**Федеральное государственное образовательное бюджетное учреждение  
высшего образования**

**«Финансовый университет при Правительстве Российской Федерации»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ПМ.02 Разработка, администрирование и защита баз данных**

**Группа: 3ПКС-115**

**УТВЕРЖДАЮ**

**Председатель цикловой комиссии**

**программирования и баз данных**

**( ) Пестов А.И.**

**\_\_\_\_.\_\_\_\_. 2018**

**ПРОЕКТ КУРСОВОЙ**

**На тему: Разработка автоматизированной информационной системы приемной комиссии колледжа на платформе 1С:Предприятие 8.3**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**Руководитель курсового проекта**

**(Костиков П. А.) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Исполнитель курсового проекта**

**(Деменчук Г.М.) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Оценка за проект:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_.\_\_\_\_.2018**

|  |
| --- |
| **2018** |

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ Ошибка! Закладка не определена.](#_Toc485812015)

[1. ОБЩИЙ РАЗДЕЛ Ошибка! Закладка не определена.](#_Toc485812016)

[1.1. Системные требования Ошибка! Закладка не определена.](#_Toc485812017)

[1.2. Характеристика системы программирования Ошибка! Закладка не определена.](#_Toc485812018)

[2. ТХНОЛОГИЧЕСКИЙ РАЗДЕЛ Ошибка! Закладка не определена.](#_Toc485812019)

[2.1. Предпроектное исследование предметной области Ошибка! Закладка не определена.](#_Toc485812020)

[2.2. Анализ требований и определение спецификаций программного обеспечения 10](#_Toc485812021)

[2.3. Проектирование программного обеспечения Ошибка! Закладка не определена.](#_Toc485812022)

[2.4. Разработка пользовательских интерфейсов программного обеспечения 16](#_Toc485812023)

[2.5. Тестирование и отладка программного обеспечения 18](#_Toc485812024)

[3. РУКОВОДСТВО ПО ИСПОЛЬЗОВАНИЮ ПРОГРАММЫ Ошибка! Закладка не определена.](#_Toc485812025)

[3.1. Руководство системного программиста 20](#_Toc485812026)

[3.2. Руководство программиста 21](#_Toc485812027)

[3.3. Руководство пользователя 22](#_Toc485812028)

[ЗАКЛЮЧЕНИЕ 25](#_Toc485812029)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 26](#_Toc485812030)

[ПРИЛОЖЕНИЕ А Ошибка! Закладка не определена.](#_Toc485812031)

# ВВЕДЕНИЕ

Автоматизированная информационная система или АИС – это информационная система, использующая ЭВМ на этапах ввода информации, ее подготовки и выдачи, то есть является неким развитием ИС, которые занимаются поиском информации, используя прикладные программные средства.

Автоматизированные информационные системы представляют собой совокупность различных средств, предназначенных для сбора, подготовки, хранения, обработки и предоставления информации, удовлетворяющей информационные потребности пользователей. АИС объединяет следующие составляющие:

* языковые средства и правила, используемые для отбора, представления и хранения информации, для отображения картины реального мира в модель данных, для представления пользователю необходимой информации;
* информационный фонд системы;
* способы и методы организации процессов обработки информации;
* комплекс программных средств, реализующих алгоритмы преобразования информации;
* комплекс технических средств, функционирующих в системе;
* персонал, обслуживающий систему.

Основными целями автоматизации деятельности предприятия являются:

1. Сбор, обработка, хранение и представление данных о деятельности организации и внешней среде в виде, удобном для финансового и любого другого анализа, и использования при принятии управленческих решений.
2. Автоматизация выполнения бизнес операций (технологических операций), составляющих целевую деятельность предприятия.
3. Автоматизация процессов, обеспечивающих выполнение основной деятельности.

Автоматизированные информационные системы можно смело отнести к классу очень сложных систем и, как правило, не столько с большой физической размерностью, а в связи с многозначностью различных структурных отношений между компонентами системы.

Автоматизированная информационная система может быть легко определена как целый комплекс современных автоматизированных информационных технологий, которые предназначены для какого-либо информационного обслуживания. Без внесения самых современных методов управления, которые базируются на АИС, невозможно и повышение эффективности функционирования предприятий.

Современные АИС позволяют:

* + 1. Повысить производительность работы всего персонала;
    2. Улучшить качество обслуживания клиентской базы;
    3. Снизить напряженность и трудоемкость труда персонала, а также минимизировать количество ошибок в его действиях.

На сегодняшний день, автоматизированная информационная система, является совокупностью технических (аппаратных), математических, телекоммуникационных, алгоритмических средств, методов описания и поиска объектов программирования и сбора и хранения информации.  
Если постараться классифицировать существующие области применения баз данных, а также оценить перспективы их развития в настоящее время, то можно получить примерный список наиболее распространенных классов, получивших распространение и применение во всех областях применения баз данных. Этот список будет выглядеть следующим образом:

* документографические и документальные применяются во всех базах органов власти и управления
* базы данных по промышленной, строительной и сельскохозяйственной продукции
* базы данных по экономической и конъюнктурной информации статистическая, кредитно-финансовая, внешнеторговая
* фактографические базы социальных данных, включающие сведения о населении и о социальной среде
* базы данных транспортных систем
* справочные данные для населения и учреждений энциклопедии и справочники, расписания самолетов и поездов, адреса и телефоны граждан и организаций
* ресурсные базы данных, включающие фактографическую информацию о природных ресурсах земля, вода, недра, биоресурсы, гидрометеорология, вторичные ресурсы и отходы, экологическая обстановка
* фактографические базы данных в области культуры и искусства
* лингвистические базы данных, то есть машинные словари разного типа и назначения.

Применение баз данных в туризме — один из способов увеличения эффективности использования данных организаций, работающих в туристической сфере.

Использование БД позволяет решить следующие важные задачи:

* хранение информации о клиентах — в данном случае БД являются источником информации с возможностью оперативного получения данных о параметрах клиентов и их заявках;
* информирование клиентов — в данном случае посредством БД пользователям предоставляется информация об услугах организации.

Целью курсового проекта является разработка АИС для регистрации и обработки информации об абитуриентах колледжей для увеличения эффективности и скорости работы приёмной комиссии. АИС должна иметь простой и интуитивно понятный интерфейс и ряд функций, таких как удаление, добавление, обновление записей в базе данных, а также печать отчетов и рейтингов.

# Основная часть

## 1. Общий раздел

### 1.1. Системные требования

Программа не предъявляет особых требований к программному и аппаратному обеспечению.

Минимальные системные требования:

* Процессор с архитектурой х86-64 (Intel, AMD);
* Оперативная память 2048мб и выше;
* Свободное место на жестком диске (HDD) от 200 мб;
* Операционная система Windows XP/Vista/7/8/10

### 1.2 Характеристика системы программирования

1С: Предприятие – программный продукт компании 1С, предназначенный для автоматизации деятельности на предприятии.

В современных условиях при автоматизации предприятий приходится сталкиваться с различными и часто диаметрально противоположными требованиями к учету одних и тех же разделов учета. Согласно современным маркетинговым исследованиям рынка   отечественных автоматизированных систем учета «1С: Предприятие» является универсальной системой автоматизации деятельности предприятия. 1С: Предприятие - это специализированная объектно-ориентированная система управления базами данных (СУБД), предназначенная для автоматизации деятельности предприятия.

1С: Предприятие представляет собой систему программ и включает в себя: технологическую платформу, созданные на ее основе прикладные решения различного масштаба и различной направленности, методологию создания прикладных решений, информационно-технологическую поддержку пользователей и разработчиков. Сама платформа не является программным продуктом для использования конечными пользователями, которые обычно работают с одним из многих прикладных решений (конфигураций), разработанных на данной платформе.

Физически серверные и клиентские части системы 1С: Предприятия могут располагаться как на разных компьютерах, так и на одном. Главное, что пользователь не имеет непосредственного доступа к серверу баз данных, это позволяет обеспечивать безопасность данных. А в файловом варианте база данных должна находиться на некотором общем сетевом ресурсе, доступном всем пользователям.

На этапе конфигурирования системы 1С: Предприятие выполняется настройка различных режимов сис­темы в соответствии с особенностями конкретного предприятия. При этом определяются структуры и свойства справочников и документов, описываются формы и алгоритмы построения отче­тов, настраивается ведение учета.

Существуют следующие версии платформы:

* Учебная
* Базовая
* Стандартная
* Профессиональная
* Сетевая

## 2. Технологический раздел

2.1 Предпроектное исследование  
  
Для оформления документов в колледж абитуриенту необходимо сдать относительно небольшое кол-во документов: аттестат за 9 или 11 класс и портфолио, но т.к. абитуриентов достаточно много, а людей в приемной комиссии, наоборот, очень мало, то данные мероприятия могут занимать достаточно продолжительное количество времени, которое можно потратить на более рациональные вещи.

Существует два варианта решения данной задачи:

1. Прием новых сотрудников для приемной комиссии
2. Оптимизация процесса приема документов в приемной комиссии

Прием новых сотрудников не особо рациональное действие: каждому следует каждый месяц выплачивать зарплату и размещать большее количество рабочих мест (стол, стул, бумага и прочие расходные материалы), поэтому мы будет действовать по второму пункту.

Оптимизировать процесс приема документов рациональнее всего с помощью автоматизации каких-либо базовых действий: занесение информации о среднем балле аттестата, формирование рейтинга абитуриентов или выставление текущей даты приема – все это можно реализовать с помощью автоматизированной системы приема абитуриентов.

Достижение цели разбивается на ряд задач. В процессе достижения основной цели создания АИС решаются следующие основные задачи:

1. Информатизация регистрации данных об абитуриентах: предполагает ведение базы данных c несколькими таблицами, содержащими информацию об абитуриентах колледжа. На основании полученных данных ведется формирование рейтинга абитуриентов с какой-либо произвольной выборкой
2. Редактирование базы данных — удаление, обновление, добавление данных абитуриентов.
3. Формирование отчетов. Создание отчетов по каждому абитуриенту при приеме документов в колледж, а также при формировании рейтинга абитуриентов c их последующим выводом на печать.
4. Оповещение пользователей и/или администраторов за счет интеграции кроссплатформенного мессенджера Telegram непосредственно в универсальный общий модуль АИС.

Данный проект был создан в целях автоматизации процесса приема абитуриентов колледжа путем разработки АИС для регистрации и обработки информации для увеличения эффективности и скорости работы приёмной комиссии.

АИС должна иметь простой и понятный интерфейс, а также включать в себя:

* Справочники
* Документы
* Перечисления
* Регистры
* Отчеты
* Общие модули

### 2.2 Анализ требований и определение спецификаций программного обеспечения Рассмотрим краткий анализ программы в целом. Все действия с программой осуществляет сотрудник приемной комиссии. Оформление абитуриента проиходит в несколько этапов:

1. Заполнение информации об абитуриенте в заявлении на поступление абитуриента, а также заполнение контактной информации о родителях абитуриента. При корректном заявлении абитуриент переносится в общий список (справочник) абитуриентов.
2. Формирование отчета о рейтинге абитуриентов, а также общей статитстики об абитуриентах.
3. При необходимости формируется заявление о возврате документов абитуриента, если абитуриент принял решение о возврате документов.
4. Формирование групп в зависимости от специальности абитуриентов, а также отчет о поступлении в конкретную группу.

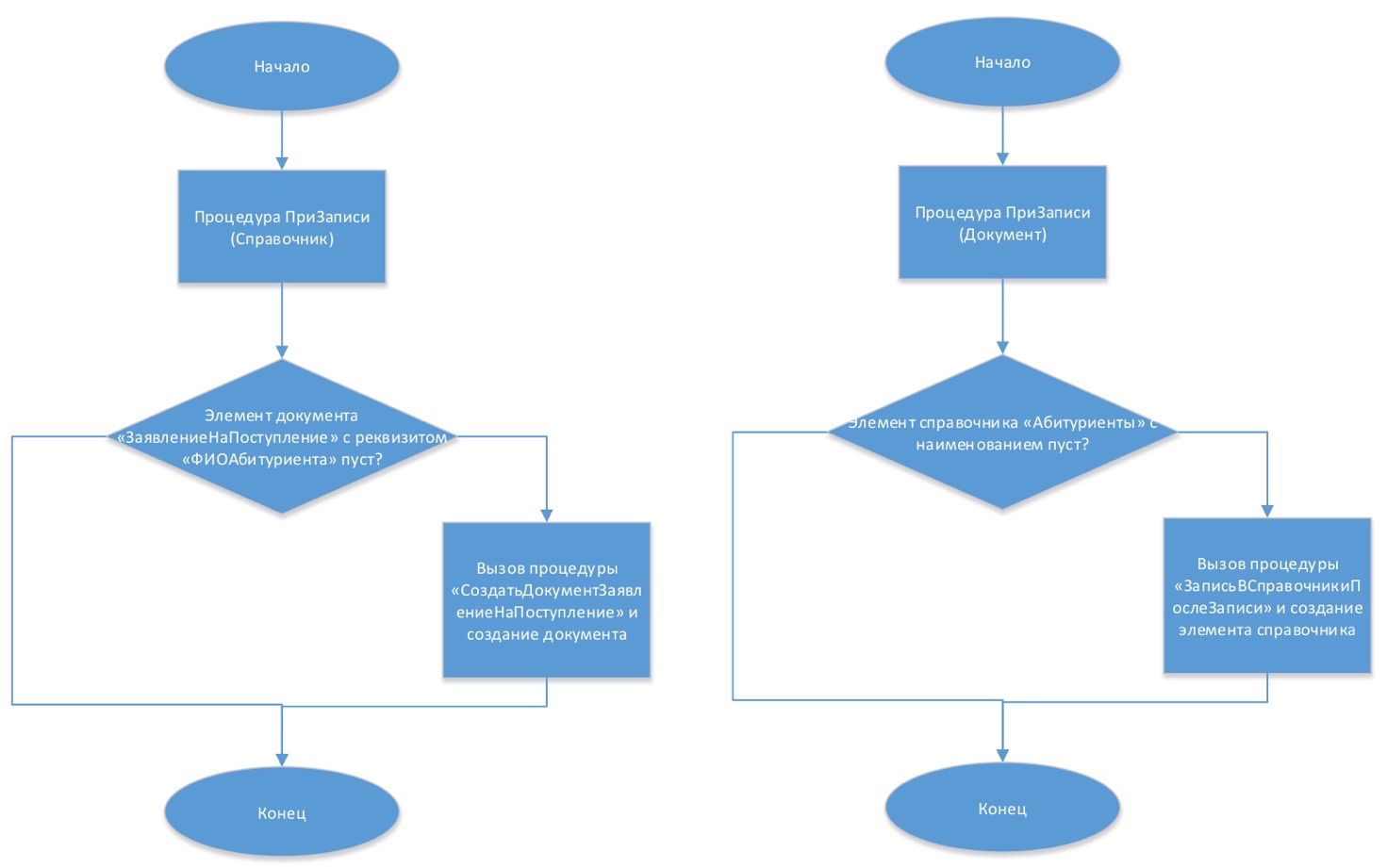
### На каждом этапе происходит информирование персонала о взаимодействии с базой данных посредством интеграции программы с мессенджером Telegram (см ниже).

2.3 Проектирование программного обеспечения

Информационная база содержит следующие справочники:

* Абитуриенты
* ФайлыСистемы
* TelegramBot
* Уведомления
* РодителиАбитуриентов
* СортировкаАбитуриентов
* НастройкаСортировкиАбитуриентов
* Сотрудники

1. Абитуриенты – основной линейный справочник, содержащий информацию об абитуриентах (Рисунки 2-3). Данные добавляются автоматически после заполнения заявления на поступление абитуриента, но также предусмотрена возможность автоматического заполнения заявления по справочнику для минимизации ошибок в программе (т. е. если мы записываем в справочник нового абитуриента, а в документе на поступление его нет, то происходит создание элемента документа с аналогичными данными справочника и наоборот) (Рисунок 1).

