Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

**Лабораторная работа №12**

По дисциплине «Основы программной инженерии»

Исполнители

Студенты 1 курса

Браганец И. А.

Глушко Д. Д.

Заяц Р. В.

Казанцева Т. И.

Сосновец М. И.

Черная Я. Р.

Минск 2022

**Содержание:**

**1.Введение1**

1.1. Наша команда 1

1.2. Цели1

1.3. Задачи1

1.4. План разработки приложения1

**2.Приложение Convert**2

2.1 Интерфейс2

2.2 Правила пользования2

**3.Техническая документация к приложению3**

3.1.Структура приложения.3

3.2. Пользовательская история и модули3

3.2.1. Модуль input.cpp3

3.2.2. Модуль verifying.cpp3

3.2.3. Модуль transformToRequest.cpp3

3.2.4. Модуль getCurrencyJson.cpp4

3.2.5. Модуль jsonParcer.cpp4

3.2.6. Модуль calcSumOut.cpp4

3.2.7. Пользовательская история проекта4

1. **Введение**

**1.1. Наша команда**

1. Браганец Иван – разработчик (модули main)

2. Глушко Денис – менеджер – модуль input

3.Заяц Ренат – разработчик (модули getCurrencyJson, jsonParcer)

4. Казанцева Татьяна – веб-разработчик – модуль verifying

5. Сосновец Мария – тестировщик – модуль transormToRequest

6. Черная Яна – тестировщик – модуль calcSumOut

**1.2. Цель**

Консольное приложение на языке C++ главная цель которого – конвертация суммы денег из одной валюты в другую. При создании данного проекта мы использовали навыки, полученные во время обучения.

**1.3. Задачи**

1. Создать алгоритм

2. Разработать приложение

3. Использовать ранее полученные знания ЯП С++

4. Изучить новые методики программирования

**1.4. План разработки приложения**

-Создать примитивный консольный интерфейс к приложению и оформление.

-Разработать:

Правила пользования

Графическое представление в консоли

Логику приложения

Выход из приложения, в зависимости от результата.

-Написать инструкцию к эксплуатации.

1. **Приложение Convert**

**2.1. Интерфейс**

Интерфейс представляет собой стандартную консоль windows, на которую выводятся пункты для ввода значений(вводимая валюта, выводимая валюта, сумма для перевода), необходимых для работы приложения и примитивно оформленный вывод результата работы приложения.

**2.2 Правила пользования**

При запуске приложения Convert перед вами появляется консоль. Тут же для начала работы у нас просят ввести поочерёдно (для подтверждения введённого значения требуется нажать клавишу Enter):

1. Валюту которую вы собирайтесь конвертировать
2. Валюту в которую будет конвертироваться ваша сумма
3. Сумму, которую требуется конвертировать

После ввода всех 3 значений консоль в отдельных строках выводится упрощённая информация о конвертации и результат перевода. После чего работа приложения считается завершённой. Для повторного запуска следует запустить приложение снова.

Уточнения по работе приложения

1. Валюты вводятся в любом регистре и должны быть в списке abbreviations.txt
2. Приложение использует курсы валют Нацбанка, обновляемые в режиме реального времени, так как используются данные напрямую с их сайта
3. **Техническая документация к приложению.**
   1. **Структура приложения**

Проект полностью написан на C++. При разработке использовались библиотеки: iostream, iomanip, curl, fstream, cstring. Основная структура проекта представляет собой один главный модуль вызывающий подключённые к нему 6 дополнительных модулей. Модули взаимосвязаны между собой. То есть существуют переменные передаваемые из одного модуля в другой. При разработке приложения использовался API для получения курсов валют.

* 1. **Пользовательская история и модули**
     1. **Модуль input.cpp**

Обеспечивает:

- ввод валют для конвертации

- перевод в верхний регистр валютных аббревиатур

- ввод суммы

- проверку на валидность введённых значений

Пользовательская история для **input.cpp**:

Как пользователь, я хочу иметь возможность ввода значений и валют поочерёдно и в любом регистре.

* + 1. **Модуль verifying.cpp**

Обеспечивает:

- проверка на валидность введенной валюты

- проверка есть ли валюта в списке abbreviations.txt

Пользовательская история для **verifying.cpp**:

Как пользователь, я хочу чтобы конвертация происходила только с существующими валютами из списка.

* + 1. **Модуль transformToRequest.cpp**

- создает ссылку-запрос для валюты будущего модуля

Я ебу что тут писать

(

Пользовательская история для **verifying.cpp**:

Как пользователь, я хочу чтобы конвертация происходила только с существующими валютами из списка.

)

* + 1. **Модуль getCurrencyJson.cpp**

Модуль запрашивает данные с сайта Нацбанка в реальном времени для удобства пользования приложением.

А именно:

- делает запрос API, который возвращает параметры валюты в виде json.

- возвращает json строку

Пользовательская история для **getCurrencyJson.cpp**:

Как пользователь, я хочу чтобы конвертация происходила с актуальными курсами валют по Нацбанку

Диаграмма использования **getCurrencyJson.cpp**:

Изображение выглядит как текст, снимок экрана, компьютер

Автоматически созданное описание

* + 1. **Модуль jsonParcer.cpp**

Модуль парсит json строку в объект структуры, для того чтобы достать нужные нам значения с возвращаемой строки

* + 1. **Модуль calcSumOut.cpp**

Модуль высчитывает по алгоритму конечную и выводимую вам сумму

* + 1. **Пользовательская история проекта:**

Как **пользователь** я хочу **использовать «Convert»**, чтобы конвертировать в нужную мне валюту сумму по актуальному курсу синхронизированному с Нацбанком.

Диаграмма использования приложения:

Изображение выглядит как текст

Автоматически созданное описание