1. Обновлена форма «О программе» для отображения новой версии.
2. Форма факторного анализа завершена. Краткая инструкция по работе:

* В области начального периода выбираем нужный год. Если в базе нет данных по указанному году, система выдаст сообщение, что за выбранный период нет данных. При этом в этот момент вызывается функция очистки всех меток в модели Дюпона.
* Аналогично делаем со конечным периодом.
* Нажимаем расчет. Заполняется модель. Расчетные формулы взять из Екселя. Все формулы вручную воспроизводил сначала в файле, потом в программе. Где-то в процентах идет расчет, поэтому нужно на 100 умножать, где-то просто расчет, где-то наоборот на 100 делится. Цветом выделяется согласно описанию. В момент запуска срабатывает проверка на наличие первоначальных показателей (первая – самая большая часть ракеты), т.к., все остальные отростки идут от них. Также проверяются все делители, чтобы не было нулей.
* Выпадающий список «сохраненные даты расчетов» позволяет пользователю увидеть все даты расчетов и при нажатии на дату удалить все связанные данные за этот момент (т.е., за каждую выбранную дату удаляется связка из 4-х показателей в базе).
* «Очистка», понятное дело очищает все метки.
* «Сохранить» выполняет сохранение видимой части модели Дюпона в формате файла \*.png, иных способов нет, т.к., модель представляет собой макет из контролов, а не готовый контрол.

Пояснение по выбору дат в периодах. В момент проверки наличия записей за указанный год, если число возвращаемых строк равно 4, то они автоматом заполняются в метках. 4 строки означает, что за указанный год было единичное сохранение и больше показатели не менялись. Если количество равно 0, значит таблица в базе пустая и ФАНЗ выдает сообщение о отсутствии данных. Если строк больше 4-х, то вызывается форма, на которой выводятся все сохраненные показатели для того, чтобы пользователь выбрал нужную четверку показателей. Фильтровать по дате. Просто строка поиска у грида включается для всех столбцов сразу. Для отдельного нельзя. На последнем видео неудачный пример показал. В базе даты хранятся вплоть до секунд, а в гриде секунды не показываются. А так как расчеты у меня за сегодня, то дата, якобы, одинаковая. Но в работе так не будет. Даты всегда будут различаться и при фильтрации по дате автоматом показывает соответствующую четверку показателей. На это надо обратить внимание. Соответственно, обновить показатели отображает в текстовых полях нужные значения, а передать выталкивает значения в модель Дюпона. При закрытии формы происходит тоже самое. При лицезрении чуда по отсутствию данных в текстовых полях программа выводит сообщение и не дает закрыть форму, чтобы пустые строки вместо чисел не ушли для расчета.

Также важный момент, на который надо обратить внимание. В выпадающих списках если значение выбрано, то повторное нажатие на нем данных не выводит. Т.е, если я нажал на 2016, а потом еще раз, то ничего не произойдет. Так работает событие EditValueChanged (). Чтобы выбрать повторно тот же год - нужно выбрать другое значение, а уже выбрать нужное. Обращаю внимание, что если при выборе значения данных нет, то вызывается очистка. Если возникают ошибки, то всегда вызывается очистка. Форма очень сложная и тяжело все уложить в простой процесс всю работу. Поэтому при любых ситуациях, когда выбора данных или расчета не происходит, то идет очистка, чтобы продолжить работу с нуля. Если этого не делать, то возникают проблемы при остальных операциях, у которых могут законфликтовать свои проверки или сами выполняемые операции. Поэтому я сделал такую тотальную защиту от дурака, так сказать. Поначалу, пока данных в базе нет будет особенно заметно, потом, конечно, количество таких очисток из-за пустых данных сократится.

Еще я намерено при очистке убираю только расчетные значения, но не сами выбранные значения в периодах. Очистка этих выпадашек вызывает EditValueChanged (), у которого свои проверки и получается какая-то неуправляемая фигня, полночи на эту убил. А вызывать это событие нам не надо. Хотя думаю, каких-то вопросов периоды не вызывают что так, что так.