  
Рисунок 1 – Пример автоматического двухстороннего заполнения между справочником и документом

Реквизиты справочника:

* СпециальностьАбитуриента – ссылка на перечисление специальностей абитуриента колледжа, в зависимости от колледжа возможны произвольные значения в перечислении, в данном случае возможен выбор между «ПКС» и «ИБАС».
* СреднийБаллАбитуриента – средний балл абитуриента по всем предметам, при вызове формы справочника и заполнении табличной части ОцениАбитуриента вычисляется автоматически.
* ТипАттестатаАбитуриента – ссылка на перечисление типов аттестата абитуриента колледжа, в зависимости от колледжа возможны произвольные значения в перечислении, в данном случае возможен выбор между «Оригинал» и «Копия».
* ПриоритетЗачисленияАбитуриента – содержит булево значение приоритета зачисления абитуриента, выставляется в зависимости от посещенных им подготовительных занятий, а также выигранных олимпиад, форумов, конгрессов и т. д.
* ФормаОплатыАбитуриента – ссылка на перечисление типов формы оплаты обучения в колледже, в зависимости от заведения возможны произвольные значения в перечислении. В данном случае возможен выбор между «Бюджет» и «Договор».
* ФотоАбитуриента – реквизит, содержащий ссылку на элемент справочника «ФайлыСистемы». Отвечает за визаулизацию фотографии абитуриента на форме справочника.
* РодительАбитуриента – ссылка на справочник «РодителиАбитуриентов». Содержит информацию о родителях абитуриента.
* АдресАбитуриента – содержит инфомацию об адресе проживания и/или регистрации абитуриента.
* EmailАбитуриента – содержит инфомацию об электронной почте абитуриента для оперативного информирования последнего.
* ТелефонАбитуриента – содержит информацию о номере телефона абитуриента также для информирования.
* ДатаРожденияАбитуриента – информация о дате рождения абитуриента

Табличные части справочника:

1. ОценкиАбитуриента – содержит информацию об успеваемости на основе аттестата абитуриента за 9 класс. Содержит следующие реквизиты:

* Предмет – полное название предмета по аттестату
* Оценка – оценка за предмет по аттестату

1. ДокументыАбитуриента – данные о документах, которые сдал абитуриент при оформлении. Содержит следующие реквизиты:

* Документ – полное название документа по аттестату. В конкретной реализации является ссылкой на перечисление «ПеречислениеДокументыАбитуриента» со значениями «Паспорт», «СНИЛС», «Аттестат».
* Номер – номер документа для его идентификации.

Также на форме абитуриента распологаются две дополнительные кнопки:

* Сортировка – при нажатии на кнопку происходит обращение к публичной процедуре «СортировкаАбитуриентовСправочник» общего модуля «СортировкаАбитуриентов» где и происходит соритировка абитуриентов программным способом и формируется справочник «СортировкаАбитуриентов» с документом «ДокументОзачисленииВГруппу», который в свою очередь заполняет регистр «РегистрОтчетаОПоступленииВГруппу» для формирования отчета «ОтчетОПоступленииВГруппу». При повторном нажатии вышесказанные объекты хранения затираются и формируются заново.
* Удаление – помечает текущего абитуриента на удаление и формирует новую запись в документе на возврат документов (СНИЛС, аттестат, паспорт), а также производит удаление элемента документа «ЗаявлениеНаПоступление», связанного с текущим документом. Если пользователь уже отмечен на удаление, то происходит обратный процесс: удаляется запись из документа о возврате документов, снимается пометка удаления с текущего абитуриента и записывается новый элемент документа «ЗаявлениеНаПоступление».

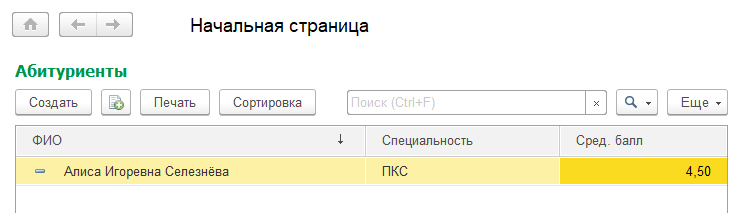


Рисунок 2 - Справочник абитуриенты

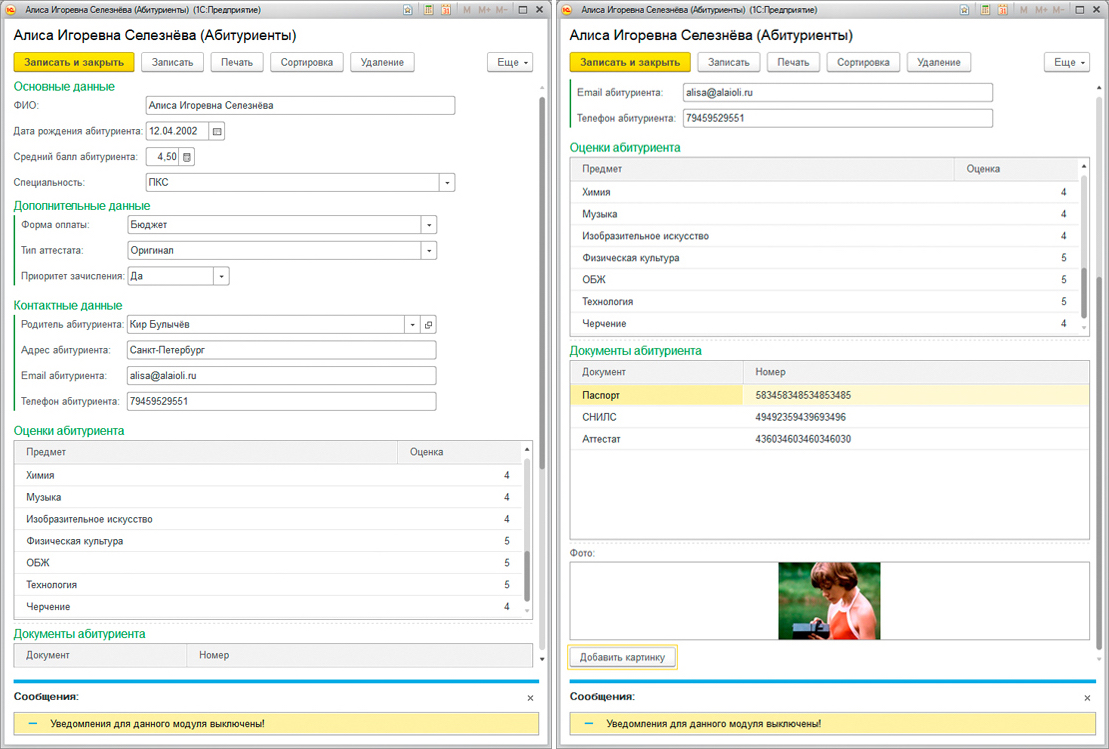


Рисунок 3 – Форма элемента справочника абитуриенты

Листинг 1. Модуль элемента справочника Абитуриенты

//Подсчет предметов для табличной части

Функция ВернутьКолвоПредметов()

Возврат Перечисления.ПеречислениеПредметыПоступленияАбитуриентов.Количество();

КонецФункции

Функция ВернутьЗначениеПредметаПоИндексу(индекс\_номер)

Возврат Перечисления.ПеречислениеПредметыПоступленияАбитуриентов.Получить(индекс\_номер);

КонецФункции

&НаСервере

Процедура СоздатьДокументЗаявлениеНаПоступление()

НовоеЗаявление = Документы.ЗаявлениеНаПоступление.СоздатьДокумент();

НовоеЗаявление.ФИОАбитуриента = Объект.Наименование;

НовоеЗаявление.СпециальностьАбитуриента = Объект.СпециальностьАбитурента;

НовоеЗаявление.СреднийБаллАбитуриента = Объект.СреднийБаллАбитуриента;

НовоеЗаявление.ТипАттестатаАбитуриента = Объект.ТипАттестатаАбитуриента;

НовоеЗаявление.ПриоритетЗачисленияАбитуриента = Объект.ПриоритетЗачисленияАбитуриента;

НовоеЗаявление.ФормаОплатыАбитуриента = Объект.ФормаОплатыАбитуриента;

НовоеЗаявление.АдресАбитуриента = Объект.АдресАбитуриента;

НовоеЗаявление.ДатаРожденияАбитуриента = Объект.ДатаРожденияАбитуриента;

НовоеЗаявление.EmailАбитуриента = Объект.EmailАбитуриента;

НовоеЗаявление.ТелефонАбитуриента = Объект.ТелефонАбитуриента;

ФИОРодителя = Объект.РодительАбитуриента;

НовоеЗаявление.ФИОРодительАбитуриента = ФИОРодителя;

ОбъектРодителя = Справочники.РодителиАбитуриентов.НайтиПоНаименованию(ФИОРодителя);

НовоеЗаявление.ДатаРожденияРодительАбитуриента = ОбъектРодителя.ГодРождения;

НовоеЗаявление.АдресРодительАбитуриента = ОбъектРодителя.Адрес;

НовоеЗаявление.EmailРодительАбитуриента = ОбъектРодителя.Email;

НовоеЗаявление.ТелефонРодительАбитуриента = ОбъектРодителя.Телефон;

Для предмет = 0 По Объект.ОценкиАбитуриента.Количество()-1 Цикл

НоваяСтрокаЗаписи = НовоеЗаявление.ОценкиАбитуриента.Добавить();

НоваяСтрокаЗаписи.Предмет = Объект.ОценкиАбитуриента[предмет].Предмет;

НоваяСтрокаЗаписи.Оценка = Объект.ОценкиАбитуриента[предмет].Оценка;

КонецЦикла;

Для документ = 0 По Объект.ДокументыАбитуриента.Количество()-1 Цикл

НоваяСтрокаЗаписи = НовоеЗаявление.ДокументыАбитуриента.Добавить();

НоваяСтрокаЗаписи.Документ = Перечисления.ПеречислениеДокументыАбитуриента.Получить(документ);

НоваяСтрокаЗаписи.Номер = Объект.ДокументыАбитуриента[документ].Номер;

КонецЦикла;

НовоеЗаявление.Дата = ТекущаяДата();

НовоеЗаявление.Записать(РежимЗаписиДокумента.Проведение);

КонецПроцедуры

//Процедуры фильтрации дублей

&наСервере

Процедура ПроверкаОтчетTelegram()

Если Справочники.Уведомления.НайтиПоРеквизиту("ФИО",Объект.Наименование).Пустая() Тогда

Сообщение = "<b>\*Добавление нового абитуриента\*</b>"+Символы.ПС+Символы.ПС+

"<b>Основная информация</b>"+Символы.ПС+

"ФИО: "+Объект.Наименование+Символы.ПС+

"Дата рождения: "+Формат(Объект.ДатаРожденияАбитуриента, "ДФ=dd.MM.yyyy")+Символы.ПС+

"Сред. балл: "+Объект.СреднийБаллАбитуриента+Символы.ПС+

"Специальность: "+Объект.СпециальностьАбитурента+Символы.ПС+Символы.ПС+

"<b>Дополнительная информация</b>"+Символы.ПС+

"Форма оплаты: "+Объект.ФормаОплатыАбитуриента+Символы.ПС+

"Тип аттестата: "+Объект.ТипАттестатаАбитуриента+Символы.ПС+

"Приоритет зачисления: "+Объект.ПриоритетЗачисленияАбитуриента+Символы.ПС+Символы.ПС+

"<b>Контактная информация</b>"+Символы.ПС+

"Родитель: "+Объект.РодительАбитуриента+Символы.ПС+

"Адрес: "+Объект.АдресАбитуриента+Символы.ПС+

"Email: "+Объект.EmailАбитуриента+Символы.ПС+

"Телефон: "+Объект.ТелефонАбитуриента+Символы.ПС;

СоздатьОтчетTelegram(Сообщение,Объект.Наименование,"Добавление нового абитуриента",0);

КонецЕсли

КонецПроцедуры

&НаСервере

Процедура ПроверкаЗаявлениеНаПоступление()

Если Документы.ЗаявлениеНаПоступление.НайтиПоРеквизиту("ФИОАбитуриента", Объект.Наименование).Пустая() Тогда

СоздатьДокументЗаявлениеНаПоступление();

КонецЕсли;

КонецПроцедуры

//Вызов процедур выше после записи

&НаКлиенте

Процедура ПослеЗаписи(ПараметрыЗаписи) Экспорт

ПроверкаОтчетTelegram();

ПроверкаЗаявлениеНаПоступление();

КонецПроцедуры

//Процедуры вывода перечислений на форму

&НаКлиенте

Процедура ПриОткрытии(Отказ)

Счетчик = 0;

Если (Объект.ОценкиАбитуриента.Количество() < ВернутьКолвоПредметов()) Тогда

Пока (Счетчик < ВернутьКолвоПредметов()) Цикл

Новаястрока = Объект.ОценкиАбитуриента.Добавить();

Новаястрока.Предмет = ВернутьЗначениеПредметаПоИндексу(Счетчик);

Счетчик = Счетчик + 1;

КонецЦикла;

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ОценкиАбитуриентаПриИзменении(Элемент)

СуммаЗначений = 0;

КоличествоЗначенийШоражников = 0;

Для пацана = 0 По Объект.ОценкиАбитуриента.Количество()-1 Цикл

Если ЗначениеЗаполнено(Объект.ОценкиАбитуриента[пацана].Оценка) Тогда

ТекущаяОценка = Объект.ОценкиАбитуриента[пацана].Оценка;

Если ((ТекущаяОценка < 2) или (ТекущаяОценка > 5)) Тогда

Сообщить("Некорректный ввод оценки по дисциплине '"+Объект.ОценкиАбитуриента[пацана].Предмет+"'!");

КонецЕсли;

КоличествоЗначенийШоражников = КоличествоЗначенийШоражников + 1;

СуммаЗначений = СуммаЗначений + ТекущаяОценка;

КонецЕсли;

КонецЦикла;

СреднееЗначение = СуммаЗначений / КоличествоЗначенийШоражников;

Объект.СреднийБаллАбитуриента = СреднееЗначение;

КонецПроцедуры

//Обработка картинок

&НаКлиенте

Процедура ДобавитьКартинку(Команда)

ИмяФайла = "";

ПутьФайла = "";

Если ПоместитьФайл(ПутьФайла, , ИмяФайла, Истина) = Истина Тогда

Фото = ПутьФайла;

УстановитьКартинку(ИмяФайла);

КонецЕсли;

КонецПроцедуры

&НаСервере

Процедура УстановитьКартинку(ИмяФайла)

НовыйФайлКартинки = Справочники.ФайлыСистемы.СоздатьЭлемент();

НовыйФайлКартинки.Файл = Новый ХранилищеЗначения(ПолучитьИзВременногоХранилища(Фото));

НовыйФайлКартинки.Наименование = ИмяФайла;

НовыйФайлКартинки.Дата = ТекущаяДата();

Фото = ПоместитьВоВременноеХранилище(НовыйФайлКартинки.Файл.Получить());

НовыйФайлКартинки.Записать();

Если Справочники.Уведомления.НайтиПоРеквизиту("ФИО",НовыйФайлКартинки.Наименование).Пустая() Тогда

Сообщение = "<b>\*Добавление нового файла\*</b>"+Символы.ПС+Символы.ПС+

"Название: "+НовыйФайлКартинки.Наименование+Символы.ПС+

"Дата добавления: "+НовыйФайлКартинки.Дата+Символы.ПС;

СоздатьОтчетTelegram(Сообщение,НовыйФайлКартинки.Наименование,"Добавление нового файла",2);

КонецЕсли;

Объект.ФотоАбитуриента = НовыйФайлКартинки.Ссылка;

КонецПроцедуры

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Фото = ПолучитьНавигационнуюСсылку(Объект.ФотоАбитуриента, "Файл");

Элементы.Фото.РазмерКартинки = РазмерКартинки.АвтоРазмер;

КонецПроцедуры

&НаСервере

Процедура СортировкаАбитуриентовСправочникСервер()

СортировкаАбитуриентовСправочник();

КонецПроцедуры

&НаКлиенте

Процедура Сортировка(Команда)

СортировкаАбитуриентовСправочникСервер();

КонецПроцедуры

&НаКлиенте

Процедура УдалениеАбитуриента(Команда);

//Объект.Наименование

УдалениеАбитуриентаНаСервере(Объект.Наименование);

ЭтаФорма.Закрыть();

КонецПроцедуры

&НаСервере

Процедура УдалениеАбитуриентаНаСервере(ТекущееНаименование)

ДокументОВозвратеДокумнтовОбращение = Документы.ДокументОВозвратеДокументов.НайтиПоРеквизиту("ФИОАбитуриентаВозврат",Объект.Наименование);

ДокументЗаявлениеНаПоступлениеОбращение = Документы.ЗаявлениеНаПоступление.НайтиПоРеквизиту("ФИОАбитуриента",Объект.Наименование);

ТекущийОбъект = Справочники.Абитуриенты.НайтиПоНаименованию(ТекущееНаименование).ПолучитьОбъект();

Если Объект.ПометкаУдаления = Ложь Тогда

Если ДокументОВозвратеДокумнтовОбращение.Пустая() Тогда

НовыйДокументВозврата = Документы.ДокументОВозвратеДокументов.СоздатьДокумент();

НовыйДокументВозврата.Дата = ТекущаяДата();

НовыйДокументВозврата.ФИОАбитуриентаВозврат = Объект.Наименование;

НовыйДокументВозврата.ДатаРожденияВозврат = Объект.ДатаРожденияАбитуриента;

НовыйДокументВозврата.АдресАбитуриентаВозврат = Объект.АдресАбитуриента;

НовыйДокументВозврата.ТелефонАбитуриентаВозврат = Объект.ТелефонАбитуриента;

НовыйДокументВозврата.EmailАбитуриентаВозврат = Объект.EmailАбитуриента;

