**Проектирование архитектуры программных систем**

**Отчёт о лабораторной работе №2**

**Выполнил Акимов Андрей, 21ПИ-1**

**Использованные приёмы программирования**

**Включение** — способ использования компонентов, при котором внешний компонент взаимодействует с внутренним компонентом, содержащим необходимую функциональность, путём получения указателей на его интерфейсы и использования их для взаимодействия. При необходимости внешний компонент может переопределить интерфейс, реализованный внутренним компонентом, передавая вызовы этого интерфейса внутреннему компоненту. Также внешний компонент может настраивать данный интерфейс, добавляя собственный код до и после вызовов внутреннего компонента.

**Агрегирование** — представляет собой специальный случай включения, при котором внешний компонент предоставляет клиенту указатель на интерфейс внутреннего компонента. В отличие от включения, внешний компонент не создаёт собственную реализацию интерфейса для передачи вызовов внутреннему компоненту. В случае агрегирования, делегирующий компонент направляет вызовы внешнему компоненту. Если компонент не является агрегированным, то его делегирующий компонент направляет вызовы неделегированному компоненту.

**Осуществление приёмов программирования**

Для осуществления включения требуется выполнить следующие шаги:

* Создать в новом компоненте поле для хранения ссылки на уже существующую реализацию нужного интерфейса и новую виртуальную таблицу для этого интерфейса.
* Запросить реализацию нужного интерфейса у существующего компонента при инициализации нового компонента с помощью метода QueryComponent интерфейсной шины и сохранить эту реализацию в соответствующее поле нового компонента.
* Реализовать методы виртуальной таблицы интерфейса, направляя их вызовы на ранее сохранённую реализацию интерфейса.

Для осуществления агрегирования требуется выполнить следующие шаги:

* Добавить в компонент поле типа IEcoUnknown \*, в котором будет храниться интерфейс агрегируемого компонента. Через этот интерфейс можно будет запросить другие интерфейсы, которые реализованы во внутреннем компоненте.
* Также необходимо добавить в компонент поле с реализацией неделегирующего собственного интерфейса IEcoUnknown
* Запросить у нужного компонента с помощью метода QueryComponent интерфейсной шины интерфейс IEcoUnknown и передать ссылку на себя при инициализации компонента. Это будет явное намерение на агрегирование. После этих действий необходимо полученный указатель сохранить в соответствующем поле компонента.
* В методе QueryInterface компонента направлять запросы на нужные интерфейсы, реализованные внутренним компонентом, с использованием сохранённого указателя.

Порядок включения и агрегирования компонентов калькулятора

Реализация интерфейса X представляется компонентами A и B, в то время как выполнение интерфейса Y возлагается на компоненты E и D. Однако возникает проблема с компонентом C из-за конфликта идентификаторов с компонентом A. Необходимо отметить, что агрегирование возможно только с компонентом B.

Для включения компонента, необходимого для работы с интерфейсом X, первым шагом является попытка агрегировать компонент B. Если это не удаётся, следующий шаг - обращение к компоненту A. Если это также не удаётся, работа с интерфейсом X становится невозможной, и все его методы будут возвращать 0.

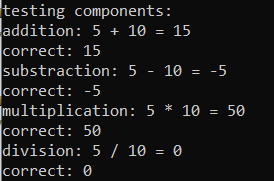
Для включения компонента, требуемого для работы с интерфейсом Y, первым шагом является подключение интерфейса Y из компонента D. Если это не удаётся, следующий шаг - обращение к компоненту E. Если это тоже не получится, работа с интерфейсом Y также становится невозможной, и все его методы будут возвращать 0.

Нужно отметить, что компонент E агрегирует компонент B, позволяя запросить интерфейс X через него. Однако, учитывая, что интерфейс X требует компонента B, такой сценарий не рассматривается.

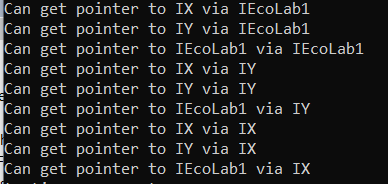
**Результат работы программы**

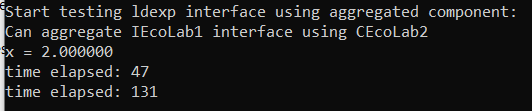
В начале из-за отсутствия DLL-файлов методы не вызывали ошибок, но возвращали нули, а после помещения файлов в папку BuildFiles результат стал верным.

Проверка функциональности компонентов калькулятора прошла успешно: получаемые ответы совпадают с обычными вычислениями.



Проверка доступа к одним интерфейсам через другие:





**Вывод:**

Были успешно реализованы механизмы включения и агрегирования для компонентов калькулятора. При таком подходе программа работает даже при частичном отсутствии DLL-файлов и не прекращается с ошибкой. При полном отсутствии DLL-файлов все методы будут возвращать нули.

Работа также показывает гибкость интерфейсов: указатель на любой из них может быть получен из любого другого. Кроме того, есть возможность ссылаться на самого себя, а также на родительский интерфейс, через который был получен текущий интерфейс. Это позволяет эффективно управлять взаимосвязями между компонентами и обеспечивать их взаимодействие в сложных системах.