**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«МАРИЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет ИиВТ

Направление подготовки

230100.62

(шифр)

Информатика и вычислительная техника

(наименование)

Студент: Фамилия Ягодаров

Имя Андрей

Отчество Евгеньевич

Тема бакалаврской работы:

«Автоматизация процесса формирования документов основной образовательной программы ФГОС ВПО»

Кафедра Информационно вычислительных систем

Заведующий кафедрой Мясников В.И., к.т.н., доцент каф. ИВС

(ф.и.о. уч. степень, звание, должность)

Руководитель Васяева Е.С,, к.т.н., доцент каф. ИВС

(ф.и.о. уч. степень, звание, должность)

Консультанты Мясников В.И., к.т.н., доцент каф. ИВС

(ф.и.о. уч. степень, звание, должность)

№ приказа о допуске к выполнению ВКР

Проект начат

Проект закончен

№ приказа (распоряжения) о допуске к защите

Оценка Государственной экзаменационной комиссии по защите

Декан факультета ( )

Секретарь Государственной

экзаменационной комиссии ( )

« » 2011г.

**АННОТАЦИЯ**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*2*

*Кнфу ХХХ.ХХХ.ПЗ*

В данной выпускной квалификационной работе (ВКР) приведены основные положения разработки программы для автоматизации формирования документов на основе имеющихся. Данная программа предназначена для автоматизации процесса составления «Программы формирования компетенций» и создания таблиц описания формируемых учебной дисциплиной компетенций в «Рабочей программе учебной дисциплины». Итоговые документы формируются исходя из анализа следующих файлов: «Учебный план», «Компетентностная модель выпускника» и набора «Паспортов компетенций» для каждой дисциплины учебного плана.

В процессе работы на ВКР были разработаны алгоритмы, на основе которых была написана программа. В ходе разработки программы использовались: среда разработки Microsoft Visual Studio 2010, динамические библиотеки Microsoft Office Word и Microsoft Office Excel платформы .NET. Программа написана на языке программирования C#.

**THE SUMMARY**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*3*

*Кнфу ХХХ.ХХХ.ПЗ*

In this final qualifying work (FQW) are the main provisions of the development of programs to automate form-based documents available. This program is designed to automate the process of compiling «Programmes competencies» and process of create a tables describing formed the discipline of competencies in the «Work programme of discipline». Final documents are formed based on the analysis of the following files: «Curriculum», «Competence model of graduates» and a set of «Passport competencies» for each of the disciplines of the curriculum.

In the process of the FQW developed algorithms on which the program was written. During the development of programs were used: the development environment Microsoft Visual Studio 2010, dynamic libraries Microsoft Office Word and Microsoft Office Excel of the platform .NET. The program is written on the programming language C#.

**СОДЕРЖАНИЕ**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*4*

*Кнфу ХХХ.ХХХ.ПЗ*

ОСНОВНЫЕ УСЛОВНЫЕ ОБОЗНАЯЕНИЯ……………………………………..5

ВВЕДЕНИЕ…………………………………………………………………...……...6

1. АНАЛИТИЧЕСКАЯЧАСТЬ……………………………………………………...7

1.1 Причины необходимости автоматизации процессов составления

документов ООП ФГОС-3 ВПО……………………………………………….7

1.2 Перечень документов ООП………………………………………………9

1.3 Анализ структуры исходных документов……………………………….12

1.4 Разработка требований к структуре исходных документов……………16

1.5 Структура документов «Программа формирования компетенций» и «Рабочая программа учебной дисциплины»………………………………...17

2. Программная часть………………………………………………………20

2.1 Особенности существующих решений автоматизации документов…..20

2.2 Анализ существующих сред разработки программного обеспечения...22

2.3 Разработка алгоритма……………………………………………………..26

2.4 Разработка форматов хранения обрабатываемых данных……………..29

2.5 Разработка и описание программного обеспечения для

автоматизации составления документов…………………………………….30

2.5.1 Документ «Программа формирования компетенций»………….41

2.5.2 Документ «Рабочая программа учебной дисциплины»…………42

3. Экспериментально – тестовая часть……………………………...43

Заключение……………………………………………………………………..47

ЛИТЕРАТУРА……………………………………………………………………...48

ИСТОЧНИКИ INTERNET…………………………………………………………49

Спецификация на программное средство………………………...50

Приложение 1. Руководство пользователя………………….………………..51

Приложение 2. Текст программы………………………………….………….55

**Основные условные обозначения**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*5*

*Кнфу ХХХ.ХХХ.ПЗ*

ООП – основная образовательная программа;

ФГОС-3 – федеральный государственный образовательный стандарт третьего поколения;

ВКР – выпускная квалификационная работа;

ГОС – государственный образовательный стандарт;

ВПО - высшее профессиональное образование;

КМВ – документ «Компетентностная модель выпускника»;

УП – документ «Учебный план»;

ПК – документ «Паспорт компетенции»;

ИГА – итоговая государственная аттестация;

БД – база данных;

COM – Complex object model – комплексная объектная модель;

ПО – программное обеспечение.

**Введение**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*6*

*Кнфу ХХХ.ХХХ.ПЗ*

*Разраб.*

*Ягодаров А.Е.*

*Провер.*

*Васяева Е.С.*

*Реценз.*

*Н. Контр.*

*Утверд.*

*Автоматизация процесса формирования документов основной образовательной программы ФГОС ВПО Пояснительная записка*

*Лит.*

*Листов*

*116*

*ИВТ-41 МарГТУ*

В настоящее время остро стоит проблема экономия трудовых затрат человека. Это касается не только физического, но и умственного труда. В подавляющем большинстве рабочие документы организации оформляются на ЭВМ.

