Написание плагина к SeePP

Плагин должен реализовывать интерфейс `DrawableDataInterface`.

Если плагин отображает представление определённых данных на каждом шаге:

1) Реализуем isRequiredCustomVariables(). Если плагин показывает представление одной или нескольких конкретных переменных (которые потом выберет пользователь), то она должна возвращать true.
2) Если isRequiredCustomVariables() возвращает true, то нужно реализовать requiredCustomVariables(), возвращающую список переменных, которые должен указать пользователь.
3) Нужно реализовать функцию setData(). Она будет вызываться на каждом шаге, и в ней надо получать все данные из отладчика. Необходимые для создания графического представления данные здесь должны запоминаться.
4) Реализовать calcGeometry(). В нём можно произвести все расчёты, если представление выбранных данных будет достаточно сложным. В результате в поле size нужно записать размер будущего графического представления данных.
5) Реализовать createGraphicsItem(). Она будет вызвана однократно после подключения плагина. После создания указатель на графическое представление лучше запомнить в поле класса для последующего обновления.
6) Реализовать updateGraphicsItem(). Она вызывается каждый раз после setData() и будет обновлять графическое представление в соответствии с изменениями в данных. Если нужно показать анимацию после предыдущего состояние, здесь же создаётся эта анимация.
7) Реализовать getAnimationAfterStep(). Она будет возвращать созданную анимацию или NULL.

Если плагин показывает последовательность вызовов функций.

Механизм работы:

Запуск анализа вызовов функций происходит по кнопке визуализатора Start session. При этом запускается проход программы до следующей точки останова, при котором на каждом вызове одной из заданных плагином функций вызывается метод плагина setFunctionParams() (далее – сессия). На момент вызова setFunctionParams() отлаживаемая программа находится на первой строчке одной из заданных функций - в этот момент из поставщика данных можно получить все необходимые сведения о параметрах, переданных в функцию.

Чтобы останавливаться в начале каждой из заданных функций перед запуском прохода на первую строку каждой из них автоматически ставятся точки останова, а по окончании сессии они убираются.

1) Реализовать метод getRepresentedFunctionNames(). Он возвращает список имён функций, в которые нужно ставить временную точку останова. Если функция принадлежит классу, пишется ClassName::methodName.

2) Реализовать setFunctionParams(). Она является аналогом setData() – здесь нужно запомнить параметры, переданные в функцию при последнем вызове.
   При реализации сперва нужно убедиться, что в functionName имя нужной функции.
   Затем через метод TraceDataProvider getFunctionArguments() получить параметры, переданные в функцию. Все данные так же нужно запомнить, для последующего создания представления.
3) Аналогично реализовать calcGeometry(), createGraphicsItem() и updateGraphicsItem().
4) Метод getAnimationAfterStep() должен возвращать анимацию за всю сессию. В результате по окончании сессии будет прокручена анимация последовательности вызовов функций.

Установка плагина: скопировать dll и lib файлы плагина, а также все используемые им картинки в папку plugins, расположенную в корне проекта SeePP (при отладке визуализатора из MSVS), или расположенную в одной папке с seepp.exe (при его обычном запуске).

Советы:

1) Два описанных вида плагина возможно совместить в одном.
2) Вместо отдельного проекта плагина можно создать такой же класс в самом проекте SeePP.
   В это случае метод isPlugin должен возвращать false.
   Тогда объект нового класса нужно будет добавить в контейнер objects в методе DataStructureRepresentation::initObjects.
3) Как создаются плагины в Qt можно почитать в Qt Assistant - How to Create Qt Plugins, примеры плагинов выложены в репозитории https://bitbucket.org/onoliferon/debug-visualization/downloads.
4) В том же репозитории можно создавать issue с любыми вопросами по разработке.