Для документ = 0 По Объект.ДокументыАбитуриента.Количество()-1 Цикл

НоваяСтрокаЗаписи = НовыйДокументВозврата.СписокДокументов.Добавить();

НоваяСтрокаЗаписи.Документ = Перечисления.ПеречислениеДокументыАбитуриента.Получить(документ);

НоваяСтрокаЗаписи.Номер = Объект.ДокументыАбитуриента[документ].Номер;

КонецЦикла; НовыйДокументВозврата.Записать(РежимЗаписиДокумента.Проведение);

КонецЕсли;

Если Не ДокументЗаявлениеНаПоступлениеОбращение.Пустая() Тогда

ДокументЗаявлениеНаПоступлениеОбращение.ПолучитьОбъект().Удалить();

КонецЕсли;

ТекущийОбъект.ПометкаУдаления = Истина;

ТекущийОбъект.Записать();

Иначе

ТекущийОбъект.ПометкаУдаления = Ложь;

ТекущийОбъект.Записать();

Если Не ДокументОВозвратеДокумнтовОбращение.Пустая() Тогда

ДокументОВозвратеДокумнтовОбращение.ПолучитьОбъект().Удалить();

КонецЕсли;

Если ДокументЗаявлениеНаПоступлениеОбращение.Пустая() Тогда

СоздатьДокументЗаявлениеНаПоступление();

КонецЕсли;

КонецЕсли;

КонецПроцедуры

Листинг 2. Модуль менеджера справочника Абитуриенты

Процедура Печать(ТабДок, Ссылка) Экспорт

//{{\_КОНСТРУКТОР\_ПЕЧАТИ(Печать)

Макет = Справочники.Абитуриенты.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Абитуриенты.Наименование,

| Абитуриенты.ПриоритетЗачисленияАбитуриента,

| Абитуриенты.СпециальностьАбитурента,

| Абитуриенты.СреднийБаллАбитуриента,

| Абитуриенты.ТипАттестатаАбитуриента,

| Абитуриенты.ФормаОплатыАбитуриента,

| Абитуриенты.ОценкиАбитуриента.(

| Предмет,

| Оценка

| )

|ИЗ

| Справочник.Абитуриенты КАК Абитуриенты

|ГДЕ

| Абитуриенты.Ссылка В (&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ОбластьОценкиАбитуриентаШапка = Макет.ПолучитьОбласть("ОценкиАбитуриентаШапка");

ОбластьОценкиАбитуриента = Макет.ПолучитьОбласть("ОценкиАбитуриента");

Подвал = Макет.ПолучитьОбласть("Подвал");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ТабДок.Вывести(ОбластьЗаголовок);

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());

ТабДок.Вывести(ОбластьОценкиАбитуриентаШапка);

ВыборкаОценкиАбитуриента = Выборка.ОценкиАбитуриента.Выбрать();

Пока ВыборкаОценкиАбитуриента.Следующий() Цикл

ОбластьОценкиАбитуриента.Параметры.Заполнить(ВыборкаОценкиАбитуриента);

ТабДок.Вывести(ОбластьОценкиАбитуриента, ВыборкаОценкиАбитуриента.Уровень());

КонецЦикла;

Подвал.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Подвал);

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

//}}

КонецПроцедуры

1. ФайлыСистемы – справочник, содержащий информацию о файлах системы (Рисунок 4), используется справочником «Абитуриенты».  
   Содержит следующие реквизиты:

* Дата – дата добавления файла в систему для большей наглядности
* Файл – непосредственно реквизит, содержащий файл

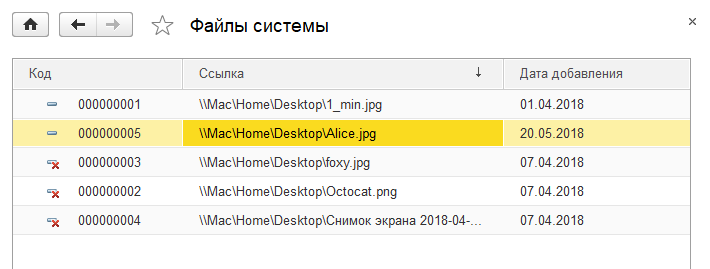


Рисунок 4 – Справочник файлов системы

Листинг 3. Модуль менеджера справочника ФайлыСистемы

Процедура Печать(ТабДок, Ссылка) Экспорт

//{{\_КОНСТРУКТОР\_ПЕЧАТИ(Печать)

Макет = Справочники.ФайлыСистемы.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ФайлыСистемы.Дата,

| ФайлыСистемы.Код,

| ФайлыСистемы.Наименование,

| ФайлыСистемы.Файл

|ИЗ

| Справочник.ФайлыСистемы КАК ФайлыСистемы

|ГДЕ

| ФайлыСистемы.Ссылка В (&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ТабДок.Вывести(ОбластьЗаголовок);

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

//}}

КонецПроцедуры

1. TelegramBot – справочник, содержащий все настройки бота мессенджера Telegram для отправки уведомлений (Рисунки 5-6). Содержит следующие реквизиты:

* Наименование – название конфигурации конкретного бота, по ней происходит обращение к конкретному боту в общем модуле программы (т.е. возможна реализация нескольких ботов для различных событий)
* Бот – реквизит с именем бота для визуализации отчетов и идентификации конкретного бота
* Токен – токен для взаимодействия с BOT API Telegram’а. Выдается при создании бота Telegram у главного бота @BotFather. Автоматически шифруется по алгоритму Base64 при записи элемента справочника.
* Сервер – сервер для обращения методов API Telegram’а. Всегда является api.telegram.org, но при появлении зеркала возможна его горячая замена для обхода каких-либо ограничений.
* Идентификатор назначения – id пользователя или название чата для отправки уведомлений ботом.

Также справочник содержит табличную часть «УведомленияБота» (Рисунок 6), позволяющую активировать оповещения Telegram для конкретного модуля и содержащую два реквизита:

* Модуль – название модуля системы только для визаулизации, обращение общего модуля «ОтчетTelegram» происходит посредством индекса табличной части.
* Значение – булево значение для активации уведомления для конкретного модуля, проверяется общим модулем «ОтчетTelegram». Истина – оповещение включено, ложь – выключено.

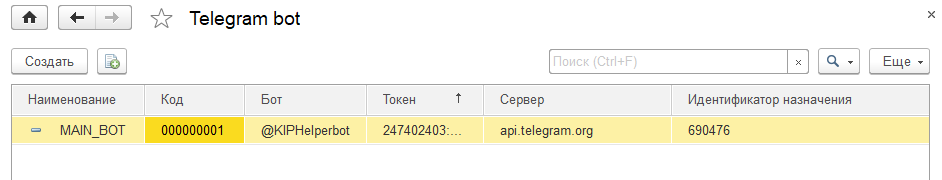


Рисунок 5 – Справочник TelegramBot

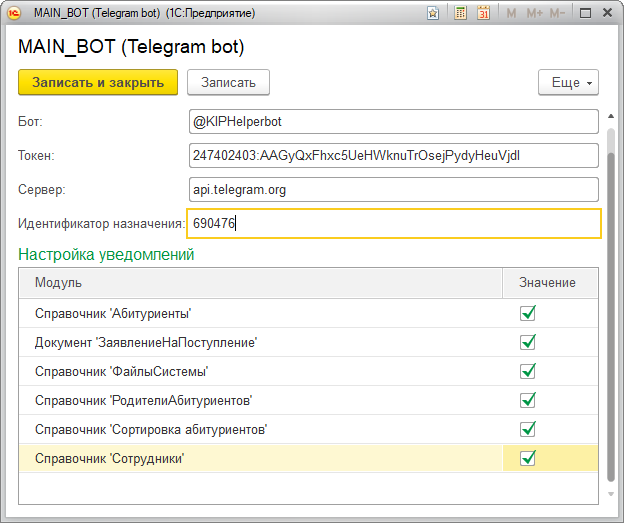


Рисунок 6 – Элемент формы справочника TelegramBot

Листинг 4. Модуль элемента справочника TelegramBot

&НаСервере

Функция СоздатьСтрокуBase64(Строка)

Если (СтрЧислоВхождений(Объект.Бот, ".")) = 0 Тогда

ВременныйФайл = ПолучитьИмяВременногоФайла();

ЗаписьТекста = Новый ЗаписьТекста(ВременныйФайл, "CESU-8");

ЗаписьТекста.Записать(Строка);

ЗаписьТекста.Закрыть();

ДД\_Файла = Новый ДвоичныеДанные(ВременныйФайл);

СтрокаBase64 = Base64Строка(ДД\_Файла);

УдалитьФайлы(ВременныйФайл);

Возврат СтрокаBase64;

Иначе

Возврат Строка;

КонецЕсли;

КонецФункции

&НаКлиенте

Процедура ПередЗаписью(Отказ, ПараметрыЗаписи)

Объект.Токен = СоздатьСтрокуBase64(Объект.Токен);

Объект.Бот = "@KIPHelperbot";

КонецПроцедуры

&НаКлиенте

Процедура УведомленияБотаПриИзменении(Элемент)

Если (СтрЧислоВхождений(Объект.Бот, ".")) = 0

Тогда

Объект.Бот = Объект.Бот+".";

КонецЕсли;

КонецПроцедуры

1. Уведомления – справочник, содержащий информацию об отправленных уведомлениях бота Telegram. Используется для статистики и защиты от дублирования уже отправленного сообщения (Рисунки 7-8).

Содержит следующие реквизиты:

* Сообщение – исходное отправленное сообщение.
* Дата – дата и время отправки исходного сообщения.
* Идентификатор – id пользователя или название группы, кому было отправленно исходное сообщение.
* ФИО – идентификатор события или фио абитуриента, а которым связано данное событие.
* Событие – название события в связи с которым было отправлено сообщение.

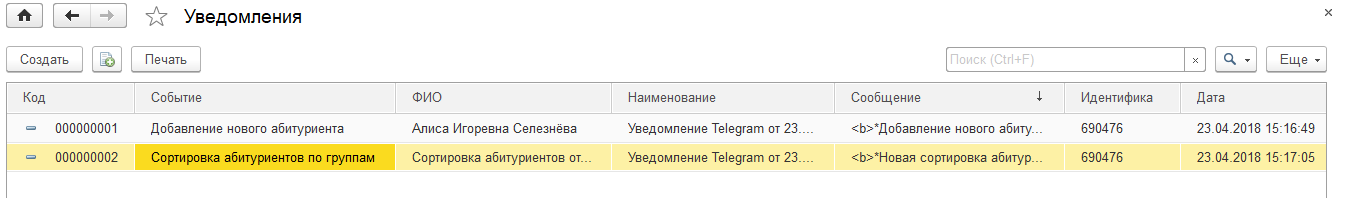


Рисунок 7 – Справочник уведомления

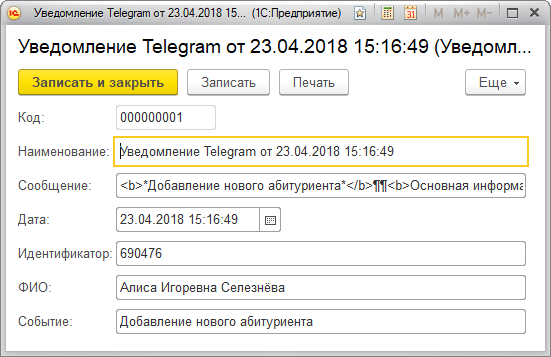


Рисунок 8 – Элемент формы справочника уведомления

Листинг 5. Модуль менеджера справочника Уведомления

Процедура Печать(ТабДок, Ссылка) Экспорт

//{{\_КОНСТРУКТОР\_ПЕЧАТИ(Печать)

Макет = Справочники.Уведомления.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Уведомления.Дата,

| Уведомления.Идентификатор,

| Уведомления.Код,

| Уведомления.Наименование,

| Уведомления.Событие,

| Уведомления.Сообщение,

| Уведомления.ФИО

|ИЗ

| Справочник.Уведомления КАК Уведомления

|ГДЕ

| Уведомления.Ссылка В (&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ТабДок.Вывести(ОбластьЗаголовок);

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

//}}

КонецПроцедуры

1. РодителиАбитуриентов – справочник, содержащий информацию о родителях абитуриента или родственников. На данный справочник ссылается реквизит «РодительАбитуриента» справочника «Абитуриенты» (Рисунок 9).

Содержит следующие реквизиты:

* ГодРождения – дата рождения родителя абитуриента
* Email – электронная почта родителя абитуриента
* Телефон – контактный телефон родителя абитуриента
* Адрес – контактный адрес родителя абитуриента

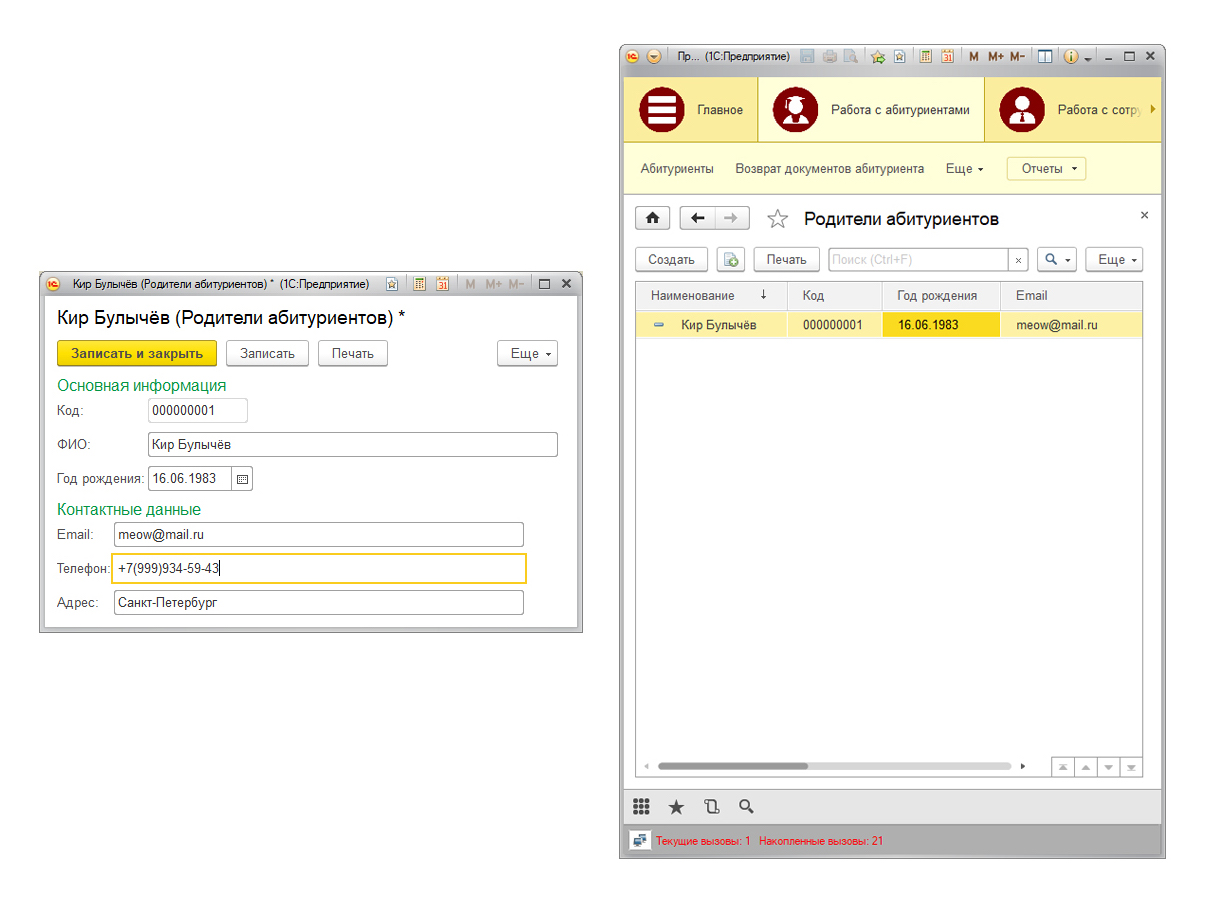


Рисунок 9 – Форма элемента справочника и список элементов справочника РодителиАбитуриентов

Листинг 6. Модуль элемента справочника РодителиАбитуриентов

&наСервере

Процедура ПроверкаОтчетTelegram()

Если Справочники.Уведомления.НайтиПоРеквизиту("ФИО",Объект.Наименование).Пустая() Тогда

Сообщение = "<b>\*Добавление нового родителя абитуриента\*</b>"+Символы.ПС+Символы.ПС+

"<b>Основная информация</b>"+Символы.ПС+

"ФИО: "+Объект.Наименование+Символы.ПС+

"Дата рождения: "+Формат(Объект.ГодРождения, "ДФ=dd.MM.yyyy")+Символы.ПС+Символы.ПС+

"<b>Дополнительная информация</b>"+Символы.ПС+

"E-mail: "+Объект.Email+Символы.ПС+

"Телефон: "+Объект.Телефон+Символы.ПС+

"Адрес: "+Объект.Адрес;

СоздатьОтчетTelegram(Сообщение,Объект.Наименование,"Добавление нового родителя абитуриента",3);

КонецЕсли

КонецПроцедуры

&НаКлиенте

Процедура ПослеЗаписи(ПараметрыЗаписи)

ПроверкаОтчетTelegram();

КонецПроцедуры

Листинг 7. Модуль менеджера справочника РодителиАбитуриентов

Процедура Печать(ТабДок, Ссылка) Экспорт

//{{\_КОНСТРУКТОР\_ПЕЧАТИ(Печать)

Макет = Справочники.РодителиАбитуриентов.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| РодителиАбитуриентов.Email,

| РодителиАбитуриентов.Адрес,

| РодителиАбитуриентов.ГодРождения,

| РодителиАбитуриентов.Код,

| РодителиАбитуриентов.Наименование,

| РодителиАбитуриентов.Телефон

|ИЗ

| Справочник.РодителиАбитуриентов КАК РодителиАбитуриентов

|ГДЕ

| РодителиАбитуриентов.Ссылка В (&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ТабДок.Вывести(ОбластьЗаголовок);

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

//}}

КонецПроцедуры;

1. СортировкаАбитуриентов – справочник, который по своей сути содержит аналогичные элементы списка справочника «Абитуриенты» (Рисунок 10), но является иерархическим (многоуровневым). Формируется программно после нажатия кнопки сортировки на форме элемента справочника «Абитуриенты», а также содержит все реквизиты, аналогичные справочнику «Абитуриенты». При желании параметры сортировки возможно изменять (Рисунок 10). Дополнительную информацию о параметрах сортировки можно посмотреть в описании справочника «НастройкиСортировкиАбитуриентов» ниже.

