

Инструкция по использованию и созданию внешних DLL для уроков комплекса КОТ

1. Использование DLL для создания уроков в комплексе КОТ

Использование внешних DLL в комплексе КОТ возможно начиная с версии 1.0.0.86 редактора тестов и версии 1.0.0.120 программы «Ученик».

Использование внешних DLL для создания заданий в уроках делает комплекс универсальным. Внешние DLL могут содержать абсолютно произвольные задания и формы. Например, может быть создано задание по математике с типовой задачей, исходные которой будут изменяться случайным образом или в соответствии с требуемым алгоритмом для каждого нового вызова этой задачи. Подобный подход затруднит «списывание» (то есть подглядывание) правильных ответов при работе в группе, а также сделает более эффективной тренинг ученика при индивидуальной работе. Помимо этого внешний вид окна задания может быть любым, какой только потребуется, даже если такой не предусмотрен в стандартных окнах заданий комплекса. На рисунке пример окна задания для тренинга при освоении формул приведения в тригонометрии 10 класса средней школы.

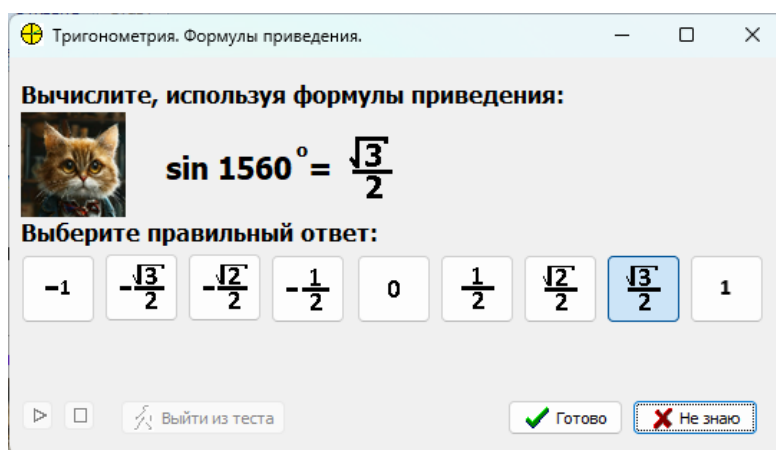


Рисунок 1. Пример окна для задания, вызванного из внешней DLL

Если DLL соответствует требованиям комплекса, то для использования её при создании урока не требуется никакой специальной подготовки. Включение DLL в урок выполняется в приложении «Редактор уроков» («Редактор тестов»), версии не ниже 1.0.0.86. Процесс подключения DLL в качестве задания в урок описано в п.2 «Подключение задания из DLL в урок (тест) комплекса КОТ».

Внимание! КОТ взаимодействует с DLL под Windows, которые соответствуют требованиям, изложенным в п.3 «Инструкция для программистов по созданию DLL для комплекса КОТ». DLL могут быть созданы в любом языке программирования под Windows в любой среде программирования, например, в Delphi, C++ Builder, C++ или C# Microsoft Visual Studio и т.д.

2. Подключение задания из DLL в урок (тест) комплекса КОТ

Подключение задания, созданного в виде DLL, в урок (тест) комплекса КОТ выполняется в приложении «Редактор уроков» («Редактор тестов») версии не ниже 1.0.0.86.

В приложении создайте новый урок или откройте существующий, в который нужно добавить задание из DLL. Создайте новое или выберите в таблице существующее задание. В выпадающем списке «Тип задания или сообщения:» выберите «Форма из DLL» (см. рис. 2).

Тип задания или сообщения:	Форма из DLL
Текст для передачи в DLL	Вопрос (задание)
Определите значение	Информационное сообщение
	Форма из DLL

Рисунок 2. Выбор типа задания «Форма из DLL»

После этого появятся редактор имени файла DLL и кнопка вызова диалога выбора имени файла DLL (см. рис. 3).

