МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ

по учебной вычислительной (ознакомительной) практике

Петрачкова Глеба

Александровича

студента 1 курса, 7 группы

специальность «Прикладная

математика»

Руководитель практики:

старший преподаватель

О.О.Колб

Минск, 2024

# ОГЛАВЛЕНИЕ

[Введение 3](#_gjdgxs)

[Глава 1 Теоретические основы 4](#_30j0zll)

[1.1 Общая характеристика курса 4](#_1fob9te)

[1.2 Модули курса 4](#_3znysh7)

[1.3 Выводы к главе 1 4](#_2et92p0)

[Глава 2 Практическая часть курса 5](#_tyjcwt)

[2.1 Решение задач 5](#_3dy6vkm)

[2.2 Дополнительный материал 5](#_1t3h5sf)

[2.3 Выводы к главе 2 5](#_4d34og8)

[Глава 3 Индивидуальный проект 6](#_2s8eyo1)

[3.1 Функциональное назначение 6](#_17dp8vu)

[3.2 Структура проекта 6](#_3rdcrjn)

[3.3 Интерфейс приложения 7](#_26in1rg)

[3.4 Описание проекта 7](#_lnxbz9)

[3.5 Выводы к главе 3 7](#_35nkun2)

[Заключение 8](#_1ksv4uv)

[Список использованных источников 9](#_44sinio)

# 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

## 1.1 Лекция компании SoftClub

SoftClub — группа компаний, которая специализируется на разработке решений для банков, финансовых компаний, электронной торговли, биржевой отрасли и цифровой экономики в Восточной Европе и странах СНГ.

В рамках лекции, проведенной представителями компании, были рассмотрены вопросы, связанные с разработкой программного обеспечения для предоставления банковских услуг, а также проведен краткий экскурс в работу компании, принципы работы банковской сферы, основы финансовой грамотности.

## 1.2 Знакомство с системой контроля версий Git. Тренинг на платформе EPAM.

В рамках выполнения первого этапа подготовительной части учебной практики рассматривался курс, подготовленный компанией EPAM, направленный на ознакомление с системой управления версиями Git.

В рамках вступительной части данного курса были рассмотрены преимущества использования системы версионного контроля, произведены установка Git на локальный компьютер (с конфигурацией имени пользователя и электронной почты) и создание собственного репозитория с открытым доступом посредством регистрации на платформе GitHub.

В рамках основной части курса был рассмотрен ряд вопросов, связанных с работой с файлами посредством системы версионного контроля Git с помощью инструкций командной строки в приложении Git Bash, а также элементов графического интерфейса (git gui и gitk). В частности, были изучены вопросы создания и изменения коммитов; просмотра истории коммитов; контроля изменений версий файлов как посредством их правки на локальном устройстве, так и при изменении версии на GitHub; работа с ветками; слияние изменений и разрешение конфликтов.

Изучения теоретического материала курса сопровождалось прохождением тестов и работой с собственным репозиторием с реализацией на практике представленных в курсе команд.

В заключительной части курса были рассмотрены и реализованы конкретные практические задания, направленные на закрепление полученных в рамках курса умений и навыков по работе с системой версионного контроля Git.

## 1.2.1 Модули курса

1. Основы системы контроля версий:

- Что такое система контроля версий и почему она необходима для разработки программного обеспечения.

- Преимущества использования системы контроля версий Git.

2. Настройка Git:

- Установка Git на локальную машину.

- Конфигурация имени пользователя и электронной почты в Git с помощью команд git config --global user.name "Your Name" и git config --global user.email "your.email@example.com".

3. Базовые операции:

- Создание и коммит изменений с использованием команд git add <file> для добавления файлов в индекс и git commit -m "Commit message" для фиксации изменений.

- Просмотр истории коммитов с помощью команды git log.

- Работа с ветками: создание новой ветки с помощью команды git branch <branchname>, переключение на другую ветку с помощью команды git checkout <branchname>.

- Восстановление предыдущих версий файлов с использованием команды git checkout <commit> -- <file>.