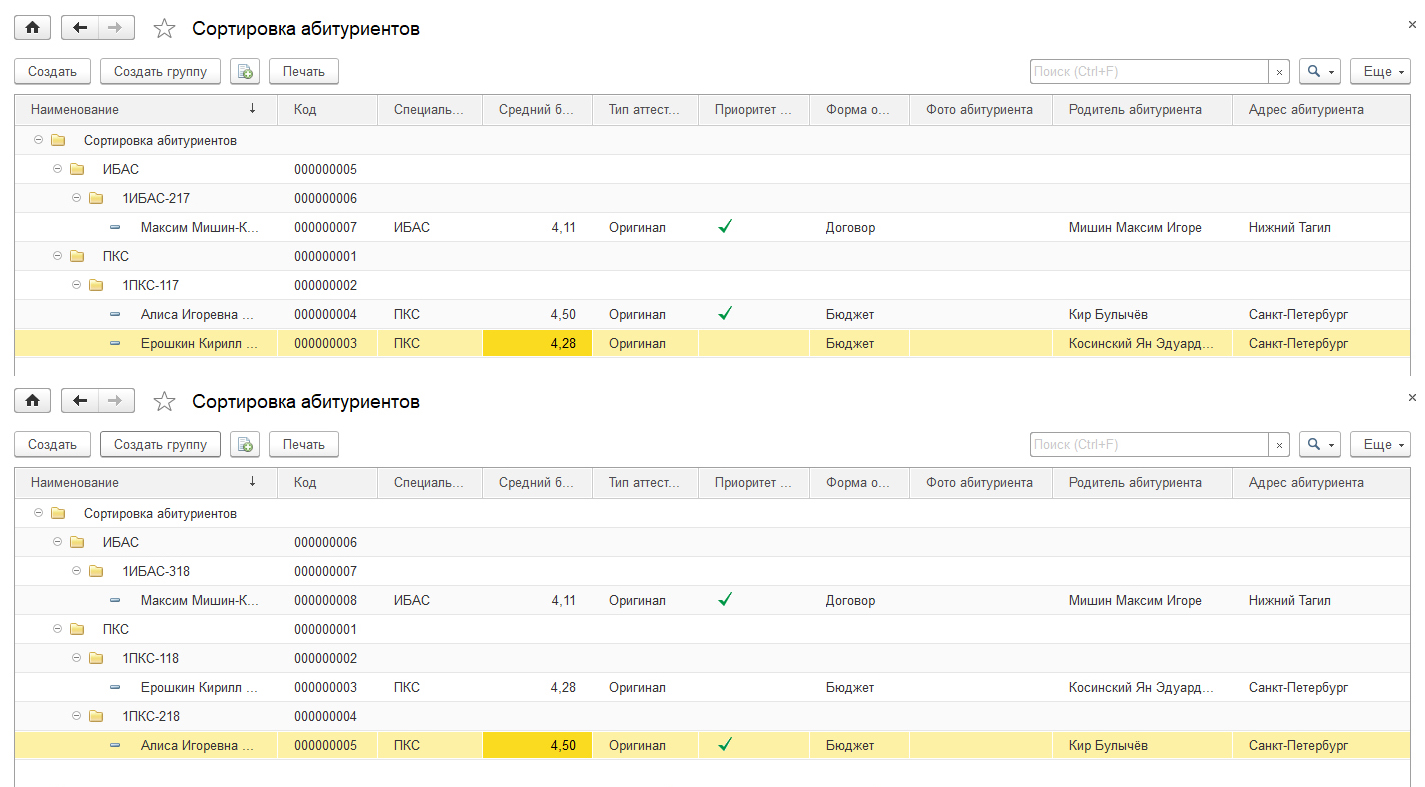


Рисунок 10 – Пример различных параметров сортировки

Листинг 8. Модуль элемента справочника СортировкаАбитуриентов

//Подсчет предметов для табличной части

Функция ВернутьКолвоПредметов()

Возврат Перечисления.ПеречислениеПредметыПоступленияАбитуриентов.Количество();

КонецФункции

Функция ВернутьЗначениеПредметаПоИндексу(индекс\_номер)

Возврат Перечисления.ПеречислениеПредметыПоступленияАбитуриентов.Получить(индекс\_номер);

КонецФункции

//Вызов процедур выше после записи

&НаКлиенте

Процедура ПослеЗаписи(ПараметрыЗаписи) Экспорт

КонецПроцедуры

//Процедуры вывода перечислений на форму

&НаКлиенте

Процедура ПриОткрытии(Отказ)

Счетчик = 0;

Если (Объект.ОценкиАбитуриента.Количество() < ВернутьКолвоПредметов()) Тогда

Пока (Счетчик < ВернутьКолвоПредметов()) Цикл

Новаястрока = Объект.ОценкиАбитуриента.Добавить();

Новаястрока.Предмет = ВернутьЗначениеПредметаПоИндексу(Счетчик);

Счетчик = Счетчик + 1;

КонецЦикла;

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ОценкиАбитуриентаПриИзменении(Элемент)

СуммаЗначений = 0;

КоличествоЗначенийШоражников = 0;

Для пацана = 0 По Объект.ОценкиАбитуриента.Количество()-1 Цикл

Если ЗначениеЗаполнено(Объект.ОценкиАбитуриента[пацана].Оценка) Тогда

ТекущаяОценка = Объект.ОценкиАбитуриента[пацана].Оценка;

Если ((ТекущаяОценка < 2) или (ТекущаяОценка > 5)) Тогда

Сообщить("Некорректный ввод оценки по дисциплине '"+Объект.ОценкиАбитуриента[пацана].Предмет+"'!");

КонецЕсли;

КоличествоЗначенийШоражников = КоличествоЗначенийШоражников + 1;

СуммаЗначений = СуммаЗначений + ТекущаяОценка;

КонецЕсли;

КонецЦикла;

СреднееЗначение = СуммаЗначений / КоличествоЗначенийШоражников;

Объект.СреднийБаллАбитуриента = СреднееЗначение;

КонецПроцедуры

//Обработка картинок

&НаКлиенте

Процедура ДобавитьКартинку(Команда)

ИмяФайла = "";

ПутьФайла = "";

Если ПоместитьФайл(ПутьФайла, , ИмяФайла, Истина) = Истина Тогда

Фото = ПутьФайла;

УстановитьКартинку(ИмяФайла);

КонецЕсли;

КонецПроцедуры

&НаСервере

Процедура УстановитьКартинку(ИмяФайла)

НовыйФайлКартинки = Справочники.ФайлыСистемы.СоздатьЭлемент();

НовыйФайлКартинки.Файл = Новый ХранилищеЗначения(ПолучитьИзВременногоХранилища(Фото));

НовыйФайлКартинки.Наименование = ИмяФайла;

НовыйФайлКартинки.Дата = ТекущаяДата();

Фото = ПоместитьВоВременноеХранилище(НовыйФайлКартинки.Файл.Получить());

НовыйФайлКартинки.Записать();

Объект.ФотоАбитуриента = НовыйФайлКартинки.Ссылка;

КонецПроцедуры

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Фото = ПолучитьНавигационнуюСсылку(Объект.ФотоАбитуриента, "Файл");

Элементы.Фото.РазмерКартинки = РазмерКартинки.АвтоРазмер;

КонецПроцедуры

Листинг 9. Модуль менеджера справочника СортировкаАбитуриентов

Процедура Печать(ТабДок, Ссылка) Экспорт

//{{\_КОНСТРУКТОР\_ПЕЧАТИ(Печать)

Макет = Справочники.СортировкаАбитуриентов.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| СортировкаАбитуриентов.EmailАбитуриента,

| СортировкаАбитуриентов.АдресАбитуриента,

| СортировкаАбитуриентов.ДатаРожденияАбитуриента,

| СортировкаАбитуриентов.Код,

| СортировкаАбитуриентов.Наименование,

| СортировкаАбитуриентов.ПриоритетЗачисленияАбитуриента,

| СортировкаАбитуриентов.РодительАбитуриента,

| СортировкаАбитуриентов.СпециальностьАбитурента,

| СортировкаАбитуриентов.СреднийБаллАбитуриента,

| СортировкаАбитуриентов.ТелефонАбитуриента,

| СортировкаАбитуриентов.ТипАттестатаАбитуриента,

| СортировкаАбитуриентов.ФормаОплатыАбитуриента,

| СортировкаАбитуриентов.ОценкиАбитуриента.(

| Предмет,

| Оценка

| ),

| СортировкаАбитуриентов.ДокументыАбитуриента.(

| Документ,

| Номер

| )

|ИЗ

| Справочник.СортировкаАбитуриентов КАК СортировкаАбитуриентов

|ГДЕ

| СортировкаАбитуриентов.Ссылка В (&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ОбластьОценкиАбитуриентаШапка = Макет.ПолучитьОбласть("ОценкиАбитуриентаШапка");

ОбластьОценкиАбитуриента = Макет.ПолучитьОбласть("ОценкиАбитуриента");

ОбластьДокументыАбитуриентаШапка = Макет.ПолучитьОбласть("ДокументыАбитуриентаШапка");

ОбластьДокументыАбитуриента = Макет.ПолучитьОбласть("ДокументыАбитуриента");

Подвал = Макет.ПолучитьОбласть("Подвал");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ТабДок.Вывести(ОбластьЗаголовок);

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());

ТабДок.Вывести(ОбластьОценкиАбитуриентаШапка);

ВыборкаОценкиАбитуриента = Выборка.ОценкиАбитуриента.Выбрать();

Пока ВыборкаОценкиАбитуриента.Следующий() Цикл

ОбластьОценкиАбитуриента.Параметры.Заполнить(ВыборкаОценкиАбитуриента);

ТабДок.Вывести(ОбластьОценкиАбитуриента, ВыборкаОценкиАбитуриента.Уровень());

КонецЦикла;

ТабДок.Вывести(ОбластьДокументыАбитуриентаШапка);

ВыборкаДокументыАбитуриента = Выборка.ДокументыАбитуриента.Выбрать();

Пока ВыборкаДокументыАбитуриента.Следующий() Цикл

ОбластьДокументыАбитуриента.Параметры.Заполнить(ВыборкаДокументыАбитуриента);

ТабДок.Вывести(ОбластьДокументыАбитуриента, ВыборкаДокументыАбитуриента.Уровень());

КонецЦикла;

Подвал.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Подвал);

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

//}}

КонецПроцедуры

1. НастройкиСортировкиАбитуриентов – справочник, содержащий основные настройки сортировки абитуриентов по группам (Рисунок 11).

Содержит следующие реквизиты:

* СсылкаНаСайтСЖурналами – ссылка на сайт с журналами для удобного получения списка группы и вслеследствии их успеваемости. Используется для уведомления Telegram (Рисунок 11).
* КоличествоСтудентовВГруппе – максимальное количество студентов в группе. Если больше – формируется следующая группа (Рисунок 10).
* ГодГруппы – год поступления группы. Используется для идентификации группы (например, 1ПКС-117, 1ПКС-217 год группы 17) (Рисунок 10).

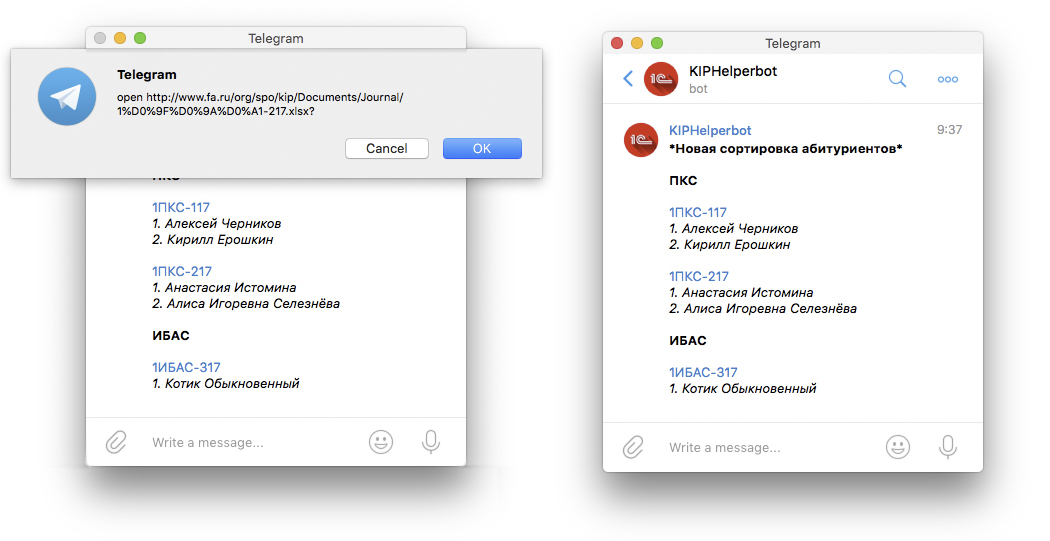


Рисунок 11 – Демонстрация использования регистра «СсылкаНаСайтСЖурналами»

1. Сотрудники – справочник, содержащий информацию о сотрудниках приемной комиссии. Является ссылкой для параметра сеанса «ТекущийПользователь» (Рисунок 12).  
   Содержит следующие реквизиты:

* ДатаРожденияСотрудника
* EmailСотрудника
* ТелефонСотрудника

Также существует табличная часть об образовании сотрудника. Содержит следующие поля:

* Учреждение – полное название учреждения обучения сотрудника
* ДатаПоступления – дата начала обучения сотрудника
* ДатаОкончания – дата окончания обучения сотрудника
* Специальность – специальность по которой обучался сотрудник

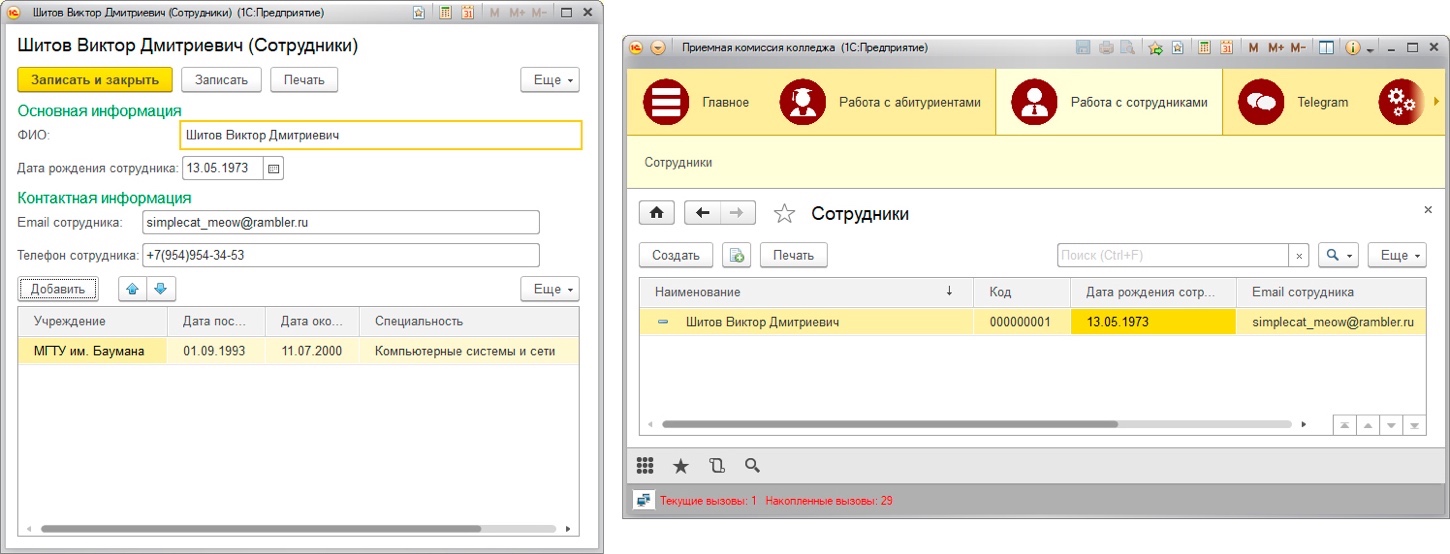


Рисунок 12 – Форма элемента справочника и список элементов справочника Сотрудники

