**Стартовое решение для создания расширений для SE.**

Пример. В дистрибутиве se есть папка extentions, в которой и находятся расширения. Когда приложение запускается оно проверяет существование папки. Если есть тогда ищет все расширения (в windows это могут быть те же DLL), загружает их и инициализирует в порядке инициализации (приоритет инициализации получают из расширения).

Чтобы например можно было отлаживать расширение.

В комплект входит самая базовая математика. Можно добавить ещё.

Моя матлиба вот она: <http://code.google.com/p/gbmath/>. Могу взять оттуда с поправками.

Что требуется от Владимира ? Совсем немного! Просто он помещает класс с реализацией абстрактников к себе и прописывает реализацию.

Класс адаптер для расширений.

Чтобы пояснить что это такое лучше примером:

//! space engine extentions

namespace sse

{

//! Это находится в extentions SDK

class IAdaptor

{

public:

//! некоторое для выполнение чего-либо, или получения.

virtual void DoSome(int param1, int param2) =0;

};

}

/\*\* это автор кладёт к себе в исходники и прописывает реализацию абстрактников \*/

class AdaptorImplement: public see::IAdaptor

{

public:

AdaptorImplement() {}

~AdaptorImplement() {}

//

virtual void DoSome(int param1, int param2)

{

// do code. Тут заполняется автором.

}

};

**Вопросы.**

Что должно быть в адаптере ? Какие методы ?

Вопрос решается в обсуждении. Чтобы не дергать инженера, можно прописать методы про запас, сразу же. Так как их наверняка будет много, то предусмотреть их группировку по префиксу .   
Например методы системные будут начинаться с SYS\_

( virtual float SYS\_GetTime() =0; )

Методы для изображений с IMG\_

(virtual void IMG\_CreateImageFromFile(void\*\* , size\_t\*, size\_t\*, const wchar\*) =0;)

**Возможность запускать приложение se с ключами.**

Например это может быть ключи, облегчающие отладку расширений, например когда не загружается/не создаётся что-либо. Например приложение после старта сразу загружает расширения