**5. Функциональные опции**

**По объему - самое минимальное. Но, на мой взгляд, очень интересное, т.к. раскрывает важную "изюминку" платформы - возможность быстрой настройки программы самим пользователем под свои конкретные нужды. Реализация данной возможности осуществляется через механизм функциональных опций.**

Функциональные опции позволяют выделить некоторую часть функциональности прикладного решения и быстро и просто управлять тем, будет эта часть использоваться, или не будет использоваться в конкретной информационной базе.

Использовать функциональные опции удобно при внедрении программ. Например, в нашей программе предусмотрено три «учетных» раздела – работа со знакомыми, событиями и финансовый учет. Если мы решим подарить нашу разработку друзьям или доработать ее функционально и продавать, всегда можно поинтересоваться, а все ли нужно потенциальным пользователям из того, что мы предлагаем. Например, далеко не все готовы набирать и отслеживать все события, которые происходят в их жизни. При этом вести список друзей и знакомых полезно практически всем, как и считать свои деньги:-).

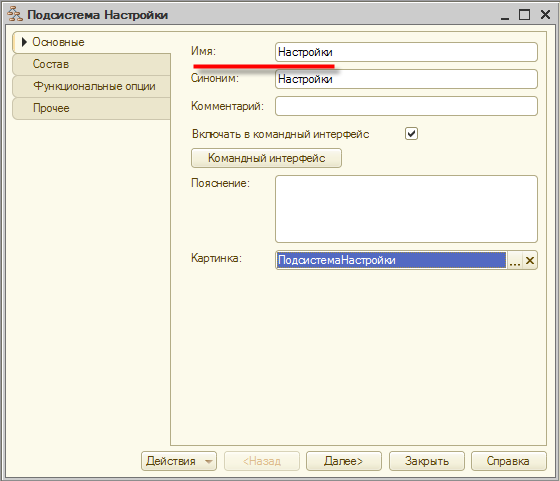
Так вот, узнав о том, что кому-то работа с событиями может показаться излишней, мы можем выделить это в функциональную опцию и предоставить пользователям самим решать, нужно им это или нет. При этом платформа «1С:Предприятие 8» отследит все упоминания событий во всех разделах конфигурации и, если это пользователю не нужно, автоматически отключит их использование и отображение.

Причем все эти изменения платформа выполнит самостоятельно, так что специалисту нет необходимости как-либо изменять конфигурацию (что-либо программировать). Он просто в пользовательском режиме *1С:Предприятие* задает нужное значение функциональной опции – и все!

Мы рассмотрим самый простой пример – когда значение функциональной опции хранится в константе, имеющей тип *Булево*. Если там значение *Истина* – значит, функциональная опция включена. Если значение *Ложь* – функциональная опция выключена.

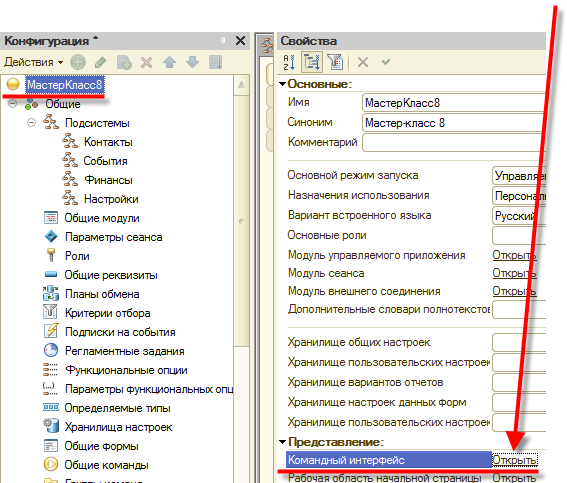
Т. к. мы создаем первую пользовательскую настройку нашей программы, логично будет создать для нее подсистему Настройки и назначить отдельную картинку «*ПодсистемаНастройки.png*».

Здесь я подготовил свою картинку (не удивляйтесь).



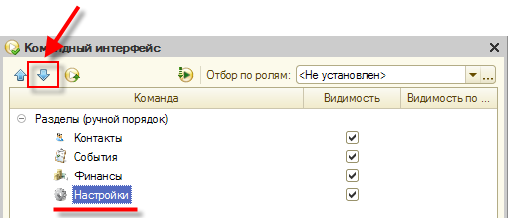
**Рис. 9-1.** Новая подсистема Настройки

Сразу расположим нашу новую «настроечную» подсистему в конце «обычных» подсистем. Для этого в свойствах конфигурации нажмем на ссылку *Командный интерфейс.*



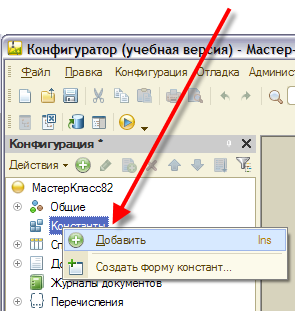
**Рис. 9-2.** Командный интерфейс

И переместим подсистему *Настройки* в самый низ.

