**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ,**

**СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»**

**(СПбГУТ)**

**Санкт-Петербургский колледж телекоммуникаций им Э.Т. Кренкеля**

**Отчёт о выполнении лабораторной работы № 15,16**

**Тема:** **Разработка приложений на базе MFC ApplicationWizard.**

**Разработка MDI-приложений**

**по дисциплине "Системное программирование"**

**Вариант№ 3**

Принял:

преподаватель Кривоносова Н.В

Выполнил: студенты группы ЗФ-053/054

Фомина Е.А.

Ткачук А.В

Шилаковски М.Э

Семененков К.И

Санкт-Петербург

2022 год

**Цель работы:**

Научиться разрабатывать и реализовывать простейшие программы на языке C++. Получить практические навыки работы по использованию графического интерфейса. Научиться связывать переменные и методы с элементами диалогового окна.

**Задание:**

Вычислить стоимость поездки на автомобили, если известно цена литра бензина, расход бензина на 100 км пути и пройденный километраж.

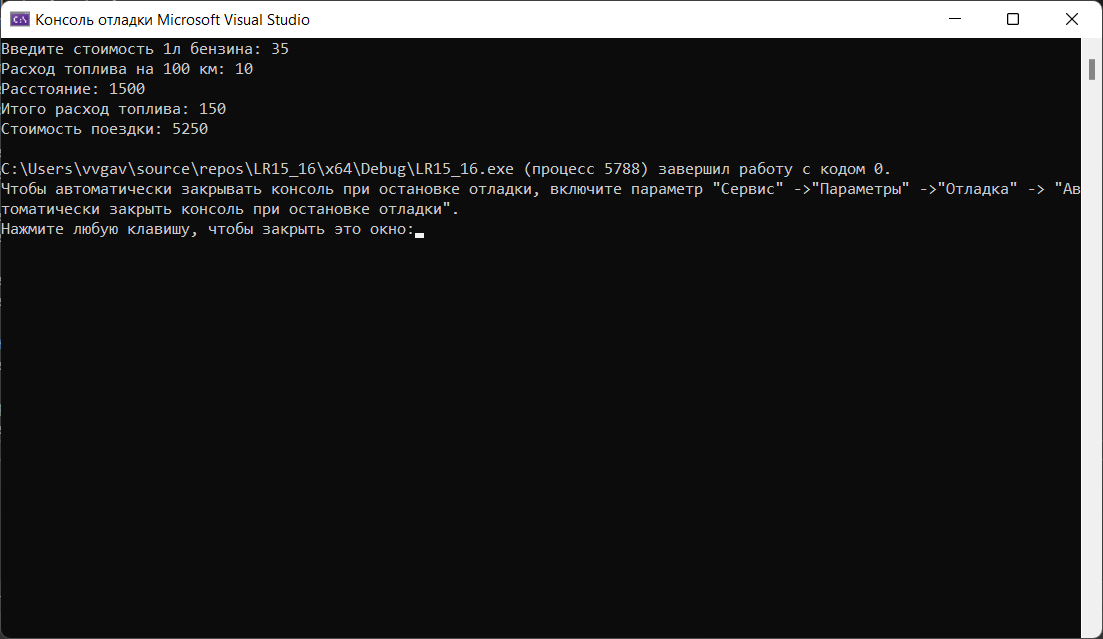
**Ссылка на гитхаб:**

**https://github.com/SPbCTTSFS/SPbCT\_TA\_SM\_FE\_SK**

**Ход выполнения работы:**

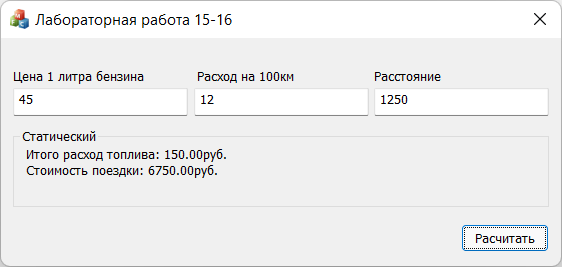
1. Создать пустой проект Console Application и реализовать в нем программу по вычислению стоимости поездки на автомобиле (исходными данными являются: километраж, расход топлива, цена единицы топлива).

Результат работы консольного приложения.