Тип задания или сообщения:	Форма из DLL			
Текст для передачи в DLL	Определите значение			
Баллов за задание	5	Секунд на просмотр	0	Посмотреть форму из DLL
Имя файла DLL				
Картинка	Звук	Видео		

Редактор имени файла DLL (указывает на поле ввода)

Кнопка вызова диалога выбора имени файла DLL (указывает на кнопку с тремя точками)

Рисунок 3. Редактор имени файла DLL

Если нажать на кнопку вызова диалога выбора имени файла DLL, появится стандартный диалог Windows выбора файла (см. рис. 4).

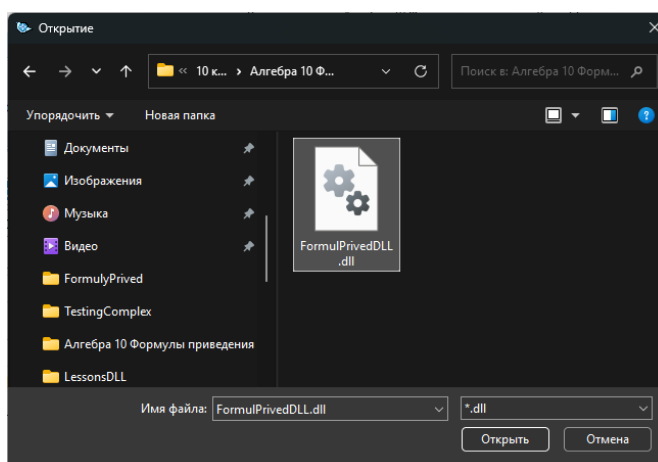


Рисунок 4. Диалог выбора имени файла DLL

После выбора имени файла DLL имя файла появится в редакторе имени файла (см. рис. 5), а если выбранная DLL находится не в папке урока, то она будет скопирована в эту папку.

Тип задания или сообщения: Форма из DLL

Текст для передачи в DLL

Определите значение

Баллов за задание 5 Секунд на просмотр 0 Посмотреть форму из DLL

Имя файла DLL FormulPrivedDLL.dll

Рисунок 5. Имя файла DLL в редакторе

Если требуется, можно задать количество баллов («Баллов за задание») за правильное выполнение задания и ограничить время выполнения задания («Секунд на просмотр»).

После того, как выбрана DLL для задания, можно посмотреть окно и задание из этой DLL. Для этого следует нажать кнопку «Посмотреть форму из DLL» (см. рис. 5). Если DLL создана согласно требованиям следующего пункта, то появится окно задания. Например, такое, как на рисунке 6.

Тригонометрия. Формулы приведения.

Вычислите, используя формулы приведения:

$\sin 1200^\circ =$

Выберите правильный ответ:

Выйти из теста Готово Не знаю

Рисунок 6. Пример окна задания, вызванного из DLL

3. Инструкция для программистов по созданию DLL для комплекса КОТ

Данный раздел предназначен для программистов и/или программирующих специалистов.

DLL можно создавать в любой среде программирования, на любом языке, позволяющем создавать стандартные DLL под Windows.

В данном документе приведены примеры заголовков функций и структур, созданных в Delphi на языке Object Pascal. Примеры для других языков запланированы и будут публиковаться по мере создания.

На нашем сайте доступны для скачивания архивы с шаблонами DLL и маленьким отладочно-проверочным приложениям. Для IDE Delphi:

<https://catcott.by/Downloads/TemplateDelphi.zip>

Для C++ Builder (версия не ниже XE5):

<https://catcott.by/Downloads/TemplateCPPBuilder.zip>

3.1. Обязательные требования к DLL, содержащей задание для комплекса КОТ

DLL обязательно должна соответствовать следующим требованиям:

1. DLL должна содержать в своём составе форму (окно) для выполнения задания. Форма должна открываться в модальном режиме и возвращать стандартные значения модальных диалогов Windows.
2. Перед открытием формы в DLL комплекс в библиотеку время, отведённое на урок в секундах, тип урока, и время начала работы с уроком. Передача осуществляется вызовом из DLL функции следующего формата (для языка Object Pascal):

```
function PrepareStartT(sr: TMyStartRecord):integer; stdcall;
```

где тип TMyStartRecord – запись (структура) следующего формата:

```
type TMyStartRecord = record
```

```
  iCurSecForTO: integer; // - количество секунд, отведённых на весь урок
```

```
  myLessonType: integer; // - тип урока
```

```
    //полная дата и время начала работы с уроком
```

```
  idtSLYear: integer; // - год
```

```
  idtSLMonth: integer; // - месяц
```

```
  idtSLDay: integer; // - день
```

```
  idtSLHour: integer; // - час
```

```
  idtSLMin: integer; // - минута
```

```
  idtSLSec: integer; // - секунда
```

```
  idtSLmSec: integer; // - миллисекунда
```

```
end;
```

В случае, если iCurSecForTO = 0, предполагается, что нет ограничений по времени на весь урок, в этом случае дата и время начала урока не имеют значения. В случае, если iCurSecForTO > 0, в форме (окне, диалоге) выполнения задания следует осуществлять сравнение количества секунд, прошедших с начала урока, и по истечении iCurSecForTO следует закрыть окно (диалог) задания с результатом, соответствующем ОК (idOK = 1). Тип урока myLessonType используется для определения активности или неактивности кнопки «Выйти из теста», если окно (диалог) планируется создать в стиле стандартных окон заданий комплекса КОТ и оно будет содержать эту кнопку. Более подробно обработка параметров этой функции рассмотрена в п.3.1.

3. Открытие окна (диалога) и начало работы задания выполняется функцией:

```
function PrepareAndShowModal(rpch: TRecTestPChar): integer; stdcall;
```

где тип TRecTestPChar – запись (структура) следующего формата:

```
Type TRecTestPChar = record //Запись описания теста для передачи в DLL
```

```
  ID : Integer; //Идентификатор теста
```

```
  MyTestType : Integer; //Тип теста
```

```
  Text : PChar; //Текст теста
```

```
  TestImageFN : PChar; //Имя файла изображения
```

```
  TestSoundFN : PChar; //Имя файла звука
```

```
  TestMovieFN : PChar; //Имя видеофайла
```

```
  AnswersType : Integer; //Тип ответов
```

```
  WeightTest : Integer; //Баллов за правильный ответ
```

```
  SecondsTO : Integer; //Время на ответ в секундах
```

```
  MyScript : PChar; //Имя файла DLL ()
```

```
iidw1,iidw2:integer;  
end;
```

Именно эта функция вызывает окно (диалог) задания из DLL.

Функция возвращает результат диалога при его закрытии кнопками «Готово», «Не знаю» или «Выйти из теста», а именно одно из следующих значений: idOK = 1, idCancel = 2 или idIgnore = 5. Остальные значения не обрабатываются приложением «Ученик» на сегодняшний день. Результат idOK = 1 соответствует кнопке «Готово», idCancel = 2 – кнопке «Не знаю», а idIgnore = 5 – кнопке «Выйти из теста». Более подробно обработка параметров этой функции рассмотрена в п.3.1.

4. Для освобождения окна в DLL следует предусмотреть функцию:

```
function DestroyMyDLGForm: Integer; stdcall;
```

Функция освобождает память объекта окна и всех его данных. Возвращает 0.

5. Для получения данных о результатах выполнения задания из DLL следует предусмотреть следующие функции:

1) function GetMyEntAnswer: PChar; stdcall; - возвращает ответ, введенный учеников в текстовом виде;

2) function GetMyCorrectAnswer: PChar; stdcall; - возвращает правильный ответ, в текстовом виде;

3) function GetMyQuestion: PChar; stdcall; - возвращает текст вопроса (задания), в текстовом виде;

4) function GetMyResult: integer; stdcall; - возвращает результат выполнения задания: 0 – неправильно, 1 – правильно.