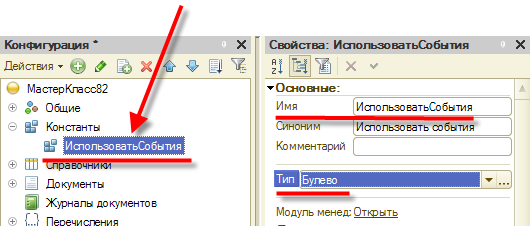


**Рис. 9-3.** Командный интерфейс

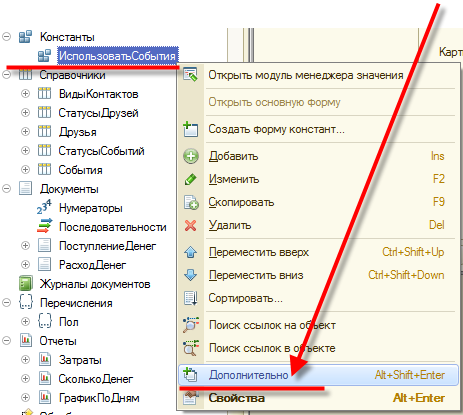
После этого создадим константу, назовем ее *ИспользоватьСобытия* и скажем, что она имеет тип *Булево*. Сразу привяжем константу к подсистеме *Настройки* (меню *Дополнительно Alt+Shift+Enter*).



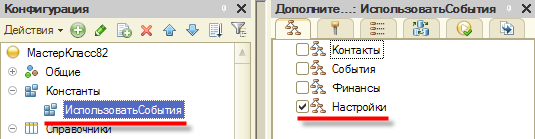
**Рис. 9-4.** Создание константы



**Рис. 9-5.** Создание константы ИспользоватьСобытия

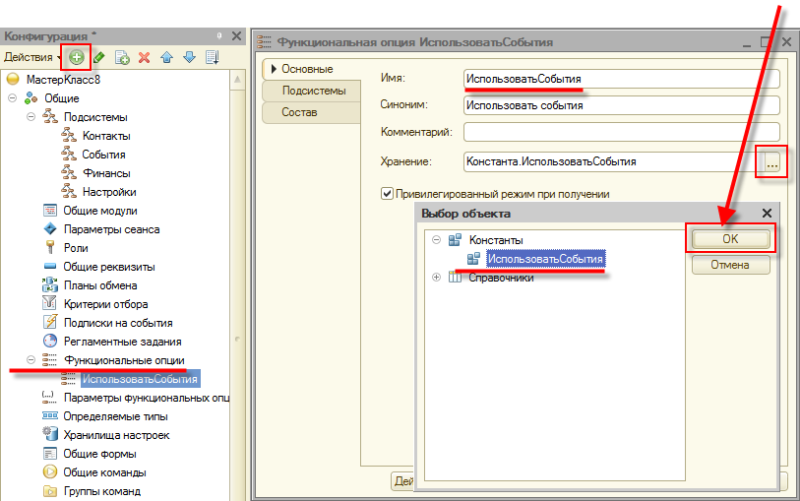


**Рис. 9-6.** Дополнительные свойства объекта



**Рис. 9-7.** Привязка константы ИспользоватьСобытия к подсистеме Настройки

Затем мы создадим функциональную опцию *ИспользоватьСобытия* и укажем, что ее значение будет храниться в константе *ИспользоватьСобытия*.

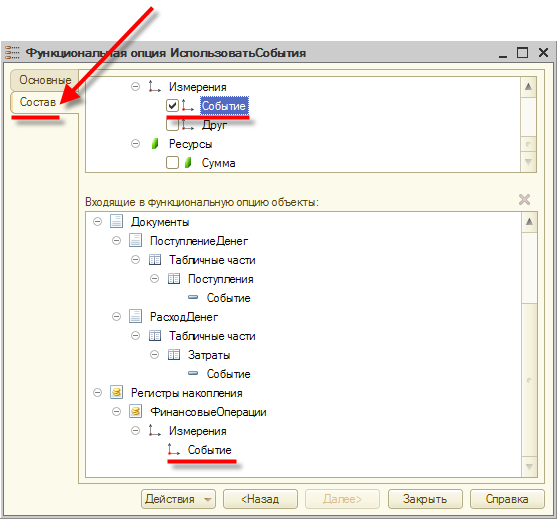


**Рис. 9-8.** Создание функциональной опции ИспользоватьСобытия

В результате этих действий в разделе *Настройки* у нас появится команда, которая позволит нам в пользовательском режиме открыть форму констант и изменить включить или отключить использование функциональной опции.

