

//ТЕКСТ МОДУЛЯ ФОРМЫ

&Сервер

Перем ТаблицаОплат;

&НаСервере

Функция ПолучитьПараметрыПодключенияСервер ()

Возврат Новый Структура ("Сервер, База, Пользователь, Пароль", Узел.Сервер, Узел.ИмяБазы, Узел.Пользователь, Узел.Пароль);

КонецФункции

&НаСервере

Функция ПодключитьБазу83()

ОК = Ложь;

CTR = Новый COMОбъект("V83.COMConnector");

ОшибкаПодключения = "";

ДанныеУзла = ПолучитьПараметрыПодключенияСервер();

//ПутьКонфигурации = """" + КаталогПрограммы() + """" + "enterprise /D" +""""+ СокрЛП(ПутьКБазе)+"""" + "\ /N" + Пользователь + " \ /P"+Пароль;

Попытка

//Если НЕ Файловая Тогда

Бух = CTR.Connect("Srvr="""+СокрЛП(ДанныеУзла.Сервер)+""";Ref="""+СокрЛП(ДанныеУзла.База)+""";Usr="""+СокрЛП(ДанныеУзла.Пользователь)+""";Pwd="""+СокрЛП(ДанныеУзла.Пароль)+""";");

//Иначе

// ОК = CTR.Connect("File="""+СокрЛП(Путь)+""";Usr="""+СокрЛП(Пользователь)+""";Pwd="""+СокрЛП(Пароль)+""";");

//КонецЕсли;

ОК = Истина;

Исключение

ОшибкаПодключения = ОписаниеОшибки();

КонецПопытки;

//ОК = Бух.Initialize(Бух.RMTrade, ПутьКонфигурации, "NO\_SPLASH\_SHOW");

Если ОК Тогда

Возврат Бух;

Иначе

Возврат Неопределено

КонецЕсли;

КонецФункции

&НаСервере

Процедура ПриСозданииНаСервере(Отказ, СтандартнаяОбработка)

Если Параметры.Свойство("Узел") Тогда

Если Параметры.Узел = Неопределено Тогда

Выборка = ПланыОбмена.цка\_Обмен\_с\_ERP.Выбрать();

Пока Выборка.Следующий() Цикл

Если НЕ Выборка.ЭтотУзел Тогда

Узел = вЫБОРКА.Ссылка;

Прервать;

КонецЕсли;

КонецЦикла;

Иначе

Узел = Параметры.Узел;

КонецЕсли;

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПодготовитьДанныеКОтправкеСервер()

ЗаписьСообщения = ПланыОбмена.СоздатьЗаписьСообщения();

ВременныйФайл = ПолучитьИмяВременногоФайла("xml");

ЗаписьXML = Новый ЗаписьXML;

ЗаписьXML.ОткрытьФайл(ВременныйФайл);

ЗаписьXML.ЗаписатьОбъявлениеXML();

ЗаписьСообщения.НачатьЗапись(ЗаписьXML, Узел);

ВыборкаИзменений = ПланыОбмена.ВыбратьИзменения(ЗаписьСообщения.Получатель, ЗаписьСообщения.НомерСообщения);

Пока ВыборкаИзменений.Следующий() Цикл

ЗаписатьXML(ЗаписьXML, ВыборкаИзменений.Получить());

КонецЦикла;

ЗаписьСообщения.ЗакончитьЗапись();

ЗаписьXML.Закрыть();

Возврат Истина;

КонецФункции

&НаСервере

Функция СоздатьЭлементСправочника(Бух, Данные, МТ, Спр)

Спр.ПометкаУдаления = Данные.ПометкаУдаления;

Спр.Наименование = Данные.Наименование;

Если МТ.Иерархический Тогда

Если МТ.ВидИерархии = Бух["Метаданные"].ObjectProperties.ВидИерархии.ИерархияГруппИЭлементов

и МТ.КоличествоУровней > 0 Тогда

Если Данные.Родитель.Пустая() Тогда

Спр.Родитель = Справочники[МТ.Имя].ПустаяСсылка();

Иначе

Спр.Родитель = ЗагрузитьОбъект(Бух, Данные.Родитель);

КонецЕсли;

КонецЕсли;

КонецЕсли;

//метаданные.Справочники.Должности.Иерархический

Если МТ.ДлинаКода > 0 Тогда

Спр.Код = Данные.Код;

КонецЕсли;

Для каждого реквизит из МТ.Реквизиты Цикл

Если Бух.XMLTypeOf(Данные[реквизит.Имя]).ИмяТипа = Бух.XMLTypeOf(Данные.Ссылка).ИмяТипа

и Бух.ЗначениеВСтрокуВнутр(Данные[реквизит.Имя]) = Бух.ЗначениеВСтрокуВнутр(Данные.Ссылка) Тогда

Спр[реквизит.Имя] = Спр.Ссылка;

Иначе

Спр[реквизит.Имя] = ЗагрузитьОбъект(Бух, Данные[реквизит.Имя]);

КонецЕсли;

КонецЦикла;

Для Каждого ТЧ из МТ.ТабличныеЧасти Цикл

Спр[ТЧ.Имя].Очистить();

Для каждого стрОЛЕ из Данные[ТЧ.Имя] Цикл

стр = Спр[ТЧ.Имя].Добавить();

Для Каждого Колонка из ТЧ.Реквизиты Цикл

стр[Колонка.Имя] = ЗагрузитьОбъект(Бух, стрОЛЕ[Колонка.Имя]);

КонецЦикла;

КонецЦикла;

КонецЦикла;

КонецФункции

&НаСервере

Функция ЗагрузитьОбъект(Бух, Данные, Загрузка = Ложь)

Перем Удаление;

Удаление = Ложь;

ТипДанных = Бух.XMLTypeOf (Данные);

ТипДанных = ТипДанных.ИмяТипа;

Если ТипДанных = "УдалениеОбъекта" Тогда

Удаление = Истина;

Данные = Данные.Ссылка;

ТипДанных = Бух.XMLTypeOf (Данные);

ТипДанных = ТипДанных.ИмяТипа;

КонецЕсли;

Если ТипДанных = Неопределено или ТипДанных = "Строка" или ТипДанных = "Булево" или ТипДанных = "Число" или ТипДанных = "Дата" Тогда

Возврат Данные;

ИначеЕсли ТипДанных = "УникальныйИдентификатор" Тогда

Возврат Бух.NewObject("УникальныйИдентификатор", Строка(Данные));

КонецЕсли;

МТ = Данные.Метаданные();

ПолноеИмя = МТ.ПолноеИмя();

Н = Найти(ПолноеИмя,".");

ТипОбъкта = Лев(ПолноеИмя, Н-1);

ИмяОбъекта = Сред(ПолноеИмя, Н+1);

Если ТипОбъкта = "Документ" Тогда

Если Данные.Ссылка.Пустая() Тогда

Возврат Бух["Документы"][ИмяОбъекта].ПустаяСсылка();

КонецЕсли;

Если ИмяОбъекта = "АпкПутевойЛистАвтотранспорта" Тогда

ФизЛицо = ЗагрузитьОбъект(Бух, Данные.Водитель);

ИначеЕсли Бух["Метаданные"]["ПланыОбмена"]["ЦКА\_Обмен\_ЗУП\_ERP"].Состав.Найти(МТ) = Неопределено Тогда

Возврат Документы[ИмяОбъекта].ПустаяСсылка();

Иначе

УИД = Новый УникальныйИдентификатор (Бух.String(Данные.Ссылка.УникальныйИдентификатор()));

ОбъектСсылка = Документы[ИмяОбъекта].ПолучитьСсылку(УИД);

Если Найти (Строка(ОбъектСсылка), "Объект не найден") > 0 Тогда

Док = Документы[ИмяОбъекта].СоздатьДокумент();

Док.УстановитьСсылкуНового (ОбъектСсылка);

Иначе

Док = ОбъектСсылка.ПолучитьОбъект();

КонецЕсли;

Док.Номер = Данные.Номер;

Док.Дата = Данные.Дата;

Док.Проведен = Данные.Проведен;

Док.ПометкаУдаления = Данные.ПометкаУдаления;