Вузы ежегодно выпускают специалистов разного профиля. Если выпускник будет подготовлен с качествами, не подходящими работодателю, то «доводка» на рабочем месте квалификации специалистов занимает не один год, как хотелось бы работодателю, а три-пять лет. При этом основная проблема состоит не в недостатке у выпускников знаний, умений и навыков, а специфического умения исполнять необходимые обязанности, связанные со степенью сформированности тех или иных компетенций. Внедрение стандартов третьего поколения на основе модульно-компетентностного подхода предполагает наличие постоянной связи разработчиков образовательных программ с требованиями работодателей и реализации их запросов в содержании образования.

В связи с этим необходимо составление документов, которые и будут отражать какие компетенции получает студент, обучив некую дисциплину. Процесс составления данных документов достаточно трудоёмок и требует автоматизации, чтобы увеличить полезное время работников кафедр высших учебных заведений. Так как работа производится на ЭВМ, то автоматизация осуществима без применения дополнительного оборудования. Процесс автоматизации создания документа «Программа формирования компетенций» и создания таблиц описания формируемых учебной дисциплиной компетенций в «Рабочей программе учебной дисциплины» рассмотрены в данной работе.

**1. Аналитическая часть**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*7*

*Кнфу ХХХ.ХХХ.ПЗ*

**1.1 Причины необходимости автоматизации процессов составления документов ООП ФГОС-3 ВПО**

Основная особенность Федеральных государственных образовательных стандартов высшего профессионального образования (ФГОС ВПО) состоит в том, что они разрабатываются на основе компетентностного подхода с участием работодателей, общественных и профессиональных объединений. ФГОС ВПО – это пакет стандартов по направлению подготовки как совокупность образовательных программ бакалавра, магистра и специалиста, объединяемых на основе общности их фундаментальной части.

В качестве основного объекта оценки выступают общекультурные и профессиональные компетенции, под которыми понимаются «способности применять знания, умения и личностные качества для успешной деятельности в определенной области» [1].

ФГОС ВПО предусматривает разработку (проектирование) основной образовательной программы (ООП), содержание и структура которой значительно усложнились по сравнению с ГОС ВПО второго поколения.

Для формирования компетенции как интегрального показателя и ее оценки в структуре ООП появились дополнительные документы:

• «Карта компетенций учебной дисциплины»;

• «Паспорт компетенций»;

• «Программа формирования компетенций»;

• «Перечень образовательных технологий» и т.п.

Составление большинства из них является трудоемким процессом. При формировании общих документов по специальности, которые по своей структуре являются громоздкими, разработчику приходится анализировать десятки таблиц, выполняя механическую рутинную работу. Это, в свою очередь, повышает утомляемость разработчика и может приводить к ошибкам. Поэтому вполне обоснованным является желание автоматизировать процесс формирования таких документов.

Можно сформулировать четыре причины необходимости автоматизации процесса создания определенных документов ООП:

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*8*

*Кнфу ХХХ.ХХХ.ПЗ*

1) снижение трудоемкости разработки ООП;

2) сокращение времени на разработку и корректировку ООП;

3) уменьшение количества ошибок;

4) существования требования ежегодного обновления основной образовательной программы.

**1.2 Перечень документов ООП**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*9*

*Кнфу ХХХ.ХХХ.ПЗ*

Основная образовательная программа – совокупность учебно-методической документации, включающей в себя учебный план, рабочие программы учебных дисциплин (модулей) и другие материалы, обеспечивающие воспитание и качество подготовки обучающихся, а также программы учебной и производственной практик, календарный учебный график и методические материалы, обеспечивающие реализацию соответствующей образовательной технологии.[2]

В состав ООП входят:

1. Общие положения (характеристика ООП): наименование ООП, профиль подготовки, квалификация (степень) выпускника, нормативные документы (перечень), миссия (цель) ООП, срок освоения ООП, трудоёмкость ООП, требования к абитуриентам.
2. Характеристика профессиональной деятельности выпускника (компетентностная модель выпускника): область, объекты, виды и задачи профессиональной деятельности выпускника вуза по направлению подготовки; компетенции выпускника как совокупный ожидаемый результат обучения (раздел 3 в ООП).
3. Документы, регламентирующие организацию учебного процесса (по формам обучения): структура (таблица ООП, учебный план (матрица соотнесения, перечень образовательных технологий), календарный учебный график.
4. Документы, регламентирующие содержание учебного процесса: программы учебных дисциплин или модулей практик.
5. Ресурсное обеспечение ООП: кадровое обеспечение, учебно-методическое и информационное, обеспечение материально –техническое обеспечение.
6. Характеристики среды вуза, обеспечивающей развитие общекультурных компетенций.
7. Нормативно-методическое обеспечение системы качества освоения обучающимися ООП ВПО (документы, описывающие систему оценки качества подготовки: фонды оценочных средств, итоговая государственная аттестация).

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*10*

*Кнфу ХХХ.ХХХ.ПЗ*

1. Другие нормативно-методические документы и материалы, обеспечивающие качество подготовки обучающихся: перечень вузовских документов и др.
2. Регламент по организации периодического обновления ООП в целом и составляющих ее документов.