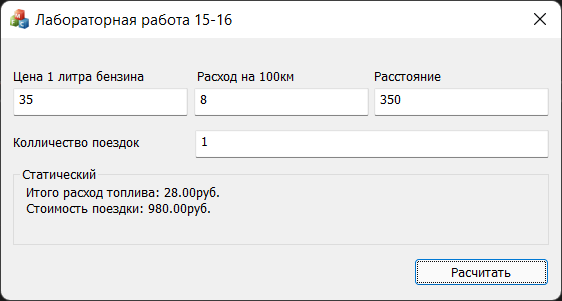
1. Создать приложение с помощью AppWizard (exe), с диалоговым окном в качестве главного. Реализовать на основе данного приложения задачу из первого пункта.

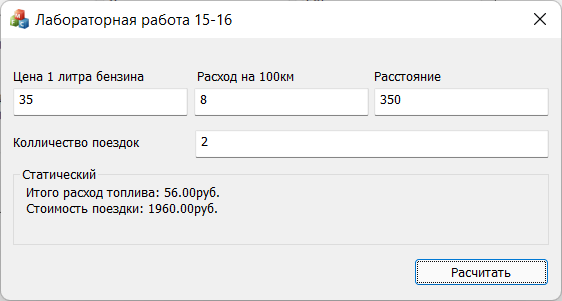
Результат работы диалогового окна на основе консольного приложения



1. Усовершенствовать полученную во втором пункте программу, добавив подсчет стоимости нескольких поездок.

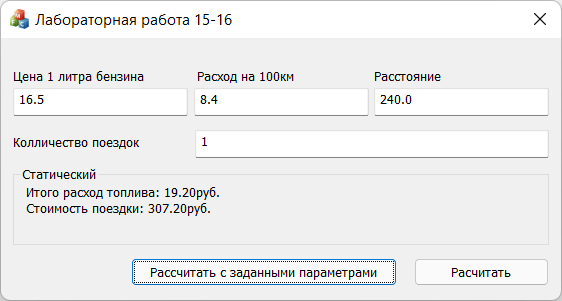
Усовершенствовать полученную программу, добавив подсчет стоимости нескольких поездок.





1. Усовершенствовать полученную программу, добавив кнопку при нажатии на которую вводятся заранее определенные значения (километраж = 240 км, расход топлива = 8.4 л/км, цена литра топлива = 16.5).

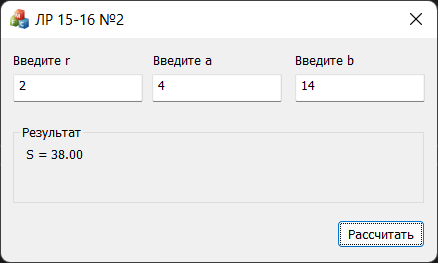
Добавляем в окно кнопку, которая будет считать с заданными параметрами



1. Создать приложение на основе диалогового окна, производящее вычисление по формуле, предложенной преподавателем.

Создать приложение на основе диалогового окна, производящее вычисление по формуле, предложенной преподавателем.

Формула : S = r^3 +a^2 +b.



Ссылка на репозиторий: <https://github.com/suai2006/SPbCT_GavrilyukVV/tree/main/LR15_16>

**Ответы на контрольные вопросы:**

1. **В чем разница между консольным приложением и приложением на основе оконного интерфейса?**

Оконное приложение – строится на базе специального набора функций API, составляющих графического пользовательского интерфейса. Оконное приложение представляет собой программу, которая весь вывод на экран производит в графическом режиме. Первым результатом работы оконного приложения является отображение на экране специального объекта – окна. После того, как окно отображено на экране, вся работа приложения направлена на то, чтобы поддержать его в актуальном состоянии

Неоконное приложение, также называемое консольным, представляет собой программу, работающую в текстовом режиме. Работа консольного приложения напоминает работу программы MS-DOS, но это лишь внешнее впечатление. Консольное приложение обеспечивается специальными функциями Windows.

Вся разница между двумя типами приложений Windows состоит в том, с каким типом информации они работают. Основной тип приложений Windows – оконные.