Док.Записать();

//Если Данные.ПометкаУдаления Тогда

// ДокОЛЕ.УстановитьПометкуУдаления(Истина);

//ИначеЕсли ДокОЛЕ.ПометкаУдаления Тогда

// ДокОЛЕ.УстановитьПометкуУдаления(Ложь);

//КонецЕсли;

Для каждого реквизит из МТ.Реквизиты Цикл

Док[реквизит.Имя] = ЗагрузитьОбъект(Бух, Данные[реквизит.Имя]);

КонецЦикла;

Для Каждого ТЧ из МТ.ТабличныеЧасти Цикл

Док[ТЧ.Имя].Очистить();

Для каждого стрОЛЕ из Данные[ТЧ.Имя] Цикл

стр = Док[ТЧ.Имя].Добавить();

Для Каждого Колонка из ТЧ.Реквизиты Цикл

стр[Колонка.Имя] = ЗагрузитьОбъект(Бух, стрОЛЕ[Колонка.Имя]);

КонецЦикла;

КонецЦикла;

КонецЦикла;

Док.Записать();

Возврат Док.Ссылка;

КонецЕсли;

ИначеЕсли ТипОбъкта = "Справочник" Тогда

Если Метаданные.Справочники.Найти(ИмяОбъекта) = Неопределено Тогда

Возврат Справочники[ИмяОбъекта].ПустаяСсылка();

КонецЕсли;

Если Данные.Ссылка.Пустая() Тогда

Возврат Справочники[ИмяОбъекта].ПустаяСсылка();

ИначеЕсли ИмяОбъекта = "Организации" или ИмяОбъекта = "КлассификаторБанков" Тогда

Возврат Справочники[ИмяОбъекта].НайтиПоРеквизиту("ИНН", Данные.ИНН);

ИначеЕсли ИмяОбъекта = "ФизическиеЛица" Тогда

УИД = Новый УникальныйИдентификатор (Бух.String(Данные.Ссылка.УникальныйИдентификатор()));

ОбъектСсылка = Справочники[ИмяОбъекта].ПолучитьСсылку(УИД);

Если Найти (Строка(ОбъектСсылка), "Объект не найден") > 0 Тогда

Если Данные.ЭтоГруппа Тогда

Спр = Справочники[ИмяОбъекта].НайтиПоНаименованию (Данные.Наименование, Истина);

Если Спр.Пустая() Тогда

Спр = Справочники[ИмяОбъекта].СоздатьГруппу();

Спр.УстановитьСсылкуНового (ОбъектСсылка);

Спр.Наименование = Данные.Наименование;

Если НЕ Данные.Родитель.Пустая() Тогда

Спр.Родитель = ЗагрузитьОбъект(Бух, Данные.Родитель);

КонецЕсли;

Спр.Записать();

КонецЕсли;

Возврат Спр.Ссылка;

КонецЕсли;

Если НЕ ПустаяСтрока(Данные.ИНН) Тогда

Спр = Справочники[ИмяОбъекта].НайтиПоРеквизиту("ИНН", Данные.ИНН);

Иначе

Спр = Справочники.ФизическиеЛица.ПустаяСсылка();

КонецЕсли;

Если Спр.ПУстая() Тогда

ЗапросОЛЕ = Бух.NewObject("Запрос");

ЗапросОЛЕ.Текст =

"ВЫБРАТЬ

| ФИОФизическихЛицСрезПоследних.Фамилия КАК Фамилия,

| ФИОФизическихЛицСрезПоследних.Имя КАК Имя,

| ФИОФизическихЛицСрезПоследних.Отчество КАК Отчество,

| ФИОФизическихЛицСрезПоследних.ФизическоеЛицо.ДатаРождения КАК ДатаРождения

|ИЗ

| РегистрСведений.ФИОФизическихЛиц.СрезПоследних(&Дата, ФизическоеЛицо = &ФизическоеЛицо) КАК ФИОФизическихЛицСрезПоследних";

ЗапросОЛЕ.УстановитьПараметр("ФизическоеЛицо", Данные.Ссылка);

ЗапросОЛЕ.УстановитьПараметр("Дата", ТекущаяДата());

ФИО = ЗапросОЛЕ.Выполнить().Выбрать();

Если ФИО.Следующий() Тогда

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ФИОФизическихЛиц.ФизическоеЛицо КАК ФизическоеЛицо

|ИЗ

| РегистрСведений.ФИОФизическихЛиц КАК ФИОФизическихЛиц

|ГДЕ

| ФИОФизическихЛиц.Фамилия = &Фамилия

| И ФИОФизическихЛиц.Имя = &Имя

| И ФИОФизическихЛиц.Отчество = &Отчество

| И ФИОФизическихЛиц.ФизическоеЛицо.ДатаРождения = &ДатаРождения";

Запрос.УстановитьПараметр("ДатаРождения", ФИО.ДатаРождения);

Запрос.УстановитьПараметр("Фамилия", ФИО.Фамилия);

Запрос.УстановитьПараметр("Имя", ФИО.Имя);

Запрос.УстановитьПараметр("Отчество", ФИО.Отчество);

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

Если Выборка.Следующий() Тогда

Спр = Выборка.ФизическоеЛицо;

Иначе

Сообщ = Новый СообщениеПользователю;

Сообщ.Текст = "Не найдено физ. лицо: "+ФИО.Фамилия+ " "+ФИО.Имя+" "+ФИО.Отчество+ " "+ФИО.ДатаРождения;

Сообщ.Сообщить();

Возврат Справочники[ИмяОбъекта].ПустаяСсылка();;

КонецЕсли;

КонецЕсли;

КонецЕсли;

Если НЕ Спр.Пустая() и Загрузка Тогда

Спр = Спр.ПолучитьОбъект();

СоздатьЭлементСправочника(Бух, Данные, МТ, Спр)

КонецЕсли;

Иначе

Спр = ОбъектСсылка.Ссылка.ПолучитьОбъект();

Если Загрузка Тогда

СоздатьЭлементСправочника(Бух, Данные, МТ, Спр)

КонецЕсли;

КонецЕсли;

Возврат Спр.Ссылка;

ИначеЕсли ИмяОбъекта = "ЗарплатныеПроекты" Тогда

Банк = ЗагрузитьОбъект(Бух, Данные.Банк);

Если Банк.Ссылка.Пустая() Тогда

Возврат Справочники[ИмяОбъекта].ПустаяСсылка();

КонецЕсли;

Возврат Справочники[ИмяОбъекта].НайтиПоРеквизиту("Банк", Банк);

ИначеЕсли ИмяОбъекта = "ПодразделенияОрганизаций" или ИмяОбъекта = "СпособыВыплатыЗарплаты" Тогда

Возврат Справочники[ИмяОбъекта].НайтиПоНаименованию(Данные.Наименование, Истина);

ИначеЕсли ИмяОбъекта = "Пользователи" Тогда

ФЛ = ЗагрузитьОбъект(Бух, Данные.ФизическоеЛицо);

Если НЕ ФЛ.Ссылка.Пустая() Тогда

Возврат Справочники[ИмяОбъекта].НайтиПоРеквизиту("ФизическоеЛицо", ФЛ);

КонецЕсли;

ИначеЕсли ИмяОбъекта = "Сотрудники" Тогда

ФЛ = ЗагрузитьОбъект(Бух, Данные.ФизическоеЛицо);

Если ФЛ.Пустая () Тогда

Возврат Справочники[ИмяОбъекта].ПустаяСсылка();

КонецЕсли;

УИД = Новый УникальныйИдентификатор (Бух.String(Данные.Ссылка.УникальныйИдентификатор()));

ОбъектСсылка = Справочники[ИмяОбъекта].ПолучитьСсылку(УИД);

Если Найти (Строка(ОбъектСсылка), "Объект не найден") > 0 Тогда

Спр = Справочники[ИмяОбъекта].НайтиПоРеквизиту("ФизическоеЛицо", ФЛ);

Если Спр.Пустая() Тогда

Сообщ = Новый СообщениеПользователю;

Сообщ.Текст = "Не найден сотрудник: "+ФЛ;

Сообщ.Сообщить();

