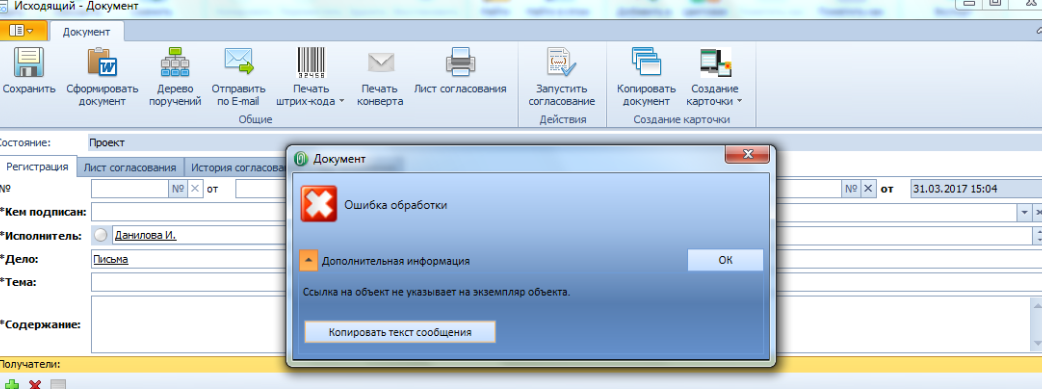
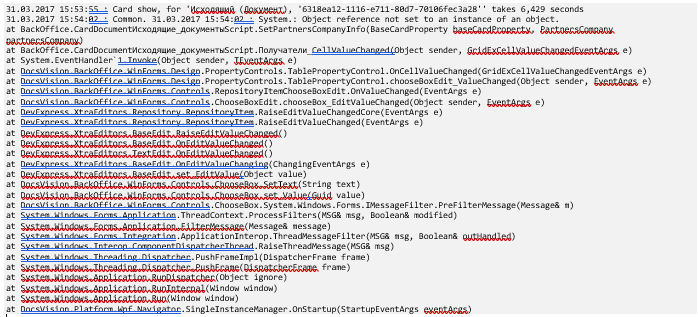
Задача 1.

Необходимо проанализировать проблему, возникшую при работе в системе, и предложить способ ее решения:

При создании исходящего документа выходит ошибка, когда заполняю поле получатели. Причем ошибка выходит не на всех пользователях. Когда заполняю поле получатели "Пелевин А.А., Смирнов А.В." выходит ошибка и программа закрывается.



В журнале ошибок приложения имеются такие записи:



Для исходящего документа в системе настроен следующий сценарий:

namespace BackOffice

{

    public class CardDocumentИсходящие\_документыScript : DocumentDMScript

    {

        private void SetPartnersCompanyInfo(BaseCardProperty baseCardProperty, PartnersCompany partnersCompany)

        {

            baseCardProperty["EmailOrganization"] = partnersCompany.Email;

            baseCardProperty["FaxOrganization"] = partnersCompany.Fax;

            string address = "";

            foreach (var partnersAddress in partnersCompany.Addresses)

            {

                if (partnersAddress.AddressType == PartnersAddresseAddressType.PostAddress)

                {

                    if (partnersAddress.ZipCode.Length > 0)

                        address = partnersAddress.ZipCode + ", ";

                    if (partnersAddress.City.Length > 0)

                        address += partnersAddress.City + ", ";

                    if (partnersAddress.Address.Length > 0)

                        address += partnersAddress.Address + ", ";

                    if (partnersAddress.Country.Length > 0)

                        address += partnersAddress.Country;

                }

            }

            baseCardProperty["Address"] = address;

        }

        #region Event Handlers

        private void Получатели\_CellValueChanged(System.Object sender, DocsVision.BackOffice.WinForms.Controls.GridExCellValueChangedEventArgs e)

        {

            ITableControl iTableControl = Customizable.FindPropertyItem<ITableControl>("Получатели");

            int index;

            index = iTableControl.FocusedRowIndex;

            BaseCardProperty baseCardProperty = iTableControl.FocusedRowItem;

            if (e.FieldName == "Recipient")

            {

                if (GetGuid(e.Value) == Guid.Empty)

                    return;

                if (e.Value != null && new Guid(e.Value.ToString()) != Guid.Empty)

                {

                    IPartnersService partnersService = ObjectContext.GetService<IPartnersService>();

                    Guid id = new Guid(e.Value.ToString());

                    PartnersEmployee partnersEmployee = partnersService.Get(id);

                    PartnersCompany partnersCompany = partnersEmployee.Unit;

                    if (partnersCompany.Type != 0) partnersCompany = GetParentPartnerOrganization(partnersCompany);

                    baseCardProperty["EmailRecipient"] = partnersEmployee.Email;

                    baseCardProperty["FaxRecipient"] = partnersEmployee.Fax;

                    baseCardProperty["Organization"] = ObjectContext.GetObjectRef<PartnersCompany>(partnersCompany).Id;

