoftClub позволяет кандидатам приобрести ценные навыки и знания в области финансового ПО.

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

Во время учебной практики был пройден курс на Epam «Контроль версий с помощью Git». Данный курс включает в себя теоретическую и практическую части, а также промежуточные проверочные тесты для закрепления теоретических знаний. Была проделана практическая работа с целью углубления навыков использования системы контроля версий Git, а также изучены основы работы с платформой GitHub.

Практические занятия позволили закрепить теоретический материал и получить практический опыт работы с Git и GitHub. По итогам прохождения курса была предоставлена информация об этапах обучения, отражающая на каких шагах возникли сложности. Это дает возможность проанализировать успеваемость и сделать выводы для дальнейшего совершенствования навыков работы в Git.

## Установка платформы Git

Была установлена и настроена платформа Git, а также выполнена авторизация на GitHub.

## 1.2.2 Изучение основных концепций системы контроля версий Git

Были изучены: commit, push, branch, remotes, merge, stash, tags

1. Сохранение изменений в локальном репозитории с комментарием, описывающим произведенные модификации, осуществляется посредством операции commit.

2. Для отправки локальных коммитов в удаленный репозиторий и синхронизации изменений с другими участниками проекта используется операция push.

3. Ветвление (brunch) позволяет разделять разработку на независимые направления, что облегчает работу над проектом.

4. Функционал remotes дает возможность синхронизировать локальный репозиторий с удаленными хранилищами, обеспечивая совместную работу над проектом.

5 Для объединения изменений разных веток и интеграции наработок в единую кодовую базу применяется операция merge.

6. Временное хранение незакоммиченных изменений с возможностью последующего восстановления для продолжения работы обеспечивается с помощью операции stash.

7. Метки (tags) позволяют помечать ключевые моменты в истории проекта.



## Работа с удаленными репозиториями

Взаимодействие с удаленными репозиториями на платформе GitHub включает в себя связывание локального репозитория с его удаленным аналогом, размещенным на GitHub. Это позволяет эффективно организовывать совместную работу над проектом, обмениваться наработками и отслеживать историю развития кодовой базы. Для этого используются операции push и pull, обеспечивающие синхронизацию изменений между локальной и удаленной версиями проектов.

## Unit тестирование

Unit-тесты представляют собой небольшие, изолированные проверки, предназначенные для оценки корректности и соответствия ожидаемым результатам отдельных компонентов программного кода. Данный инструмент широко используется разработчиками для валидации правильности работы конкретных модулей или функций программы. Применение unit-тестов позволяет удостовериться, что каждый элемент кода функционирует корректно независимо от остальных частей программы, что в свою очередь способствует выявлению и устранению ошибок на ранних этапах разработки. В целом, unit-тестирование является важным компонентом в процессе разработки программного обеспечения, поскольку оно обеспечивает стабильность и надежность конечного кода.

## Разработка простейшего GUI-приложения в QT

Разработка простейшего GUI-приложения в Qt - это процесс создания графического пользовательского интерфейса (GUI) с использованием фреймворка Qt. Qt является кроссплатформенной библиотекой для разработки приложений с графическим интерфейсом, поддерживающей различные языки программирования, такие как C++, Python, Java и другие.

Разработка простейшего GUI-приложения в Qt позволяет создавать интуитивно понятные и функциональные пользовательские интерфейсы, которые могут работать на различных платформах, включая Windows, macOS, Linux, Android и iOS. Это делает Qt популярным выбором для разработчиков, стремящихся создавать кроссплатформенные приложения с минимальными усилиями.

## Разработка приложения отражающего табличные значения

Разработка приложения, отражающего табличные данные, является важной задачей в создании современных пользовательских интерфейсов. Такие приложения позволяют эффективно представлять и взаимодействовать с большими объемами структурированной информации, что востребовано в различных областях, таких как бизнес-аналитика, научные исследования, обработка данных и многих других.

Разработка приложения, отражающего табличные данные, требует комплексного подхода, сочетающего в себе глубокое понимание пользовательских потребностей, проектирование эффективного интерфейса, надежную реализацию функциональности и тщательное тестирование. Применение передовых подходов и технологий в этой области позволяет создавать мощные и интуитивно понятные инструменты для работы с данными, повышая эффективность и производительность пользователей.

## Разработка простейшего SDI-приложения

SDI-приложения - это тип графических пользовательских интерфейсов (GUI), в котором каждый открытый документ отображается в отдельном окне. Это противоположность MDI-приложениям (Multiple Document Interface), где документы открываются во вложенных окнах внутри главного окна приложения.