Возврат Справочники[ИмяОбъекта].ПустаяСсылка();;

КонецЕсли;

Иначе

Спр = ОбъектСсылка.Ссылка.ПолучитьОбъект();

Если Загрузка Тогда

СоздатьЭлементСправочника(Бух, Данные, МТ, Спр)

КонецЕсли;

КонецЕсли;

Возврат Спр.Ссылка;

ИначеЕсли ИмяОбъекта = "Должности" или ИмяОбъекта = "ГрафикиРаботыСотрудников" Тогда

УИД = Новый УникальныйИдентификатор (Бух.String(Данные.Ссылка.УникальныйИдентификатор()));

ОбъектСсылка = Справочники[ИмяОбъекта].ПолучитьСсылку(УИД);

Если Найти (Строка(ОбъектСсылка), "Объект не найден") > 0 Тогда

Спр = Справочники[ИмяОбъекта].НайтиПоНаименованию(Данные.Наименование);

Если Спр.Пустая() Тогда

Если Данные.ЭтоГруппа Тогда

Спр = Справочники[ИмяОбъекта].СоздатьГруппу();

Иначе

Спр = Справочники[ИмяОбъекта].СоздатьЭлемент();

КонецЕсли;

Спр.УстановитьСсылкуНового (ОбъектСсылка);

СоздатьЭлементСправочника(Бух, Данные, МТ, Спр);

Спр.Записать();

ИначеЕсли Загрузка Тогда

Спр = Спр.ПолучитьОбъект();

СоздатьЭлементСправочника(Бух, Данные, МТ, Спр);

КонецЕсли;

Иначе

Спр = ОбъектСсылка.Ссылка.ПолучитьОбъект();

Если Загрузка Тогда

СоздатьЭлементСправочника(Бух, Данные, МТ, Спр)

КонецЕсли;

КонецЕсли;

Возврат Спр.Ссылка;

ИначеЕсли ИмяОбъекта = "СтруктураПредприятия" или ИмяОбъекта = "ВидыРаботСотрудников" Тогда

УИД = Новый УникальныйИдентификатор (Бух.String(Данные.Ссылка.УникальныйИдентификатор()));

ОбъектСсылка = Справочники[ИмяОбъекта].ПолучитьСсылку(УИД);

Если Найти (Строка(ОбъектСсылка), "Объект не найден") > 0 Тогда

Спр = Справочники[ИмяОбъекта].НайтиПоНаименованию(Данные.Наименование);

Если Спр.Пустая() Тогда

Если Данные.ЭтоГруппа Тогда

Спр = Справочники[ИмяОбъекта].СоздатьГруппу();

Иначе

Спр = Справочники[ИмяОбъекта].СоздатьЭлемент();

КонецЕсли;

Спр.УстановитьСсылкуНового (ОбъектСсылка);

Спр.Наименование = Данные.Наименование;

Если МТ.Иерархический Тогда

Если МТ.КоличествоУровней > 0 Тогда

Если Данные.Родитель.Пустая() Тогда

Спр.Родитель = Справочники[МТ.Имя].ПустаяСсылка();

Иначе

Спр.Родитель = ЗагрузитьОбъект(Бух, Данные.Родитель);

КонецЕсли;

КонецЕсли;

КонецЕсли;

Спр.Записать();

Если ИмяОбъекта = "ВидыРаботСотрудников" Тогда

Расценки = РегистрыСведений.РасценкиРаботСотрудников.СоздатьНаборЗаписей();

Расценки.Отбор.ВидРабот.Установить(Спр.Ссылка);

ЗаписьРегистра = Расценки.Добавить();

ЗаписьРегистра.Период = НачалоГода(ДобавитьМесяц(ТекущаяДата(), - 2));

ЗаписьРегистра.ВидРабот = Спр.Ссылка;

ЗаписьРегистра.Расценка = 1;

Расценки.Записать();

КонецЕсли;

КонецЕсли;

Иначе

Спр = ОбъектСсылка.Ссылка.ПолучитьОбъект();

КонецЕсли;

Возврат Спр.Ссылка;

Иначе

Сообщ = Новый СообщениеПользователю;

Сообщ.Текст = "не найдено правило для "+ИмяОбъекта;

Сообщ.Сообщить();;

Возврат Справочники[ИмяОбъекта].ПустаяСсылка();

КонецЕсли;

ИначеЕсли ТипОбъкта = "Перечисление" Тогда

Если Данные.Пустая() Тогда

Возврат Перечисления[ИмяОбъекта].ПустаяСсылка();

КонецЕсли;

МТ\_Перечисления = Метаданные.Перечисления[МТ.Имя].ЗначенияПеречисления[Бух.XMLString(Данные)];

Индекс = Метаданные.Перечисления[МТ.Имя].ЗначенияПеречисления.Индекс(МТ\_Перечисления);

Возврат Перечисления[МТ.Имя].Получить(Индекс);

ИначеЕсли ТипОбъкта = "РегистрНакопления" Тогда

Если ИмяОбъекта = "АпкСводОплатыТруда" Тогда

Таб = ТаблицаОплат;

Если Данные.Количество() = 0 Тогда

ДокОЛЕ = Данные.Отбор.Регистратор.Значение;

ТипОЛЕ = ДокОЛЕ.Метаданные().Имя;

СтрТаб = Таб.Добавить();

СтрТаб.Организация = ЗагрузитьОбъект(Бух, ДокОЛЕ.Организация);

СтрТаб.Период = НачалоДня(ДокОЛЕ.Дата);

Если ТипОЛЕ = "АпкПутевойЛистАвтотранспорта" Тогда

СтрТаб.Сотрудник = ЗагрузитьОбъект(Бух, ДокОЛЕ.Водитель);

ИначеЕсли ТипОЛЕ = "АпкПутевойЛистТракториста" Тогда

СтрТаб.Сотрудник = ЗагрузитьОбъект(Бух, ДокОЛЕ.Механизатор);

ИначеЕсли ТипОЛЕ = "АпкНарядНаСдельнуюРаботу" Тогда

СтрТаб.Сотрудник = ЗагрузитьОбъект(Бух, ДокОЛЕ.Сотрудник);

КонецЕсли;

СтрТаб.Удаление = Истина;

Иначе

Для каждого Запись из Данные Цикл

СтрТаб = Таб.Добавить();

СтрТаб.Организация = ЗагрузитьОбъект(Бух, Запись.Организация);

СтрТаб.Сотрудник = ЗагрузитьОбъект(Бух, Запись.Сотрудник);

СтрТаб.Период = НачалоДня(Запись.Период);

СтрТаб.ВидРаботы = ЗагрузитьОбъект(Бух, Запись.ВидРаботы);

СтрТаб.Начислено = Запись.Начислено;

СтрТаб.Подразделение = ЗагрузитьОбъект(Бух, Запись.Подразделение);

СтрТаб.СпособОтражения = Найти\_Способ\_Отражения(Бух, Запись.ТипЗатрат, Запись.АналитикаРасходов);

СтрТаб.Удаление = Ложь;

КонецЦикла;

КонецЕсли;

Иначе

Рег = РегистрыНакопления[ИмяОбъекта].СоздатьНаборЗаписей();

Для каждого Запись из Данные Цикл

КонецЦикла;

КонецЕсли;

ИначеЕсли ТипОбъкта = "РегистрСведений" Тогда

Рег = РегистрыНакопления[ИмяОбъекта].СоздатьНаборЗаписей();

Для каждого Запись из Данные Цикл

КонецЦикла;

КонецЕсли;

КонецФункции

&НаСервере

Функция Найти\_Способ\_Отражения (Бух, ТипЗатрат, АналитикаРасходов)

Имя\_Способа = "";

Имя\_Родителя = "";

Если Бух.String(ТипЗатрат) = "Прямые затраты" Тогда

Имя\_Способа = Имя\_Способа + "20 - ";

Имя\_Родителя = "Прямые расходы";

ИначеЕсли Бух.String(ТипЗатрат) = "Общепроизводственные затраты" Тогда

Имя\_Способа = Имя\_Способа + "25 - ";

Имя\_Родителя = "Общепроизводственные затраты";