В качестве приложений в состав ООП входят листы экспертной оценки; рабочие программы учебных дисциплин (модулей), учебной и производственной практик; документы по итоговой государственной аттестации и методические рекомендации по подготовке к ИГА; методические рекоменации по организации учебного процесса, в том числе по организации самостоятельной работы студентов; нормативные документы, регламентирующие учебный процесс, фонды оценочных средств и др.

В приложение выносятся таблицы, составляемые разработчиками в процессе проектирования ООП:

1. Компетентностная модель выпускника,
2. Соотнесение задач профессиональной деятельности выпускника и формируемых в ходе освоения ООП компетенций,
3. Паспорта и программы формирования у студентов компетенций,
4. Матрица соотнесения целей и содержания учебного процесса,
5. Состав, основное содержание и содержательно-логические связи учебных дисциплин (модулей), практик,
6. Описание форм текущего контроля и промежуточных аттестаций по дисциплинам учебного плана.

В результате предварительного анализа был сформирован список документов, создание которых можно автоматизировать полностью или частично: «Паспорт компетенций», «Программа формирования компетенций», «Матрица соотнесения целей и содержания учебного процесса», «Перечень образовательных технологий», «Рабочая программа дисциплины».

Формирование перечисленных документов может быть организовано как в виде отдельных программ, так и в виде модулей общего программного комплекса.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*11*

*Кнфу ХХХ.ХХХ.ПЗ*

С учётом технического задания, целью данной работы является составление алгоритма автоматического формирования двух документов:

* одного из наиболее трудоемких документов ООП – «Программа формирования компетенций».
* таблиц описания формируемых учебной дисциплиной компетенций в «Рабочей программе учебной дисциплины».

**1.3 Анализ структуры исходных документов**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*12*

*Кнфу ХХХ.ХХХ.ПЗ*

Исходными документами для программного модуля по формированию документа «Программа формирования компетенций» и таблиц документа «Рабочая программа учебной дисциплины» являются:

• «Компетентностная модель выпускника» (КМВ) (\*.doc, \*.docx);

• «Учебный план специальности» (УП) (\*.xls, \*.xlsx);

• «Паспорта компетенций для каждой дисциплины специальности» (ПК) (\*.doc, \*.docx).

1*) «Компетентностная модель выпускника».*

Документ формата \*.doc или \*.docx. Начинается с двух строчек текста, а затем таблица на весь документ. С каждой новой страницей заголовки столбцов повторяются. Вид компетенций разделяется строчкой внутри таблицы с жирным шрифтом, занимающей все столбцы. Между этими строчками-разделителями перечисляются компетенции: первый столбец – код компетенции, 2 столбец – степень значимости компетенций, 3 столбец – название компетенции, 4 столбец – характеристика обязательного уровня сформированности компетенции у выпускника. Четвёртый столбец в свою очередь разделяется на три строчки, где написано 1)знает … 2)умеет … 3) владеет … . Есть ещё строчки разделители, которые пишутся обычным шрифтом(не жирным), определяющие род деятельности, где применяется компетенция(проектно-технологическая, научно-исследовательская деятельность… и т.д.).

Структура данного документа приведена в табл. 1.

Таблица 1

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*13*

*Кнфу ХХХ.ХХХ.ПЗ*

**Компетентностная модель выпускника**

| **Код компетенции** | **Степень значимости компетенций (см. листы экспертной оценки –Приложение 1)** | | **Название компетенции** | **Характеристика обязательного уровня сформированности компетенции у выпускника вуза** |
| --- | --- | --- | --- | --- |
| **Общекультурные компетенции** | | | | |
| ОК-… |  | |  | **Знает** |
| **Умеет** |
| **Владеет** |
| **Профессиональные компетенции** | | | | |
| ПК-… |  | |  | **Знает** |
| **Умеет** |
| **Владеет** |
| **Дополнительные (профильные) компетенции (ДПК)** | | | | |
| ДПК-… | |  |  | **Знает** |
| **Умеет** |
| **Владеет** |

2) «*Учебный план специальности»*.

Учебный план представлен в виде таблицы в формате \*.xls или \*.xlsx. Содержит информацию о графике учебного процесса и распределению трудоемкости по семестрам.

Структура:

1. Реквизиты (подпись, дата)

2. Название специальности

3. График учебного процесса

4. Расшифровка условных обозначений графика учебного процесса

5. Таблица дисциплин

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*14*

*Кнфу ХХХ.ХХХ.ПЗ*

6. Реквизиты (подпись, дата)

Для автоматизации требуется лишь 2 первых столбца таблицы дисциплин:

* «№№ п/п» - уникальный код циклов, разделов ООП, модулей, дисциплин, практик
* «Наименование циклов, разделов ООП, модулей, дисциплин, практик».

Структуры этих столбцов приведена в табл. 2.

Таблица 2

**Учебный план специальности**

|  |  |
| --- | --- |
| **№№ п/п** | **Наименование циклов, разделов ООП, модулей, дисциплин, практик** |
| **X.X Название цикла** | |
| **XXX.X1.00** | **Базовая часть** |
| **XXX.X1.01** | Наименование дисциплины |
| **XXX.X2.00** | **Вариативная часть** |
| **XXX.X2.01** | Наименование дисциплины |
| **XXX.X3.00** | **Дисциплины и курсы по выбору студентов** |
| **XXX.X3.01** | Наименование дисциплины / Наименование дисциплины / Наименование дисциплины |

3) «*Паспорта компетенций для каждой дисциплины специальности»*.