Основными преимуществами SDI-приложений являются:

* Простота использования и интуитивно понятный интерфейс
* Возможность легко переключаться между открытыми документами
* Более гибкая организация рабочего пространства

## Выводы к главе 1

1. Знакомство с SoftClub:
   * SoftClub - ИТ-компания, специализирующаяся на разработке ПО и консалтинговых услугах.
   * Компания провела презентацию о работе в ИТ-сфере и возможностях стажировки.
   * Стажировка в SoftClub позволяет получить практический опыт в разработке финансового ПО.
2. Курс "Контроль версий с помощью Git":
   * Курс включает теоретическую и практическую части, а также промежуточные тесты.
   * Практические занятия позволили закрепить теоретический материал и получить опыт работы с Git и GitHub.
   * По итогам курса был предоставлен отчет об успеваемости, что позволяет проанализировать сложности и улучшить навыки.
3. Основные концепции системы контроля версий Git:
   * Commit - сохранение изменений в локальном репозитории с комментарием.
   * Push - отправка локальных коммитов в удаленный репозиторий.
   * Branch - ветвление для независимой разработки.
   * Remotes - синхронизация локального репозитория с удаленными.
   * Merge - объединение изменений разных веток.
   * Stash - временное хранение незакоммиченных изменений.
   * Tags - пометка ключевых моментов в истории проекта.
4. Работа с удаленными репозиториями

* Взаимодействие с удаленными репозиториями на платформе GitHub включает в себя связывание локального репозитория с его удаленным аналогом, размещенным на GitHub.
* Это позволяет эффективно организовывать совместную работу над проектом, обмениваться наработками и отслеживать историю развития кодовой базы.
* Для этого используются операции push и pull, обеспечивающие синхронизацию изменений между локальной и удаленной версиями проектов.

1. Unit-тестирование

* Unit-тесты представляют собой небольшие, изолированные проверки, предназначенные для оценки корректности и соответствия ожидаемым результатам отдельных компонентов программного кода.
* Данный инструмент широко используется разработчиками для валидации правильности работы конкретных модулей или функций программы.  
  Применение unit-тестов позволяет удостовериться, что каждый элемент кода функционирует корректно независимо от остальных частей программы, что в свою очередь способствует выявлению и устранению ошибок на ранних этапах разработки.
* Unit-тестирование является важным компонентом в процессе разработки программного обеспечения, поскольку оно обеспечивает стабильность и надежность конечного кода.

1. Разработка простейшего GUI-приложения в QT

* Разработка простейшего GUI-приложения в Qt - это процесс создания графического пользовательского интерфейса (GUI) с использованием фреймворка Qt.
* Qt является кроссплатформенной библиотекой для разработки приложений с графическим интерфейсом, поддерживающей различные языки программирования, такие как C++, Python, Java и другие.
* Разработка простейшего GUI-приложения в Qt позволяет создавать интуитивно понятные и функциональные пользовательские интерфейсы, которые могут работать на различных платформах, включая Windows, macOS, Linux, Android и iOS.

1. Разработка приложения, отражающего табличные данные, является важной и комплексной задачей, требующей комплексного подхода. Ключевые аспекты при разработке такого приложения включают:

* Выбор подходящей технологической платформы и фреймворка.
* Тщательное проектирование пользовательского интерфейса.
* Эффективное взаимодействие с данными из различных источников.
* Использование эффективных средств визуализации данных.
* Реализация механизмов для обработки пользовательских взаимодействий.
* Обеспечение масштабируемости и высокой производительности.
* Проведение всесторонних тестов для обеспечения стабильности и надежности.
* Создание подробной документации для поддержки и дальнейшего развития приложения.

1. Разработка простейшего SDI-приложения

* SDI-приложения - это тип графических пользовательских интерфейсов (GUI), в котором каждый открытый документ отображается в отдельном окне.
* Это противоположность MDI-приложениям (Multiple Document Interface), где документы открываются во вложенных окнах внутри главного окна приложения.
* Основными преимуществами SDI-приложений являются простота использования и интуитивно понятный интерфейс, возможность легко переключаться между открытыми документами, а также более гибкая организация рабочего пространства.

# Глава 2 Практическая часть курса

## 2.1 Unit-тестирование

Задача: составить UNIT-тест в рамках модульного тестирования.

void TestCalculator::init()

{

mCalc.SetA(A0);

mCalc.SetB(B0);

}

void TestCalculator::testDiff\_data()

{

QTest::addColumn<int>("a");

QTest::addColumn<int>("b");

QTest::addColumn<int>("result");

QTest::newRow("all 0") << 0 << 0 << 0;

QTest::newRow("same number") << 10 << 10 << 0;

QTest::newRow("opposite number") << 10 << -10 << 20;

QTest::newRow("positive res") << 10 << 5 << 5;

QTest::newRow("negative res") << 5 << 10 << -5;

}

void TestCalculator::testConstructor()

{

Calculator c1;

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

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

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");

}

void TestCalculator::testDiff()

{

QFETCH(int, a);

QFETCH(int, b);

QFETCH(int, result);

mCalc.SetA(a);

mCalc.SetB(b);

QCOMPARE(mCalc.Diff(), result);

}

void TestCalculator::testSum()

{

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

const int A = 10;

const int B = 2;

mCalc.SetA(A);

mCalc.SetB(B);