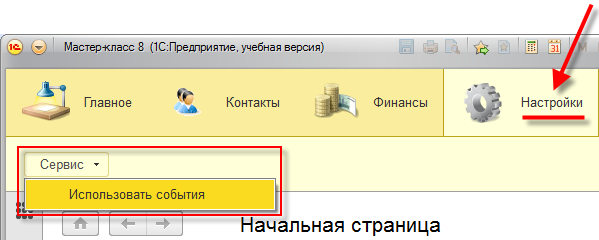
Теперь нам нужно привязать объекты конфигурации и реквизиты объектов к этой функциональной опции. Что в нашей системе обеспечивает функциональность работы с событиями? Прежде всего, это подсистема *События*; далее справочник *События* и связанный с ним справочник *СтатусыСобытий*; затем – реквизиты табличных частей документов *Событие*; и, наконец, измерение *Событие* регистра накопления.

Очень просто привязать все перечисленные выше элементы конфигурации к функциональной опции *ИспользоватьСобытия*. Достаточно в окне параметров функциональной опции перейти на закладку *Состав*. После чего последовательно отметить флажки у всех объектов, относящихся к функциональной опции. Для контроля система будет выводить в нижнем окне список того, что мы отметили.

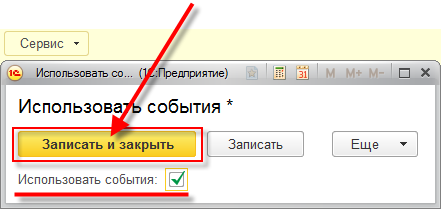


**Рис. 9-9.** Привязка объектов к функциональной опции

Теперь запускаем пользовательский режим, сразу идем в раздел *Настройки* и включаем использование *Событий* (в секции *Сервис*), чтобы проверить, что создание функциональной опции ни на что в нашей системе не повлияло.



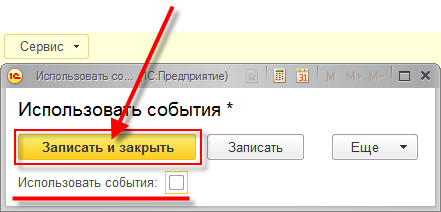
**Рис. 9-10.** Включение функциональной опции



**Рис. 9-11.** Включение функциональной опции

Перезапускаем пользовательский режим и видим, что все на своих местах (осталось, как и было до введения функциональной опции).

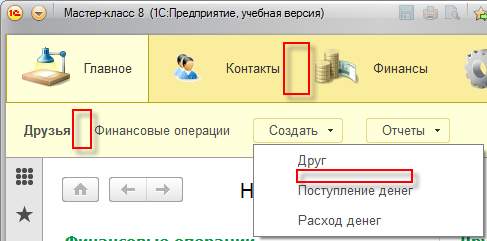
Теперь идем в *Настройки* и отключаем использование событий. Сохраняем изменения. Перезапускаем пользовательский режим.



**Рис. 9-12.** Отключение функциональной опции

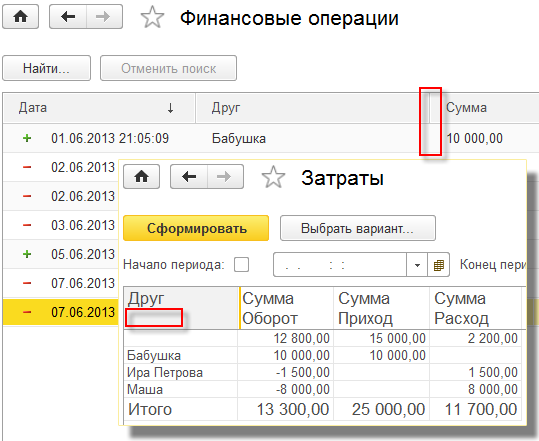
Внимательно смотрим на наше приложение.

Во-первых, «пропал» раздел *События* и упоминания объектов, связанных с событиями на рабочем столе.



**Рис. 9-13.** Отсутствие событий

Во-вторых, исчезли все упоминания событий во всех используемых в системе объектах, включая информацию, отображаемую отчетами.



**Рис. 9-14.** Отсутствие событий

Мы получили то, что хотели – использование учета событий в нашей CRM-системе стало опциональным и зависит только от желания конкретного пользователя этой системы.