Документ формата \*.docx. Начинается с двух строчек – заголовков, где пишется название дисциплины. Далее таблица до конца документа: 1 столбец – учебная дисциплина, планируемый семестр(столбец из одной ячейки на всю таблицу, не считая разрыва страниц); 2 – код и название компетенции(код пишется в скобках); 3 – способы(признаки) проявления компетенции; 4- образовательные технологии; 5 – средства оценки уровня сформированности компетенции. Существуют строчки – разделители, которые занимают все столбцы, не считая первого левого. Эти строчки разделяют компетенции по виду(общекультурные, профессиональные и т.д.).

Структура данного документа приведена в таблице 3.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*15*

*Кнфу ХХХ.ХХХ.ПЗ*

Таблица 3

**Паспорт компетенции**

**Учебная дисциплина «Наименование дисциплины»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Учебная дисциплина; планируемый семестр | Код и название компетенции | Способы (признаки) проявления компетенции | Образовательные  технологии | Средства оценки уровня сформированности компетенции |
| Наименование дисциплины,  Семестр – Х, Х | **Общекультурная компетенция** | | | |
| Описание компетенции (код); | **Знать:**  **Уметь:**  **Владеть:**  **Личн.хар.:** |  |  |
| **Профессиональная компетенция** | | | |
| Описание компетенции (код); | **Знать:**  **Уметь:**  **Владеть:**  **Личн.хар.:** |  |  |
| **Дополнительные профессиональные компетенции** | | | |
| Описание компетенции (код); | **Знать:**  **Уметь:**  **Владеть:**  **Личн.хар.:** |  |  |

**1.4 Разработка требований к структуре исходных документов**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*16*

*Кнфу ХХХ.ХХХ.ПЗ*

Чтобы разрабатываемое программное обеспечение работало исправно, необходимо исполнение следующих нежёстких требований к структуре исходных документов:

1) Требования к КМВ.

Документ должен быть формата Microsoft Office Word (\*.doc, \*.docx). До строки «*Компетентностная модель выпускника»* может быть любая последовательность символов, после этой строки – должна начинаться таблица. Желательно, чтобы таблица была одна (разделённая разрывами страниц). Столбец «*Код компетенции*» должен быть крайним слева.

2) Требования к УП.

Документ должен быть формата Microsoft Office Excel (\*.xls, \*.xlsx). В данном документе инетересуют только два столбца в главной таблице – это *«№№ п/п»* и *«Наименование циклов, разделов ООП, модулей, дисциплин, практик».*  Первый столбец должен располагаться первым слева, второй – вторым. Название столбцов не должно меняться. Структура кодов дисциплин не должна меняться: циклы дисциплин определяются кодами «Б.Х», части циклов Х.Х.Х.00, а дисциплины Х.Х.Х.ХХ. Последняя запись первого столбца, по которому определяется конец анализа документа должна быть: *«Общая трудоемкость основной образовательной программы»*.

3) Требования к ПК.

Документ должен быть формата Microsoft Office Word (\*.doc, \*.docx). Требуется наличие обязательного заголовка «*Паспорт компетенции Учебная дисциплина «ХХХ»*». Между кавычками пишется название учебной дисциплины. Требуется, чтобы столбец «*Учебная дисциплина; планируемый семестр*» был первым слева, «*Код и название компетенции*» - вторым. В столбце «*Учебная дисциплина; планируемый семестр*» требуется наличие всего одной ячейки, где после символов «Семестр –» перечисляются номера семестров через запятую. В столбце «*Код и название компетенции*» код компетенции записывается в скобках «(ХХХ)». Больше в этом столбце скобки применяться не должны.

**1.5 Структура документов «Программа формирования компетенций» и «Рабочая программа учебной дисциплины»**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*17*

*Кнфу ХХХ.ХХХ.ПЗ*

*«Программа формирования компетенций»* составляется на основе установленного структурой (таблицей) ООП перечня учебных дисциплин, паспортов компетенций (с указанием учебных дисциплин, планируемых семестров изучения учебной дисциплины), рабочих программ учебных дисциплин (модулей) и практик, учебного плана.

Анализ «Программы формирования компетенций» позволит разработчикам внести коррективы в учебный план: распределение учебных дисциплин по семестрам с целью планомерного формирования компетенций. Структура «Программы формирования компетенций» приведена в табл. 4 [2].

Таблица 4

**Программа формирования компетенций**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Компетенция**  **(код)**  **Дисциплины**  **уч. плана** | **Семестры (годы) обучения** | | | | | | |
| 1 семестр | | | | 2 семестр | 3 семестр | … семестр |
| код |  |  |  |  |  |  |
| 1. | + |  |  |  |  |  |  |
| 2. |  | + | + |  |  |  |  |
| 3. | + |  |  | + |  |  |  |
| Сумма | = | = | = | = |  |  |  |

*«Рабочие программы учебных дисциплин»* составляется для дисциплин как базовой, так и вариантной части, в том числе дисциплин по выбору обучающихся.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*18*

*Кнфу ХХХ.ХХХ.ПЗ*

Формирование РП начинается с определения и прописывания знаний, умений и навыков, которые должен освоить студент при изучении данной дисциплины. Эти конкретные результаты обучения должны быть направлены на формирование у студента определённых копетенций, общекультурных, профессиональных, дополнительных (профильных), содержание и состав которых соответствуют планируемым преподавателем результатам обучения. Паспорта компетенций, созданные разработчиками на предварительном этапе проектирования ООП, дают возможность преподавателю правильно выбрать компетенции, на формирование которых и должна быть нацелена конкретная дисциплина.

