1. ***Хранение данных о курсах, преподавателях и студентах***

1.1. Создаём Справочники с названиями: Курсы, Преподаватели, Студенты, Файлы, ФайлыЛоготипов, Занятия.

1.2.В Справочнике Курсы Создаём следующие реквизиты:

Логотип(Ссылка на справочник Файлы);

Описание(Строка, неограниченно);

ЛогоЗначение(Строка, неограниченно);

ДатаНачала(Дата, дата);

ДатаОкончания(Дата, дата);

Владелец – Справочник Преподаватели

Табличные части:

Тесты:

Вопрос(Строка , неограниченно);

ОтветВерный(Строка, неограниченно);

ОтветНеверный1(Строка, неограниченно);

ОтветНеверный2(Строка, неограниченно);

Занятия:

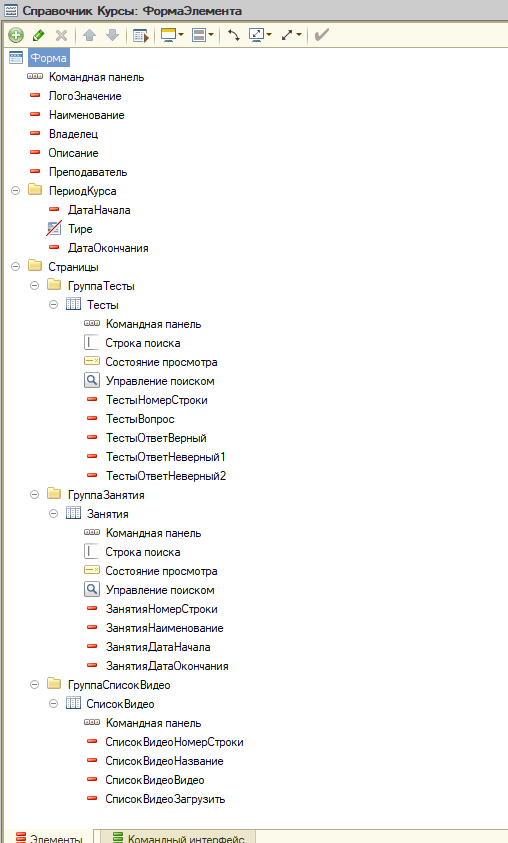
Наименование(Ссылка на справочник Занятия);

ДатаНачала(Дата, ДатаИВремя);

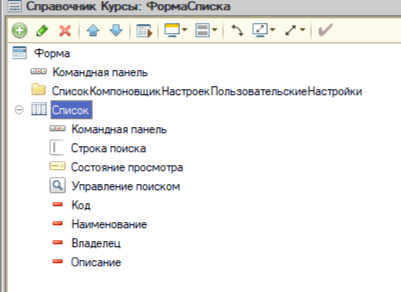
ДатаОкончания(Дата, ДатаИВремя);

Формы:

ФормаЭлемента:



ФормаСписка:



1.3.В Справочнике Преподаватели:

Меняем заголовок для реквизита наименование на ФИО.

1.4.В Справочнике Студенты:

Создаём реквизит Пароль(Строка, неограниченно);

Меняем заголовок наименования на ФИО;

1.5.В справочнике Файлы и ФайлыЛоготипов:

Создаём реквизит Значение(ХранилищеЗначения);

1.6.В справочнике Занятия:

Создаём реквизиты:

Логотип(Строка, неограниченно);

ЛоготипЗначение(Ссылка на справочник ФайлыЛоготипов);

Табличные части:

СписокВидео:

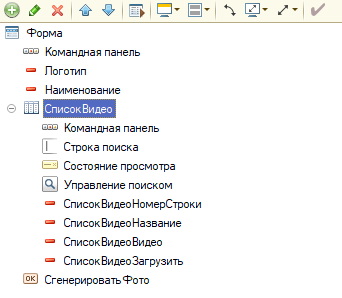
Название(Строка, неограниченно);

Видео(Ссылка на справочник Файлы);

Загрузить(Строка, неограниченно);

Формы:

ФормаЭлемента:



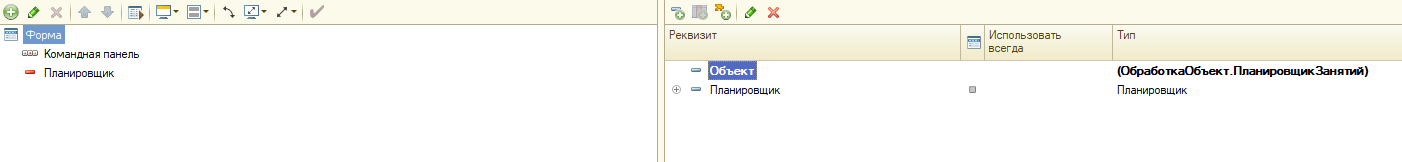
**ВНИМАНИ: ВЕСЬ НЕОБХОДИМЫЙ КОД БУДЕТ ВЫЛОЖЕН В САМОМ КОНЦЕ В ДОПОЛНИТЕЛЬНЫХ МАТЕРИАЛАХ!!!**

2. ***Планирование курса***

2.1. Создаём обработку ПланировщикЗанятий;

2.2. В Данной Обработке:

Форма:



3. ***Учёт взаиморасчётов между школой и студентами***

3.1. Создаём Документы: Оплата, ЗаписьНаКурс, ОтказОтКурса; А так же создаём Регистр накопления: ДоходСКурсов; А так же периодический регистр сведений: ЦеныКурсов и СтудентыНаКурсах.

