МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

ОТЧЕТ ПО учебной вычислительной

практике

Артимени Ивана Сергеевича

студента 1(2) курса,

специальность «Прикладная математика»

Руководитель практики:

старший преподаватель

Е.В. Пазюра

Минск, 2024

ОГЛАВЛЕНИЕ

[Введение 3](#_Toc170721660)

[Глава 1 Теоретические основы выбранного курса (КУРСОВ) 4](#_Toc170721661)

[1.1 Общая характеристика курса 4](#_Toc170721662)

[1.2 Модули курса 4](#_Toc170721663)

[Глава 2 Практическая часть курса 4](#_Toc170721664)

[2.1 Решение задач 4](#_Toc170721665)

[2.2 Дополнительный материал 14](#_Toc170721666)

[Список использованных источников 15](#_Toc170721667)

Введение

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

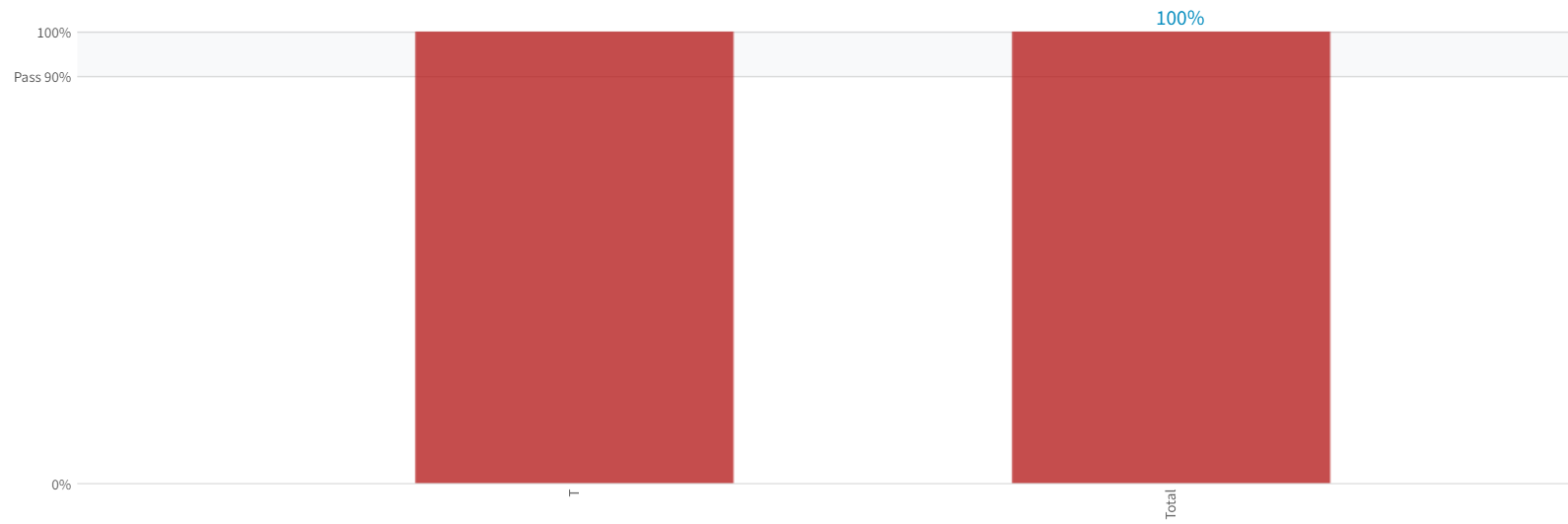
* цель практики, задачи практики;
* выбранные направления, их актуальность;
* логин/email регистрации с указанием системы тестирования (training/learning/autocode/codewars другой);
* краткое представление (наличие) Индивидуального проекта.

*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»)

Глава 1 Теоретические основы выбранного курса (КУРСОВ)

1.1 Общая характеристика курса

Прослушан курс Epam по системе контроля версий git [[1]](#Закладка1) и Создан свой аккаунт на GitHub [[2]](#Закладка2). Результаты курса показаны на слайде:

Также была посещена и прослушана презентация компании Softclub, в которой подробно описывались направления и нюансы её работы. Также, помимо работы, нас знакомили с её особенностями. Про компанию рассказывали не только очно, но и по сети.