Ниже приведена требуемая для автоматизации структура Рабочей программы учебной дисциплины.

**Структура «Рабочей программы учебной дисциплины»**

*Титульный лист*

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

МАРИЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Декан

(назв. факультета)

(подпись, Ф.И.О. декана)

« » 2011г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**шифр наименование дисциплины по учебному плану**

*(указывается шифр и наименование дисциплины по учебному плану)*

*Раздел 1*

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*19*

*Кнфу ХХХ.ХХХ.ПЗ*

Раздел 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

**общекультурных:**

|  |  |
| --- | --- |
| **Код компетенции** | **Название компетенции** |
| код | название |

**профессиональных:**

|  |  |
| --- | --- |
| **Код компетенции** | **Название компетенции** |
| код | название |

*Раздел 2*

Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП.

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Название компетенции** | **Предшествующие дисциплины** | **Последующие дисциплины** |
| *Общекультурные компетенции* | | | |
| код | название |  |  |
| *Профессиональные компетенции* | | | |
| код | название |  |  |

*Раздел 3*

Раздел 3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

(матрица распределения компетенций по разделам и темам дисциплины)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ТЕМЫ,  РАЗДЕЛЫ  ДИСЦИПЛИНЫ | КОЛИЧЕСТВО ЧАСОВ  (АУД/СРС) | КОМПЕТЕНЦИИ | | | | | | |
| код | код | код | код | код | код | Σ  ОБЩЕЕ КОЛИЧЕСТВО КОМПЕТЕНЦИЙ |
|  |  |  |  |  |  |  |  |  |
| ИТОГО |  |  |  |  |  |  |  |  |

**2. Программная часть**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*20*

*Кнфу ХХХ.ХХХ.ПЗ*

**2.1 Особенности существующих решений автоматизации документов**

Для решения задачи автоматического формирования документов существует несколько путей и подходов. Ниже проведён анализ существующих систем.

В настоящее время создано достаточно большое количество программ, реализующих автоматизацию формирования документов.

Например «АвтоДок»1. Логика работы рассматриваемой программы основывается на понятии «бизнес-сценарий». Под данным термином разработчики продукта «АвтоДок» понимают набор действий пользователя, которые приводят к генерации конечного документа. Неотъемлемой частью бизнес-сценариев являются шаблоны. Это специальные файлы форматов RTF, DOC (для текстовых документов) или XLS (для электронных таблиц). В принципе они отличаются от привычных для всех пользователей ПК документов только наличием так называемых переменных - специальных наборов символов, вместо которых при генерации конечного файла будет вставляться нужная информация. Создаваться шаблоны могут по-разному. Так, например, самым простым способом является использование уже существующих документов. Делается это так. Пользователь открывает нужный файл и постепенно удаляет из него всю изменяющуюся информацию, заменяя ее переменными, условными предложениями, функциями и т. п. Особенно удобно делать это с помощью специальной панели, которая появляется в Microsoft Word и Excel после установки «АвтоДок»`а. Обработав, таким образом, весь файл, пользователь получит из него готовый шаблон, содержащий только типовые, то есть не меняющиеся во всех подобных документах данные, и переменные.

Главный минус этой программы в том, что она не позволяет делать какой

1 - <http://www.auto-doc.ru/> - официальный сайт программы «АвтоДок»

либо анализ документов. Работа основана исключительно на шаблонах и замене ключевых слов текстом.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*21*

*Кнфу ХХХ.ХХХ.ПЗ*

Другой разновидностью программ, автоматизирующих создание документов является «AVA ERP»1. Данная программа предлагает все необходимые инструменты для организации и управления документооборотом на предприятии. Используя «AVA ERP» появляется возможность автоматизировать все необходимые бизнес-процессы компании, и, как следствие, весь документооборот. Таким образом, формирование документов будет неразрывно связано с логистикой, финансами и другими бизнес-процессами. Вы получите не просто документы в базе данных, а полноценную причинно-следственную связь «процесс-документ».

Оценив существующие решения, был выбран вариант, схожий с «AVA ERP». Исходные документы будут играть роль специализированной полнотекстовой БД, анализируя которую программа будет формировать документы. Это накладывает некоторые правила оформления на исходные документы, но поскольку они формируются по государственным стандартам, это не будет большой проблемой. Интуитивно понятный интерфейс позволит пользователю легко разобраться с работой программы. На выходе пользователь получит знакомые ему документы в формате \*.doc и \*.xls или \*.docx и \*.xlsx – форматы Microsoft Office Word и Microsoft Office Excel.

Данный способ реализации является одним из наиболее эффективных со всех точек зрения. Разработанный модуль можно установить на любой специально выбранной машине (соответствующей минимальным системным требованиям).

В результате разработки должен получиться функциональный программно-аппаратный комплекс формирования документов ФГОС - 3.

1- http://avasystems.ru/ - официальный сайт программы «AVA ERP»

**2.2 Анализ существующих сред разработки программного обеспечения**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*22*

*Кнфу ХХХ.ХХХ.ПЗ*

Для рациональной работы по созданию программного обеспечения для процесса автоматического формирования документов требуется выбрать подходящую среду разработки программного обеспечения. Рассмотрено несколько таких сред: Visual Studio 2010, Borland C++ Builder 2007, Borland Delphi 7.

*Среда разработки Visual Studio 2010*

Visual Studio 2010 – многообещающая версия популярной платформы разработки, в которой реализованы новейшие достижения Microsoft для разработчиков, руководителей разработки, тестировщиков и дизайнеров.

