**Проектирование архитектуры программных систем**

**Отчёт по лабораторной работа № 2**

**Тема: Работа с приемами включения\агрегирования в компоненте Bubble Sort**

**Выполнила:**

**Сологуб Алина Денисовна**

**Группа 21 ПИ-1**

**Ход выполнения работы**

1. В структуру CBubleSort\_B4ED74BE в файле CBubleSort.h были добавлены таблицы функций интерфейсов X и Y, а также указатели на интерфейсы включаемых компонентов и указатель на IEcoUnknown внутреннего компонента.

/\* Таблица функций интерфейса IBubleSort \*/

IBubleSortVTbl\* m\_pVTblIBubleSort;

/\* Таблица функций интерфейса IEcoCalculatorX \*/

IEcoCalculatorXVTbl\* m\_pVTblIX;

/\* Таблица функций интерфейса IEcoCalculatorY \*/

IEcoCalculatorYVTbl\* m\_pVTblIY;

…

/\* Указатель на интерфейс IEcoCalculatorX включаемого компонента \*/

IEcoCalculatorX\* m\_pIX;

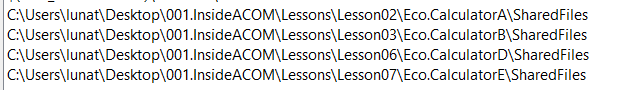
/\* Указатель на интерфейс IEcoCalculatorY включаемого компонента \*/

IEcoCalculatorY\* m\_pIY;

/\* Указатель на IEcoUnknown внутреннего компонента \*/

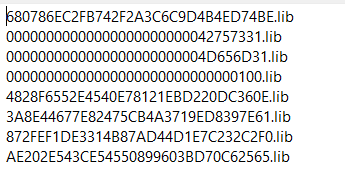
IEcoUnknown\* m\_pInnerUnknown;

1. Были подключены Shared files в настройках проекта.



BubleSort->Properties->C/C++-> General->Additional Include Directories

1. В Linker были добавлены макросы для либ.



BubleSortUnitTest->Properties->Linker->Input-> Additional Dependencies

1. В файл CbubleSort.c были добавлены функции QueryInterface, AddRef и Release для компонентов X и Y. Также были созданы функции Addition, Subtraction, Multiplication, Division.
2. Пример функции QueryInterface

int16\_t ECOCALLMETHOD CBubleSort\_B4ED74BE\_QueryInterface(/\* in \*/ struct IBubleSort\* me, /\* in \*/ const UGUID\* riid, /\* out \*/ void\*\* ppv) {

CBubleSort\_B4ED74BE\* pCMe = (CBubleSort\_B4ED74BE\*)me;

int16\_t result = -1;

/\* Проверка указателей \*/

if (me == 0 || ppv == 0) {

return result;

}

/\* Проверка и получение запрошенного интерфейса \*/

if ( IsEqualUGUID(riid, &IID\_IBubleSort) ) {

\*ppv = &pCMe->m\_pVTblIBubleSort;

pCMe->m\_pVTblIBubleSort->AddRef((IBubleSort\*)pCMe);

}

else if ( IsEqualUGUID(riid, &IID\_IEcoCalculatorX) ) {

if (pCMe->m\_pInnerUnknown != 0) {

/\* Запрашиваем интерфейс IEcoCalculatorX внутреннего компонента \*/

result = pCMe->m\_pInnerUnknown->pVTbl->QueryInterface(pCMe->m\_pInnerUnknown, riid, ppv);

return result;

}

else {

\*ppv = &pCMe->m\_pVTblIX;

pCMe->m\_pVTblIBubleSort->AddRef((IBubleSort\*)pCMe);

}

}

else if ( IsEqualUGUID(riid, &IID\_IEcoCalculatorY) ) {

\*ppv = &pCMe->m\_pVTblIY;

pCMe->m\_pVTblIBubleSort->AddRef((IBubleSort\*)pCMe);

}

else if ( IsEqualUGUID(riid, &IID\_IEcoUnknown) ) {

\*ppv = &pCMe->m\_pVTblIBubleSort;

pCMe->m\_pVTblIBubleSort->AddRef((IBubleSort\*)pCMe);

}

else {

\*ppv = 0;

return -1;

}

return 0;

}

1. Добавлены виртуальные таблицы:

IBubleSortVTbl g\_x87CE395400A14271A107EB2788B31103VTbl\_B4ED74BE = {

CBubleSort\_B4ED74BE\_QueryInterface,

CBubleSort\_B4ED74BE\_AddRef,

CBubleSort\_B4ED74BE\_Release,

CBubleSort\_B4ED74BE\_MyFunction

};

IEcoCalculatorXVTbl g\_x9322111622484742AE0682819447843DVTblD\_B4ED74BE = {

CBubleSort\_B4ED74BE\_IEcoCalculatorX\_QueryInterface,

CBubleSort\_B4ED74BE\_IEcoCalculatorX\_AddRef,

CBubleSort\_B4ED74BE\_IEcoCalculatorX\_Release,

CBubleSort\_B4ED74BE\_Addition,

CBubleSort\_B4ED74BE\_Subtraction

};

IEcoCalculatorYVTbl g\_xBD6414C29096423EA90C04D77AFD1CADVTblD\_B4ED74BE = {

CBubleSort\_B4ED74BE\_IEcoCalculatorY\_QueryInterface,

CBubleSort\_B4ED74BE\_IEcoCalculatorY\_AddRef,

CBubleSort\_B4ED74BE\_IEcoCalculatorY\_Release,

CBubleSort\_B4ED74BE\_Multiplication,

CBubleSort\_B4ED74BE\_Division

};

1. В функцию initCBubleSort\_B4ED74BE были добавлены строки для получения интерфейсов по работе с арифметическими операциями. Причем компонент B был добавлен с поддержкой агрегирования.

/\* Создание внутреннего компонента "B" c поддержкой агрегирования \*/

/\* ВАЖНО: При агрегировании мы передаем IID IEcoUnknown \*/

result = pIBus->pVTbl->QueryComponent(pIBus, &CID\_EcoCalculatorB, pOuterUnknown, &IID\_IEcoUnknown, (void\*\*) &pCMe->m\_pInnerUnknown);

/\* Проверка \*/

if (result != 0 || pCMe->m\_pIX == 0) {

/\* Получение интерфейса по работе со сложением и вычитанием у компонента "A" \*/

result = pIBus->pVTbl->QueryComponent(pIBus, &CID\_EcoCalculatorA, 0, &IID\_IEcoCalculatorX, (void\*\*) &pCMe->m\_pIX);

}

/\* Получение интерфейса по работе с умножением и делением у компонента "E" c поддержкой агрегирования \*/

result = pIBus->pVTbl->QueryComponent(pIBus, &CID\_EcoCalculatorE, 0, &IID\_IEcoCalculatorY, (void\*\*) &pCMe->m\_pIY);

/\* Проверка \*/

if (result != 0 || pCMe->m\_pIX == 0) {

/\* Получение интерфейса по работе с умножением и делением у компонента "D" \*/

result = pIBus->pVTbl->QueryComponent(pIBus, &CID\_EcoCalculatorD, 0, &IID\_IEcoCalculatorY, (void\*\*) &pCMe->m\_pIY);

}

1. В функции create было прописано создание таблиц функций интерфейсов IbubleSort, IecoCalculatorX, IecoCalculatorY.

/\* Создание таблицы функций интерфейса IBubleSort \*/

pCMe->m\_pVTblIBubleSort = &g\_x87CE395400A14271A107EB2788B31103VTbl\_B4ED74BE;

pCMe->m\_pVTblIX = &g\_x9322111622484742AE0682819447843DVTblD\_B4ED74BE;

pCMe->m\_pVTblIY = &g\_xBD6414C29096423EA90C04D77AFD1CADVTblD\_B4ED74BE;

1. В файл BubleSort.c были добавлены переменные:

IEcoCalculatorX\* pIX = 0;

IEcoCalculatorY\* pIY = 0;

int32\_t answer1 = 0;

1. Регистрация статических компонентов A, B, D, E

/\* Регистрация статического компонента A \*/

result = pIBus->pVTbl->RegisterComponent(pIBus, &CID\_EcoCalculatorA, (IEcoUnknown\*)GetIEcoComponentFactoryPtr\_4828F6552E4540E78121EBD220DC360E);

if (result != 0 ) {

/\* Освобождение в случае ошибки \*/

goto Release;

}

/\* Регистрация статического компонента B \*/

result = pIBus->pVTbl->RegisterComponent(pIBus, &CID\_EcoCalculatorB, (IEcoUnknown\*)GetIEcoComponentFactoryPtr\_AE202E543CE54550899603BD70C62565);

if (result != 0 ) {

/\* Освобождение в случае ошибки \*/

goto Release;