6. В случае, если в задании используются символы национального алфавита, например - кириллицы, следует учесть и проверить корректность передачи символов в форму DLL и из формы в комплекс.

3.2. Необязательные рекомендации для DLL, содержащей задание для комплекса КОТ

Все требования предыдущего пункта 3.1 обязательны, поскольку комплекс КОТ использует именно перечисленные функции и форматы для работы с внешними DLL. Рекомендации, перечисленные ниже не обязательны, но желательны для сохранения однообразного подхода к организации уроков, тестов и заданий.

1. Все стандартные окна заданий комплекса КОТ поддерживают возможность ограничения времени на выполнение как задания, так и урока в целом. В случае, если параметр iCurSecForTO > 0 (одно из полей параметра функции PrepareStartT(sr: TMyStartRecord):integer; см.п.3.1.2), в окне, вызванном из DLL, следует производить контроль времени. Если разница в секундах между текущим временем и временем начала урока (также поля параметра функции PrepareStartT(sr: TMyStartRecord):integer; см.п.3.1.2) превысила iCurSecForTO, следует автоматически закрыть окно задания, при этом вернуть результат диалога idOK = 1. Если iCurSecForTO = 0, то нет ограничений на время выполнения урока в целом. Ограничение на время выполнения задания передается параметром SecondsTO (одно из полей параметра функции PrepareAndShowModal(rpch: TRecTestPChar): integer; см.п.3.1.3). В случае, если SecondsTO > 0, следует сравнивать текущее время со временем начала текущего задания (открытия формы из DLL). Когда разница в секундах между текущим временем и временем начала задания превысит

- SecondsTO, следует автоматически закрыть окно задания, при этом вернуть результат диалога idOK = 1. Если SecondsTO = 0, то нет ограничений на время выполнения задания.
2. В окне выполнения задания рекомендуется предусмотреть следующие кнопки: «Готово», «Не знаю» и «Выйти из теста» (см. рис. 7). Результат idOK = 1 соответствует кнопке «Готово», idCancel = 2 – кнопке «Не знаю», а idIgnore = 5 – кнопке «Выйти из теста». Кнопка «Выйти из теста» активна только в случае типов урока (теста) «Тренировочный тест простой» и «Тренировочный тест до победы», при этом параметр MyTestType имеет значение 0 или 1, соответственно. MyTestType - одно из полей параметра функции PrepareAndShowModal(rpch: TRecTestPChar): integer; см.п.3.1.3.

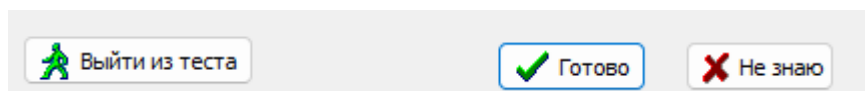


Рисунок 7. Стандартные кнопки окон заданий комплекса

3.3. Объявление соответствующих структур и функций для DLL в C++ Builder

1. Объявление структур и глобальных переменных:

```
struct TMyStartRecord{
    int iCurSecForTO;
    int myLessonType;
    int idtSLYear;
    int idtSLMonth;
    int idtSLDay;
    int idtSLHour;
    int idtSLMin;
    int idtSLSec;
    int idtSLmSec;
};

struct TRecTestPChar{
    int ID; //Идентификатор теста
    int MyTestType; //Тип теста
    PChar Text; //Текст теста
    PChar TestImageFN; //Имя файла изображения
    PChar TestSoundFN; //Имя файла звука
    PChar TestMovieFN; //Имя видеофайла
    int AnswersType; //Тип ответов
    int WeightTest; //Баллов за правильный ответ
    int SecondsTO; //Время на ответ в секундах
    PChar MyScript; //Имя файла DLL ()
    int iidw1;
    int iidw2;
};

struct TRecTest{ //Запись описания теста
    int ID; //Идентификатор теста
    int MyTestType; //Тип теста
    String Text; //Текст теста
```

```

TFileName TestImageFN; //Имя файла изображения
TFileName TestSoundFN; //Имя файла звука
TFileName TestMovieFN; //Имя видеофайла
int AnswersType; //Тип ответов
int WeightTest; //Баллов за правильный ответ
int SecondsTO; //Время на ответ в секундах
String MyScript; //Скрипт
int iidw1;
int iidw2;
};