1. **Какие преимущества и недостатки, по вашему мнению, имеет консольное приложение и приложение на основе оконного интерфейса?**

Одним из основных преимуществ работы с MFC является возможность многократного использования одного и того же кода. Так как библиотека содержит много элементов, общих для всех Windows-приложений, нет необходимости каждый раз писать их заново. Вместо этого их можно просто наследовать (говоря языком объектно-ориентированного программирования). Кроме того, интерфейс, обеспечиваемый библиотекой, практически независим от конкретных деталей, его реализующих. Поэтому программы, написанные на основе MFC, могут быть легко адаптированы к новым версиям Windows (в отличие от большинства программ, написанных обычными методами).

Еще одним существенным преимуществом MFC является упрощение взаимодействия с прикладным программным интерфейсом (API) Windows. Любое приложение взаимодействует с Windows через API, который содержит несколько сот функций. Внушительный размер API затрудняет попытки понять и изучить его целиком. Зачастую даже сложно проследить, как отдельные части API связанны друг с другом! Но поскольку библиотека MFC объединяет (путем инкапсуляции) функции API в логически организованное множество классов, интерфейсом становится значительно легче управлять.

1. **Для чего служит ClassWizard.?**

Он предназначен для работы с классами - для добавки в них методов и свойств, для добавления переменных к элементам диалогового окна и для других целей. ClassWizard автоматически подсоединится к вашему проекту, если вы запускаете какой-нибудь мастер для создания заготовки для вашего приложения.

1. **Для чего нужен AppWizard?**

AppWizard является специальным инструментом для того, чтобы генерировать программные тексты. Во время его работы на дисплее появляется очередность диалоговых окошек, в которых задают главные вопросы, которые относятся желаемых характеристик создаваемых программ. Получая ответ, AppWizard может генерировать текст базовых программ (скелет приложений), который содержит все ее непременные детали.

1. **Как осуществляется конструирование графического интерфейса?**

Система объектно-ориентированного программирования позволяет визуализировать процесс создания графического интерфейса разрабатываемого проекта. Графический интерфейс необходим для реализации интерактивного диалога пользователя с исполняемым проектом.

Форма. Основой для создания графического интерфейса разрабатываемого проекта является форма, представляющая собой окно, в котором размещаются элементы управления. Необходимо отметить, что графический интерфейс проекта может включать в себя несколько форм.

Форма — это объект, представляющий собой окно на экране, в котором размещаются элементы управления.

Элементы управления. Визуальное конструирование графического интерфейса проекта состоит в том, что на форму с помощью мыши помещаются и «рисуются» те или иные элементы управления.

• текстовые поля TextBox (Edit в языке Lazarus), надписи Label и списки ListBox и ComboBox для ввода и вывода данных;

• графические поля PictureBox (Image в языке Lazarus) для вывода графики;

• командные кнопки Button, переключатели RadioButton и флажки CheckBox для организации интерактивного диалога пользователя с проектом;

• главное меню MainMenu для создания меню формы;

• панель инструментов ToolBar для создания панели инструментов формы;

• коллекция изображений ImageList для хранения изображений;

• диалоги ColorDialog и FontDialog для выбора цвета и шрифта;

• диалоги OpenFileDialog и SaveFileDialog для выбора файла при открытии и сохранении.

На форму может быть помещено несколько экземпляров одного класса элементов управления. Например, несколько кнопок, каждая из которых обладает индивидуальными значениями свойств (надпись, размеры и др.).

1. **Как связать переменную с элементом графического интерфейса?**

В состав компилятора Microsoft Visual Studio встроены средства, позволяющие программисту облегчить разработку приложений. В первую очередь к ним относятся MFC AppWisard, ClassWizard и редактор ресурсов.

Благодаря MFC AppWizard среда разработчика позволяет быстро создавать шаблоны новых приложений. При этом программисту не приходится писать ни одной строчки кода. Достаточно ответить на ряд вопросов, касающихся того, какое приложение требуется создать, и исходные тексты шаблона приложения вместе с файлами ресурсов готовы. Эти тексты можно оттранслировать и получить готовый загрузочный модуль приложения.

Для создания ресурсов приложения предназначен редактор ресурсов. Он позволяет быстро создавать новые меню, диалоговые панели, добавлять кнопки к панели управления toolbar и т.д.

Средство ClassWizard позволяет подключить к созданным и отредактированным ресурсам управляющий ими код. Большую часть работы по описанию и определению функций, обрабатывающих сообщения от меню, органов управления диалоговых панелей и т.д., также берет на себя средство ClassWizard.