3.2.В Документе оплата:

Создаём Реквизиты:

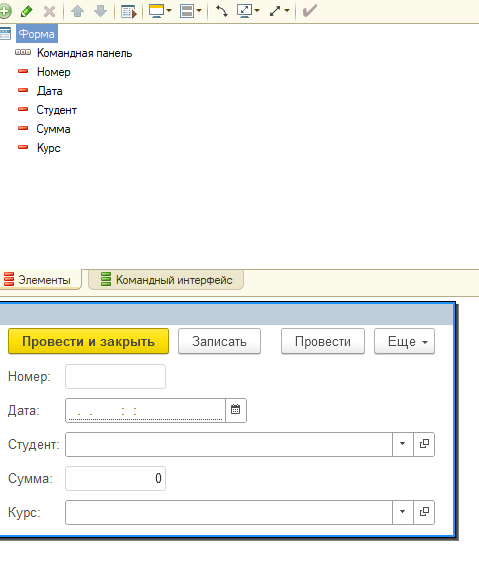
Студент(Ссылка на справочник Студенты);

Сумма(Число, длинна 10, с точностью до 2, неотрицательное);

Курс(Ссылка на справочник Курсы);

Формы:

Форма:



3.3. В Документах ЗаписьНаКурс и ОтказОтКурса:

Создаём Реквизиты:

Студент(Ссылка на справочник Студенты);

Курс(Ссылка на справочник Курсы);

3.4. В Регистре сведений ЦеныКурсов:

Создаём Измерения:

НаименованиеКурса(Ссылка на справочник Курсы);

Создаём Ресурсы:

Цена(Число, длинна 10, точность 2, неотрицательное);

3.5. В Регистре сведений СтудентыНаКурсах:

Создаём Измерения:

Студент(Ссылка на справочник Студенты);

Курсы(Ссылка на справочник Курсы);

Создаём Ресурсы:

Записан(Булево);

ДатаЗаписи(Дата, Дата);

ДатаПоследнейВозможностиОтказа(Дата, Дата);

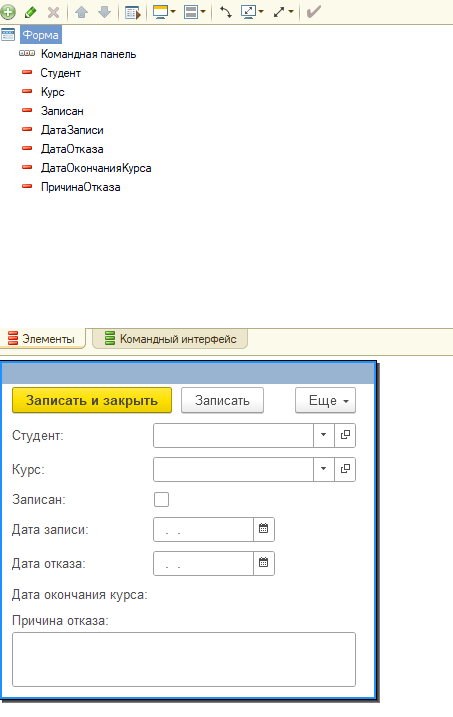
ДатаОтказа(Дата, Дата);

ДатаОкончанияКурса(Дата, Дата)

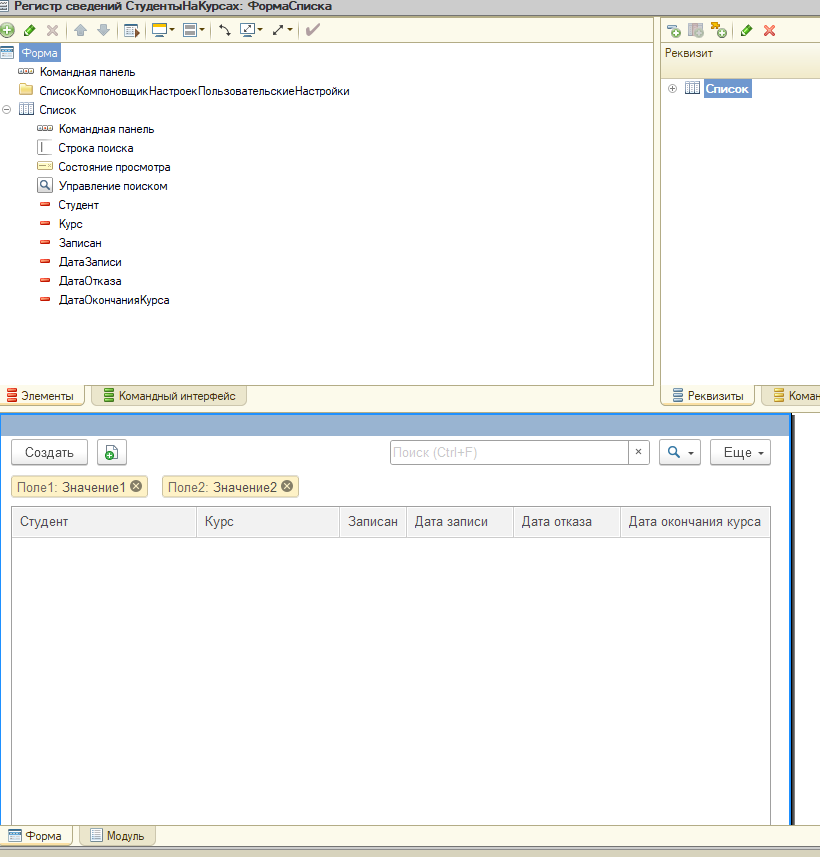
ПричинаОтказа(Строка, неограниченно);

Формы:

ФормаЗаписи:



ФормаСписка:



3.6. В Регистре накоплений ДоходСКурсов:

Создаём Измерения:

Студент(Ссылка на справочник Студенты);

Курс(Ссылка на справочник Курсы);

Создаём Ресурсы:

Сумма(Число, 10, точность 2, неотрицательное);

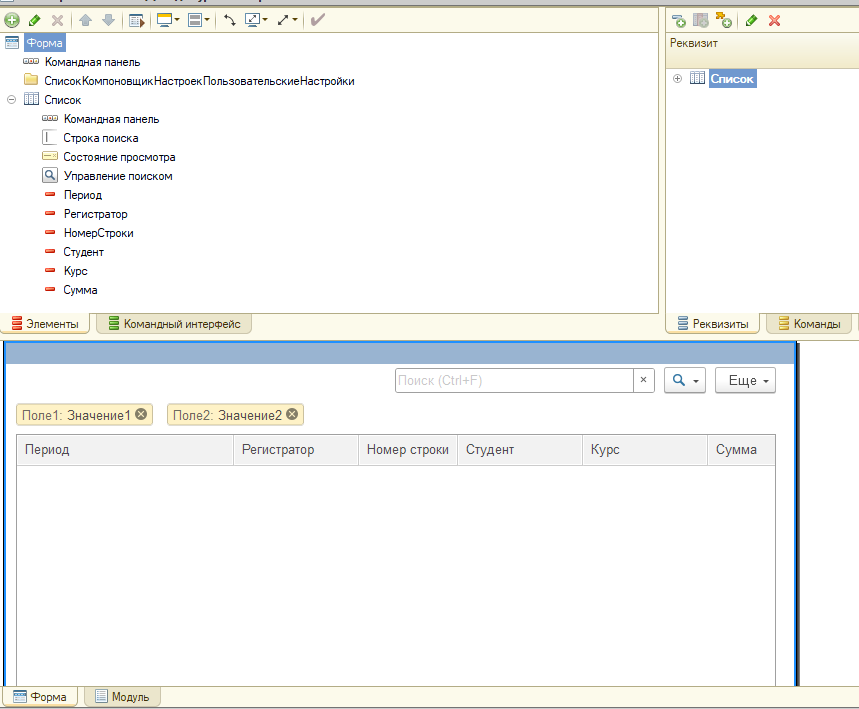
Добавляем Регистраторы:

Документ.ЗаписьНаКурс;

Документ.ОтказОтКурса;

Формы:

ФормаСписка:



4. ***Хранение электронных материалов для занятий и т.п.***

Это мы создаём в пунктах 1.

**ДОПОЛНИТЕЛЬНЫЙ МАТЕРИАЛ**

HTTPСервисы:

Обмен(КорневойURL: obmen):

Создаём:

УчебныеКурсы(URL: trainingcourses):

Отправить;

УчебныеКурсыПоЗаписи(URL: trainingcourseswithstudy):

Отправить;

Модуль:

**Внимание, копируйте код только если уверены, что использовали названия как они были указаны, иначе посмотрите структуру и просто сделайте по такому же принципу!!!**

Функция УчебныеКурсыОтправить(Запрос)

Ответ = Новый HTTPСервисОтвет(200);

Курсы = Новый Массив;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Курсы.Наименование КАК Наименование,

| Курсы.Логотип.Значение КАК ЛоготипЗначение,

| Курсы.Описание КАК Описание,

| Курсы.ДатаНачала КАК ДатаНачала,

| Курсы.ДатаОкончания КАК ДатаОкончания,

| Курсы.Ссылка КАК Ссылка,

| Курсы.Владелец.Наименование КАК ВладелецНаименование

|ИЗ

| Справочник.Курсы КАК Курсы";

РезультатЗапроса = Запрос.Выполнить();

Выборка = РезультатЗапроса.Выбрать();

Если не РезультатЗапроса.Пустой()

Тогда

Пока Выборка.Следующий() Цикл

Данные = Новый Структура;

Тесты = Новый Массив;

Занятия = Новый Массив;

ПрошлыеВидео = Новый Массив;

ВидеоЕсть = Ложь;

Данные.Вставить("НазваниеКурса", Выборка.Наименование);

Данные.Вставить("ОписаниеКурса", Выборка.Описание);

Данные.Вставить("ЗначениеЛоготипа", Выборка.ЛоготипЗначение);

Данные.Вставить("Преподаватель", Выборка.ВладелецНаименование);

Данные.Вставить("ДатаНачалаКурса", Выборка.ДатаНачала);

Данные.Вставить("ДатаОкончанияКурса", Выборка.ДатаОкончания);

Для Каждого СтрокаТЧ из Выборка.Ссылка.Занятия

Цикл

Занятие = Новый Структура;

Занятие.Вставить("НазваниеЗанятия", СтрокаТЧ.Наименование.Наименование);

СпрЗан = Справочники.Занятия.НайтиПоНаименованию(СтрокаТЧ.Наименование.Наименование).ПолучитьОбъект();

Занятие.Вставить("Логотип", СпрЗан.ЛоготипЗначение.Значение);

Занятие.Вставить("НачалоЗанятия", СтрокаТЧ.ДатаНачала);

Занятие.Вставить("КонецЗанятия", СтрокаТЧ.ДатаОкончания);

СписокВидео = Новый Массив;

ЗапросВидео = Новый Запрос;

ЗапросВидео.Текст =

"ВЫБРАТЬ

| ЗанятияСписокВидео.Название КАК Название,

| ЗанятияСписокВидео.Видео КАК Видео

|ИЗ

| Справочник.Занятия.СписокВидео КАК ЗанятияСписокВидео

|ГДЕ

| ЗанятияСписокВидео.Ссылка.Наименование = &Наименование";

ЗапросВидео.УстановитьПараметр("Наименование", СтрокаТЧ.Наименование.Наименование);

РезультатЗапросаВидео = ЗапросВидео.Выполнить();

ВыборкаВидео = РезультатЗапросаВидео.Выбрать();

Пока ВыборкаВидео.Следующий()

Цикл

Видео = Новый Структура;

Видео.Вставить("НазваниеВидео", ВыборкаВидео.Название);

Видео.Вставить("ЗначениеВидео", ВыборкаВидео.Видео.Значение);

СписокВидео.Добавить(Видео);

КонецЦикла;

Занятие.Вставить("СписокВидео", СписокВидео);

Занятия.Добавить(Занятие);

КонецЦикла;

Данные.Вставить("Занятия", Занятия);

Для Каждого ТекСтрока из Выборка.Ссылка.Тесты

Цикл

Тест = Новый Структура;

Тест.Вставить("ВопросТеста", ТекСтрока.Вопрос);

Тест.Вставить("ПравильныйОтвет", ТекСтрока.ОтветВерный);

Тест.Вставить("НеправильныйОтвет1", ТекСтрока.ОтветНеверный1);

Тест.Вставить("НеправильныйОтвет2", ТекСтрока.ОтветНеверный2);

Тесты.Добавить(Тест)

КонецЦикла;

Данные.Вставить("Тесты", Тесты);

Курсы.Добавить(Данные);

КонецЦикла;

ОтветСтрока = XMLСтрока(Новый ХранилищеЗначения(Курсы));

Ответ.УстановитьТелоИзСтроки(ОтветСтрока);

КонецЕсли;

Возврат Ответ;

КонецФункции

Функция УчебныеКурсыПоЗаписиОтправить(Запрос)

Ответ = Новый HTTPСервисОтвет(200);

Студент = Запрос.ПолучитьТелоКакСтроку();