ИначеЕсли Бух.String(ТипЗатрат) = "Общехозяйственные затраты" Тогда

Имя\_Способа = Имя\_Способа + "26 - ";

Имя\_Родителя = "Общехозяйственные затраты";

ИначеЕсли Бух.String(ТипЗатрат) = "Затраты на ремонт" Тогда

Имя\_Способа = Имя\_Способа + "23 - ";

Имя\_Родителя = "Затраты на ремонт";

ИначеЕсли Бух.String(ТипЗатрат) = "Коммерческие расходы" Тогда

Имя\_Способа = Имя\_Способа + "44 - ";

Имя\_Родителя = "Коммерческие расходы";

КонецЕсли;

Имя\_Способа = Имя\_Способа + АналитикаРасходов.Наименование;

Способ = Справочники.СпособыОтраженияЗарплатыВБухУчете.НайтиПоНаименованию(Имя\_Способа, Истина);

Если Способ.Пустая() Тогда

Способ = Справочники.СпособыОтраженияЗарплатыВБухУчете.СоздатьЭлемент();

Способ.Наименование = Имя\_Способа;

Способ.Родитель = Справочники.СпособыОтраженияЗарплатыВБухУчете.НайтиПоНаименованию(Имя\_Родителя,Истина);

Иначе

Возврат Способ;

КонецЕсли;

УИД\_ОЛЕ = Бух.String(АналитикаРасходов.Ссылка.УникальныйИдентификатор());

стр = Способ.ДополнительныеРеквизиты.Добавить();

стр.Свойство = ПланыВидовХарактеристик.ДополнительныеРеквизитыИСведения.НайтиПоНаименованию("УИД\_Подразделения");

стр.Значение = УИД\_ОЛЕ;

стр.ТекстоваяСтрока = УИД\_ОЛЕ;

Способ.Записать();

Возврат Способ.Ссылка;

КонецФункции

&НаСервере

Функция ОтправитьДанныеНаСервере()

ДанныеОВыполнении = Новый Структура ("Индикатор, Состояние, Ошибка, ВидимостьИндикатора, ОбработкаЗавершена", 0, "", Ложь, Ложь, Ложь);

АдресХранилища = ПоместитьВоВременноеХранилище(ДанныеОВыполнении, ЭтаФорма.УникальныйИдентификатор);

МассивПараметров = Новый Массив;

МассивПараметров.Добавить(Узел);

МассивПараметров.Добавить(АдресХранилища);

ФЗ = ФоновыеЗадания.Выполнить("цка\_ОбщиеПроцедуры\_Сервер.Выполнить\_Выгрузку\_Данных\_ERP",МассивПараметров, "Выгрузка в ERP");

УИД\_ФоновогоЗадания = ФЗ.УникальныйИдентификатор;

КонецФункции

&НаСервере

Процедура ЗавершитьОперациюОтправки()

// Вставить содержимое обработчика.

//УдалитьФайлы(ВременныйФайл);

КонецПроцедуры

&НаКлиенте

Процедура ОтправитьДанные(Команда)

Элементы.Индикатор.Видимость = Ложь;

Элементы.Состояние.Заголовок = "";

Элементы.Состояние.ЦветТекста = WebЦвета.Черный;

ОтправитьДанныеНаСервере();

ПодключитьОбработчикОжидания("ОбработчикОжидания",5,Ложь);

Элементы.ОтправитьДанные.Доступность = Ложь;

КонецПроцедуры

&НаСервере

Процедура ЗагрузитьДанныеНаСервере()

Бух = ПодключитьБазу83 ();

Если НЕ ПустаяСтрока(ОшибкаПодключения) Тогда

Возврат;

КонецЕсли;

ВыборкаУзлов = ПланыОбмена.цка\_Обмен\_с\_ERP.Выбрать();

Пока ВыборкаУзлов.Следующий() Цикл

Если ВыборкаУзлов.ЭтотУзел Тогда

ЭтотУзел = ВыборкаУзлов.Ссылка;

Прервать;

КонецЕсли;

КонецЦикла;

ВыборкаОЛЕ = Бух["ПланыОбмена"].ЦКА\_Обмен\_ЗУП\_ERP.Выбрать();

Пока ВыборкаОЛЕ.Следующий() Цикл

Если ВыборкаОЛЕ.ЭтотУзел Тогда

Продолжить;

КонецЕсли;

Если ВыборкаОЛЕ.Код = ЭтотУзел.Код Тогда

УзелОЛЕ = ВыборкаОЛЕ.Ссылка;

Прервать;

КонецЕсли;

КонецЦикла;

Если НЕ Бух.ValueIsFilled(УзелОЛЕ.ссылка) Тогда

Сообщ = Новый СообщениеПользователю;

Сообщ.Текст = "Не найден узел обмена в ERP с кодом "+ЭтотУзел.Код;

Сообщ.Сообщить();

Возврат;

КонецЕсли;

ВыборкаИзменений = Бух["ПланыОбмена"].ВыбратьИзменения(УзелОЛЕ, 1);

//ВыборкаИзменений = Бух.ExchangePlans.ВыбратьИзменения(УзелОЛЕ.Ссылка, 1, Неопределено);

ТаблицаОплат = Новый ТаблицаЗначений;

ТаблицаОплат.Колонки.Добавить("Организация");

ТаблицаОплат.Колонки.Добавить("Сотрудник");

ТаблицаОплат.Колонки.Добавить("Период");

ТаблицаОплат.Колонки.Добавить("ВидРаботы");

ТаблицаОплат.Колонки.Добавить("Подразделение");

ТаблицаОплат.Колонки.Добавить("Начислено");

ТаблицаОплат.Колонки.Добавить("СпособОтражения");

ТаблицаОплат.Колонки.Добавить("Удаление");

Пока ВыборкаИзменений.Следующий() Цикл

Попытка

Данные = ВыборкаИзменений.Получить();

ЗагрузитьОбъект(Бух, Данные, Истина);

Бух["ПланыОбмена"].УдалитьРегистрациюИзменений(УзелОЛЕ, Данные);

Исключение

Сообщ = Новый СообщениеПользователю;

Сообщ.Текст = ОписаниеОшибки();

Сообщ.Сообщить();

КонецПопытки;

КонецЦикла;

ЗагрузитьНаряды();

Бух = Неопределено;

КонецПроцедуры

&НаСервере

Процедура ЗагрузитьНаряды ()

ТаблицаОплат.Колонки.Добавить("ФизЛицо");

ТаблицаОплат.Колонки.Добавить("КодФизЛица");

Для каждого стр из ТаблицаОплат Цикл

ДанныкеСотрудника = НайтиСотрудника (стр.Сотрудник, стр.Период, стр.Организация);

Если ДанныкеСотрудника = Неопределено Тогда

стр.ФизЛицо = Справочники.ФизическиеЛица.ПустаяСсылка();

стр.КодФизЛица = "";

ПродолжитЬ;

КонецЕсли;

стр.ФизЛицо = ДанныкеСотрудника.Сотрудник.ФизическоеЛицо;

стр.КодФизЛица = стр.ФизЛицо.Код;

КонецЦикла;

ТаблицаОплат.Сортировать("Организация, Период, КодФизЛица");

Док = Неопределено;

ВидДокумента = Справочники.ВидыДокументовВводДанныхДляРасчетаЗарплаты.НайтиПоНаименованию("Сдельные работы",Истина);

ТЗ = Неопределено;

ТЗ\_ФЛ = Неопределено;

Для каждого стр из ТаблицаОплат Цикл

Если ДатаНачала <> Дата("00010101") и стр.Период < ДатаНачала Тогда

Продолжить;

КонецЕсли;

Если ДатаОкончания <> Дата("00010101") и стр.Период > КонецДня(ДатаОкончания) Тогда

Продолжить;

КонецЕсли;

Если Док = Неопределено или НачалоДня(Док.Дата) <> стр.Период или Док.Организация <> стр.Организация Тогда

Если Док <> Неопределено Тогда

Док.ВыполненныеРаботы.Загрузить(ТЗ);

Док.ФизическиеЛица.Загрузить(ТЗ\_ФЛ);