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

}

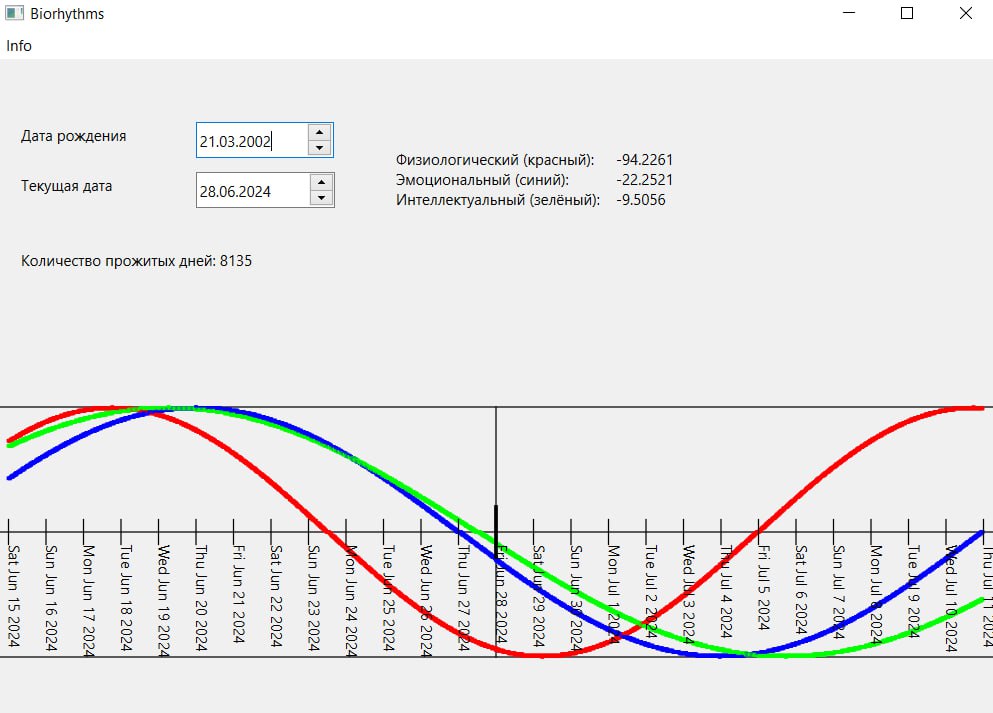
QTEST\_APPLESS\_MAIN(TestCalculator)

Этот код демонстрирует основы модульного тестирования класса Calculator с использованием фреймворка Qt Testing. Он проверяет различные аспекты функциональности класса, такие как конструктор, вычисление разности и суммы.

## 2.2 Разработка простейшего GUI-приложения в QT

Задача: разработать простейшее GUI-приложения в QT

Реализация: реализует визуализацию биологических ритмов человека в зависимости от количества прожитых дней. Пользователь может видеть графическое отображение физиологического, эмоционального и интеллектуального состояния на заданную дату.



## 2.3 Разработка приложения, отражающего табличные данные

## 2.4 Разработка простейшего SDI-приложения

Задача: Разработать простейшее SDI-приложение в QT

QMyForm::QMyForm(QWidget \*parent)

: QMainWindow(parent)

{

QMenu\* file = new QMenu("&File");

QMenu\* info = new QMenu("&Info");

QWorkForm\* form = new QWorkForm;

file->addAction("&Open", form, SLOT(slotLoad()));

info->addAction("&Help", this, SLOT(slotInfo()));

file->addAction("&Save", form, SLOT(slotSave()));

file->addAction("&Save as", form, SLOT(slotSaveAs()));

file->addSeparator();

file->addAction("&Quit", qApp, SLOT(quit()));

menuBar()->addMenu(file);

menuBar()->addMenu(info);

statusBar()->setVisible(false);

setCentralWidget(form);

connect(form, SIGNAL(ChangeTitle(const QString&)), SLOT(slotChangeTitle(const QString&)));

}  
  
этот конструктор настраивает главное окно приложения, добавляя в него меню "File" и "Info", а также устанавливая центральный виджет QWorkForm. Также устанавливается обработчик сигнала ChangeTitle(const QString&), испускаемого объектом QWorkForm.



# 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!

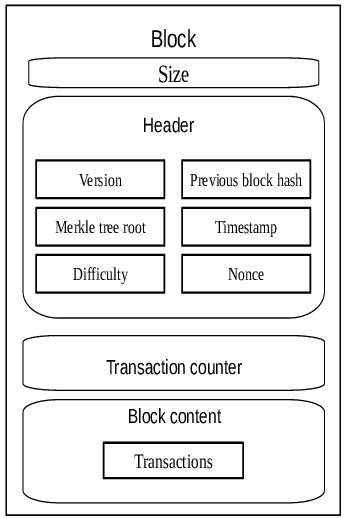
Недопустим в главе только один раздел, в разделе – только один подраздел.

## 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)