Требования и рекомендации

# Введение

Настоящий документ определяет правила выполнения индивидуального задания на тему: «Сравнение калькулятора калорий на двух ЯП (C++, C#)» по МДК.02.01 Технология разработки программного обеспечения.

Целью данного задания является развитие навыков программирования, сравнение особенностей двух языков программирования (C++ и C#) при реализации одной и той же задачи, а также анализ их преимуществ и недостатков в контексте разработки прикладных программ.

# Название приложения

Названия файлов (программы) одной из программ (написанное на ЯП C++) не соответствует требованиям.

# Файловая структура

Файловая структура проекта должна отражать логику, заложенную в приложение. Например, все формы содержатся в одной директории, пользовательские визуальные компоненты – в другой, классы сущностей – в третьей.

В случае второй программы, все условия по названиям соблюдены.

# Структура проекта

В обеих программах присутствует более чем одна функция, в которых прописаны свои задачи для корректной работы приложения. Как таковых классов и\или их файлов – нет.

# Логическая структура

Логика представления не перемешана с бизнес-логикой, а также не перемешана с логикой доступа к базе данных и т.д.

(не понял что это - В идеале это должны быть три независимых модуля.)

# Руководство по стилю

Визуальные компоненты соответствуют руководству по стилю, предоставленному в качестве ресурсов к заданию в соответствующем файле. Соблюдены требований всех компонентов в следующих областях:

* цветовая схема,
* использование шрифтов (используется шрифт программой по умолчанию, пользовательские шрифты не включены в проект)

Не соблюдены\включены следующие требования:

* размещение логотипа,
* установка иконки приложения.

# Макет и технические характеристики

Компоненты обоих систем имеют единый согласованный внешний вид, соответствующий руководству по стилю, а также следующим требованиям:

* разметка и дизайн;
* использование соответствующих элементов управления
* расположение и выравнивание элементов
* последовательный переход фокуса по элементам
* общая компоновка логична, понятна и проста в использовании;
* последовательный пользовательский интерфейс, позволяющий перемещаться между существующими окнами в приложении (как таковой кнопки «Назад» – нет, в моих программах она не нуждается в использовании, т.к. логическую задачу этой кнопки выполняет другая.)
* соответствующий заголовок на каждом окне приложения (на файлах заголовок присутствует по умолчанию, на окнах программы – все соблюдено).

Требования и руководства по стилю, которые не соблюдены\включены в проект:

* ограничение на минимальный размер окна;
* изменения размеров окна, где это необходимо;
* увеличение размеров окна должно увеличивать размер контентной части
* группировка элементов (в логические категории);

# Обратная связь с пользователем

Включены следующие требования:

* Происходит уведомление пользователя о совершаемых им ошибках или о запрещенных в рамках задания действиях.
* Окна сообщений соответствующих типов, отображаются с соответствующим заголовком и пиктограммой.
* Присутствует информирование об отсутствии результатов.
* Текст сообщения полезным и информативным

Не включённые требования:

* Текст сообщения не содержит полную информацию о совершенных ошибках пользователя и порядка действий для их исправления. Визуальные подсказки для пользователя при вводе данных не включены.
* Запрашивание подтверждения перед удалением, и о неотвратимых операциях.

# Обработка ошибок

Программы, написанные мной, не позволяют пользователю вводить некорректные значения в текстовые поля сущностей. При неверном вводе\обращении в программе, происходит оповещение пользователя о совершенной им ошибке.

Использование абсолютных и относительных путей к изображениям не является необходимым в моем ИП. Приложение корректно работает, в том числе и при перемещении папки с исполняемым файлом.

При возникновении непредвиденной ошибки приложение аварийно не завершает работу.

# Оформление кода

Идентификаторы переменных, методов и классов должны отражают суть и/или цель их использования, в том числе и наименования элементов управления.

Идентификаторы соответствуют соглашению об именовании и стилю CamelCase (для C# и C++, используемых в моем ИП)

Допустимо использование не более одной команды в строке.

# Комментарии

В моем проекте используются немного комментариев для пояснения очевидных и неочевидных фрагментов кода.

Не использовал в своей работе специальные комментарии с соответствующими тегами.

# Оценка

Каждая задача оценивается путем тестирования реализации требуемого функционала. Так как требования к реализуемой системе очень высоки, возможно, будут использоваться средства для автоматизированного тестирования приложения. В связи с этим, в ходе разработки, может возникнуть необходимость следовать определенным правилам именования и структурирования проекта.

# Предоставление результатов

Все практические результаты переданы заказчику путем загрузки файлов на репозиторий системы контроля версий git. Переданы следующие результаты:

* исходный код приложения (первой программы и второй),
* исполняемые файлы (вся обязательная документация, включающая в себя: ТЗ, Руководство пользователя и т.п.)
* прочие графические файлы (диаграммы)

Проект содержит описание в формате. Так же заполнена дополнительная информацию о проекте и способе запуска приложения в файле readme.md.