Док.Записать();

Попытка

Док.Записать(РежимЗаписиДокумента.Проведение);

Исключение

Сообщ = Новый СообщениеПользователю;

Сообщ.Текст = ОписаниеОшибки();

Сообщ.Сообщить();

КонецПопытки;

КонецЕсли;

Док = НайтиНаряд (стр.Период, стр.Организация, ВидДокумента);

Если Док <> Неопределено Тогда

Док = Док.Ссылка.ПолучитьОбъект();

Иначе

Док = Документы.ДанныеДляРасчетаЗарплаты.СоздатьДокумент();

КонецЕсли;

Док.Дата = стр.Период;

Док.Период = стр.Период;

Док.ПериодОкончания = стр.период;

Док.Организация = стр.Организация;

Док.ВидДокумента = ВидДокумента;

Док.ИспользоватьПриРасчетеПервойПоловиныМесяца = ?(День(Док.Дата) < 16, Истина, Ложь);

Док.Ответственный = ПараметрыСеанса.ТекущийПользователь;

//--------------------------------------

Док.ВыполненныеРаботы.Очистить();

Док.ФизическиеЛица.Очистить();

//--------------------------------------

ТЗ = Док.ВыполненныеРаботы.Выгрузить();

ТЗ\_ФЛ = Док.ФизическиеЛица.Выгрузить();

ФизЛицл = "";

КонецЕсли;

Если НЕ ЗначениеЗаполнено(стр.Сотрудник) Тогда

Продолжить;

КонецЕсли;

Если ФизЛицл <> стр.ФизЛицо Тогда

ДанныкеСотрудника = НайтиСотрудника (стр.Сотрудник, стр.Период, стр.Организация);

Если ДанныкеСотрудника = Неопределено Тогда

Продолжить;

КонецЕсли;

ФизЛицл = стр.ФизЛицо;

//--------------------------------------

//Отбор = Новый Структура("ФизическоеЛицо", ФизЛицл);

//МассивСтрок = ТЗ\_ФЛ.НайтиСтроки(Отбор);

//Для каждого стрДок из МассивСтрок Цикл

// ТЗ\_ФЛ.Удалить(стрДок);

//КонецЦикла;

//Отбор = Новый Структура("Сотрудник", ДанныкеСотрудника.Сотрудник);

//МассивСтрок = ТЗ.НайтиСтроки(Отбор);

//Для каждого стрДок из МассивСтрок Цикл

// ТЗ.Удалить(стрДок);

//КонецЦикла;

//--------------------------------------

КонецЕсли;

Если стр.Удаление Тогда

Продолжить;

КонецЕсли;

Если ЗначениеЗаполнено(ДанныкеСотрудника) и ЗначениеЗаполнено(ДанныкеСотрудника.Сотрудник) Тогда

стрДок = ТЗ\_ФЛ.Добавить();

стрДок.ФизическоеЛицо = ФизЛицл;

стрДок = ТЗ.Добавить();

стрДок.Сотрудник = ДанныкеСотрудника.Сотрудник;

//стр.Сотрудник = ДанныкеСотрудника.Сотрудник;

стрДок.ВидРабот = стр.ВидРаботы;

стрДок.ОбъемВыполненныхРабот = стр.Начислено;

стрДок.Расценка = 1;

стрДок.Период = стр.Период;

стрДок.Сумма = стр.Начислено;

стрДок.ПодразделениеУчетаЗатрат = ДанныкеСотрудника.Подразделение;

стрДок.СпособОтраженияЗарплатыВБухучете = стр.СпособОтражения;

КонецЕсли;

КонецЦикла;

Если Док <> Неопределено Тогда

Док.ВыполненныеРаботы.Загрузить(ТЗ);

Док.ФизическиеЛица.Загрузить(ТЗ\_ФЛ);

Док.Записать();

Попытка

Док.Записать(РежимЗаписиДокумента.Проведение);

Исключение

Сообщ = Новый СообщениеПользователю;

Сообщ.Текст = ОписаниеОшибки();

Сообщ.Сообщить();

КонецПопытки;

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция НайтиНаряд (ДатаДок, Организация, ВидДокумента)

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ДанныеДляРасчетаЗарплаты.Ссылка КАК Наряд

|ИЗ

| Документ.ДанныеДляРасчетаЗарплаты КАК ДанныеДляРасчетаЗарплаты

|ГДЕ

| НАЧАЛОПЕРИОДА(ДанныеДляРасчетаЗарплаты.Дата, ДЕНЬ) = &Дата

| И ДанныеДляРасчетаЗарплаты.Организация = &Организация

| И ДанныеДляРасчетаЗарплаты.ВидДокумента = &ВидДокумента";

Запрос.УстановитьПараметр("Дата", ДатаДок);

Запрос.УстановитьПараметр("Организация", Организация);

Запрос.УстановитьПараметр("ВидДокумента", ВидДокумента);

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

Если Выборка.Следующий() Тогда

Возврат Выборка.Наряд;

КонецЕсли;

Возврат Неопределено;

КонецФункции

&НаСервере

Функция НайтиСотрудника (ФизЛицо, ДатаДок, организация)

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| КИ.Сотрудник КАК Сотрудник,

| КИ.Подразделение КАК Подразделение,

| КИ.ВидСобытия КАК ВидСобытия,

| БУ.СпособОтраженияЗарплатыВБухучете КАК СпособОтражения

|ИЗ

| РегистрСведений.КадроваяИсторияСотрудников.СрезПоследних(

| &Дата,

| Активность

| И Организация = &Организация

| И Сотрудник.ФизическоеЛицо = &ФизическоеЛицо) КАК КИ

| ЛЕВОЕ СОЕДИНЕНИЕ РегистрСведений.БухучетЗарплатыПодразделений.СрезПоследних(&Дата, ) КАК БУ

| ПО КИ.Подразделение = БУ.Подразделение

|ГДЕ

| КИ.ВидСобытия <> ЗНАЧЕНИЕ(Перечисление.ВидыКадровыхСобытий.Увольнение)";

Запрос.УстановитьПараметр("Дата", КонецДня(ДатаДок));

Запрос.УстановитьПараметр("Организация", Организация);

Запрос.УстановитьПараметр("ФизическоеЛицо", ФизЛицо);

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

Если Выборка.Следующий() Тогда

Возврат Выборка;

КонецЕсли;

Сообщ = Новый СообщениеПользователю;

Сообщ.Текст = "Не найден сотрудник "+ФизЛицо +" на дату "+ДатаДок;

Сообщ.Сообщить();

Возврат Неопределено;

КонецФункции

&НаКлиенте

Процедура ЗагрузитьДанные(Команда)

ЗагрузитьДанныеНаСервере();

КонецПроцедуры

&НаКлиенте

Процедура ОбработчикОжидания() Экспорт

Если ПустаяСтрока(АдресХранилища) Тогда

ОтключитьОбработчикОжидания("ОбработчикОжидания");

АдресХранилища = "";

Элементы.ОтправитьДанные.Доступность = Истина;

Возврат;

КонецЕсли;

ДанныеОВыполнении = ПолучитьИзВременногоХранилища(АдресХранилища);

Если ТипЗнч(ДанныеОВыполнении) = Тип("Структура") Тогда

ПозицияИндикатора = ДанныеОВыполнении.Индикатор;

Элементы.Индикатор.Видимость = ДанныеОВыполнении.ВидимостьИндикатора;

Элементы.Состояние.Заголовок = ДанныеОВыполнении.Состояние;

Если ДанныеОВыполнении.Ошибка Тогда

Элементы.Состояние.ЦветТекста = WebЦвета.Красный;

КонецЕсли;

Ошибка = Ложь;

Если ДанныеОВыполнении.ОбработкаЗавершена или ПроверитьСостояниеФоновогоЗадания(Ошибка) = Ложь Тогда

ОтключитьОбработчикОжидания("ОбработчикОжидания");

АдресХранилища = "";

Элементы.Индикатор.Видимость = Ложь;

Элементы.ОтправитьДанные.Доступность = Истина;

Если Ошибка Тогда

Элементы.Состояние.Заголовок = "Обработка завершена с ошибками";