4. Слияние изменений (Merge):

- Понимание процесса слияния веток.

- Создание слияний (merge) веток в Git с помощью команды git merge <branchname>.

- Разрешение конфликтов при слиянии путем ручного редактирования файлов.

5. Разрешение конфликтов:

- Понимание возможных конфликтов при слиянии.

- Ручное разрешение конфликтов в Git путем редактирования конфликтных файлов.

- Использование инструментов для автоматического разрешения конфликтов.

## 1.2.2 Выполнение тестовых заданий

В рамках прохождения курса были выполнены шесть тестов по следующим темам:

- Настройка Git

- Базовые операции

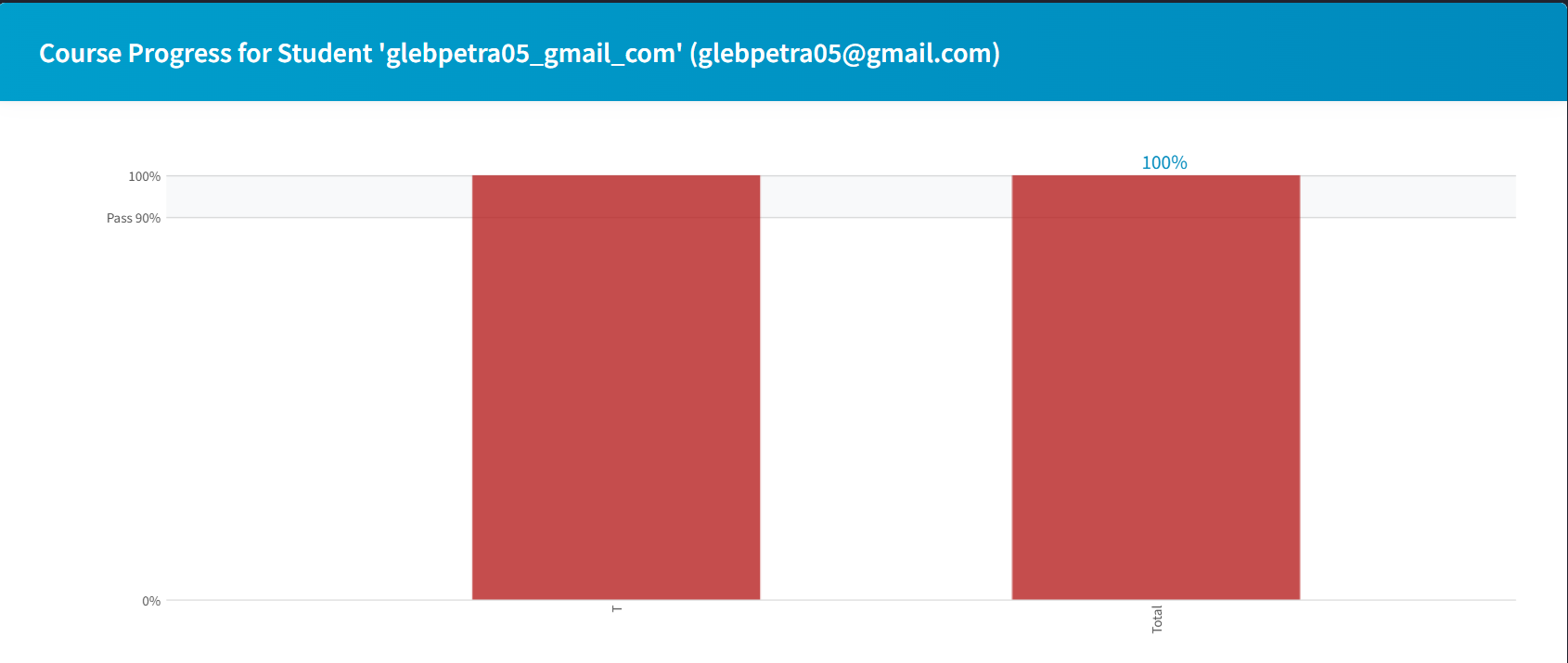
- Внутреннее устройство Git

- Отмена действий

- Ветки и merge

- Stash, tags, remotes

Итоговый результат прохождения курса:



## 1.3 Выводы к главе 1

Практически каждый современный проект использует системы версионного контроля. Git - самый популярный инструмент для контроля версий. Курс от компании EPAM помог научиться использовать git для решения ежедневных задач. Прохождение курса также улучшило понимание различных стратегий организации ветвления в проекте и выбора оптимальной стратегии.