Курсы = Новый Массив;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| СтудентыНаКурсах.Студент КАК Студент,

| СтудентыНаКурсах.Курс КАК Курс

|ИЗ

| РегистрСведений.СтудентыНаКурсах КАК СтудентыНаКурсах

|ГДЕ

| СтудентыНаКурсах.Записан = ИСТИНА

| И СтудентыНаКурсах.Студент.Наименование = &Студент";

Запрос.УстановитьПараметр("Истина", Истина);

Запрос.УстановитьПараметр("Студент", Студент);

РезультатЗапроса = Запрос.Выполнить();

Выборка = РезультатЗапроса.Выбрать();

Пока Выборка.Следующий() Цикл

ЗапросКурса = Новый Запрос;

ЗапросКурса.Текст =

"ВЫБРАТЬ

| Курсы.Наименование КАК Наименование,

| Курсы.Логотип.Значение КАК ЛоготипЗначение,

| Курсы.Описание КАК Описание,

| Курсы.ДатаНачала КАК ДатаНачала,

| Курсы.ДатаОкончания КАК ДатаОкончания,

| Курсы.Ссылка КАК Ссылка,

| Курсы.Владелец.Наименование КАК ВладелецНаименование

|ИЗ

| Справочник.Курсы КАК Курсы

|ГДЕ

| Курсы.Ссылка = &Ссылка";

ЗапросКурса.УстановитьПараметр("Ссылка", Выборка.Курс);

РезультатЗапросаКурса = ЗапросКурса.Выполнить();

ВыборкаКурса = РезультатЗапросаКурса.Выбрать();

Если не РезультатЗапросаКурса.Пустой()

Тогда

Пока ВыборкаКурса.Следующий() Цикл

Данные = Новый Структура;

Тесты = Новый Массив;

Занятия = Новый Массив;

ПрошлыеВидео = Новый Массив;

ВидеоЕсть = Ложь;

Данные.Вставить("НазваниеКурса", ВыборкаКурса.Наименование);

Данные.Вставить("ОписаниеКурса", ВыборкаКурса.Описание);

Данные.Вставить("ЗначениеЛоготипа", ВыборкаКурса.ЛоготипЗначение);

Данные.Вставить("Преподаватель", ВыборкаКурса.ВладелецНаименование);

Данные.Вставить("ДатаНачалаКурса", ВыборкаКурса.ДатаНачала);

Данные.Вставить("ДатаОкончанияКурса", ВыборкаКурса.ДатаОкончания);

Для Каждого СтрокаТЧ из ВыборкаКурса.Ссылка.Занятия

Цикл

Занятие = Новый Структура;

Занятие.Вставить("НазваниеЗанятия", СтрокаТЧ.Наименование.Наименование);

Занятие.Вставить("НачалоЗанятия", СтрокаТЧ.ДатаНачала);

Занятие.Вставить("КонецЗанятия", СтрокаТЧ.ДатаОкончания);

СписокВидео = Новый Массив;

ЗапросВидео = Новый Запрос;

ЗапросВидео.Текст =

"ВЫБРАТЬ

| ЗанятияСписокВидео.Название КАК Название,

| ЗанятияСписокВидео.Видео КАК Видео

|ИЗ

| Справочник.Занятия.СписокВидео КАК ЗанятияСписокВидео

|ГДЕ

| ЗанятияСписокВидео.Ссылка.Наименование = &Наименование";

ЗапросВидео.УстановитьПараметр("Наименование", СтрокаТЧ.Наименование.Наименование);

РезультатЗапросаВидео = ЗапросВидео.Выполнить();

ВыборкаВидео = РезультатЗапросаВидео.Выбрать();

Пока ВыборкаВидео.Следующий()

Цикл

Видео = Новый Структура;

Видео.Вставить("НазваниеВидео", ВыборкаВидео.Название);

Видео.Вставить("ЗначениеВидео", ВыборкаВидео.Видео.Значение);

СписокВидео.Добавить(Видео);

КонецЦикла;

Занятие.Вставить("СписокВидео", СписокВидео);

Занятия.Добавить(Занятие);

КонецЦикла;

Данные.Вставить("Занятия", Занятия);

Для Каждого ТекСтрока из ВыборкаКурса.Ссылка.Тесты

Цикл

Тест = Новый Структура;

Тест.Вставить("ВопросТеста", ТекСтрока.Вопрос);

Тест.Вставить("ПравильныйОтвет", ТекСтрока.ОтветВерный);

Тест.Вставить("НеправильныйОтвет1", ТекСтрока.ОтветНеверный1);

Тест.Вставить("НеправильныйОтвет2", ТекСтрока.ОтветНеверный2);

Тесты.Добавить(Тест)

КонецЦикла;

Данные.Вставить("Тесты", Тесты);

Курсы.Добавить(Данные);

КонецЦикла;

КонецЕсли;

КонецЦикла;

ОтветСтрока = XMLСтрока(Новый ХранилищеЗначения(Курсы));

Ответ.УстановитьТелоИзСтроки(ОтветСтрока);

Возврат Ответ;

КонецФункции

Авторизация(КорневойURL: auth):

Студенты(URL: students):

Отправить;

Модуль:

**Внимание, копируйте код только если уверены, что использовали названия как они были указаны, иначе посмотрите структуру и просто сделайте по такому же принципу!!!**

Функция СтудентыПроверить(Запрос)

Ответ = Новый HTTPСервисОтвет(200);

ПолученныеДанные = СтрРазделить(Запрос.ПолучитьТелоКакСтроку(), ";");

Студент = ПолученныеДанные[0];

Пароль = ПолученныеДанные[1];

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Студенты.Наименование КАК Наименование,

| Студенты.Пароль КАК Пароль

|ИЗ

| Справочник.Студенты КАК Студенты

|ГДЕ

| Студенты.Наименование = &Студент

| И Студенты.Пароль = &Пароль";

Запрос.УстановитьПараметр("Пароль", Пароль);

Запрос.УстановитьПараметр("Студент", Студент);

РезультатЗапроса = Запрос.Выполнить();

Если не РезультатЗапроса.Пустой()

Тогда

Ответ.УстановитьТелоИзСтроки("1");

Иначе

Ответ.УстановитьТелоИзСтроки("0");

КонецЕсли;

Возврат Ответ;

КонецФункции

Справочники:

Курсы:

ФормаЭлемента.Модуль:

&НаКлиенте

Процедура ФотоНажатие(Элемент, СтандартнаяОбработка)

СтандартнаяОбработка = Ложь;

Диалог = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Открытие);

ОП = Новый ОписаниеОповещения("ОбработкаФото", ЭтотОбъект);

Диалог.Показать(ОП);

КонецПроцедуры

&НаКлиенте

Процедура ОбработкаФото(Результат, ДП) Экспорт

Если не Результат = Неопределено

Тогда

Картинка = Новый Картинка(Результат[0]);

СохранитьКартинку(Картинка)

КонецЕсли;

КонецПроцедуры

Процедура СохранитьКартинку(Картинка)

Файл = Справочники.ФайлыЛоготипов.СоздатьЭлемент();

Файл.Наименование = Объект.Наименование;

Файл.Значение = Новый ХранилищеЗначения(Картинка);

Файл.Записать();

Объект.Логотип = Файл.Ссылка;

Объект.ЛогоЗначение = ПолучитьНавигационнуюСсылку(Объект.Логотип, "Значение");

КонецПроцедуры

Занятие:

ФормаЭлемента.Модуль:

&НаКлиенте

Процедура СписокВидеоЗагрузитьНачалоВыбора(Элемент, ДанныеВыбора, ВыборДобавлением, СтандартнаяОбработка)

СтандартнаяОбработка = Ложь;

Диалог = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Открытие);

ОП = Новый ОписаниеОповещения("ОбработкаВидео", ЭтотОбъект);

Диалог.МножественныйВыбор = Ложь;

Диалог.Показать(ОП);

КонецПроцедуры

&НаКлиенте

Процедура ОбработкаВидео(Результат, ДП) Экспорт

Если не Результат = Неопределено

Тогда

ДД = Новый ДвоичныеДанные(Результат[0]);

ТД = Элементы.СписокВидео.ТекущиеДанные;

ТД.Видео = ЗагрузитьВидео(ТД.Название, ДД)

КонецЕсли;

КонецПроцедуры

Функция ЗагрузитьВидео(Название, ДД)

Спр = Справочники.Файлы.СоздатьЭлемент();

Спр.Наименование = Название;

Спр.Значение = Новый ХранилищеЗначения(ДД);

Спр.Записать();

Возврат Спр.Ссылка

КонецФункции

&НаКлиенте

Процедура ЛоготипНажатие(Элемент, СтандартнаяОбработка)

СтандартнаяОбработка = Ложь;

Диалог = Новый ДиалогВыбораФайла(РежимДиалогаВыбораФайла.Открытие);

Диалог.МножественныйВыбор = Ложь;

ОП = Новый ОписаниеОповещения("ОбработкаФотографии", ЭтотОбъект);

Диалог.Показать(ОП);

КонецПроцедуры

Процедура ОбработкаФотографии(Результат, ДП) Экспорт

Если не Результат = Неопределено

Тогда

Картинка = Новый Картинка(Результат[0]);

СохранитьКартинку(Картинка);

КонецЕсли;

КонецПроцедуры

Процедура СохранитьКартинку(Картинка)

Спр = Справочники.ФайлыЛоготипов.СоздатьЭлемент();

Спр.Наименование = Объект.Наименование;

Спр.Значение = Новый ХранилищеЗначения(Картинка);

Спр.Записать();

Объект.ЛоготипЗначение = Спр.Ссылка;

Объект.Логотип = ПолучитьНавигационнуюСсылку(Объект.ЛоготипЗначение, "Значение");

КонецПроцедуры

&НаКлиенте

Процедура СгенерироватьФото(Команда)

СгенерироватьФотоНаСервере();

КонецПроцедуры

&НаСервере

Процедура СгенерироватьФотоНаСервере()

ЗапуститьПриложение("cmd /k ");

КонецПроцедуры

Документы:

Оплата:

МодульОбъекта:

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

РегистрНач = РегистрыСведений.УчётВзаиморасчётовСоСтудентами;

Документы1 = Документы.Оплата.Выбрать();

ЗначениеЕсть = Ложь;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| УчётВзаиморасчётовСоСтудентами.Студент КАК Студент,

| УчётВзаиморасчётовСоСтудентами.ВсегоПотратил КАК ВсегоПотратил,

| УчётВзаиморасчётовСоСтудентами.Документы КАК Документы

|ИЗ

| РегистрСведений.УчётВзаиморасчётовСоСтудентами КАК УчётВзаиморасчётовСоСтудентами";

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Регистр = РегистрНач.СоздатьМенеджерЗаписи();

Регистр.Студент = ВыборкаДетальныеЗаписи.Студент;

Регистр.ВсегоПотратил = ВыборкаДетальныеЗаписи.ВсегоПотратил;

Регистр.Документы = ВыборкаДетальныеЗаписи.Документы;

Регистр.Прочитать();

Если Регистр.Выбран()

Тогда

Регистр.Удалить();

КонецЕсли;

КонецЦикла;

Пока Документы1.Следующий()

Цикл

Документ = Документы1.ПолучитьОбъект();

Регистр = РегистрНач.СоздатьМенеджерЗаписи();

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| УчётВзаиморасчётовСоСтудентами.Студент КАК Студент,

| УчётВзаиморасчётовСоСтудентами.ВсегоПотратил КАК ВсегоПотратил,

| УчётВзаиморасчётовСоСтудентами.Документы КАК Документы

|ИЗ

| РегистрСведений.УчётВзаиморасчётовСоСтудентами КАК УчётВзаиморасчётовСоСтудентами

|ГДЕ

| УчётВзаиморасчётовСоСтудентами.Студент = &Студент";

Запрос.УстановитьПараметр("Студент", Документ.Студент);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Если не РезультатЗапроса.Пустой()

Тогда

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Регистр = РегистрНач.СоздатьМенеджерЗаписи();

Массив = СтрРазделить(ВыборкаДетальныеЗаписи.Документы, ";");

Для Каждого Элемент Из Массив

Цикл

Если Элемент = Документ.Номер

Тогда

ЗначениеЕсть = Истина;

КонецЕсли;

КонецЦикла;

Если ЗначениеЕсть = Ложь

Тогда

Регистр.ВсегоПотратил = ВыборкаДетальныеЗаписи.ВсегоПотратил;

Регистр.Студент = Документ.Студент;