Элементы.Состояние.ЦветТекста = WebЦвета.Красный;

КонецЕсли;

КонецЕсли;

Иначе

ОтключитьОбработчикОжидания("ОбработчикОжидания");

Элементы.Состояние.Заголовок = "Обработка завершена";

Элементы.Состояние.ЦветТекста = WebЦвета.Красный;

Элементы.Индикатор.Видимость = Ложь;

АдресХранилища = "";

Элементы.ОтправитьДанные.Доступность = Истина;

КонецЕсли;

КонецПроцедуры

&НаСервере

Процедура ЗавершитьФоновоеЗадание()

Если ЗначениеЗаполнено(УИД\_ФоновогоЗадания) Тогда

ФЗ = ФоновыеЗадания.НайтиПоУникальномуИдентификатору(УИД\_ФоновогоЗадания);

ФЗ.Отменить();

УИД\_ФоновогоЗадания = Неопределено;

КонецЕсли;

КонецПроцедуры

&НаСервере

Функция ПроверитьСостояниеФоновогоЗадания(Ошибка)

Если ЗначениеЗаполнено(УИД\_ФоновогоЗадания) Тогда

ФЗ = ФоновыеЗадания.НайтиПоУникальномуИдентификатору(УИД\_ФоновогоЗадания);

Сообщения = ФЗ.ПолучитьСообщенияПользователю(Истина);

Если Сообщения.Количество() Тогда

Для каждого Эл из Сообщения Цикл

Эл.Сообщить();

КонецЦикла;

КонецЕсли;

Если НЕ ФЗ.Состояние = СостояниеФоновогоЗадания.Активно Тогда

УИД\_ФоновогоЗадания = Неопределено;

Если ФЗ.Состояние = СостояниеФоновогоЗадания.ЗавершеноАварийно Тогда

Ошибка = Истина;

Конецесли;

Возврат Ложь;

КонецЕсли;

Возврат Истина

КонецЕсли;

возврат Ложь;

КонецФункции

&НаКлиенте

Процедура ПередЗакрытием(Отказ, ЗавершениеРаботы, ТекстПредупреждения, СтандартнаяОбработка)

Если НЕ ПустаяСтрока(АдресХранилища) Тогда

ДанныеОВыполнении = ПолучитьИзВременногоХранилища(АдресХранилища);

Если ТипЗнч(ДанныеОВыполнении) = Тип("Структура") Тогда

Если НЕ ДанныеОВыполнении.ОбработкаЗавершена Тогда

ДиалогСВопросом();

Отказ = Истина;

КонецЕсли;

КонецЕсли;

КонецЕсли;

КонецПроцедуры

&НаКлиенте

Процедура ДиалогСВопросом()

Оповещение = Новый ОписаниеОповещения("ПослеЗакрытияВопроса",

ЭтотОбъект);

ПоказатьВопрос(Оповещение,

"Прервать обработку?",

РежимДиалогаВопрос.ДаНет,

60, // таймаут в секундах

КодВозвратаДиалога.Нета, // (необ.) кнопка по умолчанию

"Завершение работы выгрузки" // (необ.) заголовок

);

КонецПроцедуры

&НаКлиенте

Процедура ПослеЗакрытияВопроса(Результат, Параметры) Экспорт

Если Результат = КодВозвратаДиалога.Да Тогда

ЗавершитьФоновоеЗадание();

ЭтаФорма.Закрыть(0);

КонецЕсли;

КонецПроцедуры

&НаСервере

Процедура РегистрацияНарядовНаСервере()

Бух = ПодключитьБазу83 ();

Если НЕ ПустаяСтрока(ОшибкаПодключения) Тогда

Возврат;

КонецЕсли;

ВыборкаУзлов = ПланыОбмена.цка\_Обмен\_с\_ERP.Выбрать();

Пока ВыборкаУзлов.Следующий() Цикл

Если ВыборкаУзлов.ЭтотУзел Тогда

ЭтотУзел = ВыборкаУзлов.Ссылка;

Прервать;

КонецЕсли;

КонецЦикла;

ВыборкаОЛЕ = Бух["ПланыОбмена"].ЦКА\_Обмен\_ЗУП\_ERP.Выбрать();

Пока ВыборкаОЛЕ.Следующий() Цикл

Если ВыборкаОЛЕ.ЭтотУзел Тогда

Продолжить;

КонецЕсли;

Если ВыборкаОЛЕ.Код = ЭтотУзел.Код Тогда

УзелОЛЕ = ВыборкаОЛЕ.Ссылка;

Прервать;

КонецЕсли;

КонецЦикла;

Если НЕ Бух.ValueIsFilled(УзелОЛЕ.ссылка) Тогда

Сообщ = Новый СообщениеПользователю;

Сообщ.Текст = "Не найден узел обмена в ERP с кодом "+ЭтотУзел.Код;

Сообщ.Сообщить();

Возврат;

КонецЕсли;

ТекстЗапроса = "";

МассивТипов = Бух["Метаданные"]["РегистрыНакопления"]["АпкСводОплатыТруда"].StandardAttributes.Recorder.Тип.Типы();

Для каждого Рег из МассивТипов Цикл

ТипСтрокой = Бух.XMLType(Рег).TypeName;

Имя = Сред(ТипСтрокой, Найти(ТипСтрокой, ".")+1);

ТекстЗапроса = ТекстЗапроса + ?(ПустаяСтрока(ТекстЗапроса), "", " ОБЪЕДИНИТЬ ВСЕ ") + "

| ВЫБРАТЬ Док.Ссылка ИЗ Документ."+Имя +" КАК Док ГДЕ Док.Дата МЕЖДУ &НачПериода и &КонПериода

|";

КонецЦикла;

ЗапросОЛЕ = Бух.NewObject("Запрос");

ЗапросОЛЕ.Текст = ТекстЗапроса;

ЗапросОЛЕ.УстановитьПараметр("НачПериода", ДатаНачала);

ЗапросОЛЕ.УстановитьПараметр("КонПериода", КонецДня(ДатаОкончания));

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

//РегОЛЕ.Прочитать();

//ТЗ = РегОЛЕ.Выгрузить();

//ТЗ.Свернуть("Регистратор");

//Н = 0;

//Пока Н < ТЗ.Количество() Цикл

// Если ТЗ[Н].Регистратор.Дата < ДатаНачала или ТЗ[Н].Регистратор.Дата > ДатаОкончания Тогда

// ТЗ.Удалить(ТЗ[Н]);

// Продолжить;

// КонецЕсли;

// Н = Н + 1;

//КонецЦикла;

//Для каждого стр из ТЗ Цикл

РегОЛЕ = Бух["РегистрыНакопления"]["АпкСводОплатыТруда"].СоздатьНаборЗаписей();

Пока Выборка.Следующий() Цикл

РегОЛЕ.Отбор.Регистратор.Установить(Выборка.Ссылка);

РегОЛЕ.Прочитать();

Бух["ПланыОбмена"].ЗарегистрироватьИзменения(УзелОЛЕ, РегОЛЕ);

КонецЦикла;

//ВыборкаИзменений = Бух["ПланыОбмена"].ЗарегистрироватьИзменения(УзелОЛЕ, Бух["Метаданные"]["РегистрыНакопления"]["АпкСводОплатыТруда"]);

КонецПроцедуры

&НаКлиенте

Процедура РегистрацияНарядов(Команда)

РегистрацияНарядовНаСервере();

КонецПроцедуры

&НаСервере

Процедура СоздатьГрафикиНаСервере()

// Вставить содержимое обработчика.