Основная цель прохождения курса - подготовить фундамент для изучения программирования, эффективного использования современных инструментов разработчика.

# Глава 2 Практическая часть курса

## 2.1 Решение задач

**2.1.1 Unit-тестирование**

Постановка задачи: дополнение ранее разработанных классов системой Unit-тестов (для консольных приложений).

Реализация: Для приобретения навыков работы с модульным тестированием на практике были реализовано создание юнит-тестов для примитивных классов (в частности, класса Calculator, реализующего основные арифметические операции), с помощью использования среды для модульного тестирования Qt Test.

Реализация юнит-тестов для описанного класса осуществлялась как посредством ручного сравнения результатов каждого теста с различными входными значениями, так и посредством тестирования на основе данных с созданием функций данных. Реализация осуществлялась с использованием основных макросов (QVERIFY, QVERIFY2, QCOMPARE и т.д.)

Класс Калькулятор (исходный класс):

class Calculator

{

public:

Calculator(int a = 0, int b = 0) : mA(a), mB(b) { }

int GetA() const { return mA; }

void SetA(int a) { mA = a; }

int GetB() const { return mB; }

void SetB(int b) { mB = b; }

bool isNull() { return mB == 0; }

int Sum() const { return mA + mB; }

int Diff() const { return mA - mB; }

int Mult() const { return mA \* mB; }

int Div() const { return mA / mB; }

private:

int mA;

int mB;

};

Класс тестирования (заголовочный файл):

#include <QtTest>

class TestCalculating : public QObject

{

Q\_OBJECT

// public:

// TestCalculating();

// ~TestCalculating();

private slots:

void init();

void testSum();

void testSum\_data();

void testDiv();

private:

Calculator mCalc;

const int A0 = 0;

const int B0 = 0;

};

Класс тестирования (исполняемый файл с реализацией конструктора):

#include "tst\_testcalculator.h"

void TestCalculator::init()

{

mCalc.SetA(A0);

mCalc.SetB(B0);

}

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::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)

Класс тестирования (исполняемый файл с реализацией через функции данных):

#include <QtTest>

#include "tst\_testcalculating.h"

void TestCalculating::init()

{

mCalc.SetA(A0);

mCalc.SetB(B0);

}

void TestCalculating::testSum\_data()

{

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

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

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

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

QTest::newRow("10 and 10") << 10 << 10 << 20;

QTest::newRow("20 and 4") << 20 << 4 << 24;

}

void TestCalculating::testSum()

{

QFETCH(int, a);

QFETCH(int, b);

QFETCH(int, result);

mCalc.SetA(a);

mCalc.SetB(b);

QCOMPARE(mCalc.Sum(), result);

}

void TestCalculating::testDiv()

{

const int A = 20;

const int B = 10;

mCalc.SetA(A);

mCalc.SetB(B);

QCOMPARE(mCalc.Div(), A / B);

// QSKIP("Are you sure we need continue?");

QEXPECT\_FAIL("", "we must avoid dividing on 0", Abort);

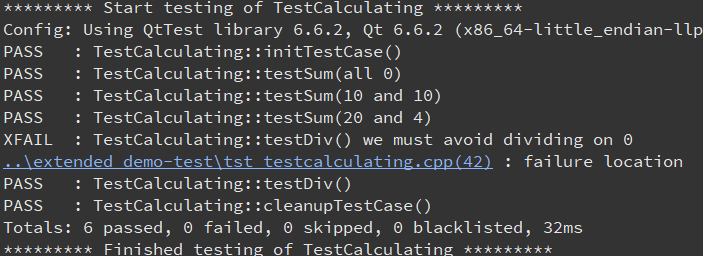
QVERIFY(mCalc.isNull());