1. В динамическом отчете поправил вывод сообщения о построении отчета. Если данных нет по указанным показателям, то сначала выходило сообщение, что данных нет, а потом, что отчет построен, а как же он построен, если данных нет.
2. Для таблицы отклонений «поднял» зум бар для таблицы, как на остальных вкладках. Он был, но уполз вниз, из-за перекреста нескольких контролов, которые определяют макет.
3. Согласно замечанию от 31 мая: «*На вкладке Результат оптимизации... последние два поля (Чистая прибыль и нераспределенная прибыль) для фактических значений нужно сделать просто редактируемыми и без всяких в них формул. С учетом того что налоговое законодательство постоянно меняется нет смысла там формулы делать. Проще вручную туда внести данные. Подсказку на нераспределенной прибыли конечно оставить*» - открыл редактирования последних двух строк. Подсказка сохранена, все остальное модификации не потребовало. По таблице оптимизации обращаю внимание на один момент. В гриде devexpress нет понятия ячейки с набором привычных свойств. Тут нельзя сделать режим редонли для одной ячейки, и наоборот для другой. Есть события, которые обрабатывают строки, и там можно задать режим редонли для строки. Таким же образом настраиваются и остальные свойства. В принципе половина того, что видно на экране – это обработка событий, а не назначение свойств.
4. Синхронность записи данных в базу для динамического отчета и для прогнозирования доделал. Оказывается, я это уже начинал делать. Раньше было как. При расчете итогового отклонения автоматом улетало в базу и на форме прогнозирования при множественных имитационных расчетов тоже было много данных. Сейчас запись в обе таблицы улетает из одной функции и тем самым реализовывается соответствие между количеством данных для прогнозирования и динамического отчета.

Разумеется, удалить записи из каждой таблицы пользователь может и там, и там, но это нормально и уже на совести пользователя вести соответствие по датам. Т.е, если среди 10-ти сохраненных расчетов в прогнозировании удалят 5 дат, а в динамическом 7, то это на совесть юзера. К сожалению, синхронно при удалении из одной таблицы подхватывать и из другой, неважно с какой формы удаляем нельзя. Я попробовал. Тут нюанс в том, что при удалении я передаю полную дату, которая записана в базе, это формат datetime, который учитывает вплоть до секунд. А вот время записи по секундам и различаются. На домашнем компе у меня разница в датах в одну секунду. Комп очень быстрый. На менее шустрой машине разница может составить и более секунды. Тут не угадаешь, в итоге.

Поэтому я переделал механизм удаления из модуля прогнозирования. Вместо ID, пользователь видит выпадашку с сохраненными датами и удаляет уже по дате. Так же, как и в динамическом отчете. Вроде, красиво и удобно вышло. В табличке ненужные столбцы скрыл.

1. Обновил все расширения для ФАНЗ, пересобрал все dll и саму ФАНЗ с учетом параметра оптимизации кода при построении в самой студии. Там, где можно было (без последствий) собирал под платформу x86, отключил (как оказалось установлено по умолчанию) загрузку шрифтов при отладке и после шрифтов с серверов MS. Также средствами студии перед сборкой поудалял везде неиспользуемые сборки. Почистил остатки контролов, свойств, методов, которые уже не актуальны, тем самым уменьшая обработку и сокращая процент возникновения ошибок программы при различных кейсах работы пользователей. Попытки использовать оптимизацию кода вкупе с использованием платформы x86 позволило сократить ошибки MS, которые иногда возникают после закрытия ФАНЗ (а у меня еще и при каждой компиляции). Вот эти ошибки. Месячная попытки и исследования привели к установлению причины – windows, различные версии net. framework и приложения под них фигово работают при разных разрядностях системы. [**DLL hell**](https://ru.wikipedia.org/wiki/DLL_hell)во всем своем великолепии. MS тупит, в devexpress достаточно багов, + все это тянуть через x86 и x64, при этом далеко не все ошибки имеют какие-то стабильные варианты решения, даже формулировку и причину проблемы порой тяжело найти.
2. Прошелся по всему коду (и в ФАНЗ и в dll, когда пересобирал), где не хватало обработки исключений и останова кода добавил, т.е., по сути где-то дополнил, где-то добавил try- catch. Тем самым количество возможных ошибок программы или необычных действий пользователя, которые могут привести к ошибкам – постарался сократить. По большей части, надеюсь без фокусов MS, все исключения по программе полностью обработаны.