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| Наряды.Сотрудник КАК Сотрудник

|ПОМЕСТИТЬ ВТ\_Сотрудники

|ИЗ

| Документ.ДанныеДляРасчетаЗарплаты.ВыполненныеРаботы КАК Наряды

|ГДЕ

| Наряды.Ссылка.Проведен

| И Наряды.Ссылка.Дата МЕЖДУ &НачПериода И &КонПериода

| И Наряды.Ссылка.ВидДокумента = &ВидДокумента

|

|СГРУППИРОВАТЬ ПО

| Наряды.Сотрудник

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| СостоянияСотрудников.Состояние КАК Состояние,

| СостоянияСотрудников.Сотрудник КАК Сотрудник,

| СостоянияСотрудников.Период КАК Период

|ПОМЕСТИТЬ ВТ\_Состояния

|ИЗ

| РегистрСведений.СостоянияСотрудников КАК СостоянияСотрудников

|ГДЕ

| СостоянияСотрудников.Период МЕЖДУ &НачПериода И &КонПериода

| И СостоянияСотрудников.Сотрудник В

| (ВЫБРАТЬ

| Таб.Сотрудник

| ИЗ

| ВТ\_Сотрудники КАК Таб)

|

|ОБЪЕДИНИТЬ ВСЕ

|

|ВЫБРАТЬ

| СостоянияСотрудниковСрезПоследних.Состояние,

| СостоянияСотрудниковСрезПоследних.Сотрудник,

| &НачПериода

|ИЗ

| РегистрСведений.СостоянияСотрудников.СрезПервых(

| &НачПериода,

| Сотрудник В

| (ВЫБРАТЬ

| Таб.Сотрудник

| ИЗ

| ВТ\_Сотрудники КАК Таб)) КАК СостоянияСотрудниковСрезПоследних

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| ДанныеПроизводственногоКалендаря.Дата КАК Дата

|ПОМЕСТИТЬ ВТ\_Дни

|ИЗ

| РегистрСведений.ДанныеПроизводственногоКалендаря КАК ДанныеПроизводственногоКалендаря

|ГДЕ

| ДанныеПроизводственногоКалендаря.Год = &Год

| И ДанныеПроизводственногоКалендаря.Дата МЕЖДУ &НачПериода И &КонПериода

|

|СГРУППИРОВАТЬ ПО

| ДанныеПроизводственногоКалендаря.Дата

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| ВТ\_Дни.Дата КАК Дата,

| ВТ\_Состояния.Сотрудник КАК Сотрудник,

| ВТ\_Состояния.Период КАК ПериодСостояния,

| ВТ\_Состояния.Состояние КАК Состояние

|ПОМЕСТИТЬ ВТ\_СводПоСотрудникам

|ИЗ

| ВТ\_Дни КАК ВТ\_Дни

| ВНУТРЕННЕЕ СОЕДИНЕНИЕ ВТ\_Состояния КАК ВТ\_Состояния

| ПО ВТ\_Дни.Дата >= ВТ\_Состояния.Период

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| СостояниеНаДату.Дата КАК Дата,

| СостояниеНаДату.Сотрудник КАК Сотрудник,

| СостояниеНаДату.ПериодСостояния КАК ПериодСостояния,

| СостояниеНаДату.Состояние КАК Состояние

|ПОМЕСТИТЬ ВТ\_СостояниеПоДням

|ИЗ

| (ВЫБРАТЬ

| ВТ\_СводПоСотрудникам.Дата КАК Дата,

| ВТ\_СводПоСотрудникам.Сотрудник КАК Сотрудник,

| ВТ\_СводПоСотрудникам.ПериодСостояния КАК ПериодСостояния,

| ВТ\_СводПоСотрудникам.Состояние КАК Состояние

| ИЗ

| ВТ\_СводПоСотрудникам КАК ВТ\_СводПоСотрудникам

| ВНУТРЕННЕЕ СОЕДИНЕНИЕ (ВЫБРАТЬ

| ВТ\_СводПоСотрудникам.Дата КАК Дата,

| МАКСИМУМ(ВТ\_СводПоСотрудникам.ПериодСостояния) КАК ПериодСостояния,

| ВТ\_СводПоСотрудникам.Сотрудник КАК Сотрудник

| ИЗ

| ВТ\_СводПоСотрудникам КАК ВТ\_СводПоСотрудникам

|

| СГРУППИРОВАТЬ ПО

| ВТ\_СводПоСотрудникам.Дата,

| ВТ\_СводПоСотрудникам.Сотрудник) КАК ОтборПоПериодам

| ПО ВТ\_СводПоСотрудникам.Сотрудник = ОтборПоПериодам.Сотрудник

| И ВТ\_СводПоСотрудникам.Дата = ОтборПоПериодам.Дата

| И ВТ\_СводПоСотрудникам.ПериодСостояния = ОтборПоПериодам.ПериодСостояния) КАК СостояниеНаДату

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| ГрафикиРаботыПоВидамВремени.Дата КАК Дата,

| ВЫБОР

| КОГДА ГрафикиРаботыПоВидамВремени.ВидУчетаВремени = ЗНАЧЕНИЕ(Справочник.ВидыИспользованияРабочегоВремени.Явка)

| ТОГДА ИСТИНА

| ИНАЧЕ ЛОЖЬ

| КОНЕЦ КАК Явка,

| СУММА(ГрафикиРаботыПоВидамВремени.ДополнительноеЗначение) КАК Часы,

| ГрафикиРаботыПоВидамВремени.ГрафикРаботы КАК ГрафикРаботы,

| ГрафикРаботыСотрудниковСрезПоследних.Сотрудник КАК Сотрудник,

| ГрафикиРаботыПоВидамВремени.ВидУчетаВремени КАК ВидВремени,

| ГрафикРаботыСотрудниковСрезПоследних.ГоловнаяОрганизация КАК Организация

|ПОМЕСТИТЬ ВТ\_ГрафикиРаботы

|ИЗ

| РегистрСведений.ГрафикРаботыСотрудников.СрезПоследних(

| &КонПериода,

| Сотрудник В

| (ВЫБРАТЬ

| Таб.Сотрудник

| ИЗ

| ВТ\_Сотрудники КАК Таб)) КАК ГрафикРаботыСотрудниковСрезПоследних

| ВНУТРЕННЕЕ СОЕДИНЕНИЕ РегистрСведений.ГрафикиРаботыПоВидамВремени КАК ГрафикиРаботыПоВидамВремени

| ПО ГрафикРаботыСотрудниковСрезПоследних.ГрафикРаботы = ГрафикиРаботыПоВидамВремени.ГрафикРаботы

|ГДЕ

| ГрафикиРаботыПоВидамВремени.Дата МЕЖДУ &НачПериода И &КонПериода

| И (ГрафикиРаботыПоВидамВремени.ВидУчетаВремени = ЗНАЧЕНИЕ(Справочник.ВидыИспользованияРабочегоВремени.Явка)

| ИЛИ ГрафикиРаботыПоВидамВремени.ВидУчетаВремени = ЗНАЧЕНИЕ(Справочник.ВидыИспользованияРабочегоВремени.ВыходныеДни))

|

|СГРУППИРОВАТЬ ПО

| ГрафикиРаботыПоВидамВремени.Дата,

| ГрафикиРаботыПоВидамВремени.ГрафикРаботы,

| ГрафикРаботыСотрудниковСрезПоследних.Сотрудник,

| ГрафикиРаботыПоВидамВремени.ВидУчетаВремени,

| ГрафикРаботыСотрудниковСрезПоследних.ГоловнаяОрганизация,

| ВЫБОР

| КОГДА ГрафикиРаботыПоВидамВремени.ВидУчетаВремени = ЗНАЧЕНИЕ(Справочник.ВидыИспользованияРабочегоВремени.Явка)

| ТОГДА ИСТИНА

| ИНАЧЕ ЛОЖЬ

| КОНЕЦ

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников.Сотрудник КАК Сотрудник,

| ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников.Период КАК Период,

| СУММА(ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников.Значение) КАК Значение,

| ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников.Организация КАК Организация

|ПОМЕСТИТЬ ВТ\_Показатели

|ИЗ

| РегистрНакопления.ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников КАК ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников

|ГДЕ

| ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников.Период МЕЖДУ &НачПериода И &КонПериода

| И ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников.Активность

| И ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников.Сотрудник В

| (ВЫБРАТЬ

| Таб.Сотрудник

| ИЗ

| ВТ\_Сотрудники КАК Таб)

| И ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников.Показатель = &Показатель

|

|СГРУППИРОВАТЬ ПО

| ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников.Сотрудник,

| ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников.Период,

| ЗначенияОперативныхПоказателейРасчетаЗарплатыСотрудников.Организация

|;

