

Отчет по реализации компонентной архитектуры с механизмами включения и агрегирования

1. Общая постановка задачи

Необходимо модифицировать компонента Eco.Lab1 для поддержки интерфейсов EcoCalculatorX и IEcoCalculatorY с использованием включения и агрегирования.

Компонент должен объединять функциональность нескольких независимых компонентов (выступать в роли фасада и агрегатора) и предоставлять единую точку доступа к операциям сортировки и арифметическим операциям.

2. Архитектура и реализация компонента

Структура компонента EcoLab1 была расширена для поддержки множественных интерфейсов:

```
typedef struct CEcoLab1 {  
    ...  
  
    /* Таблица функций интерфейса IEcoCalculatorX */  
    IEcoCalculatorXVTbl* m_pVTblIEcoCalculatorX;  
  
    /* Таблица функций интерфейса IEcoCalculatorY */  
    IEcoCalculatorYVTbl* m_pVTblIEcoCalculatorY;  
  
    /* Указатель на интерфейс IEcoCalculatorX включаемого компонента (цель  
    делегирования) */  
    IEcoCalculatorX* m_pIEcoCalculatorX;  
  
    /* Указатель на интерфейс IEcoCalculatorY включаемого компонента (цель  
    делегирования, если нет агрегации) */  
    IEcoCalculatorY* m_pIEcoCalculatorY;  
  
    /* Указатель на интерфейс внутреннего агрегируемого компонента*/  
    IEcoUnknown* m_pInnerUnknown;  
} CEcoLab1, *CEcoLab1Ptr;
```

Реализуемые интерфейсы:

1. IEcoLab1 — операции Gnome сортировки (прямая реализация)
2. IEcoCalculatorX — операции сложения и вычитания (агрегирование + включение)
3. IEcoCalculatorY — операции умножения и деления (включение)

Используемые внутренние компоненты:

- CID_EcoCalculatorA
- CID_EcoCalculatorB
- CID_EcoCalculatorC

- CID_EcoCalculatorD
- CID_EcoCalculatorE

3. Реализация механизмов

3.1 Механизм включения

Механизм включения основан на прямом создании внутренних компонентов и хранении их интерфейсов. Все вызовы методов делегируются включенным объектам.

Пример инициализации включенных компонентов:

```
result = pIBus->pVTbl->QueryComponent(
    pIBus,
    &CID_EcoCalculatorA,
    0,
    &IID_IEcoCalculatorX,
    (void**)&pCMe->m_pIEcoCalculatorX
);
if (result != 0 || pCMe->m_pIEcoCalculatorX == 0) {
    result = pIBus->pVTbl->QueryComponent(
        pIBus,
        &CID_EcoCalculatorB,
        0,
        &IID_IEcoCalculatorX,
        (void**)&pCMe->m_pIEcoCalculatorX
    );
}
if (result != 0 || pCMe->m_pIEcoCalculatorX == 0) {
    result = pIBus->pVTbl->QueryComponent(
        pIBus,
        &CID_EcoCalculatorC,
        0,
        &IID_IEcoCalculatorX,
        (void**)&pCMe->m_pIEcoCalculatorX
    );
}
```

Делегирование вызовов:

```
static int32_t ECOCALLMETHOD CEcoLab1_Addition(IEcoLab1Ptr_t me, int16_t x, int16_t
y) {
    CEcoLab1* pCMe = (CEcoLab1*)me;
    int32_t result = -1;
    /* Проверка указателей */
    if (me == 0) {
        return result;
    }
```

```

    }
    if (pCMe->m_pIEcoCalculatorX != NULL) {
        return pCMe->m_pIEcoCalculatorX->pVTbl->Addition(pCMe->m_pIEcoCalculatorX, x,
y);
    }
    return result;
}

```

Таким образом, компонент SEcoLab1 выступает фасадом, перенаправляя вызовы клиентского кода к внутренним компонентам.

3.2 Механизм агрегирования

Агрегирование применяется для IEcoCalculatorX, чтобы встроить внутренний компонент как часть внешнего.

Инициализация агрегированного компонента:

```

result = pIBus->pVTbl->QueryComponent(
    pIBus,
    &CID_EcoCalculatorB,
    pOuterUnknown,
    &IID_IEcoUnknown,
    (void**)&pCMe->m_pInnerUnknown
);

```

Особенность агрегирования заключается в том, что внутренний компонент получает указатель на IEcoUnknown внешнего объекта, что обеспечивает общее управление временем жизни и единый счетчик ссылок.

Логика QueryInterface для агрегирования:

```

if (IsEqualUGUID(riid, &IID_IEcoCalculatorX)) {
    if (pCMe->m_pInnerUnknown != 0) {
        return pCMe->m_pInnerUnknown->pVTbl->QueryInterface(
            pCMe->m_pInnerUnknown,
            riid,
            ppv
        );
    }
    *ppv = &pCMe->m_pVTblIEcoCalculatorX;
    pCMe->m_pVTblIEcoLab1->AddRef((IEcoLab1*)pCMe);
}

```

Если агрегирование не удалось, реализуется резервный механизм включения, что повышает устойчивость компонента.

4. Тестирование

Для проверки корректности были проведены проверки на:

- Вызов арифметических операций через интерфейс IEcoLab1
- Получение и проверка интерфейсов калькуляторов:
 - Запрос интерфейса IEcoCalculatorX через IEcoLab1
 - Запрос интерфейса IEcoCalculatorY через IEcoLab1
 - Проверка работоспособности арифметических функций через полученные интерфейсы
- Тестирование агрегации IEcoCalculatorX
- Тестирование передачи интерфейсов:
 - Проверка получения IEcoLab1 из IEcoCalculatorX
 - Проверка получения IEcoLab1 из IEcoCalculatorY
 - Выполнение операций GnomeSort через полученные интерфейсы

=== Lab2 ===

Addition: $10 + 101 = 111$

Subtraction: $19 - 4 = 15$

Multiplication: $10 * 15 = 150$

Division: $300 / 100 = 3$

IEcoCalculatorX Addition: $10 + 19 = 29$

IEcoCalculatorY Multiplication: $15 * 3 = 45$

Aggragation IEcoCalculatorX: PASS

CalculatorX -> IEcoLab1(Lab2): PASS

CalculatorY -> IEcoLab1(Lab2): PASS

5. Заключение

В ходе работы был модифицирован компонент SEcoLab1, в котором реализованы механизмы включения и агрегирования. Включение обеспечило делегирование вызовов к внутренним компонентам через фасад, а агрегирование позволило объединить интерфейсы и управление временем жизни в одном объекте.