1.2 Модули курса

Перечислить модули, которые были изучены.

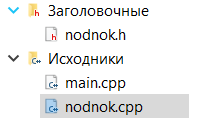
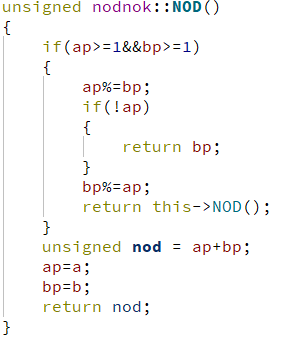
Здесь следует перечислить и кратко описать разделы изученного материала со ссылками на источники.

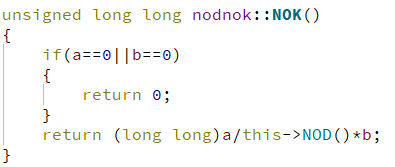
Глава 2 Практическая часть курса

2.1 Решение задач

2.1.1 Выполнена задача по написанию программы для нахождения НОД и НОК двух чисел.

Для нахождения НОД и НОК двух чисел использовался алгоритм Евклида. Программа была разделена на несколько частей:

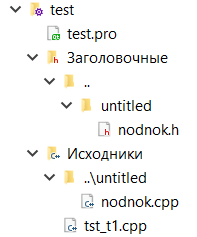
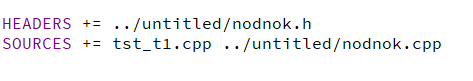
* nodnok.h
* main.cpp
* nodnok.cpp

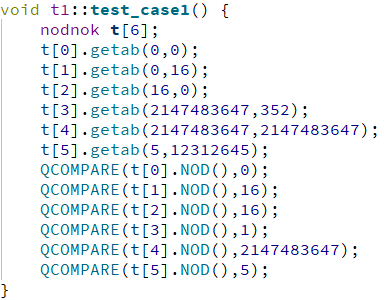
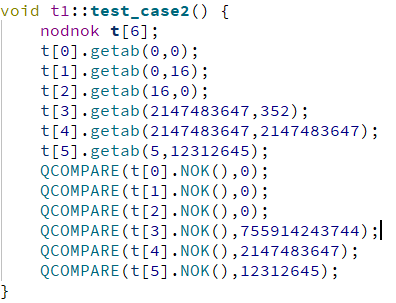


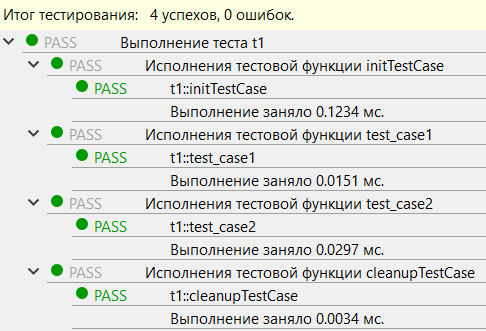
Также был добавлен подпроект test для тестирования функций.

В части test.pro были добавлены следующие строки кода:

(Примечание: нужно убрать лишнюю строку HEADERS)

В tst\_t1.cpp были написаны тесты



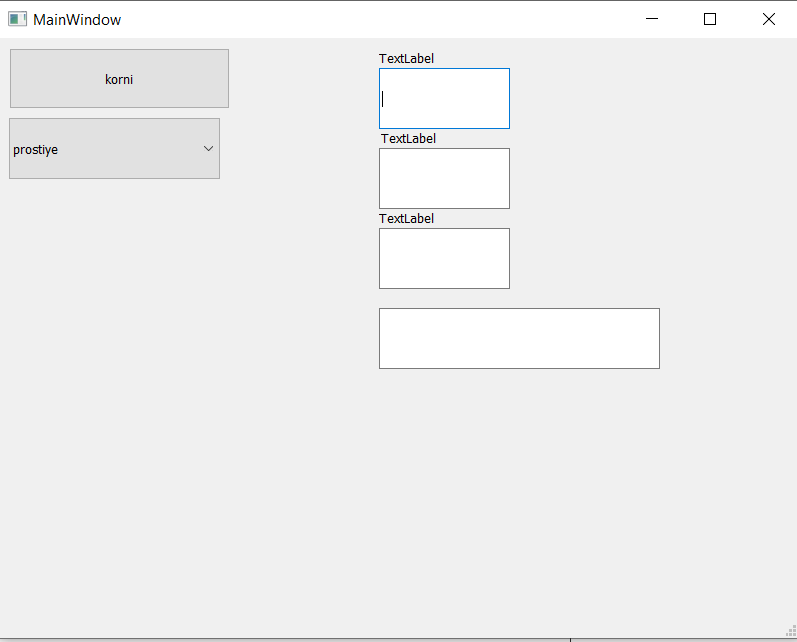
Тесты были успешно пройдены:

2.1.2 Выполнена задача по созданию графического интерфейса для трех функций:

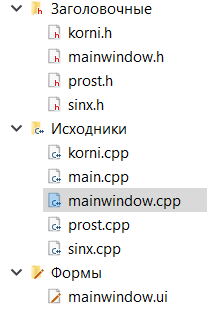
- проверка числа на простоту

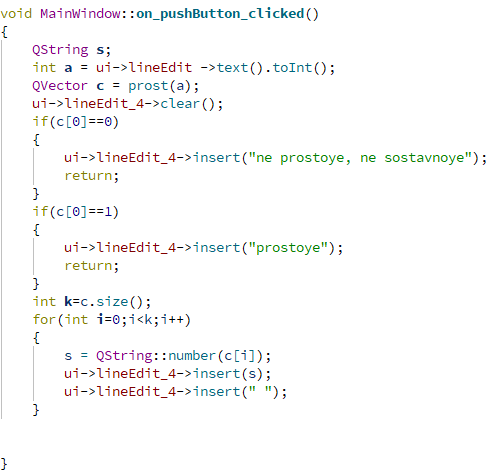
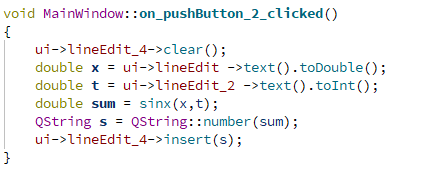
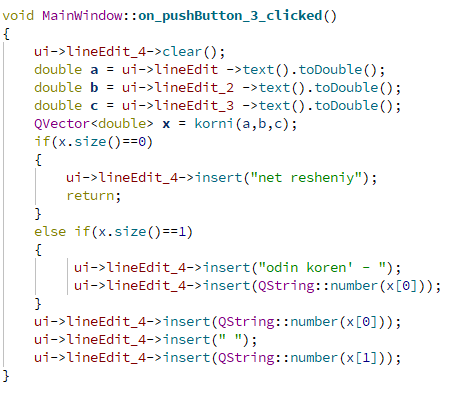
- подсчет синуса числа с заданной точностью

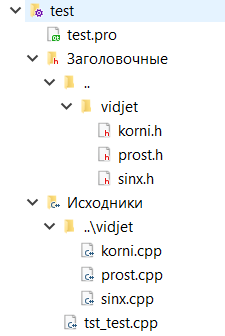
- подсчет корней квадратного уравнения

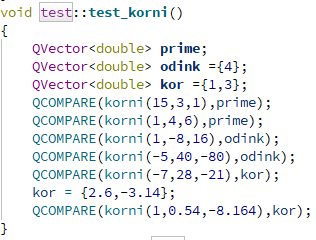
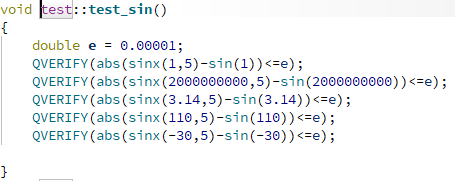
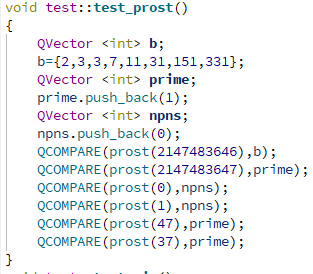
Вид графического интерфейса представлен на скриншоте:

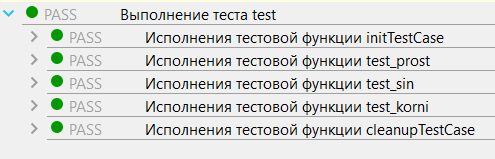
В графическом интерфейсе присутствует до 3 полей ввода данных, кпопка активации, окно вывода и comboBox.