|

|////////////////////////////////////////////////////////////////////////////////

|ВЫБРАТЬ

| ВТ\_ГрафикиРаботы.Дата КАК Дата,

| ВТ\_ГрафикиРаботы.Сотрудник КАК Сотрудник,

| СУММА(ВТ\_ГрафикиРаботы.Часы) КАК Часы,

| ВЫБОР

| КОГДА ВТ\_Показатели.Значение ЕСТЬ NULL

| ТОГДА ВЫБОР

| КОГДА ВТ\_СостояниеПоДням.Сотрудник.Наименование = ""Работает""

| ТОГДА ЗНАЧЕНИЕ(Справочник.ВидыИспользованияРабочегоВремени.ВыходныеДни)

| ИНАЧЕ ВТ\_ГрафикиРаботы.ВидВремени

| КОНЕЦ

| ИНАЧЕ ЗНАЧЕНИЕ(Справочник.ВидыИспользованияРабочегоВремени.Явка)

| КОНЕЦ КАК ВидВремени,

| НАЧАЛОПЕРИОДА(ВТ\_ГрафикиРаботы.Дата, МЕСЯЦ) КАК ПериодРегистрации,

| ВТ\_Показатели.Организация КАК Организация

|ИЗ

| ВТ\_ГрафикиРаботы КАК ВТ\_ГрафикиРаботы

| ВНУТРЕННЕЕ СОЕДИНЕНИЕ ВТ\_СостояниеПоДням КАК ВТ\_СостояниеПоДням

| ПО ВТ\_ГрафикиРаботы.Сотрудник = ВТ\_СостояниеПоДням.Сотрудник

| И ВТ\_ГрафикиРаботы.Дата = ВТ\_СостояниеПоДням.Дата

| ЛЕВОЕ СОЕДИНЕНИЕ ВТ\_Показатели КАК ВТ\_Показатели

| ПО ВТ\_ГрафикиРаботы.Сотрудник = ВТ\_Показатели.Сотрудник

| И ВТ\_ГрафикиРаботы.Дата = ВТ\_Показатели.Период

|

|СГРУППИРОВАТЬ ПО

| ВТ\_ГрафикиРаботы.Дата,

| ВТ\_ГрафикиРаботы.Сотрудник,

| НАЧАЛОПЕРИОДА(ВТ\_ГрафикиРаботы.Дата, МЕСЯЦ),

| ВТ\_Показатели.Организация,

| ВЫБОР

| КОГДА ВТ\_Показатели.Значение ЕСТЬ NULL

| ТОГДА ВЫБОР

| КОГДА ВТ\_СостояниеПоДням.Сотрудник.Наименование = ""Работает""

| ТОГДА ЗНАЧЕНИЕ(Справочник.ВидыИспользованияРабочегоВремени.ВыходныеДни)

| ИНАЧЕ ВТ\_ГрафикиРаботы.ВидВремени

| КОНЕЦ

| ИНАЧЕ ЗНАЧЕНИЕ(Справочник.ВидыИспользованияРабочегоВремени.Явка)

| КОНЕЦ

|

|УПОРЯДОЧИТЬ ПО

| ПериодРегистрации,

| Сотрудник,

| Дата

|ИТОГИ

| СУММА(Часы)

|ПО

| ПериодРегистрации,

| Организация,

| Сотрудник";

//МассивСотров = ТаблицаОплат.ВыгрузитьКолонку("Сотрудник");

//СписокСотров = Новый СписокЗначений;

//СписокСотров.ЗагрузитьЗначения(МассивСотров);

//ТалицаПериодов = ТаблицаОплат.Выгрузить("Период");

//ТалицаПериодов.Свернуть("Период");

//ТалицаПериодов.Сортировать("Период");

Показатель = Справочники.ПоказателиРасчетаЗарплаты.НайтиПоНаименованию("Сдельный заработок",Истина);

ВидДокумента = Справочники.ВидыДокументовВводДанныхДляРасчетаЗарплаты.НайтиПоНаименованию("Сдельные работы",Истина);

//Запрос.УстановитьПараметр("Сотрудник", СписокСотров);

Запрос.УстановитьПараметр("Год", 2019);

Запрос.УстановитьПараметр("Показатель", Показатель);

Запрос.УстановитьПараметр("ВидДокумента", ВидДокумента);

Запрос.УстановитьПараметр("НачПериода", Дата("20190301000000"));

Запрос.УстановитьПараметр("КонПериода", Дата("20190331235959"));

Рез = Запрос.Выполнить();

Таб = Рез.Выгрузить();

//Запрос.УстановитьПараметр("НачПериода", ТалицаПериодов[0].Период);

//Запрос.УстановитьПараметр("КонПериода", КонецДня(ТалицаПериодов[ТалицаПериодов.Количество()-1].Период));

ВыборкаПериода = Рез.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам, "ПериодРегистрации");

Пока ВыборкаПериода.Следующий() Цикл

ВыборкаОрганизаций = ВыборкаПериода.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам, "Организация");

Пока ВыборкаОрганизаций.Следующий() Цикл

Если НЕ ЗначениеЗаполнено(ВыборкаОрганизаций.Организация) Тогда

Продолжить;

КонецЕсли;

Док = Неопределено;

ВыборкаДок = Документы.ИндивидуальныйГрафик.Выбрать(НачалоМесяца(ВыборкаПериода.ПериодРегистрации), КонецМесяца(ВыборкаПериода.ПериодРегистрации));

Пока ВыборкаДок.Следующий() Цикл

Если ВыборкаДок.ПериодРегистрации = ВыборкаПериода.ПериодРегистрации

и ВыборкаДок.Комментарий = "# сформирован автоматически #"

и ВыборкаДок.Организация = ВыборкаОрганизаций.Организация Тогда

Док = ВыборкаДок.Ссылка;

Прервать;

КонецЕсли;

КонецЦикла;

Если Док <> Неопределено Тогда

Док = Док.ПолучитьОбъект();

Иначе

Док = Документы.ИндивидуальныйГрафик.СоздатьДокумент();

Док.Комментарий = "# сформирован автоматически #";

Док.ПериодРегистрации = ВыборкаПериода.ПериодРегистрации;

Док.Дата = КонецМесяца(ВыборкаПериода.ПериодРегистрации);

Док.Организация = ВыборкаОрганизаций.Организация;

КонецЕсли;

Док.ДатаНачалаПериода = НачалоМесяца(ВыборкаПериода.ПериодРегистрации);

Док.ДатаОкончанияПериода = КонецМесяца(ВыборкаПериода.ПериодРегистрации);

Док.ДанныеОВремени.Очистить();

ВыборкаСотр = ВыборкаОрганизаций.Выбрать(ОбходРезультатаЗапроса.ПоГруппировкам, "Сотрудник");

Пока ВыборкаСотр.Следующий() Цикл

Если НЕ ЗначениеЗаполнено(ВыборкаСотр.Сотрудник) Тогда

Продолжить;

КонецЕсли;

стр = Док.ДанныеОВремени.Добавить();

стр.Сотрудник = ВыборкаСотр.Сотрудник;

ВыборкаВремени = ВыборкаСотр.Выбрать();

Пока ВыборкаВремени.Следующий() Цикл

Если НЕ ЗначениеЗаполнено(ВыборкаВремени.ВидВремени) Тогда

Продолжить;

КонецЕсли;

Н = День(ВыборкаВремени.Дата);

стр["ВидВремени"+Н] = ВыборкаВремени.ВидВремени;

Если ВыборкаВремени.ВидВремени = Справочники.ВидыИспользованияРабочегоВремени.Явка Тогда

стр["Часов"+Н] = ?(ВыборкаВремени.Часы = 0, 8, ВыборкаВремени.Часы);

КонецЕсли;

Н = Н+1;

КонецЦикла;

КонецЦикла;

Док.Записать();

Док.Записать(РежимЗаписиДокумента.Проведение);

КонецЦикла;

КонецЦикла;

КонецПроцедуры

&НаКлиенте

Процедура СоздатьГрафики(Команда)

СоздатьГрафикиНаСервере();

КонецПроцедуры