Регистр.Документы = ВыборкаДетальныеЗаписи.Документы;

Регистр.Прочитать();

Если Регистр.Выбран()

Тогда

Регистр.Документы = Регистр.Документы + Документ.Номер + ";";

Регистр.ВсегоПотратил = Регистр.ВсегоПотратил + Документ.Сумма;

Регистр.Записать();

Иначе

Регистр.ВсегоПотратил = ВыборкаДетальныеЗаписи.ВсегоПотратил;

Регистр.Студент = Документ.Студент;

Регистр.Документы = ВыборкаДетальныеЗаписи.Документы;

Регистр.Записать();

КонецЕсли;

КонецЕсли;

КонецЦикла;

Иначе

Регистр = РегистрНач.СоздатьМенеджерЗаписи();

Регистр.Студент = Документ.Студент;

Регистр.ВсегоПотратил = Документ.Сумма;

Регистр.Документы = Регистр.Документы + Документ.Номер + ";";

Регистр.Записать();

КонецЕсли;

КонецЦикла;

КонецПроцедуры

Процедура ОбработкаПроведения(Отказ, РежимПроведения)

РегистрНач = РегистрыСведений.УчётВзаиморасчётовСоСтудентами;

Документы1 = Документы.Оплата.Выбрать();

ЗначениеЕсть = Ложь;

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| УчётВзаиморасчётовСоСтудентами.Студент КАК Студент,

| УчётВзаиморасчётовСоСтудентами.ВсегоПотратил КАК ВсегоПотратил,

| УчётВзаиморасчётовСоСтудентами.Документы КАК Документы

|ИЗ

| РегистрСведений.УчётВзаиморасчётовСоСтудентами КАК УчётВзаиморасчётовСоСтудентами";

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Регистр = РегистрНач.СоздатьМенеджерЗаписи();

Регистр.Студент = ВыборкаДетальныеЗаписи.Студент;

Регистр.ВсегоПотратил = ВыборкаДетальныеЗаписи.ВсегоПотратил;

Регистр.Документы = ВыборкаДетальныеЗаписи.Документы;

Регистр.Прочитать();

Если Регистр.Выбран()

Тогда

Регистр.Удалить();

КонецЕсли;

КонецЦикла;

Пока Документы1.Следующий()

Цикл

Документ = Документы1.ПолучитьОбъект();

Регистр = РегистрНач.СоздатьМенеджерЗаписи();

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| УчётВзаиморасчётовСоСтудентами.Студент КАК Студент,

| УчётВзаиморасчётовСоСтудентами.ВсегоПотратил КАК ВсегоПотратил,

| УчётВзаиморасчётовСоСтудентами.Документы КАК Документы

|ИЗ

| РегистрСведений.УчётВзаиморасчётовСоСтудентами КАК УчётВзаиморасчётовСоСтудентами

|ГДЕ

| УчётВзаиморасчётовСоСтудентами.Студент = &Студент";

Запрос.УстановитьПараметр("Студент", Документ.Студент);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Если не РезультатЗапроса.Пустой()

Тогда

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Регистр = РегистрНач.СоздатьМенеджерЗаписи();

Массив = СтрРазделить(ВыборкаДетальныеЗаписи.Документы, ";");

Для Каждого Элемент Из Массив

Цикл

Если Элемент = Документ.Номер

Тогда

ЗначениеЕсть = Истина;

КонецЕсли;

КонецЦикла;

Если ЗначениеЕсть = Ложь

Тогда

Регистр.ВсегоПотратил = ВыборкаДетальныеЗаписи.ВсегоПотратил;

Регистр.Студент = Документ.Студент;

Регистр.Документы = ВыборкаДетальныеЗаписи.Документы;

Регистр.Прочитать();

Если Регистр.Выбран()

Тогда

Регистр.Документы = Регистр.Документы + Документ.Номер + ";";

Регистр.ВсегоПотратил = Регистр.ВсегоПотратил + Документ.Сумма;

Регистр.Записать();

Иначе

Регистр.ВсегоПотратил = ВыборкаДетальныеЗаписи.ВсегоПотратил;

Регистр.Студент = Документ.Студент;

Регистр.Документы = ВыборкаДетальныеЗаписи.Документы;

Регистр.Записать();

КонецЕсли;

КонецЕсли;

КонецЦикла;

Иначе

Регистр = РегистрНач.СоздатьМенеджерЗаписи();

Регистр.Студент = Документ.Студент;

Регистр.ВсегоПотратил = Документ.Сумма;

Регистр.Документы = Регистр.Документы + Документ.Номер + ";";

Регистр.Записать();

КонецЕсли;

КонецЦикла;

Движения.ДоходСКурсов.Записывать = Истина;

Движение = Движения.ДоходСКурсов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение.Студент = Студент;

Движение.Курс = Курс;

Движение.Период = Дата;

Движение.Сумма = Сумма;

КонецПроцедуры

ЗаписьНаКурс:

МодульОбъекта:

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| СтудентыНаКурсах.Студент КАК Студент,

| СтудентыНаКурсах.Курс КАК Курс,

| СтудентыНаКурсах.Записан КАК Записан,

| СтудентыНаКурсах.ДатаЗаписи КАК ДатаЗаписи,

| СтудентыНаКурсах.ДатаПоследнейВозможностиОтказа КАК ДатаПоследнейВозможностиОтказа,

| СтудентыНаКурсах.ДатаОтказа КАК ДатаОтказа,

| СтудентыНаКурсах.ДатаОкончанияКурса КАК ДатаОкончанияКурса,

| СтудентыНаКурсах.ПричинаОтказа КАК ПричинаОтказа

|ИЗ

| РегистрСведений.СтудентыНаКурсах КАК СтудентыНаКурсах

|ГДЕ

| СтудентыНаКурсах.Студент = &Студент

| И СтудентыНаКурсах.Курс = &Курс";

Запрос.УстановитьПараметр("Студент", Студент);

Запрос.УстановитьПараметр("Курс", Курс);

РезультатЗапроса = Запрос.Выполнить();

Если не РезультатЗапроса.Пустой()

Тогда

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Регистр = РегистрыСведений.СтудентыНаКурсах.СоздатьМенеджерЗаписи();

КурсС = ВыборкаДетальныеЗаписи.Курс;