                    SetPartnersCompanyInfo(baseCardProperty, partnersCompany);

                    iTableControl.RefreshRow(index);

                }

                else

                {

                    baseCardProperty["EmailRecipient"] = "";

                    baseCardProperty["FaxRecipient"] = "";

                    baseCardProperty["Organization"] = null;

                    baseCardProperty["EmailOrganization"] = "";

                    baseCardProperty["FaxOrganization"] = "";

                    baseCardProperty["Address"] = "";

                    baseCardProperty["SendingType"] = null;

                    baseCardProperty["SendingState"] = false;

                    iTableControl.RefreshRow(index);

                }

            }

            if (e.FieldName == "Organization")

            {

                if (GetGuid(e.Value) == Guid.Empty)

                    return;

                if (e.Value != null && new Guid(e.Value.ToString()) != Guid.Empty)

                {

                    Guid id = new Guid(e.Value.ToString());

                    PartnersCompany partnersCompany = ObjectContext.GetObject<PartnersCompany>(id);

                    SetPartnersCompanyInfo(baseCardProperty, partnersCompany);

                    iTableControl.RefreshRow(index);

                }

                else

                {

                    baseCardProperty["EmailOrganization"] = "";

                    baseCardProperty["FaxOrganization"] = "";

                    baseCardProperty["Address"] = "";

                    baseCardProperty["SendingType"] = null;

                    baseCardProperty["SendingState"] = false;

                    iTableControl.RefreshRow(index);

                }

            }

        }

        #endregion

    }

}

В чем может заключаться ошибка? Какой алгоритм поиска этой ошибки вы можете предложить? Как можно исправить эту ошибку?

**Задание 2.**

Написать программу для обработки и валидации списка клиентов организации.

Входные параметры:

\* xml-файл со списком клиентов (см. вложение)

На выходе:

\* xml-файл со списком клиентов, записи которых не содержат ошибки. Запись о клиенте считается ошибочной, если отсутствуют или не заполнены поля ФИО, Рег. номер, DiasoftID, Регистратор. Каждая запись клиента должна содержать дополнительный атрибут RegistratorID - числовой идентификатор каждого уникального регистратора в исходном списке. Идентификатор нужно сгенерировать самостоятельно.

\* xml-файл со списком регистраторов с полями Name и ID, соответствующих полям Registrator и RegistratorID из предыдущего файла.

\* текстовый файл с описанием ошибок, сгруппированных по типу, отсортированных по количеству ошибочных записей данного типа в обратном порядке, а также с общем количеством ошибочных записей. Пример файла отчета об ошибках:

Не указан DiasoftID: 100 записей

Не указан Регистратор: 15 записей

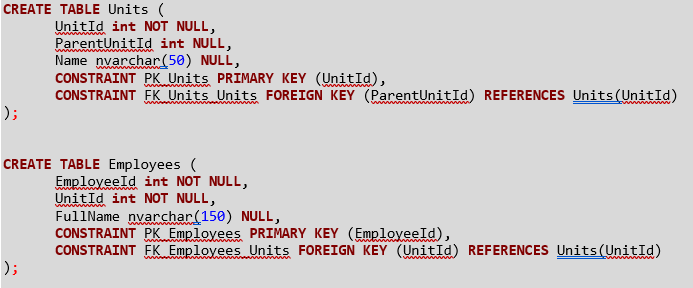
Не указано ФИО: 5 записей

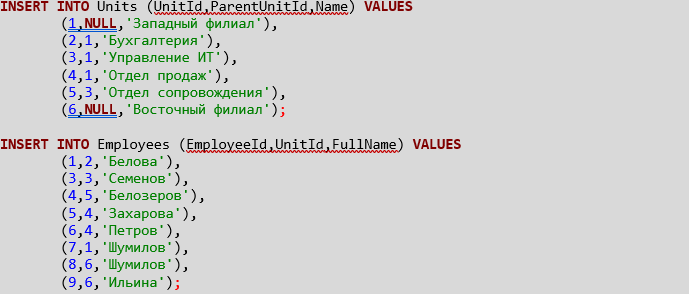
Всего ошибочных записей: 105

Программа должна быть написана на языке C#. Исходный код решения нужно выслать по электронной почте в виде архива БЕЗ исполняемых файлов. Допускается отправка решения в виде ссылки на публичный репозиторий. xml-файлы в репозитории не публиковать.

**Задание 3.**

Существует база данных со структурой организации. Сценарий для создания БД следующий:



Необходимо составить SQL-запрос, который выводит фамилии сотрудников, которые находятся в западном филиале (включая подчиненные подразделения).

По скрипту создания сотрудников видно, что Шумилов входит в 2 подразделения. Подходит ли созданная структура таблиц для таких случаев? Если нет – как можно изменить структуру таблиц для исправления?