Листинг 10. Модуль элемента справочника Сотрудники

//Процедура формирования отчета Telegram

&наСервере

Процедура ПроверкаОтчетTelegram()

Если Справочники.Уведомления.НайтиПоРеквизиту("ФИО",Объект.Наименование).Пустая() Тогда

СтрокаОбразования = "";

Для НомерОбразования = 0 По Объект.ТабличнаяЧасть.Количество()-1 Цикл

СтрокаОбразования = СтрокаОбразования + "<b>" + строка(НомерОбразования+1) +".</b> " +

"Учреждение: " + Объект.ТабличнаяЧасть[НомерОбразования].Учреждение + Символы.ПС+

"Специальность: " + Объект.ТабличнаяЧасть[НомерОбразования].Специальность + Символы.ПС+

"Дата поступления: " + Формат(Объект.ТабличнаяЧасть[НомерОбразования].ДатаПоступления, "ДФ=dd.MM.yyyy") + Символы.ПС+

"Дата окончания: " + Формат(Объект.ТабличнаяЧасть[НомерОбразования].ДатаОкончания, "ДФ=dd.MM.yyyy") + Символы.ПС + Символы.ПС;

КонецЦикла;

Сообщение = "<b>\*Добавление нового сотрудника\*</b>"+Символы.ПС+Символы.ПС+

"<b>Основная информация</b>"+Символы.ПС+

"ФИО: "+Объект.Наименование+Символы.ПС+

"Дата рождения: "+Формат(Объект.ДатаРожденияСотрудника, "ДФ=dd.MM.yyyy")+Символы.ПС+Символы.ПС+

"<b>Контактная информация</b>"+Символы.ПС+

"E-mail: " +Объект.EmailСотрудника+Символы.ПС+

"Телефон: "+Объект.ТелефонСотрудника+Символы.ПС+Символы.ПС+

"<b>Образование</b>"+Символы.ПС + СтрокаОбразования;

СоздатьОтчетTelegram(Сообщение,Объект.Наименование,"Добавление нового сотрудника",5);

КонецЕсли

КонецПроцедуры

//СоздатьОтчетСотрудниковвTelegram

&НаКлиенте

Процедура ПослеЗаписи(ПараметрыЗаписи)

ПроверкаОтчетTelegram();

КонецПроцедуры

Листинг 11. Модуль менеджера справочника Сотрудники

Процедура Печать(ТабДок, Ссылка) Экспорт

//{{\_КОНСТРУКТОР\_ПЕЧАТИ(Печать)

Макет = Справочники.Сотрудники.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Сотрудники.EmailСотрудника,

| Сотрудники.ДатаРожденияСотрудника,

| Сотрудники.Код,

| Сотрудники.Наименование,

| Сотрудники.ТелефонСотрудника,

| Сотрудники.ТабличнаяЧасть.(

| Учреждение,

| ДатаПоступления,

| ДатаОкончания,

| Специальность

| )

|ИЗ

| Справочник.Сотрудники КАК Сотрудники

|ГДЕ

| Сотрудники.Ссылка В (&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ОбластьТабличнаяЧастьШапка = Макет.ПолучитьОбласть("ТабличнаяЧастьШапка");

ОбластьТабличнаяЧасть = Макет.ПолучитьОбласть("ТабличнаяЧасть");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ТабДок.Вывести(ОбластьЗаголовок);

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());

ТабДок.Вывести(ОбластьТабличнаяЧастьШапка);

ВыборкаТабличнаяЧасть = Выборка.ТабличнаяЧасть.Выбрать();

Пока ВыборкаТабличнаяЧасть.Следующий() Цикл

ОбластьТабличнаяЧасть.Параметры.Заполнить(ВыборкаТабличнаяЧасть);

ТабДок.Вывести(ОбластьТабличнаяЧасть, ВыборкаТабличнаяЧасть.Уровень());

КонецЦикла;

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

//}}

КонецПроцедуры

Документы, которые содержит информационная база:

1. ЗаявлениеНаПоступление – документ, содержащий заявления на поступления абитуриентов, содержит реквизиты аналогичные справочнику «Абитуриенты», а также реквизиты справочника «РодителиАбитуриентов» (Рисунки 13-14). Возможно два типа заполнения:

* Программное заполнение по справочнику (см. описание кнопки «Сортировка» справочника «Абитуриенты» на стр 13)
* Заполненние элемента вручную и формирование элементов справочников «Абитуриенты» и «РодителиАбитуриентов» из реквизитов элемента данного документа.

При проведении документа происходят движения по регистрам накопления «РегистрСтатистикиАбитуриентов» и «РегистрРейтингаАбитуриентов», которые в свою очередь используются для формирования отчетов «ОтчетСтатистикиАбитуриентов» и «ОтчетОРейтингеАбитуриентов» соответственно.

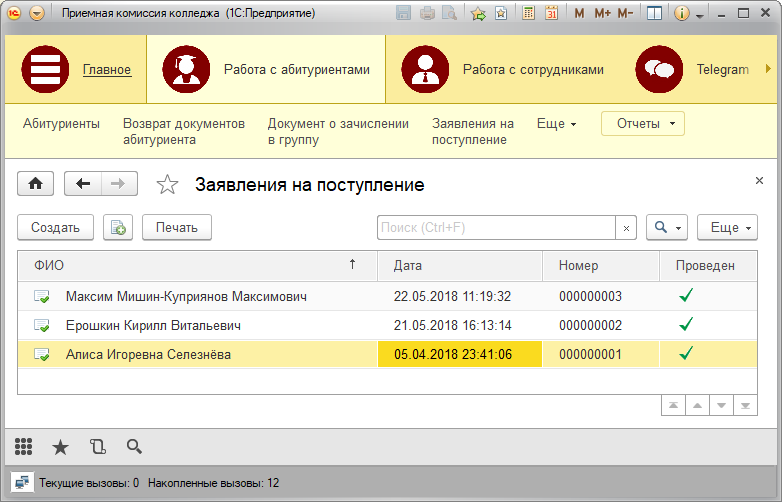


Рисунок 13 – Список элементов документа ЗаявлениеНаПоступление

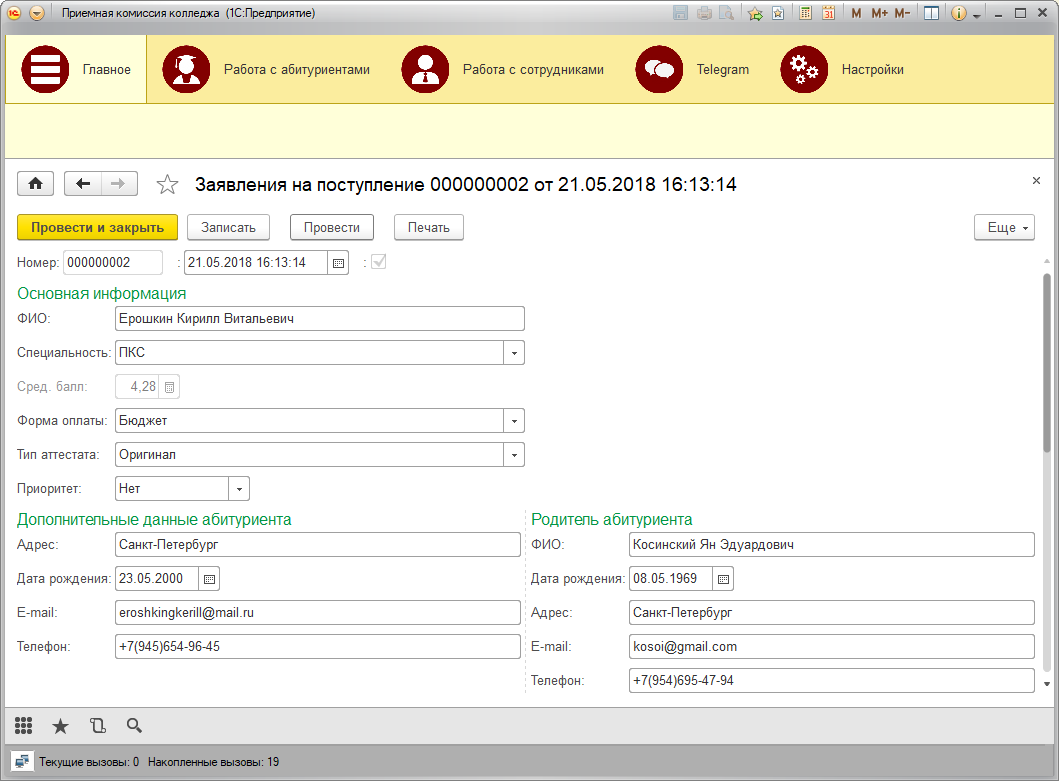


Рисунок 14 – Форма элемента документа ЗаявлениеНаПоступление

Листинг 12. Модуль элемента документа ЗаявлениеНаПоступление

//Подсчет предметов для табличной части

Функция ВернутьКолвоПредметов()

Возврат Перечисления.ПеречислениеПредметыПоступленияАбитуриентов.Количество();

КонецФункции

Функция ВернутьКолвоДокументов()

Возврат Перечисления.ПеречислениеДокументыАбитуриента.Количество();

КонецФункции

Функция ВернутьЗначениеПредметаПоИндексу(индекс\_номер)

Возврат Перечисления.ПеречислениеПредметыПоступленияАбитуриентов.Получить(индекс\_номер);

КонецФункции

Функция ВернутьЗначениеДокументаПоИндексу(индекс\_номер)

Возврат Перечисления.ПеречислениеДокументыАбитуриента.Получить(индекс\_номер);

КонецФункции

//Отображение табличной части

&НаКлиенте

Процедура ПриОткрытии(Отказ)

СчетчикОценок = 0;

Если (Объект.ОценкиАбитуриента.Количество() < ВернутьКолвоПредметов()) Тогда

Пока (СчетчикОценок < ВернутьКолвоПредметов()) Цикл

НоваяCтрока = Объект.ОценкиАбитуриента.Добавить();

НоваяCтрока.Предмет = ВернутьЗначениеПредметаПоИндексу(СчетчикОценок);

СчетчикОценок = СчетчикОценок + 1;

КонецЦикла;

КонецЕсли;

СчетчикДокументов = 0;

Если (Объект.ДокументыАбитуриента.Количество() < ВернутьКолвоДокументов()) Тогда

Пока (СчетчикДокументов < ВернутьКолвоДокументов()) Цикл

НоваяCтрока = Объект.ДокументыАбитуриента.Добавить();

НоваяCтрока.Документ = ВернутьЗначениеДокументаПоИндексу(СчетчикДокументов);

СчетчикДокументов = СчетчикДокументов + 1;

КонецЦикла;

КонецЕсли

КонецПроцедуры

//Высчитваем сред. балл

&НаКлиенте

Процедура ОценкиАбитуриентаОценкаПриИзменении(Элемент)

СуммаЗначений = 0;

КоличествоЗначений = 0;

Для пацана = 0 По Объект.ОценкиАбитуриента.Количество()-1 Цикл

Если ЗначениеЗаполнено(Объект.ОценкиАбитуриента[пацана].Оценка) Тогда

ТекущаяОценка = Объект.ОценкиАбитуриента[пацана].Оценка;

Если ((ТекущаяОценка < 2) или (ТекущаяОценка > 5)) Тогда

Сообщить("Некорректный ввод оценки по дисциплине '"+Объект.ОценкиАбитуриента[пацана].Предмет+"'!");

КонецЕсли;

КоличествоЗначений = КоличествоЗначений + 1;

СуммаЗначений = СуммаЗначений + ТекущаяОценка;

КонецЕсли;

КонецЦикла;

СреднееЗначение = СуммаЗначений / КоличествоЗначений;

Объект.СреднийБаллАбитуриента = СреднееЗначение;

КонецПроцедуры

//Запись данных в справочники

&НаСервере

Процедура ЗаписьВСправочникиПослеЗаписи()

Если Справочники.РодителиАбитуриентов.НайтиПоНаименованию(Объект.ФИОРодительАбитуриента).Пустая() Тогда

//Работа со справочником РодителиАбитуриентов

НоваяЗаписьРодителя = Справочники.РодителиАбитуриентов.СоздатьЭлемент();

НоваяЗаписьРодителя.Наименование = Объект.ФИОРодительАбитуриента;

НоваяЗаписьРодителя.ГодРождения = Объект.ДатаРожденияРодительАбитуриента;

НоваяЗаписьРодителя.Email = Объект.EmailРодительАбитуриента;

НоваяЗаписьРодителя.Телефон = Объект.ТелефонРодительАбитуриента;

НоваяЗаписьРодителя.Адрес = Объект.АдресРодительАбитуриента;

НоваяЗаписьРодителя.Записать();

//Дубликат функции ПроверкаОтчетTelegram т.к. не хочу выносить в общий модуль

Если Справочники.Уведомления.НайтиПоРеквизиту("ФИО",Объект.ФИОРодительАбитуриента).Пустая() Тогда

Сообщение = "<b>\*Добавление нового родителя абитуриента\*</b>"+Символы.ПС+Символы.ПС+

"<b>Основная информация</b>"+Символы.ПС+

"ФИО: "+Объект.ФИОРодительАбитуриента+Символы.ПС+

"Дата рождения: "+Формат(Объект.ДатаРожденияРодительАбитуриента, "ДФ=dd.MM.yyyy")+Символы.ПС+Символы.ПС+

"<b>Дополнительная информация</b>"+Символы.ПС+

"E-mail: "+Объект.EmailРодительАбитуриента+Символы.ПС+

"Телефон: "+Объект.ТелефонРодительАбитуриента+Символы.ПС+

"Адрес: "+Объект.АдресРодительАбитуриента;

СоздатьОтчетTelegram(Сообщение,Объект.ФИОРодительАбитуриента,"Добавление нового родителя абитуриента",1);

КонецЕсли

КонецЕсли;

Если Справочники.Абитуриенты.НайтиПоНаименованию(Объект.ФИОАбитуриента).Пустая() Тогда

//Работа со справочником Абитуриенты

НоваяЗаписьАбитуриента = Справочники.Абитуриенты.СоздатьЭлемент();

НоваяЗаписьАбитуриента.Наименование = Объект.ФИОАбитуриента;

НоваяЗаписьАбитуриента.СреднийБаллАбитуриента = Объект.СреднийБаллАбитуриента;

НоваяЗаписьАбитуриента.СпециальностьАбитурента = Объект.СпециальностьАбитуриента;

НоваяЗаписьАбитуриента.ФормаОплатыАбитуриента = Объект.ФормаОплатыАбитуриента;

НоваяЗаписьАбитуриента.ТипАттестатаАбитуриента = Объект.ТипАттестатаАбитуриента;

НоваяЗаписьАбитуриента.ПриоритетЗачисленияАбитуриента = Объект.ПриоритетЗачисленияАбитуриента;

НоваяЗаписьАбитуриента.РодительАбитуриента = Справочники.РодителиАбитуриентов.НайтиПоНаименованию(Объект.ФИОРодительАбитуриента);

НоваяЗаписьАбитуриента.АдресАбитуриента = Объект.АдресАбитуриента;

НоваяЗаписьАбитуриента.EmailАбитуриента = Объект.EmailАбитуриента;

НоваяЗаписьАбитуриента.ТелефонАбитуриента = Объект.ТелефонАбитуриента;

НоваяЗаписьАбитуриента.ДатаРожденияАбитуриента = Объект.ДатаРожденияАбитуриента;

Для предмет = 0 По Объект.ОценкиАбитуриента.Количество()-1 Цикл

НоваяСтрокаЗаписи = НоваяЗаписьАбитуриента.ОценкиАбитуриента.Добавить();

НоваяСтрокаЗаписи.Предмет = Объект.ОценкиАбитуриента[предмет].Предмет;

НоваяСтрокаЗаписи.Оценка = Объект.ОценкиАбитуриента[предмет].Оценка;

КонецЦикла;