СтудентС = ВыборкаДетальныеЗаписи.Студент;

ЗаписанС = ВыборкаДетальныеЗаписи.Записан;

ДатаЗаписиС = ВыборкаДетальныеЗаписи.ДатаЗаписи;

ДатаПоследнейВозможностиОтказаС = ВыборкаДетальныеЗаписи.ДатаПоследнейВозможностиОтказа;

ДатаОтказаС = ВыборкаДетальныеЗаписи.ДатаОтказа;

ДатаОкончанияКурсаС = ВыборкаДетальныеЗаписи.ДатаОкончанияКурса;

ПричинаС = ВыборкаДетальныеЗаписи.ПричинаОтказа;

Регистр.Курс = КурсС;

Регистр.Студент = СтудентС;

Регистр.Записан = ЗаписанС;

Регистр.ДатаЗаписи = ДатаЗаписиС;

Регистр.ДатаПоследнейВозможностиОтказа = ДатаПоследнейВозможностиОтказаС;

Регистр.ДатаОтказа = ДатаОтказаС;

Регистр.ДатаОкончанияКурса = ДатаОкончанияКурсаС;

Регистр.ПричинаОтказа = ПричинаС;

Регистр.Прочитать();

Если Регистр.Выбран()

Тогда

Регистр.Записан = Истина;

Регистр.ДатаЗаписи = Дата;

Регистр.ДатаПоследнейВозможностиОтказа = Регистр.ДатаЗаписи + 3600\*Константы.ЧислоДнейДоДатыПоследнейВозможностиОтказа.Получить()\*24;

Регистр.ДатаОкончанияКурса = Справочники.Курсы.НайтиПоНаименованию(Строка(Регистр.Курс)).ДатаОкончания;

Регистр.Записать();

КонецЕсли;

КонецЦикла;

Иначе

Регистр = РегистрыСведений.СтудентыНаКурсах.СоздатьМенеджерЗаписи();

Регистр.Курс = Курс;

Регистр.Студент = Студент;

Регистр.Записан = Истина;

Регистр.ДатаЗаписи = Дата;

Регистр.ДатаПоследнейВозможностиОтказа = Регистр.ДатаЗаписи + 3600\*Константы.ЧислоДнейДоДатыПоследнейВозможностиОтказа.Получить()\*24;

Регистр.Записать();

КонецЕсли;

КонецПроцедуры

ОтказОтКурса:

МодульОбъекта:

Процедура ПередЗаписью(Отказ, РежимЗаписи, РежимПроведения)

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| СтудентыНаКурсах.Студент КАК Студент,

| СтудентыНаКурсах.Курс КАК Курс,

| СтудентыНаКурсах.Записан КАК Записан,

| СтудентыНаКурсах.ДатаЗаписи КАК ДатаЗаписи,

| СтудентыНаКурсах.ДатаПоследнейВозможностиОтказа КАК ДатаПоследнейВозможностиОтказа,

| СтудентыНаКурсах.ДатаОтказа КАК ДатаОтказа,

| СтудентыНаКурсах.ДатаОкончанияКурса КАК ДатаОкончанияКурса,

| СтудентыНаКурсах.ПричинаОтказа КАК ПричинаОтказа

|ИЗ

| РегистрСведений.СтудентыНаКурсах КАК СтудентыНаКурсах

|ГДЕ

| СтудентыНаКурсах.Студент = &Студент

| И СтудентыНаКурсах.Курс = &Курс";

Запрос.УстановитьПараметр("Студент", Студент);

Запрос.УстановитьПараметр("Курс", Курс);

РезультатЗапроса = Запрос.Выполнить();

Если не РезультатЗапроса.Пустой()

Тогда

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Регистр = РегистрыСведений.СтудентыНаКурсах.СоздатьМенеджерЗаписи();

КурсС = ВыборкаДетальныеЗаписи.Курс;

СтудентС = ВыборкаДетальныеЗаписи.Студент;

ЗаписанС = ВыборкаДетальныеЗаписи.Записан;

ДатаЗаписиС = ВыборкаДетальныеЗаписи.ДатаЗаписи;

ПоследняяДата = ВыборкаДетальныеЗаписи.ДатаПоследнейВозможностиОтказа;

ДатаОтказаС = ВыборкаДетальныеЗаписи.ДатаОтказа;

ДатаОкончанияКурсаС = ВыборкаДетальныеЗаписи.ДатаОкончанияКурса;

ПричинаС = ВыборкаДетальныеЗаписи.ПричинаОтказа;

Регистр.Курс = КурсС;

Регистр.Студент = СтудентС;

Регистр.Записан = ЗаписанС;

Регистр.ДатаЗаписи = ДатаЗаписиС;

Регистр.ДатаПоследнейВозможностиОтказа = ПоследняяДата;

Регистр.ДатаОтказа = ДатаОтказаС;

Регистр.ДатаОкончанияКурса = ДатаОкончанияКурсаС;

Регистр.ПричинаОтказа = ПричинаС;

Регистр.Прочитать();

Если Регистр.Выбран()

Тогда

Если Число(Год(ПоследняяДата) - Год(ТекущаяДата())) >= 0 и Число(Месяц(ПоследняяДата) - Месяц(ТекущаяДата())) >= 0 и Число(День(ПоследняяДата) - День(ТекущаяДата())) >= 0

Тогда

Отказ = Ложь;

Регистр.Записан = Ложь;

Регистр.ДатаОтказа = Дата;

Регистр.ДатаОкончанияКурса = Дата("00010101");

Регистр.Записать();

Иначе

Отказ = Истина;

Сообщить("Студент " + Студент + " не может отказаться от курса " + Курс + ", так как дата возможности отказа истекла в " + Строка(Формат(ПоследняяДата, "ДФ=; ДЛФ=DD")));

КонецЕсли;

КонецЕсли;

КонецЦикла;

Иначе

Сообщить("Студент " + Студент + " не был записан на курс " + Курс);

Отказ = Истина;

КонецЕсли;

КонецПроцедуры

Процедура ОбработкаПроведения()

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЦеныКурсов.Цена КАК Цена

|ИЗ

| РегистрСведений.ЦеныКурсов КАК ЦеныКурсов

|ГДЕ

| ЦеныКурсов.НаименованиеКурса = &НаименованиеКурса";