Данная версия включает Visual Basic .NET, Visual C++, Visual C#, а также новый функциональный язык Visual F#. В комплекте с программой идет усовершенствованная 4-ая версия .NET Framework. Visual Studio 2010 поддерживает JavaScript и jQuery.

Интерфейс программы и редактор кода были переработаны и реализованы на Windows Presentation Foundation (WPF). Это позволило сделать рабочую среду более функциональной и гибкой. Среди новых функций, стоит отметить возможность использования нескольких мониторов, а также возможность отображения информации о коде в разных представлениях. Программа также имеет усовершенствованный редактор для Silverlight.

Отдельного внимания заслуживает новый инструмент отладки Visual Studio 2010 – IntelliTrace или «историческая отладка» (Historical Debugging). Благодаря этому элементу, все действия программы, включая вызовы методов и обработку событий, отслеживаются и сохраняются. А при поиске ошибки, достаточно прервать работу программы отладчиком, просмотреть историю исполнения и перейти к нужной точке. Для уменьшения нагрузки на систему, историческая отладка позволяет настраивать максимальный размер протокола и имеет гибкую настройку.

Одним из нововведений программы является модуль Architecture Explorer, который позволяет анализировать сложный программный код в графической форме. Такая форма облегчает чтение стороннего кода и упрощает коллективную работу над большим проектом. Также имеется возможность настраивать детализацию отображения информации.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*23*

*Кнфу ХХХ.ХХХ.ПЗ*

Visual Studio 2010 позволяет разрабатывать программное обеспечение, которое использует возможности многопроцессорных платформ, распределяя вычислительную нагрузку между множеством процессоров. При этом работу с потоками программа берет на себя.

Новая версия программы позволяет легко создавать приложения для последних версий Windows, Office и SharePoint. С помощью имеющихся в арсенале Visual Studio 2010 и .NET Framework 4.0 инструментов, разработчики могут создавать не только приложения для ПК, бизнеса и мобильных телефонов, но и «облачные» (Azure) приложения.

Платформа доступна в четырех редакциях – Professional, Premium, Ultimate и Test Professional. Visual Studio 2010 Professional является базовой комплектацией продукта.

*Среда разработки Borland C++ Builder 2007*

C++ Builder —интегрированная среда программирования (IDE), система, используемая программистами для быстрой разработки программного обеспечения на языке C++.

C++ Builder представляет собой приложение, главное окно которого содержит настраиваемую инструментальную панель (слева) и палитру компонентов (справа). Помимо этого, по умолчанию при запуске C++ Builder появляются окно инспектора объектов (слева) и форма нового приложения (справа). Под окном формы приложения находится окно редактора кода.

Формы являются основой приложений C++ Builder. Создание пользовательского интерфейса приложения заключается в добавлении в окно формы элементов объектов C++ Builder, называемых компонентами. Компоненты C++ Builder располагаются на палитре компонентов, выполненной в виде многостраничного блокнота. Важная особенность C++ Builder состоит в том, что он позволяет создавать собственные компоненты и настраивать палитру компонентов, а также создавать различные версии палитры компонентов для разных проектов.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*24*

*Кнфу ХХХ.ХХХ.ПЗ*

*Среда разработки Borland Delphi 7*

Среда Delphi 7 представляет собой интегрированную оболочку разработчика, в которую входит набор специализированных программ, ответственных за разные этапы создания готового приложения.

Delphi – язык и среда программирования, относящаяся к классу RAD- (Rapid Application Development – «Средство быстрой разработки приложений. Мощность и гибкость языка программирования Delphi – безусловное достоинство языка, выгодно отличающее эту систему программирования от других инструментов RAD. Ядром языка Delphi является язык Паскаль.

Визуальное проектирование избавляет программиста от многих аспектов разработки интерфейса программы, так как Delphi автоматически готовит необходимые программные заготовки и соответствующий файл ресурсов.

Среда разработки Delphi предоставляет разработчику большой набор простых в использовании инструментов, позволяющих быстро разрабатывать сложные проекты, создавая приятный и удобный пользовательский интерфейс.

Использование компонентов визуального программирования не только во много раз сокращает сроки разработки программ, но и существенно снижает вероятность случайных программных ошибок.

Таким образом, Borland Delphi обеспечивает высокую эффективность процесса разработки и большую производительность результирующих приложений. Большое число компонентов позволяет организовать эффективную обработку данных. Borland Delphi можно рекомендовать как универсальное средство разработки приложений различного уровня сложности, начиная от простейших задач до крупных проектов.

Выбор остановлен на среде разработки Visual Studio 2010. Навыбор повлияли следующие преимущества:

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*25*

*Кнфу ХХХ.ХХХ.ПЗ*

- высокая скорость разработки приложения;

- высокая производительность разработанного приложения;

- низкие требования разработанного приложения к ресурсам компьютера;

- наращиваемость за счет встраивания новых компонент и инструментов в среду Visual Studio 2010;

- нулевая стоимость среды разработки, так как её можно приобрести по специальной программе Microsoft для студентов1.

1 - Для подробной информации посетите http://www.dreamspark.com/

**2.3 Разработка алгоритма**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*26*

*Кнфу ХХХ.ХХХ.ПЗ*

В первую очередь пользователь задаёт файлы на обработку в графической оболочке программы. УП и КМВ выбираются одним файлом, а ПК по предметам – папкой, в которой хранятся все паспорта на обработку.