Каждая функция написана в отдельном файле:

В mainwindow.cpp все три функции используются в функциях, срабатывающих при нажатии кнопок

Как видим, в программе существуют 3 кнопки. ComboBox позволяет использовать нужную, скрывая остальные:

Для этой программы написаны тесты:

Тесты были успешно пройдены:

2.1.3 Задача для программирования шаблона модель – вид

Создать два списка разной структуры:

Разные названия

Разные элементы структур

Дополнение, сортировка, поиск

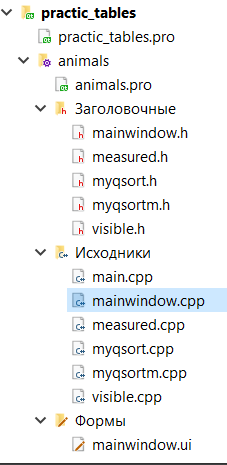
Вторая часть задачи – визуализация

Корректировка(при нажатии кнопки обновляет список)

Файлы для хранения:

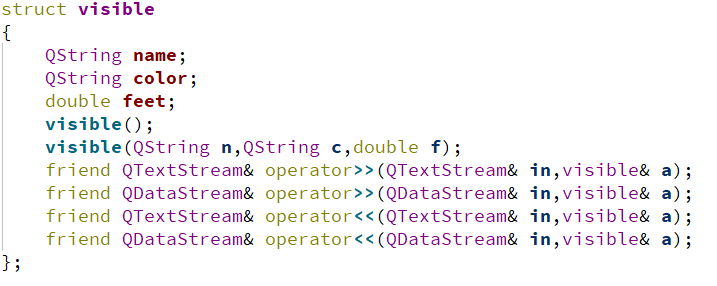
Текстовый

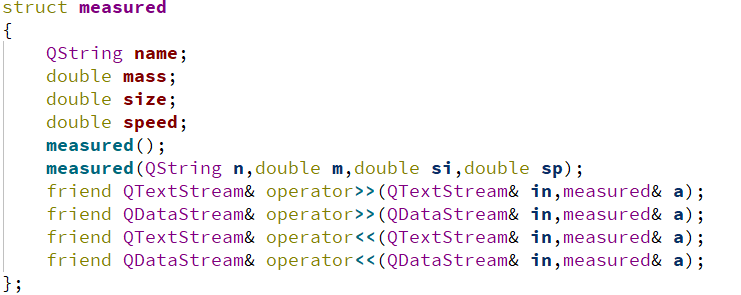
Бинарный

Задача была разбита на несколько файлов

Объектами задачи были выбраны животные

Созданы две структуры: visible и measured.

Элементами структуры visible являются название животного, количество его ног и цвет.

Структура measured - название, масса, размер и скорость.

Сам графический интерфейс выглядит следующим образом:

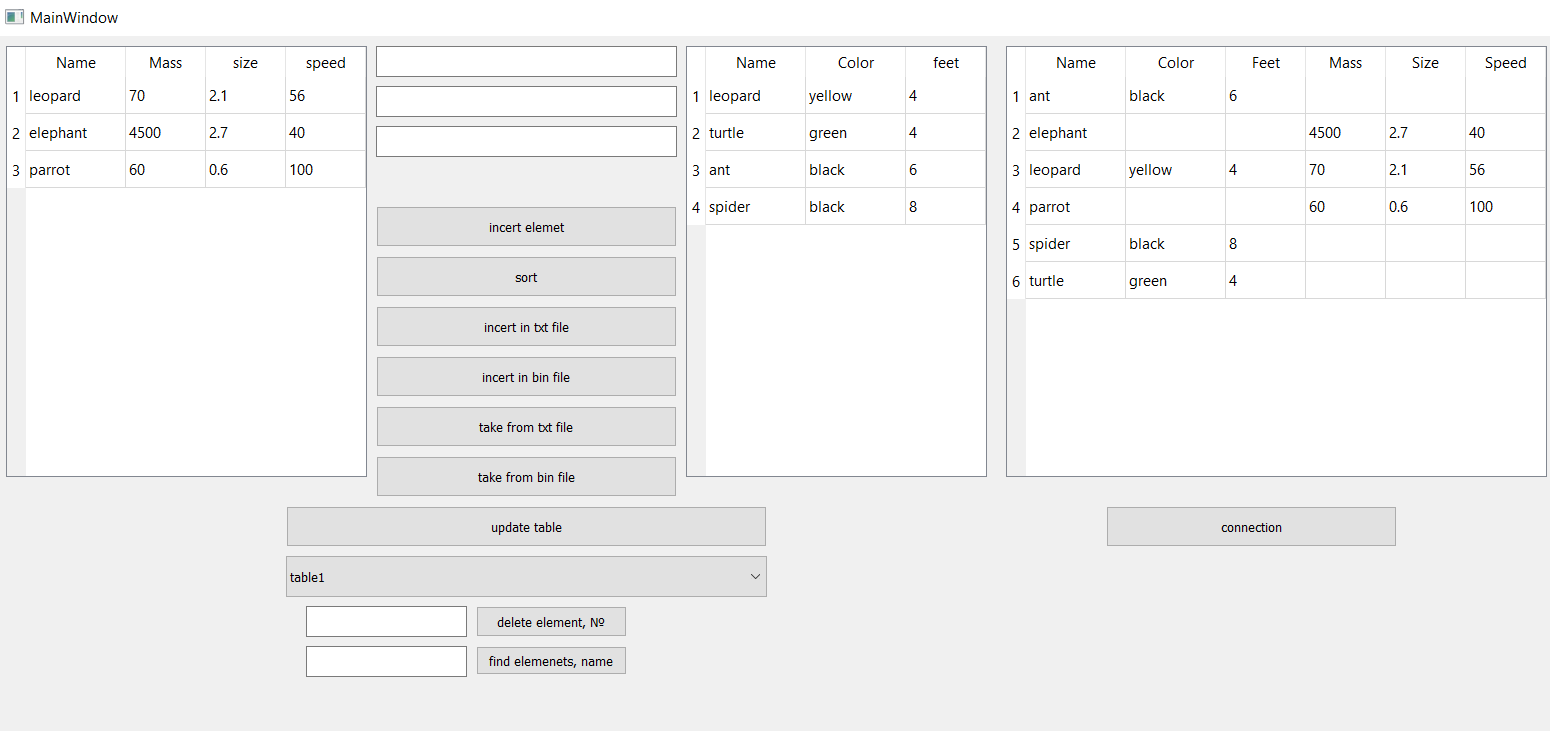
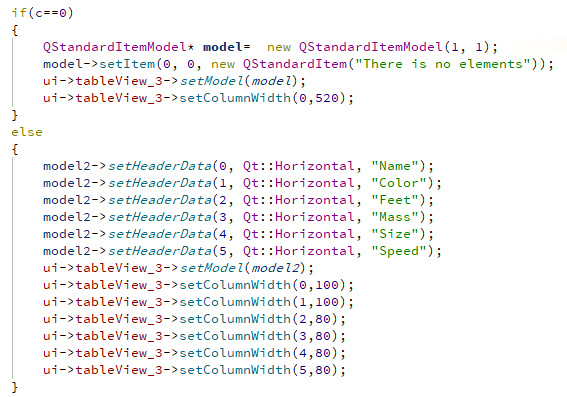
Таблица measured находится слева, visible находится в центре. Правая – дополнение этих двух таблиц.

 Таблица дополнения вписывает только известные данные о животных, с левой или средней таблицы. Однако, если это животное есть в обеих таблицах, данные вводятся из двух. Реализация представлена на следующих скриншотах:

В цикле проходит проверка на подобность имен в двух массивах данных visibles и measureds. Если имена не одинаковы, то вводится тот элемент, который идет раньше по алфавиту. Цикл заканчивается, когда проверены все элементы хотя бы одного массива.

 После этого цикла идет добавление оставшегося:

После этого идет вывод этого дополнения



2.2 Дополнительный материал

Список использованных источников

1. <https://elearn.epam.com/courses/course-v1:EPAM+VCG+RU/courseware/8a58c84fd1d2474b8f69a15171f524ae/ab5e12cb97ca4c3983af26e9bd1f715c/>

[обратно](#Закладка1обр)

1. <https://github.com/Artimenia-Ivan?tab=repositories>

[обратно](#Закладка2обр)