Запрос.УстановитьПараметр("НаименованиеКурса", Курс.Ссылка);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Сумма = ВыборкаДетальныеЗаписи.Цена;

КонецЦикла;

Движения.ДоходСКурсов.Записывать = Истина;

Движение = Движения.ДоходСКурсов.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Расход;

Движение.Период = Дата;

Движение.Студент = Студент;

Движение.Курс = Курс;

Движение.Сумма = Сумма;

КонецПроцедуры

Отчёты:

Доходы:

ОсновнаяСхема:

ВЫБРАТЬ

ДоходСКурсовОстаткиИОбороты.Студент КАК Студент,

ДоходСКурсовОстаткиИОбороты.Курс КАК Курс,

ДоходСКурсовОстаткиИОбороты.СуммаОборот КАК СуммаОборот

ИЗ

РегистрНакопления.ДоходСКурсов.ОстаткиИОбороты КАК ДоходСКурсовОстаткиИОбороты

ЗаписьНаКурсы:

ОсновнаяСхема:

ВЫБРАТЬ

СтудентыНаКурсах.Студент КАК Студент,

СтудентыНаКурсах.Курс КАК Курс

ИЗ

РегистрСведений.СтудентыНаКурсах КАК СтудентыНаКурсах

ГДЕ

СтудентыНаКурсах.Записан = Истина

ОтказОтКурсов:

ОсновнаяСхема:

ВЫБРАТЬ

СтудентыНаКурсах.Студент КАК Студент,

СтудентыНаКурсах.Курс КАК Курс,

СтудентыНаКурсах.ПричинаОтказа КАК ПричинаОтказа

ИЗ

РегистрСведений.СтудентыНаКурсах КАК СтудентыНаКурсах

ГДЕ

СтудентыНаКурсах.Записан = ЛОЖЬ

Обработки:

ПланировщикЗанятий:

Создаём РеквизитФормы:

Планировщик(Планировщик);

Форма.Модуль:

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

ЗаполнитьПланировщикНаСервере();

Планировщик.ЕдиницаПериодическогоВарианта = ТипЕдиницыШкалыВремени.Час;

Планировщик.КратностьПериодическогоВарианта = 24;

Планировщик.ОтступСНачалаПереносаШкалыВремени = 8;

Планировщик.ОтступСКонцаПереносаШкалыВремени = 9;

Планировщик.ТекущиеПериодыОтображения.Очистить();

НачалоПериода = НачалоНедели(ТекущаяДата());

КонецПериода = КонецНедели(ТекущаяДата());

Планировщик.ТекущиеПериодыОтображения.Добавить(НачалоПериода, КонецПериода);

КонецПроцедуры

Процедура ЗаполнитьПланировщикНаСервере()

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Курсы.Наименование КАК Наименование,

| КурсыЗанятия.Наименование КАК НаименованиеЗанятия,

| КурсыЗанятия.ДатаНачала КАК ДатаНачала,

| КурсыЗанятия.ДатаОкончания КАК ДатаОкончания

|ИЗ

| Справочник.Курсы.Занятия КАК КурсыЗанятия

| ЛЕВОЕ СОЕДИНЕНИЕ Справочник.Курсы КАК Курсы

| ПО КурсыЗанятия.Ссылка = Курсы.Ссылка";

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

ДатаНачала = СтрРазделить(Лев(Прав(Строка(ВыборкаДетальныеЗаписи.ДатаНачала), 8), 5), ":");

ДатаОкончания = Лев(Прав(Строка(ВыборкаДетальныеЗаписи.ДатаОкончания), 8), 5);

ДатаССек = Прав(Строка(ВыборкаДетальныеЗаписи.ДатаНачала), 8);

ЧасыНач = Число(ДатаНачала[0]);

МинутыНач = Число(ДатаНачала[1]);

ГодНач = Год(ВыборкаДетальныеЗаписи.ДатаНачала);

МесяцНач = Месяц(ВыборкаДетальныеЗаписи.ДатаНачала);

ДеньНач = День(ВыборкаДетальныеЗаписи.ДатаНачала);

ДатаОкончания = СтрРазделить(Лев(Прав(Строка(ВыборкаДетальныеЗаписи.ДатаОкончания), 8), 5), ":");

ЧасыКон = Число(ДатаОкончания[0]);

МинутыКон = Число(ДатаОкончания[1]);

ГодКон = Год(ВыборкаДетальныеЗаписи.ДатаОкончания);

МесяцКон = Месяц(ВыборкаДетальныеЗаписи.ДатаОкончания);

ДеньКон = День(ВыборкаДетальныеЗаписи.ДатаОкончания);

ДатаНачала = Дата(ГодНач, МесяцНач, ДеньНач, ЧасыНач, МинутыНач, 00);

ДатаОкончания = Дата(ГодКон, МесяцКон, ДеньКон, ЧасыКон, МинутыКон, 00);

НовыйЭлемент = Планировщик.Элементы.Добавить(ДатаНачала, ДатаОкончания);

НовыйЭлемент.Текст = Строка(ВыборкаДетальныеЗаписи.Наименование) + Символы.ПС + Строка(ВыборкаДетальныеЗаписи.НаименованиеЗанятия);

КонецЦикла;

КонецПроцедуры

УстановкаКоличестваДнейДляВозврата:

Создаём РеквизитыФормы:

КоличествоДней(Число, 10, неотрицательное);

Создаём Команду:

Занести;

Форма.Модуль:

&НаКлиенте

Процедура Занести(Команда)

Проверка = ЗанестиНаСервере();

Если Проверка = 0

Тогда

ПредупреждениеАсинх("Количество дней не может равняться 0", 5, "Ошибка");

ИначеЕсли Проверка = 1

Тогда

ПредупреждениеАсинх("Количество дней не должно быть меньше 7", 5, "Ошибка");

Иначе

Сообщить("Количество дней для возврата установленно на " + КоличествоДней);

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ЗанестиНаСервере()

Если не КоличествоДней = 0

Тогда

Если не КоличествоДней < 7

Тогда

Константы.ЧислоДнейДоДатыПоследнейВозможностиОтказа.Установить(КоличествоДней);

Возврат КоличествоДней;

Иначе

Возврат 1;

КонецЕсли;

Иначе

Возврат 0;

КонецЕсли;

КонецФункции