Если хоть один путь не будет указан, либо директория ПК будет пуста, программа предупредит об этом пользователя и предложит ему вновь указать пути. Отсутствие файла какого-либо ПК не прерывает программу. После вывода предупреждения о том, что ПК для текущей дисциплины нет, программа продолжит свою работу, не заполнив этот предмет в Программе формирования компетенций. Если будет найден ПК дисциплины, которой нет в учебном плане, программа также предупредит пользователя об этом и продолжит работу. Поскольку все пути и список файлов будут формироваться в начале работы программы, не рекомендуется удалять, перемещать рабочие директории и файлы, находящиеся в них. Если ранее найденный файл будет не доступен по какой-либо причине, программа предложит пользователю завершить или продолжить работу.

Основная трудоемкость алгоритма (рис. 1) формирования Программы формирования компетенций заключается в:

1) анализе файла КМВ, в результате которого составляется перечень формируемых компетенций;

2) анализе файла УП, в результате которого формируется перечень учебных дисциплин;

3) анализе файлов ПК, в результате которого определяется дисциплина, перечень формируемых ею компетенций.

После выбора файлов и папки открывается файл УП для составления списка предметов. Список хранится в памяти компьютера и удаляется по завершению работы программы. После этого файл УП закрывается.

По введённой папке ПК составляется массив из названий файлов ПК, которые хранятся в этой папке. По названиям файлы открываются поочерёдно. В обрабатываемом файле выделяются компетенции определённых семестров. Номер семестра и соответствующие этому номеру семестра компетенции записывается в результирующий массив.

После обработки файла ПК определённого предмета, он закрывается и открывается следующий файл. Когда все файлы из массива названий файлов ПК обработаются, открывается файл КМВ. По этому файлу составляется массив всех существующих кодов компетенций и соответствующих названий компетенций, после чего файл закрывается.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*27*

*Кнфу ХХХ.ХХХ.ПЗ*

Проанализировав все документы, программа создаст следующие файлы, заполнив их полученными данными из двух массивов и добавив необходимые поля и формулы для подсчета сумм и т.п.:

* «Программу формирования компетенций»,
* «Рабочие программы учебных дисциплин» (таблицы описания формируемых учебной дисциплиной компетенций).

Работу приложений Word и Excel целесообразно осуществлять в скрытом режиме для увеличения быстродействия.

На рис. 1 представлен общий алгоритм формирования «Программы формирования компетенций» и таблиц описания формируемых учебной дисциплиной компетенций в «Рабочих программах учебных дисциплин».



*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*28*

*Кнфу ХХХ.ХХХ.ПЗ*

Рис. 1 Общий алгоритм формирования «Программы формирования компетенций» и таблиц описания формируемых учебной дисциплиной компетенций в «Рабочих программах учебных дисциплин».

**2.4 Разработка форматов хранения обрабатываемых данных**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*29*

*Кнфу ХХХ.ХХХ.ПЗ*

Вся извлеченная в процессе анализа документов информация будет находиться в оперативной памяти до момента создания документов «Программы формирования компетенций» и таблиц «Рабочих программ учебных дисциплин». Информация представляет собой три массива:

• массив для перечня названий файлов ПК NKOM (String);

• массив для перечня названий компетенций KOM (String);

• массив результатов анализа REZ (String).

После запуска приложения и открытия УП программа заполняет первый столбец массива REZ названиями предметов, после чего закрывает этот документ. Массив NKOM заполняется на основе анализа директории документов ПК: названия файлов, размещённые в данной директории, а также поддиректорий записываются в массив. Далее из массива последовательно извлекается название файла и он обрабатывается. В результате обработки массив REZ переписывается:

* в соответствии с названием предмета(первый столбец) во второй столбец записываются номера семестров этого предмета;
* в третий – код компетенции соответствующего номера семестра;
* в четвёртый - номер листа книги Excel «Программы формирования компетенций», в который будет внесена запись об этом предмете;
* в пятый – адрес ячейки(строки) соответствующего листа, которую нужно пометить, которая обозначит всю информацию об этом предмете;
* в шестой – адрес ячейки(столбца) соответствующего листа, которую нужно пометить, которая обозначит всю информацию об этом предмете;
* в седьмой - код соответствующего предмета.

После того как все файлы ПК обработаются, открывается КМВ и массив KOM заполняется кодами(первый столбец) и названиями компетенций(второй столбец).

**2.5 Разработка и описание программного обеспечения для автоматизации составления документов**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*30*

*Кнфу ХХХ.ХХХ.ПЗ*

Office – это среда, в которой большинство задач можно решать без какого-либо программирования. Но вся ценность приложений Office для разработчика заключается в том, что все, что можно сделать вручную - можно сделать программным путем. Кроме того, приложения Office поставляют сервера COM, которые предоставляют интерфейс доступа к приложению и его объектам. Благодаря этому, разработчик в среде Visual Studio имеет возможность, создав контроллер автоматизации, управлять сервером.

Приложение рассматривается как совокупность объектов со своими методами, свойствами, событиями, которые обеспечивают скелет приложения. Таким образом, можно управлять COM серверами Word и Excel извне.

В Visual Studio для работы с COM-серверами приложений Microsoft Office используется язык программирования C#. Соответственно, используя этот язык и было разработано ПО.

*Сфера применения*

Разрабатываемая система ориентирована на пользователей работающих в сфере образования.

*Вычислительная среда*

Система способна работать на любой платформе с необходимым программным обеспечением (Microsoft Office, Framework 2.0) под управлением Windows XP, Windows Vista, Windows 7. Для работы с программой необходимы клавиатура или мышь а также монитор.