```

```

extern PACKAGE TFormFromDLL *FormFromDLL;//это сама форма, её имя может быть иным
extern TMyStartRecord MyStartRecord;
extern String MyEntAnswer;
extern String MyCorrectAnswer;
extern String MyQuestion;
extern int MyResult;

```

2. Объявление экспортируемых функций:

```

extern "C" __declspec(dllexport) int __stdcall PrepareAndShowModal(TRecTestPChar rpch);
extern "C" __declspec(dllexport) int __stdcall PrepareStartT(TMyStartRecord sr);
extern "C" __declspec(dllexport) int __stdcall DestroyMyDLGForm(void);

extern "C" __declspec(dllexport) wchar_t* __stdcall GetMyEntAnswer(void);
extern "C" __declspec(dllexport) wchar_t* __stdcall GetMyCorrectAnswer(void);
extern "C" __declspec(dllexport) wchar_t* __stdcall GetMyQuestion(void);

extern "C" __declspec(dllexport) int __stdcall GetMyResult(void);

```

3. Примеры этих функций:

```

wchar_t* __stdcall GetMyEntAnswer(void){
    wchar_t* bufret = MyEntAnswer.w_str();
    return bufret;
}

wchar_t* __stdcall GetMyCorrectAnswer(void){
    wchar_t* bufret = MyCorrectAnswer.w_str();
    return bufret;//debug
}

wchar_t* __stdcall GetMyQuestion(void){
    wchar_t* bufret = MyQuestion.w_str();
    return bufret;
}

int __stdcall PrepareAndShowModal(TRecTestPChar rpch){

```

```

FormFromDLL = new TFormFromDLL(NULL);
int bufret = FormFromDLL->PrepareAndShowModal(rpch);
    return bufret;
}

```

```

int __stdcall PrepareStartT(TMyStartRecord sr){
    MyStartRecord.iCurSecForTO = sr.iCurSecForTO;
    MyStartRecord.myLessonType = sr.myLessonType;
    MyStartRecord.idtSLYear = sr.idtSLYear;
    MyStartRecord.idtSLMonth = sr.idtSLMonth;
    MyStartRecord.idtSLDay = sr.idtSLDay;
    MyStartRecord.idtSLHour = sr.idtSLHour;
    MyStartRecord.idtSLMin = sr.idtSLMin;
    MyStartRecord.idtSLSec = sr.idtSLSec;
    MyStartRecord.idtSLmSec = sr.idtSLmSec;
    return 1;
}

```

```

int __stdcall DestroyMyDLGForm(void){
    delete FormFromDLL;
    return 1;
}

```

```

int __stdcall GetMyResult(void) {
    return MyResult;
}

```

4. Глобальные переменные:

```

TFormFromDLL *FormFromDLL;
TMyStartRecord MyStartRecord;
String MyEntAnswer = "";
String MyCorrectAnswer = "";
String MyQuestion = "";
int MyResult = 0;

```

3.4. Отладочное приложение

Для проверки и отладки DLL с заданием и его формой вышеупомянутые архивы укомплектованы простейшим приложением, позволяющим загрузить DLL и открыть форму задания из неё. Файл ProjectFormDLL.exe находится в папке проекта. Архив для IDE Delphi также укомплектован исходными кодами этого приложения.