Для документ = 0 По Объект.ДокументыАбитуриента.Количество()-1 Цикл

НоваяСтрокаЗаписи = НоваяЗаписьАбитуриента.ДокументыАбитуриента.Добавить();

НоваяСтрокаЗаписи.Документ = Перечисления.ПеречислениеДокументыАбитуриента.Получить(документ);

НоваяСтрокаЗаписи.Номер = Объект.ДокументыАбитуриента[документ].Номер;

КонецЦикла;

НоваяЗаписьАбитуриента.Записать();

//Дубликат функции ПроверкаОтчетTelegram т.к. не хочу выносить в общий модуль

Если Справочники.Уведомления.НайтиПоРеквизиту("ФИО",Объект.ФИОАбитуриента).Пустая() Тогда

Сообщение = "<b>\*Добавление нового абитуриента\*</b>"+Символы.ПС+Символы.ПС+

"<b>Основная информация</b>"+Символы.ПС+

"ФИО: "+Объект.ФИОАбитуриента+Символы.ПС+

"Дата рождения: "+Формат(Объект.ДатаРожденияАбитуриента, "ДФ=dd.MM.yyyy")+Символы.ПС+

"Сред. балл: "+Объект.СреднийБаллАбитуриента+Символы.ПС+

"Специальность: "+Объект.СпециальностьАбитуриента+Символы.ПС+Символы.ПС+

"<b>Дополнительная информация</b>"+Символы.ПС+

"Форма оплаты: "+Объект.ФормаОплатыАбитуриента+Символы.ПС+

"Тип аттестата: "+Объект.ТипАттестатаАбитуриента+Символы.ПС+

"Приоритет зачисления: "+Объект.ПриоритетЗачисленияАбитуриента+Символы.ПС+Символы.ПС+

"<b>Контактная информация</b>"+Символы.ПС+

"Родитель: "+Объект.ФИОРодительАбитуриента+Символы.ПС+

"Адрес: "+Объект.АдресАбитуриента+Символы.ПС+

"Email: "+Объект.EmailАбитуриента+Символы.ПС+

"Телефон: "+Объект.ТелефонАбитуриента+Символы.ПС;

СоздатьОтчетTelegram(Сообщение,Объект.ФИОАбитуриента,"Добавление нового абитуриента",1);

КонецЕсли

КонецЕсли;

КонецПроцедуры

//После записи формы

&НаКлиенте

Процедура ПослеЗаписи(ПараметрыЗаписи)

ЗаписьВСправочникиПослеЗаписи();

КонецПроцедуры

Листинг 13. Модуль объекта документа ЗаявлениеНаПоступление

Процедура ОбработкаПроведения(Отказ, Режим)

//{{\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

// Данный фрагмент построен конструктором.

// При повторном использовании конструктора, внесенные вручную изменения будут утеряны!!!

// регистр РегистрРейтингаАбитуриентов

Движения.РегистрРейтингаАбитуриентов.Записывать = Истина;

Движение = Движения.РегистрРейтингаАбитуриентов.Добавить();

Движение.Период = Дата;

Движение.ФИО = ФИОАбитуриента;

Движение.Специальность = СпециальностьАбитуриента;

Движение.СредБалл = СреднийБаллАбитуриента;

// регистр РегистрСтатистикиАбитуриентов Приход

Движения.РегистрСтатистикиАбитуриентов.Записывать = Истина;

Движение = Движения.РегистрСтатистикиАбитуриентов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение.Период = Дата;

Если (СпециальностьАбитуриента = Перечисления.ПеречислениеСпециальностейАбитуринта.ПКС) Тогда

Движение.КолвоСпециальностиПКС = Движение.КолвоСпециальностиПКС + 1;

Иначе

Движение.КолвоСпециальностиПКС = Движение.КолвоСпециальностиПКС;

КонецЕсли;

Если (СпециальностьАбитуриента = Перечисления.ПеречислениеСпециальностейАбитуринта.ИБАС) Тогда

Движение.КолвоСпециальностиИБАС = Движение.КолвоСпециальностиИБАС + 1;

Иначе

Движение.КолвоСпециальностиИБАС = Движение.КолвоСпециальностиИБАС;

КонецЕсли;

Если (ПриоритетЗачисленияАбитуриента = Истина) Тогда

Движение.КолвоАбитуриентовСПриоритетом = Движение.КолвоАбитуриентовСПриоритетом + 1;

Иначе

Движение.КолвоАбитуриентовСПриоритетом = Движение.КолвоАбитуриентовСПриоритетом;

КонецЕсли;

Если (ПриоритетЗачисленияАбитуриента = Ложь) Тогда

Движение.КолвоАбитуриентовБезПриоритета = Движение.КолвоАбитуриентовБезПриоритета + 1;

Иначе

Движение.КолвоАбитуриентовБезПриоритета = Движение.КолвоАбитуриентовБезПриоритета;

КонецЕсли;

Если (ФормаОплатыАбитуриента = Перечисления.ПеречислениеФормыОплатыАбитуриента.Бюджет) Тогда

Движение.КолвоФормыОплатыБюджет = Движение.КолвоФормыОплатыБюджет + 1;

Иначе

Движение.КолвоФормыОплатыБюджет = Движение.КолвоФормыОплатыБюджет;

КонецЕсли;

Если (ФормаОплатыАбитуриента = Перечисления.ПеречислениеФормыОплатыАбитуриента.Договор) Тогда

Движение.КолвоФормыОплатыДоговор = Движение.КолвоФормыОплатыДоговор + 1;

Иначе

Движение.КолвоФормыОплатыДоговор = Движение.КолвоФормыОплатыДоговор;

КонецЕсли;

//}}\_\_КОНСТРУКТОР\_ДВИЖЕНИЙ\_РЕГИСТРОВ

КонецПроцедуры

Листинг 14. Модуль менеджера документа ЗаявлениеНаПоступление

Процедура Печать(ТабДок, Ссылка) Экспорт

//{{\_КОНСТРУКТОР\_ПЕЧАТИ(Печать)

Макет = Документы.ЗаявлениеНаПоступление.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЗаявлениеНаПоступление.EmailАбитуриента,

| ЗаявлениеНаПоступление.EmailРодительАбитуриента,

| ЗаявлениеНаПоступление.АдресАбитуриента,

| ЗаявлениеНаПоступление.АдресРодительАбитуриента,

| ЗаявлениеНаПоступление.Дата,

| ЗаявлениеНаПоступление.ДатаРожденияАбитуриента,

| ЗаявлениеНаПоступление.ДатаРожденияРодительАбитуриента,

| ЗаявлениеНаПоступление.Номер,

| ЗаявлениеНаПоступление.ПриоритетЗачисленияАбитуриента,

| ЗаявлениеНаПоступление.СпециальностьАбитуриента,

| ЗаявлениеНаПоступление.СреднийБаллАбитуриента,

| ЗаявлениеНаПоступление.ТелефонАбитуриента,

| ЗаявлениеНаПоступление.ТелефонРодительАбитуриента,

| ЗаявлениеНаПоступление.ТипАттестатаАбитуриента,

| ЗаявлениеНаПоступление.ФИОАбитуриента,

| ЗаявлениеНаПоступление.ФИОРодительАбитуриента,

| ЗаявлениеНаПоступление.ФормаОплатыАбитуриента,

| ЗаявлениеНаПоступление.ОценкиАбитуриента.(

| Предмет,

| Оценка

| ),

| ЗаявлениеНаПоступление.ДокументыАбитуриента.(

| Документ,

| Номер

| )

|ИЗ

| Документ.ЗаявлениеНаПоступление КАК ЗаявлениеНаПоступление

|ГДЕ

| ЗаявлениеНаПоступление.Ссылка В (&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ОбластьОценкиАбитуриентаШапка = Макет.ПолучитьОбласть("ОценкиАбитуриентаШапка");

ОбластьОценкиАбитуриента = Макет.ПолучитьОбласть("ОценкиАбитуриента");

ОбластьДокументыАбитуриентаШапка = Макет.ПолучитьОбласть("ДокументыАбитуриентаШапка");

ОбластьДокументыАбитуриента = Макет.ПолучитьОбласть("ДокументыАбитуриента");

Подвал = Макет.ПолучитьОбласть("Подвал");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ТабДок.Вывести(ОбластьЗаголовок);

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());

ТабДок.Вывести(ОбластьОценкиАбитуриентаШапка);

ВыборкаОценкиАбитуриента = Выборка.ОценкиАбитуриента.Выбрать();

Пока ВыборкаОценкиАбитуриента.Следующий() Цикл

ОбластьОценкиАбитуриента.Параметры.Заполнить(ВыборкаОценкиАбитуриента);

ТабДок.Вывести(ОбластьОценкиАбитуриента, ВыборкаОценкиАбитуриента.Уровень());

КонецЦикла;

ТабДок.Вывести(ОбластьДокументыАбитуриентаШапка);

ВыборкаДокументыАбитуриента = Выборка.ДокументыАбитуриента.Выбрать();

Пока ВыборкаДокументыАбитуриента.Следующий() Цикл

ОбластьДокументыАбитуриента.Параметры.Заполнить(ВыборкаДокументыАбитуриента);

ТабДок.Вывести(ОбластьДокументыАбитуриента, ВыборкаДокументыАбитуриента.Уровень());

КонецЦикла;

Подвал.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Подвал);

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

//}}

КонецПроцедуры

1. ДокументОЗачисленииВГруппу – документ, содержащий информацию о распределении абитуриентов группы, формируется только автоматически (см. описание кнопки «Сортировка» справочника «Абитуриенты» на стр 13). При проведении документа происходят движения по регистру «РегистрОтчетаОПоступленииВГруппу», который служит основой для формирования отчета «ОтчетОПоступленииВГруппу» (Рисунок 15) Содержит следующие реквизиты:

* СпециальностьАбитурента
* СреднийБаллАбитуриента
* ФормаОплатыАбитуриента
* ДатаРожденияАбитуриента
* ГруппаАбитуриента
* ФИОАбитуриента

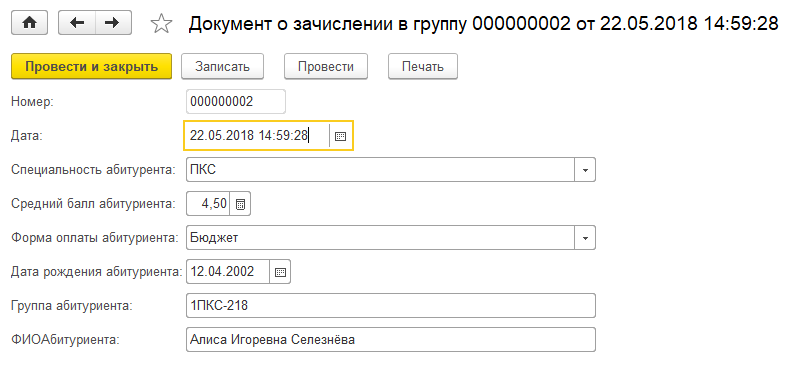


Рисунок 15 – Форма элемента документа ДокументОЗачисленииВГруппу

Листинг 15. Модуль объекта документа ДокументОЗачисленииВГруппу

Процедура ОбработкаПроведения(Отказ, Режим)

// регистр РегистрОтчетаОПоступленииВГруппу

Движения.РегистрОтчетаОПоступленииВГруппу.Записывать = Истина;

Движение = Движения.РегистрОтчетаОПоступленииВГруппу.Добавить();

Движение.Период = Дата;

Движение.Группа = ГруппаАбитуриента;

Движение.Специальность = СпециальностьАбитурента;

Движение.СредБалл = СреднийБаллАбитуриента;

Движение.ФИО = ФИОАбитуриента;

Движение.ДатаРождения = ДатаРожденияАбитуриента;

КонецПроцедуры

Листинг 16. Модуль менеджера документа ДокументОЗачисленииВГруппу

Процедура Печать(ТабДок, Ссылка) Экспорт

//{{\_КОНСТРУКТОР\_ПЕЧАТИ(Печать)

Макет = Документы.ДокументОЗачисленииВГруппу.ПолучитьМакет("Печать");

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ДокументОЗачисленииВГруппу.ГруппаАбитуриента,

| ДокументОЗачисленииВГруппу.Дата,

| ДокументОЗачисленииВГруппу.ДатаРожденияАбитуриента,

| ДокументОЗачисленииВГруппу.Номер,

| ДокументОЗачисленииВГруппу.СпециальностьАбитурента,

| ДокументОЗачисленииВГруппу.ФИОАбитуриента

|ИЗ

| Документ.ДокументОЗачисленииВГруппу КАК ДокументОЗачисленииВГруппу

|ГДЕ

| ДокументОЗачисленииВГруппу.Ссылка В (&Ссылка)";

Запрос.Параметры.Вставить("Ссылка", Ссылка);

Выборка = Запрос.Выполнить().Выбрать();

ОбластьЗаголовок = Макет.ПолучитьОбласть("Заголовок");

Шапка = Макет.ПолучитьОбласть("Шапка");

ТабДок.Очистить();

ВставлятьРазделительСтраниц = Ложь;

Пока Выборка.Следующий() Цикл

Если ВставлятьРазделительСтраниц Тогда

ТабДок.ВывестиГоризонтальныйРазделительСтраниц();

КонецЕсли;

ТабДок.Вывести(ОбластьЗаголовок);

Шапка.Параметры.Заполнить(Выборка);

ТабДок.Вывести(Шапка, Выборка.Уровень());

ВставлятьРазделительСтраниц = Истина;

КонецЦикла;

//}}

КонецПроцедуры

1. ДокументОВозвратеДокументов – также является документом только с автоматическим заполнением. Элементы добавляются при нажатии на кнопку «Удаление» элемента справочника «Абитуриенты» (Рисунок 16). Более подоробно см. описание кнопки «Сортировка» справочника «Абитуриенты» на стр 13.

Содержит следующие реквизиты:

* ФИОАбитуриентаВозврат
* EmailАбитуриентаВозврат
* ДатаРожденияВозврат
* ТелефонАбитуриентаВозврат
* АдресАбитуриентаВозврат

Табличная часть «СписокДокументов» содержит реквизиты и свойства, аналогичные табличной части «ФормаДокумента» формы элемента документа «ЗаявлениеНаПоступление».

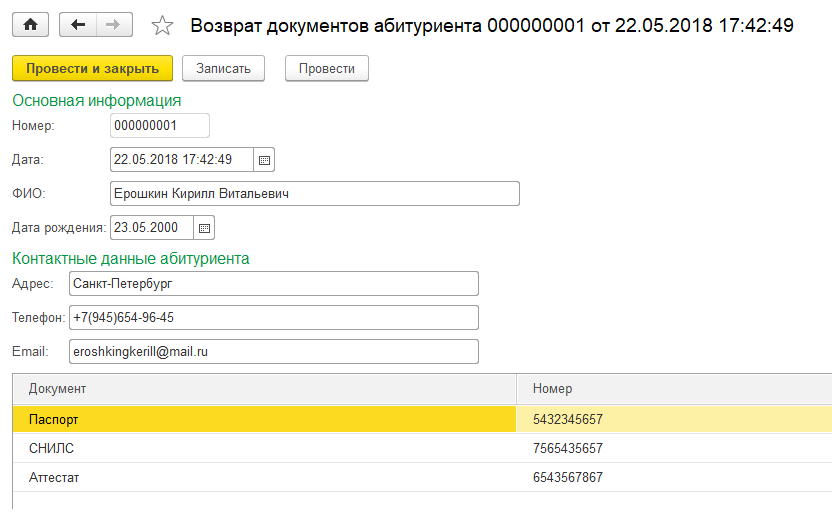


Рисунок 16 – Форма элемента документа ДокументОВозвратеДокументов

Также существуют общие модули системы:

* СортировкаАбитуриентов – сортирует абитуриентов по группам, согласно настройкам справочника «НастройкаСортировкиАбитуриентов»
* ОтчетTelegram – (см. ниже).