QCOMPARE(mCalc.Div(), A0 / B0);

}

QTEST\_APPLESS\_MAIN(TestCalculating)

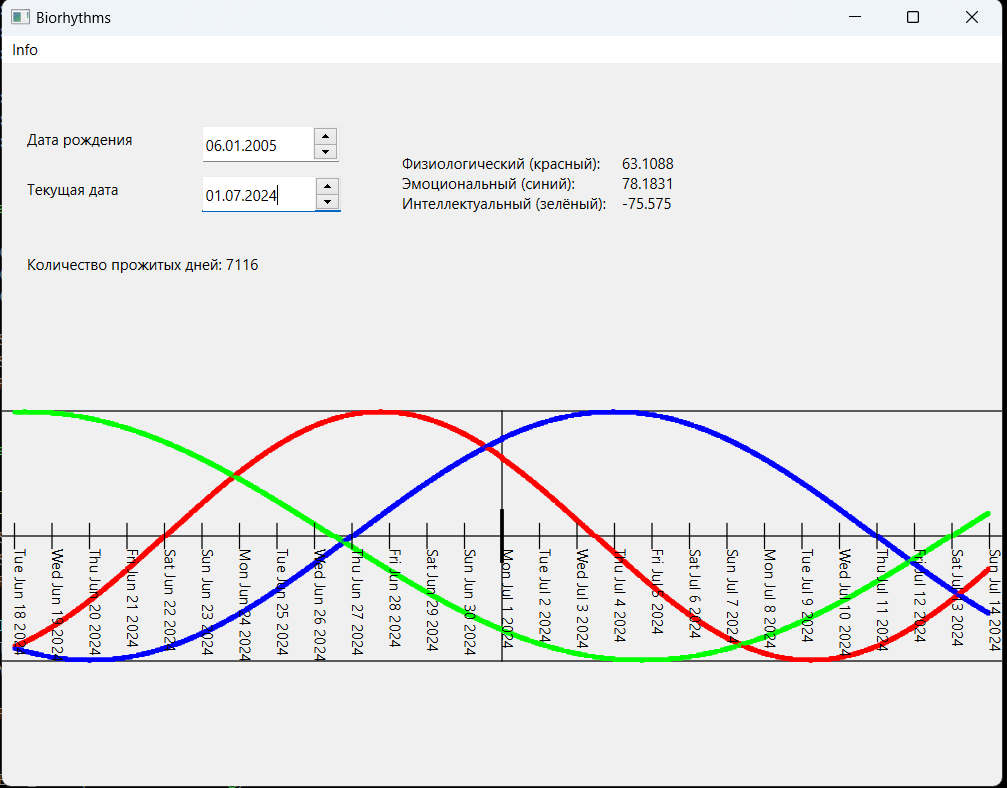
Вывод приложения:



**2.1.2 Разработка простейшего GUI-приложения в QT**

Постановка задачи: Диалоговое окно с виджетами, обеспечивающими ввод информации, вызов расчетной части приложения и вывод результатов.

Реализация:



## 2.2 Дополнительный материал

Перечислить выполненные дополнительные работы

## 2.3 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!

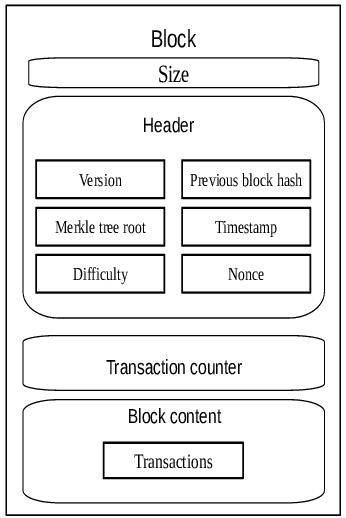
Недопустим в главе только один раздел, в разделе – только один подраздел.

## 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)