

ПРИЛОЖЕНИЕ В. КРАТКИЕ СВЕДЕНИЯ О LIRA API

Общие сведения

Если при работе с ПК Лира 10.4 Вам потребовалось выполнить экспорт исходных данных или результатов расчета в нестандартный формат или выполнить дополнительные расчеты, опираясь на данные, уже существующие в рамках расчетной схемы, и при этом Вы обладаете навыками программирования это приложение поможет Вам познакомиться с возможностями LiraAPI.

При разработке расширения ПК Лира 10.4 Вам понадобится выполнить два действия:

1. Разработать расширение, опираясь на возможности LiraAPI;
2. Выполнить регистрацию разработанного расширения в среде ПК Лира 10.4.

После успешного выполнения этих действий в главном меню ПК Лира 10.4 появится пункт меню Расширения, в котором Вы найдете подпункт меню, вызывающий расширение, реализованное Вами (рисунок В.1).

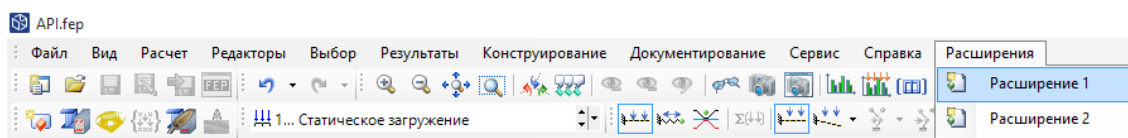



Рисунок В.1

 Вы можете реализовать и зарегистрировать множество расширений, для каждого из них будет генерироваться отдельный подпункт меню.

Регистрация расширения

При первом запуске Лира 10.4 создает файл с глобальными настройками:

```
[ApplicationData]+"\\Lira Soft\\Lira10.4\\VariableEnvironment_x86.xml"  
[ApplicationData]+"\\Lira Soft\\Lira10.4\\VariableEnvironment_x64.xml"
```

Среди прочих настроек в этом файле есть параметр `AddinsPath`, содержащий путь к папке, в которой должны находиться xml файлы регистрации расширений, по умолчанию это `[ApplicationData]+"\\Lira Soft\\Lira10.4\\Addins"`.

Файлы регистрации расширения должны иметь следующую структуру:

```
<?xml version="1.0" encoding="utf-8"?>  
<LiraAddIns>  
  <AddIn Type="PROLONGATION">  
    <AssemblyPath>путь к dll</AssemblyPath>  
    <CommandName>Имя команды</CommandName>  
    <CommandDescription>Описание команды</CommandDescription>  
    <ImagePath>Путь к изображению</ImagePath>  
    <Vendor>Название организации</Vendor>  
    <VendorDescription>описание</VendorDescription>  
  </AddIn>  
</LiraAddIns>
```

AddIn содержит атрибут Type, который может принимать одно из трех значений: PRIME, PROLONGATION или ALL. Этот атрибут указывает, в каком режиме будет доступна команда этого расширения: в режиме редактирования исходных данных, в режиме анализа результатов расчета или в обоих режимах. В текущей версии доступно только значение PROLONGATION.

AssemblyPath - абсолютный путь к файлу расширения *.dll;

ImagePath - абсолютный путь к файлу, содержащему изображение для иконки в меню (стандартный размер Width=24, Height=20);

CommandName - имя команды в меню Лира 10.4;

Vendor и VendorDescription – информация о разработчике расширения.

При запуске Лира 10.4 добавляет пункт меню для каждого найденного и удачно прочитанного xml файла регистрации расширений.

Разработка расширения

Рекомендуемой средой разработки расширений Лира 10.4 является **Microsoft Visual Studio 2010**. К проекту расширения нужно добавить как минимум две References ссылки на библиотеки из установочного дистрибутива Лира 10.4 (LiraAPI.dll и FEModel.dll). В проекте расширения должен быть реализован один class, наследованный от интерфейса LiraAPI.ILiraAPI, который описан в динамической библиотеке LiraAPI.dll:

```
public ref class CSampleLiraAPI : public LiraAPI::ILiraAPI
{
    public: virtual LiraAPI::ReturnCodes ExecuteProgram_Result (LiraAPI::IResultLiraAPI ^pResultLiraAPI,
        int NodesNumber, int ElementsNumber , List<List<FEModel::Results_Key^>>^ pAllCases,
        FEModel::Results_Key ^pCurentCase);
}
```

NodesNumber и ElementsNumber - количество узлов и элементов в расчетной схеме;

pCurentCase- информация о текущем загрузении;

pAllCases - информация о всех доступных в задаче загрузениях;

pResultLiraAPI – объект, позволяющий получить таблицы результатов расчета.

Объект, описывающий загрузение имеет следующий вид:

```
ref class FEModel::Results_Key
{
    //индекс загрузения, истории нагужений, номер РСН,...
    long m_IndexLoadingCase;
    // индекс сопутствующего загрузения, номер варианта РСН,...
    short m_SubIndexLoadingCase;
    // индекс формы, шага нелинейного загрузения, момента времени,...
    long m_IndexForm;
};
```

Объект List<List<FEModel::Results_Key^>>^ pAllCases содержит до 5 массивов вида List<FEModel::Results_Key^>, каждый из которых описывает перечень доступных загрузений для различных типов таблиц.

pAllCases[0] – загрузения и составляющие;

pAllCases[1] – формы собственных колебаний;

pAllCases[2] – формы потери устойчивости от загрузений;

pAllCases[3] – расчетные сочетания нагрузок (РСН);

pAllCases[4] – формы потери устойчивости от РСН.

Интерфейс LiraAPI::IResultLiraAPI в Лире 10.4 имеет вид:

```
public interface class IResultLiraAPI
{
virtual int getLiraApiVersion();
virtual DataTable ^getTableResult(FEModel::e_Results_TableType rtt,
    System::Collections::Generic::List<int> ^pObjArr,
    System::Collections::Generic::List<FEModel::Results_Key ^> ^pKeyArr,
    array<e_Results_ColumnType> ^%pTypeColumns,
    array<System::String ^> ^%pNameColumns);
};
```

Функция getLiraApiVersion() возвращает номер версии текущего объекта LiraApi.

Функция get_TableResult(...) имеет три входных параметра FEModel::e_Results_TableType rtt, System::Collections::Generic::List<int> ^pObjArr, System::Collections::Generic::List<FEModel::Results_Key ^> ^pKeyArr

и три выходных array<e_Results_ColumnType> ^%pTypeColumns, array< System::String ^> ^%pNameColumns и DataTable с результатами запроса.

System::Collections::Generic::List<int> ^pObjArr – массив индексов узлов или элементов (начиная от 0);

System::Collections::Generic::List<FEModel::Results_Key ^> ^pKeyArr – массив объектов FEModel::Results_Key, описывающих загрузки;

FEModel::e_Results_TableType rtt – тип таблицы, подробную информацию о доступных типах таблиц Вы можете найти в файле LiraAPIHelp.pdf, который копируется на жесткий диск в папку [INSTALLDIR] + "\\LiraAPI" в момент установки Лире 10.4.

Пример расширения

В процессе установки Лире 10.4 на жесткий диск в папку [INSTALLDIR] + "\\LiraAPI" копируется архивный файл с проектом Visual Studio, в котором продемонстрированы примеры с объектами LiraAPI.