Листинг 17. Общий модуль СортировкаАбитуриентов

Процедура СортировкаАбитуриентовСправочник() Экспорт

//Удаление

Выборка = Справочники.СортировкаАбитуриентов.ВыбратьИерархически();

Пока Выборка.Следующий() Цикл

ОбъектСправочника = Выборка.ПолучитьОбъект();

ОбъектСправочника.Удалить();

КонецЦикла;

Выборка = Документы.ДокументОЗачисленииВГруппу.Выбрать();

Пока Выборка.Следующий() Цикл

ОбъектДокумента = Выборка.ПолучитьОбъект();

ОбъектДокумента.Удалить();

КонецЦикла;

//Первоначальные настройки

НастройкиСортировки = Справочники.НастройкиСортировкиАбитуриентов.НайтиПоНаименованию("MAIN\_CONFIG");

КолвоЧеловекВГруппе = НастройкиСортировки.КоличествоСтудентовВГруппе;

ГлобальнаяНумирацияГрупп = 0;

СтрокаДляОтчетаTelegram = "<b>\*Новая сортировка абитуриентов\*</b>"+Символы.ПС;

Для ИндексСчетчика = 0 по Перечисления.ПеречислениеСпециальностейАбитуринта.Количество()-1 Цикл

ИмяГруппы = Перечисления.ПеречислениеСпециальностейАбитуринта.Получить(ИндексСчетчика);

НоваяГруппа = Справочники.СортировкаАбитуриентов.СоздатьГруппу();

НоваяГруппа.Наименование = ИмяГруппы;

СтрокаДляОтчетаTelegram = СтрокаДляОтчетаTelegram + Символы.ПС + "<b>" + ИмяГруппы + "</b>"+Символы.ПС;

НоваяГруппа.Записать();

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Наименование, СпециальностьАбитурента, СреднийБаллАбитуриента, ТипАттестатаАбитуриента, ПриоритетЗачисленияАбитуриента,

| ФормаОплатыАбитуриента, ФотоАбитуриента, РодительАбитуриента, АдресАбитуриента, EmailАбитуриента, ТелефонАбитуриента, ДатаРожденияАбитуриента

|ИЗ

| Справочник.Абитуриенты КАК Абитуриенты

|УПОРЯДОЧИТЬ ПО

| СреднийБаллАбитуриента";

РезультатЗапроса = Запрос.Выполнить();

ВыборкаАбитуриентов = РезультатЗапроса.Выбрать();

ОбщийСчетчикАбитуриентовДляГрупп = 0;

Пока ВыборкаАбитуриентов.Следующий() Цикл

Если (ВыборкаАбитуриентов.СпециальностьАбитурента = ИмяГруппы) Тогда

ОбщийСчетчикАбитуриентовДляГрупп = ОбщийСчетчикАбитуриентовДляГрупп + 1;

КонецЕсли;

КонецЦикла;

ВыборкаАбитуриентов.Сбросить();

Для СчетчикГрупп = 1 По (Окр(ОбщийСчетчикАбитуриентовДляГрупп/КолвоЧеловекВГруппе+0.5,0,0)) Цикл

НоваяПодгруппа = Справочники.СортировкаАбитуриентов.СоздатьГруппу();

ГлобальнаяНумирацияГрупп = ГлобальнаяНумирацияГрупп + 1;

ТекущееНаименованиеГруппы = "1"+ИмяГруппы+"-"+ГлобальнаяНумирацияГрупп+НастройкиСортировки.ГодГруппы;

НоваяПодгруппа.Наименование = ТекущееНаименованиеГруппы;

СтрокаДляОтчетаTelegram = СтрокаДляОтчетаTelegram + Символы.ПС + "<a href="""+НастройкиСортировки.СсылкаНаСайтСЖурналами+ТекущееНаименованиеГруппы+".xlsx"">"+ТекущееНаименованиеГруппы+"</a>" + Символы.ПС;

НоваяПодгруппа.Родитель = НоваяГруппа.Ссылка;

НоваяПодгруппа.Записать();

СчетчикЧеловекВГруппе = 0;

Пока ВыборкаАбитуриентов.Следующий() Цикл

Если (ВыборкаАбитуриентов.СпециальностьАбитурента = ИмяГруппы) Тогда

//Структура для записи в документ "ДокументОЗачисленииВГруппу"

НовыйДокументЗачисления = Документы.ДокументОЗачисленииВГруппу.СоздатьДокумент();

НовыйДокументЗачисления.ФИОАбитуриента = ВыборкаАбитуриентов.Наименование;

НовыйДокументЗачисления.ГруппаАбитуриента = ТекущееНаименованиеГруппы;

НовыйДокументЗачисления.СпециальностьАбитурента = ВыборкаАбитуриентов.СпециальностьАбитурента;

НовыйДокументЗачисления.СреднийБаллАбитуриента = ВыборкаАбитуриентов.СреднийБаллАбитуриента;

НовыйДокументЗачисления.ФормаОплатыАбитуриента = ВыборкаАбитуриентов.ФормаОплатыАбитуриента;

НовыйДокументЗачисления.ДатаРожденияАбитуриента = ВыборкаАбитуриентов.ДатаРожденияАбитуриента;

НовыйДокументЗачисления.Дата = ТекущаяДата(); НовыйДокументЗачисления.Записать(РежимЗаписиДокумента.Проведение);

//Структура для записи в справочник "СортировкаАбитуриентов"

Абитуриент = Справочники.СортировкаАбитуриентов.СоздатьЭлемент();

Абитуриент.Родитель = НоваяПодгруппа.Ссылка;

Абитуриент.Наименование = ВыборкаАбитуриентов.Наименование;

Абитуриент.СпециальностьАбитурента = ВыборкаАбитуриентов.СпециальностьАбитурента;

Абитуриент.СреднийБаллАбитуриента = ВыборкаАбитуриентов.СреднийБаллАбитуриента;

Абитуриент.ТипАттестатаАбитуриента = ВыборкаАбитуриентов.ТипАттестатаАбитуриента;

Абитуриент.ПриоритетЗачисленияАбитуриента = ВыборкаАбитуриентов.ПриоритетЗачисленияАбитуриента;

Абитуриент.ФормаОплатыАбитуриента = ВыборкаАбитуриентов.ФормаОплатыАбитуриента;

Абитуриент.ФотоАбитуриента = ВыборкаАбитуриентов.ФотоАбитуриента;

Абитуриент.РодительАбитуриента = Справочники.РодителиАбитуриентов.НайтиПоНаименованию(ВыборкаАбитуриентов.РодительАбитуриента).Ссылка;

Абитуриент.АдресАбитуриента = ВыборкаАбитуриентов.АдресАбитуриента;

Абитуриент.EmailАбитуриента = ВыборкаАбитуриентов.EmailАбитуриента;

Абитуриент.ТелефонАбитуриента = ВыборкаАбитуриентов.ТелефонАбитуриента;

Абитуриент.ДатаРожденияАбитуриента = ВыборкаАбитуриентов.ДатаРожденияАбитуриента;

Абитуриент.Записать();

СтрокаДляОтчетаTelegram = СтрокаДляОтчетаTelegram + "<i>" + Строка(Число(СчетчикЧеловекВГруппе)+1) + ". " + Абитуриент.Наименование + "</i>" + Символы.ПС;

СчетчикЧеловекВГруппе = СчетчикЧеловекВГруппе + 1;

Если (СчетчикЧеловекВГруппе = КолвоЧеловекВГруппе) Тогда

Прервать;

КонецЕсли;

КонецЕсли;

КонецЦикла;

КонецЦикла;

КонецЦикла;

СоздатьОтчетTelegram(СтрокаДляОтчетаTelegram,"Сортировка абитуриентов от "+ТекущаяДата(),"Сортировка абитуриентов по группам",4);

КонецПроцедуры

В качестве отличительной особенности данного курсового проекта, как уже упоминалось раннее, выступает глубокая интеграция с мессенджером Telegram.

Список объектов системы, связанных с интеграцией Telegram:

* Справочник «TelegramBot» - содержит настройки бота оповещений (см. описание справочника «TelegramBot»)
* Справочник «Уведомления» - список отправленных ботом уведомлений (см. описание справочника «Уведомления»)
* Подсистема «Telegram» - содержит в себе два вышеприведенных справочника для визаулизации пользовательского интерфейса системы.
* Общий модуль «ОтчетTelegram» - содержит универсальную процедуру «СоздатьОтчетTelegram», реализующую механику взаимодействия 1С: Предприятие с Telegram.

Сам механизм работы формы справочника/документа/отчета с общим модулем является достаточно простым (Рисунок 17):

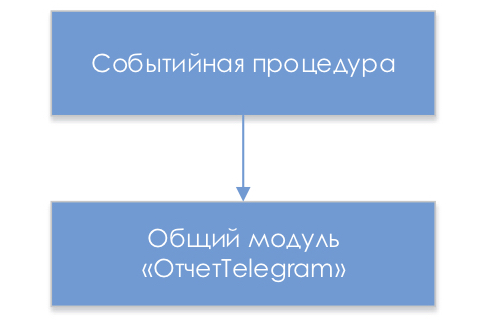


Рисунок 17 – Взаимодействие между модулями системы

Процедура формы справочника/документа:

* Фильтрация на уже отправленное сообщение
* Формирование сообщения для дальнейшей передачи

Общий модуль:

* Проверка на активацию сообщений для текущего события
* Отправка полученного сообщения https-запросом
* Запись в справочник информации об отправленном сообщении

В качестве примера посмотрим взаимодействие модуля элемента справочника «Сотрудники» с общим модулем «ОтчетTelegram»:

Листинг 18. Процедура «ПроверкаОтчетTelegram» элемента справочника «Сотрудники»

//Процедура формирования отчета Telegram

&наСервере

Процедура ПроверкаОтчетTelegram()

Если Справочники.Уведомления.НайтиПоРеквизиту("ФИО",Объект.Наименование).Пустая() Тогда

СтрокаОбразования = "";

Для НомерОбразования = 0 По Объект.ТабличнаяЧасть.Количество()-1 Цикл

СтрокаОбразования = СтрокаОбразования + "<b>" + строка(НомерОбразования+1) +".</b> " +

"Учреждение: " + Объект.ТабличнаяЧасть[НомерОбразования].Учреждение + Символы.ПС+

"Специальность: " + Объект.ТабличнаяЧасть[НомерОбразования].Специальность + Символы.ПС+

"Дата поступления: " + Формат(Объект.ТабличнаяЧасть[НомерОбразования].ДатаПоступления, "ДФ=dd.MM.yyyy") + Символы.ПС+

"Дата окончания: " + Формат(Объект.ТабличнаяЧасть[НомерОбразования].ДатаОкончания, "ДФ=dd.MM.yyyy") + Символы.ПС + Символы.ПС;

КонецЦикла;

Сообщение = "<b>\*Добавление нового сотрудника\*</b>"+Символы.ПС+Символы.ПС+

"<b>Основная информация</b>"+Символы.ПС+

"ФИО: "+Объект.Наименование+Символы.ПС+

"Дата рождения: "+Формат(Объект.ДатаРожденияСотрудника, "ДФ=dd.MM.yyyy")+Символы.ПС+Символы.ПС+

"<b>Контактная информация</b>"+Символы.ПС+

"E-mail: " +Объект.EmailСотрудника+Символы.ПС+

"Телефон: "+Объект.ТелефонСотрудника+Символы.ПС+Символы.ПС+

"<b>Образование</b>"+Символы.ПС + СтрокаОбразования;

СоздатьОтчетTelegram(Сообщение,Объект.Наименование,"Добавление нового сотрудника",5);

КонецЕсли

КонецПроцедуры

Рассмотрим исходный код общего модуля «ОтчетTelegram»:

Листинг 19. Общий модуль ОтчетTelegram

Функция Base64Декодинг(ВходнаяСтрока, КодировкаТекста=Неопределено)

ДвДанные = Base64Значение(ВходнаяСтрока);

Если ДвДанные = Неопределено Тогда

Возврат Неопределено;

КонецЕсли;

Если КодировкаТекста = Неопределено Тогда

ЧтениеДвДанных = Новый ЧтениеДанных(ДвДанные);

Иначе

ЧтениеДвДанных = Новый ЧтениеДанных(ДвДанные, КодировкаТекста);

КонецЕсли;

СтрокаВозврата = ЧтениеДвДанных.ПрочитатьСимволы();

ЧтениеДвДанных.Закрыть();

Возврат СокрЛП(СтрокаВозврата);

КонецФункции

Процедура СоздатьОтчетTelegram(Сообщение,ФИОИмя,СобытиеИмя,ИндексЭлементаВызова) Экспорт

Данные = Справочники.TelegramBot.НайтиПоНаименованию("MAIN\_BOT");

Если (Данные.УведомленияБота.Получить(ИндексЭлементаВызова).Значение = Истина) Тогда

Ресурс = "bot" + Base64Декодинг(Данные.Токен) + "/sendMessage?chat\_id="+Данные.ИдентификаторНазначения+"&text="+Сообщение+"&parse\_mode=HTML";

Соединение = Новый HTTPСоединение(Данные.Сервер,443,,,,,Новый ЗащищенноеСоединениеOpenSSL());

Запрос = Новый HTTPЗапрос(Ресурс);

Ответ = Соединение.Получить(Запрос);

НовыйОтчетTelegram = Справочники.Уведомления.СоздатьЭлемент();

НовыйОтчетTelegram.Событие = СобытиеИмя;

НовыйОтчетTelegram.Наименование = "Уведомление Telegram от "+ТекущаяДата();

НовыйОтчетTelegram.Сообщение = Сообщение;

НовыйОтчетTelegram.ФИО = ФИОИмя;

НовыйОтчетTelegram.Идентификатор = Данные.ИдентификаторНазначения;

НовыйОтчетTelegram.Дата = ТекущаяДата();

НовыйОтчетTelegram.Записать();

Иначе

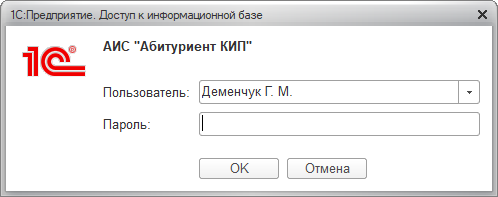
Сообщить("Уведомления для данного модуля выключены!");

КонецЕсли;

КонецПроцедуры

2.4 Разработка пользовательских интерфейсов программного обеспечения

При запуске программы возникает окно авторизации пользователя/сотрудника приемной комиссии колледжа (Рисунок 18).



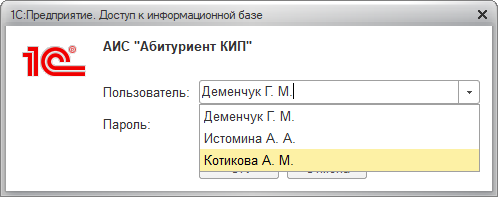


Рисунок 18 – Доступ к информационной базе

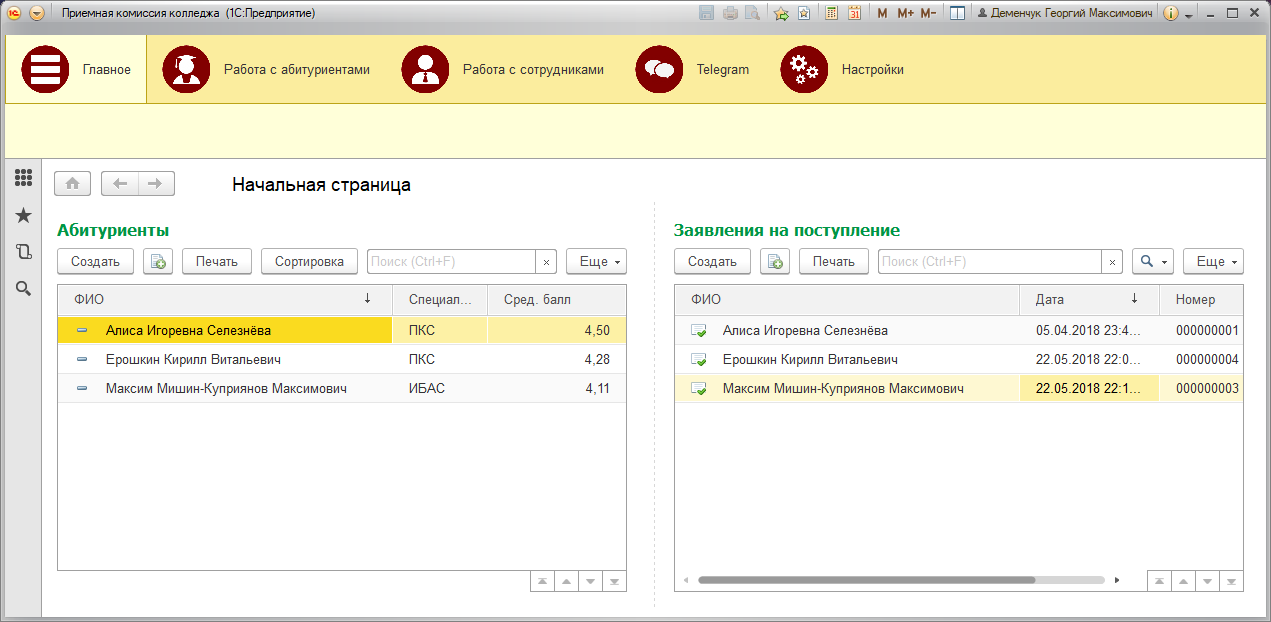
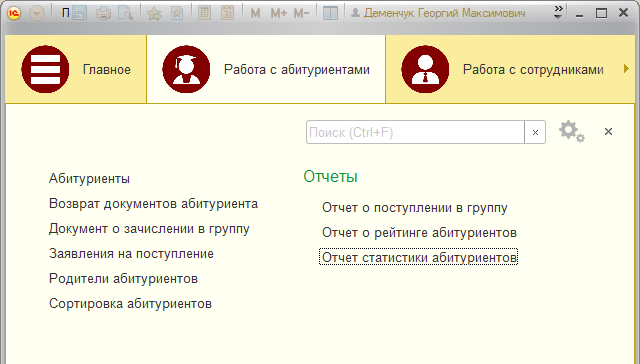
При успешной авторизации перед администратором слева возникает список абитуриентов, а справа – заявления на поступления (Рисунок 19).