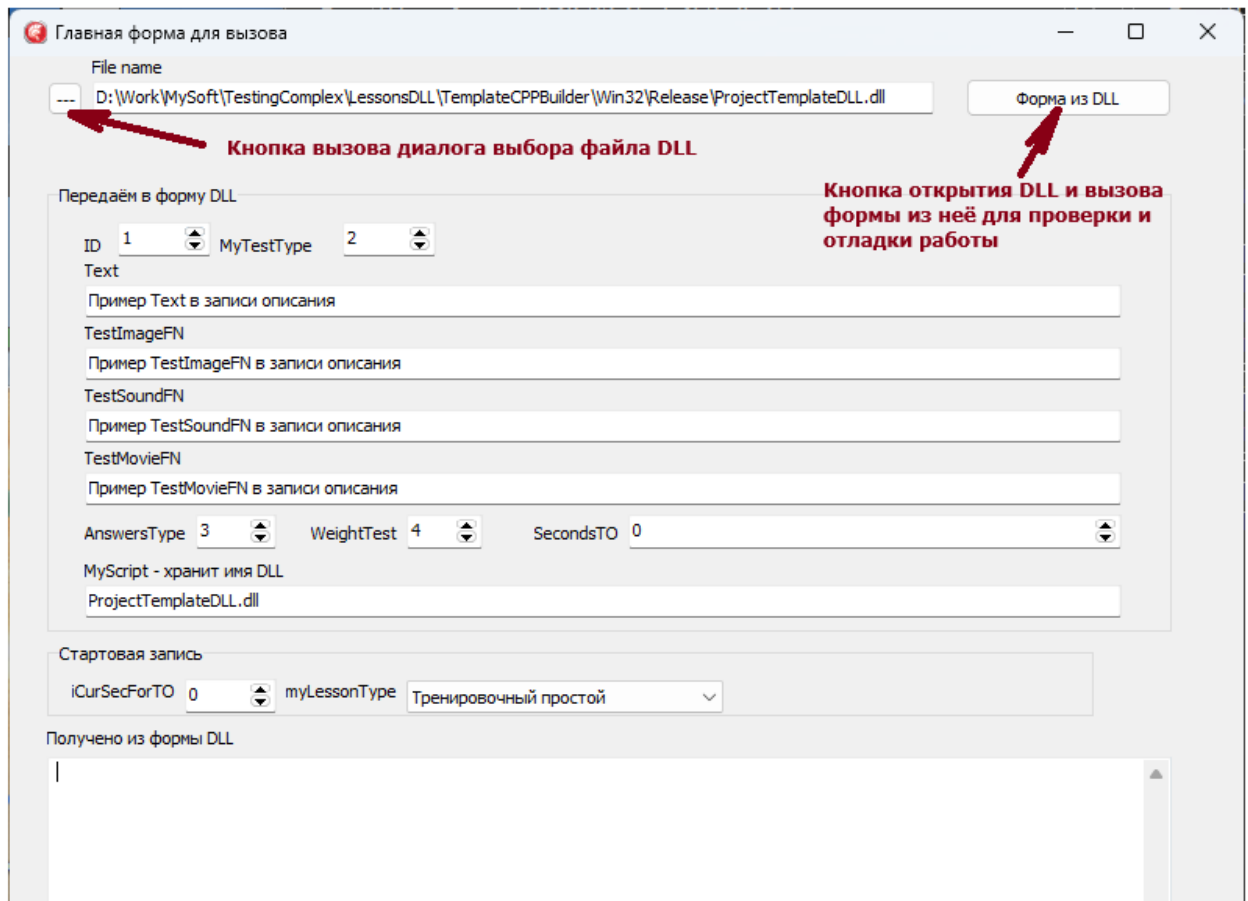


Рисунок 8. Отладочное приложение