}

/\* Регистрация статического компонента D \*/

result = pIBus->pVTbl->RegisterComponent(pIBus, &CID\_EcoCalculatorD, (IEcoUnknown\*)GetIEcoComponentFactoryPtr\_3A8E44677E82475CB4A3719ED8397E61);

if (result != 0 ) {

/\* Освобождение в случае ошибки \*/

goto Release;

}

/\* Регистрация статического компонента E \*/

result = pIBus->pVTbl->RegisterComponent(pIBus, &CID\_EcoCalculatorE, (IEcoUnknown\*)GetIEcoComponentFactoryPtr\_872FEF1DE3314B87AD44D1E7C232C2F0);

if (result != 0 ) {

/\* Освобождение в случае ошибки \*/

goto Release;

}

1. Получение тестируемых интерфейсов

result = pIBus->pVTbl->QueryComponent(pIBus, &CID\_EcoCalculatorA, 0, &IID\_IEcoCalculatorX, (void\*\*) &pIX);

if (result != 0 || pIX == 0) {

/\* Освобождение интерфейсов в случае ошибки \*/

goto Release;

}

result = pIBus->pVTbl->QueryComponent(pIBus, &CID\_EcoCalculatorB, 0, &IID\_IEcoCalculatorX, (void\*\*) &pIX);

if (result != 0 || pIX == 0) {

/\* Освобождение интерфейсов в случае ошибки \*/

goto Release;

}

result = pIBus->pVTbl->QueryComponent(pIBus, &CID\_EcoCalculatorD, 0, &IID\_IEcoCalculatorY, (void\*\*) &pIY);

if (result != 0 || pIX == 0) {

/\* Освобождение интерфейсов в случае ошибки \*/

goto Release;

}

result = pIBus->pVTbl->QueryComponent(pIBus, &CID\_EcoCalculatorE, 0, &IID\_IEcoCalculatorY, (void\*\*) &pIY);

if (result != 0 || pIX == 0) {

/\* Освобождение интерфейсов в случае ошибки \*/

goto Release;

}

1. Тесты для проверки работы алгоритма включения компонентов X и Y и доступности функций сложения, вычитания, умножения и деления.

result = pIBubleSort->pVTbl->QueryInterface(pIBubleSort, &IID\_IEcoCalculatorX, (void\*\*)&pIX);

answer1 = pIX->pVTbl->Addition(pIX, 10, 2);

printf("Variables: 10 2\n");

printf("The expected result of the addition: 12\n");

printf("The real result of the addition: %d\n", answer1);

result = pIBubleSort->pVTbl->QueryInterface(pIBubleSort, &IID\_IEcoCalculatorX, (void\*\*)&pIX);

answer1 = pIX->pVTbl->Subtraction(pIX, 10, 2);

printf("The expected result of the subtraction: 8\n");

printf("The real result of the subtraction: %d\n", answer1);

result = pIBubleSort->pVTbl->QueryInterface(pIBubleSort, &IID\_IEcoCalculatorY, (void\*\*)&pIY);

answer1 = pIY->pVTbl->Multiplication(pIY, 10, 2);

printf("The expected result of the multiplication: 20\n");

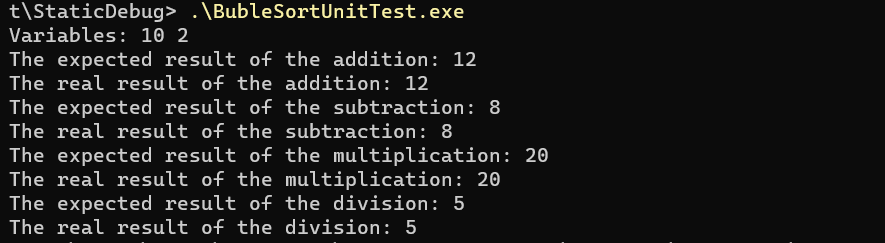
printf("The real result of the multiplication: %d\n", answer1);

result = pIBubleSort->pVTbl->QueryInterface(pIBubleSort, &IID\_IEcoCalculatorY, (void\*\*)&pIY);

answer1 = pIY->pVTbl->Division(pIY, 10, 2);

printf("The expected result of the division: 5\n");

printf("The real result of the division: %d\n", answer1);



**Итог**

Реализация процесса включения компонентов X и Y в компонент BubleSort была успешно завершена. Теперь компонент BubleSort имеет возможность вызывать функции сложения, вычитания, умножения и деления, которые реализованы в других компонентах.