Рисунок 19 – Пользовательский интерфейс начальной страницы

В зависимости от ролей сотрудников, некоторые подсистемы скрыты:

* При роли «Сотрудник» скрываются все подсистемы, кроме «РаботаСАбитуриентами»
* При роли «Менеджер» скрывается подсистема «Telegram» (но соответственно сами заранее сконфигурироанные оповещения активированы)
* При роли «Администратор» все подсистемы отображаются на форме.

Подсистема «РаботаСАбитуриентами» является основной и содержит все необходимые модули, предназначенные для работы с абитуриентами (Рисунок 20).



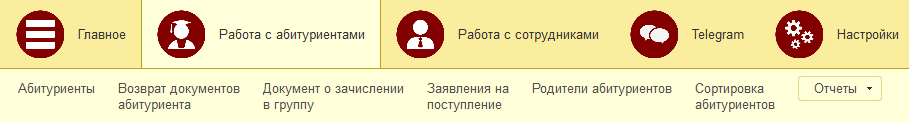


Рисунок 20 - Подсистема «РаботаСАбитуриентами»

Подсистема «РаботаССотрудниками» содержит единственный справочник «Сотрудники» и предназначена для добавления сотрудников (Рисунок 21).

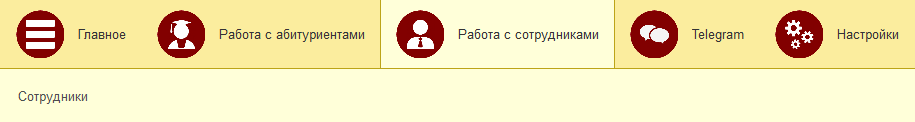


Рисунок 21 - Подсистема «РаботаССотрудниками»

Подсистема «Telegram» предназначена для объединения всех модулей, связанных с Telegram (Рисунок 22).

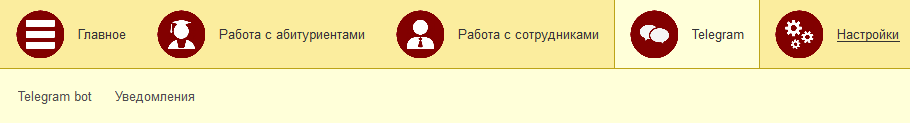


Рисунок 22 - Подсистема «Telegram»

Подсистема «Настройки» предназначена для объединения всех настроек системы, а также вывода регистров для анализа и предотвращения ошибок (Рисунок 23).

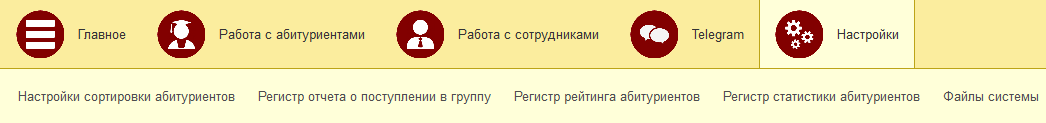


Рисунок 23 - Подсистема «Настройки»

2.5. Организация защиты базы данных 1С

* 1. . Тестирование и отладка программного обеспечения

Один из самых активных модулей, который будут использован чаще всего и должен работать без ошибок – бизнес процесс по ремонту (Рисунок 14).

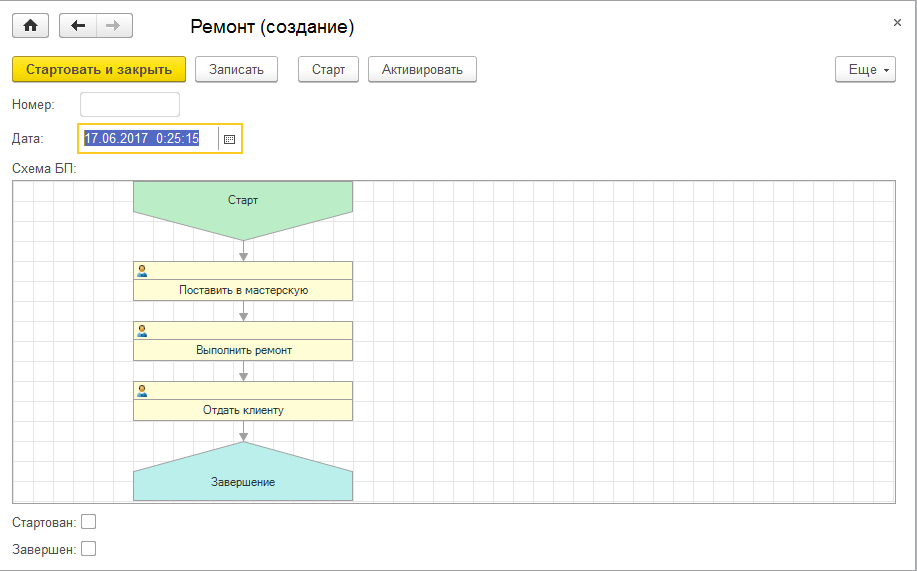


Рисунок 14 - Создание бизнес процесса

Предположим, что был заказ на ремонт, тогда менеджер либо мастер создает бизнес процесс, далее бизнес процесс выполняет мастер. Во вкладке «Задача ремонт» создаются задачи по бизнес процессу. Пример работы бизнес процесса.(Рисунок 15).

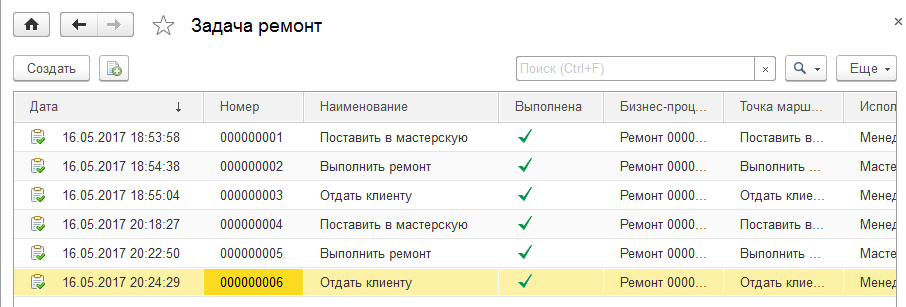


Рисунок 15 - Пример работы программы

Второй модуль, это поступление на ремонт. В этом модуле заполняется бланк на заказ ремонта, выбираются необходимые детали, а далее система автоматически считает стоимость детали вместе с ремонтом. Далее создается чек, который мы можем распечатать нажав на кнопку «Печать».

Проиллюстрированную работу вышеперечисленных модулей можно рассмотреть в руководстве пользователя.

* 1. Руководство системного программиста

Программа поддерживает полную привязку к пользовательской системе. Пользователи могут быть настроены через меню конфигуратора: Администрирование, Пользователи (Рисунок 16).

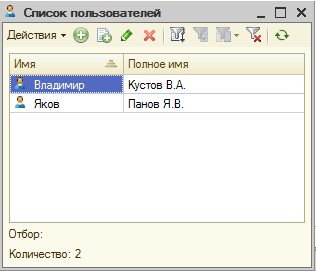


Рисунок 16 - Список пользователей в конфигураторе

Перед добавлением нового пользователя в конфигураторе, необходимо занести данные о нем в базу данных через 1С предприятие: Справочники, Сотрудники (Рисунок 19). После того как Вы добавили данные сотрудника, можно добавлять его в список пользователей и присваивать роли.

Полное имя пользователя должно соответствовать наименованию его строки в справочнике сотрудники (Рисунок 17).

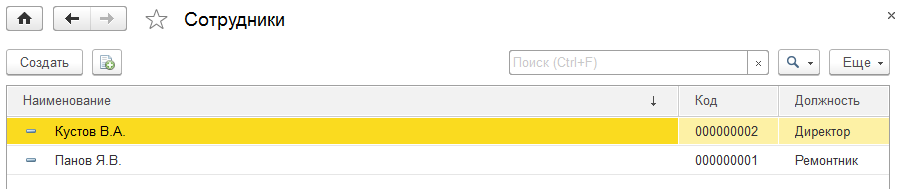


Рисунок 17 - Справочник сотрудники

В программе имеются автоматизированные процессы:

1. При заполнении поступления на ремонт – автоматически считается стоимость ремонта и создается чек.
   1. Руководство программиста

Программа предназначена для организации ремонта устройств, так же она осуществляет автоматизацию подсчета стоимости и создания чека(как описано в руководстве системного программиста).

Подробное описание автоматизированных модулей:

1. Подсчет стоимости. Менеджер при заполнении выбирает детали необходимые для ремонта устройства и их количество. Система считает общую сумму всех деталей и прибавляет стоимость ремонта, залоденную непосредственно в коде программы.

&НаКлиенте

Процедура НеобходимыеДеталиПриИзменении(Элемент)

НеобходимыеДеталиПриИзмененииНаСервере();

КонецПроцедуры

&НаСервере

Процедура НеобходимыеДеталиПриИзмененииНаСервере()

Объект.СтоимостьРемонта = 0;

ТЧ = Объект.НеобходимыеДетали;

стоимость = 0.00;

Для Каждого СтрокаТЧ Из ТЧ Цикл

регистр = РегистрыСведений.ЦеныДеталей.Выбрать();

Пока регистр.Следующий() Цикл

Если СтрокаТЧ.Деталь = регистр.Номенклатура Тогда

Если СтрокаТЧ.Количество = 0 Тогда

Иначе

стоимость = стоимость + (регистр.Стоимость \* СтрокаТЧ.Количество );

КонецЕсли

КонецЕсли

КонецЦикла;

КонецЦикла;

Объект.СтоимостьРемонта = стоимость + 5000;

КонецПроцедуры

* 1. Руководство пользователя

Как только Вы запустите 1С предприятие, Вам предложат ввести логин и пароль для идентификации личности. После авторизации Вы можете приступать к работе.

Перед началом использование программы, обязательно изучите технический раздел, пункт 2.3 «Проектирование программного обеспечения», где описаны все справочники, документы и прочие модули информационной базы!

1. Первым шагом необходимо создать номенклатуру в справочнике Номенклатура.
2. Далее нужно создать деталь в справочнике Детали.
3. Далее в регистре сведений «Цены деталей» нужно записать ее стоимость за 1 штуку.
4. Далее нужно занести клиентов в базу данных.

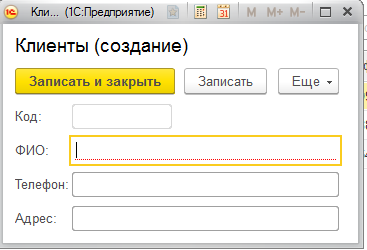


Рисунок 18 - Добавление нового клиента

1. Почти все готово. Создадим поступление на ремонт. (Рисунок 19).

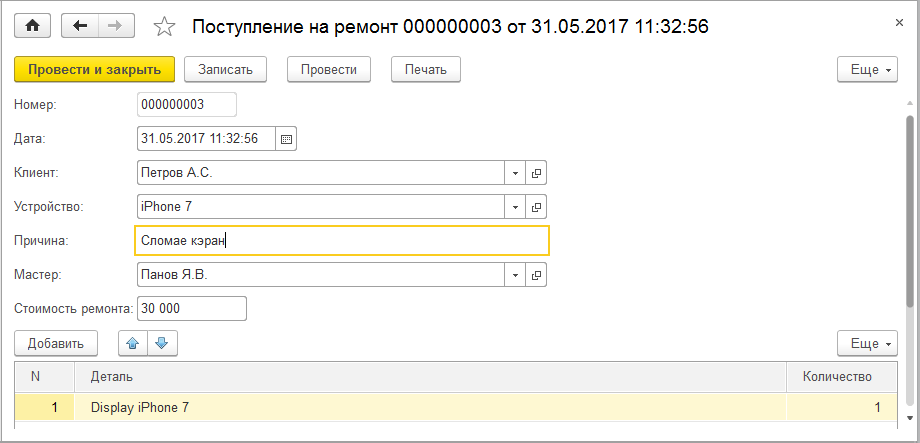


Рисунок 19 - Оформление заказа

Данные документа написаны прямолинейно и не вызывают вопросов. Нажимаем провести и закрыть.

1. Далее создаем бизнес процесс во вкладке «Ремонт» (Рисунок 14) и нажимаем «Стартовать и закрыть».
2. Далее создается задача по-нашему бизнес процессу» (Рисунок 20). Здесь мы продвигаем наш бизнес процесс по кнопке «Выполнить, закрыть».

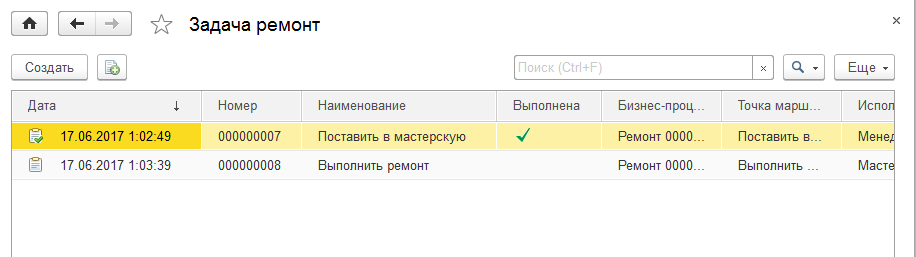


Рисунок 20 - Пример заполнения продажи товара

Никаких сложностей в заполнении документа возникнуть не должно.

1. После создания поступления на ремонт создается чек, его мы печатаем после выполнения ремонта, заходим снова в «Поступление на ремонт» открываем заказ и нажимаем кнопку «Печать» далее видим чек (Рисунок 21).



Рисунок 21 - Чек продажи

ЗАКЛЮЧЕНИЕ

В ходе работы над курсовым проектом были получены знания о том, как работает внутренняя система ремонтной мастерской и необходимые базисы для создания программ в русскоязычной среде 1С.

Были достигнуты цели, которые были поставлены:

- Получение знаний 1С: Предприятие.

- Получение навыков работы с 1С: Предприятие.

- Написание ПО для 1С: Предприятие «Ремонтная мастерская».

- Простота и удобство использования ПО для 1С: Предприятие «ремонтная мастерская».

Разработанный продукт полностью выполняет все поставленные цели, а к его достоинствам можно простоту использования.

Достоинства 1С: предприятие 8.3. Версия для обучения программированию:

- Удобный интерфейс конфигуратора.

- Много инструментов для конфигурирования базы данных и самого ПО без применения кода.

В процессе выполнения курсового проекта были выполнены основные цели, а именно – были закреплены знания совершенного нового языка программирования «1С» и навыки по проектированию баз данных.

Программа включает в себя весь необходимый функционал и отлично подходит не только к выбранному предприятию, но и для других ремонтных мастерских, выполняющих ремонт мобильных гаджетов.

Были выполнены все поставленные задачи и смоделирована схема работы ремонтной мастерской, которая отражает все процессы, происходящие в реальности. Поставленная задача выполнена, и организованы основные цели данной работы.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Стандарты

* 1. ГОСТ 7.32 – 2001. Система стандартов по информацию, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления. – М.: ИПК Издательство стандартов, 2001. – 21 с.
  2. ГОСТ 7.1. – 2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2004. – 169 с.
  3. ГОСТ 7.82 – 2001. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2001. – 21 с.
  4. Единая система программной документации. – М.: Стандартинформ, 2005. – 128 с

Учебники

* 1. Радченко М.Г., Хрусталева Е.Ю. 1С: Предприятие 8.3. Практическое пособие разработчика. - 1С-Паблишинг, 2014.
  2. Габец А.П., Гончаров Д.И. 1С:предприятие 8.0 Простые примеры разработки. - 1С-Паблишинг, 2014.
  3. Хрусталева Е.Ю. Разработка сложных отчётов в 1С:Предприятии 8. - 1С-Паблишинг, 2014.
  4. Ажеронок В.А., Островерх А.В., Радченко М.Г., Хрусталева Е.Ю. Разработка управляемого интерфейса. - 1С-Паблишинг, 2015.
  5. Габец А.П., Козырев Д.В., Кухлевский Д.С., Хрусталева Е.Ю. Реализация прикладных задача в системе «1СПредприятие 8.2» - 1С-Паблишинг, 2014.

Интернет-ресурсы

* 1. Интернет справочник 1С:Предприятие 8. – [Электронный ресурс]. – Режим доступа: <http://v8.1c.ru/>, свободный. – (Дата обращения: 01.06.17).
  2. Обзор архитектуры платформы – [Электронный ресурс]. – Режим доступа: <http://v8.1c.ru/overview/Term_000000877.htm>, свободный. – (Дата обращения: 03.06.17).