На конечный программный продукт налагаются *ограничения*:

1) ПО должно иметь интуитивно-понятный интерфейс, чтобы пользователь мог сосредоточиться на создании документов;

2) ПО должно представлять результат в виде готовых документов \*.doc и \*.xls или \*.docx и \*.xlsx формата;

3) ПО должно работать эффективно и с максимальным быстродействием;

4) ПО должно по возможности не зависеть от типа операционной системы терминала и его аппаратного обеспечения;

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*31*

*Кнфу ХХХ.ХХХ.ПЗ*

5) ПО должно обеспечивать возможность корректировать результаты выполнения.

*Функции системы*

В соответствии с требованиями, система предназначена для автоматического формирования документов ФГОС-3 в среде Microsoft Office.

Главное окно программы представляет собой интерфейс управления путями к файлам и папке, необходимым для работы программы. Их задает пользователь. На рис. 2 представлена графическая оболочка разрабатываемого ПО.

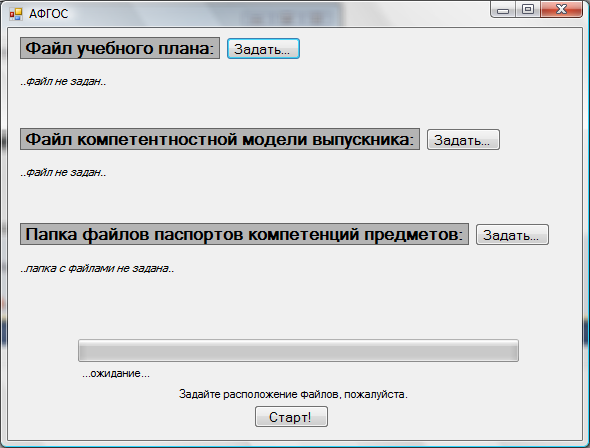


Рис. 2 Графическая оболочка разрабатываемого ПО

*Структура системы*

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*32*

*Кнфу ХХХ.ХХХ.ПЗ*

Условно систему можно разбить на несколько модулей.

Модуль генерирования документов

Модуль обработки и анализа документов

Модуль начальных настроек

*Модуль начальных настроек* включает в себя процедуры задания файлов и папки на обработку, проверку достоверности введения исходных данных.

*Основные процедуры*

Для работы с COM серверами Office и обработки документов были разработаны следующие процедуры:

Основные процедуры для работы с сервером Excel:

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

//Создание файла Excel

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

//Создание объекта Excel.Application - равносильно запуску Microsoft Office Excel

excelApp = new Excel.Application();

//Определение числа листов в создаваемой книге(файле)

excelApp.SheetsInNewWorkbook = 1;

//Добавление листов к книге

excelApp.Workbooks.Add(Type.Missing);

//Присвоение значания excelApp.Workbooks - совокупности ссылок на открытые рабочие книги, объекту excelWorkbooks

excelWorkbooks = excelApp.Workbooks;

//Определение, что работа производится с первой книгой

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*33*

*Кнфу ХХХ.ХХХ.ПЗ*

excelWorkbook = excelWorkbooks[1];

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

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

//Открытие файла Excel

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

//Создание объекта Excel.Application - равносильно запуску Microsoft Office Excel

excelApp = new Excel.Application();

//Присвоение значания excelApp.Workbooks - совокупности ссылок на открытые рабочие книги, объекту excelWorkbooks

excelWorkbooks = excelApp.Workbooks;

//Открытие книги(файла) Excel

excelWorkbook = excelApp.Workbooks.Open(…);

//Присвоение значания excelWorkbook.Worksheets - массив ссылок на листы книги, объекту excelSheets

excelSheets = excelWorkbook.Worksheets;

//Получение ссылки на первый лист книги

excelWorksheet = (Excel.Worksheet)excelSheets.get\_Item(1);

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

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

//Сохранение файла Excel

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

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*34*

*Кнфу ХХХ.ХХХ.ПЗ*

//Строка, содержащая имя сохраняемого файла "Программы формирования компетенций"

string savePath = "";

//Объявление булевой переменной для записи состояния сохранённости файла

bool resOfSave = false;

//Пока файл не сохранён, выполнять..

while (resOfSave == false)

{

//..отображение диалога сохранения файла. Если нажата кнопка "Сохранить", то..

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

//.. запись пути и имени файла в соответствующую строку

}

else

{

//Создание папки АФГОС\_резервное\_сохранение в папке "Мои документы" и запись пути в соответствующую строку

}

//Попытаться сделать..

try

{

//..сохранение файла "Программа формирования компетенций"..

excelWorkbook.SaveAs(…);

}

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*35*

*Кнфу ХХХ.ХХХ.ПЗ*

//.. и если возникло исключение, то..

catch (COMException err)

{

//..обработка исключений

}

//Сохранение удачно, запись информации об этом

resOfSave = true;

}

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

//Закрытие файла Excel

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

//Отсечение запроса на сохранение для книги

excelWorkbook.Saved = true;

//Закрытие книги без сохранения

excelWorkbook.Close(Excel.XlSaveAction.xlDoNotSaveChanges, Type.Missing, Type.Missing);

//Закрытие приложения

excelApp.Quit();

//Обнуление объекта Excel.Application

excelApp = null;

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

Основные процедуры для работы с сервером Word:

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*36*

*Кнфу ХХХ.ХХХ.ПЗ*

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

//Создание файла Word

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

//Создаем объект Word - равносильно запуску Word

wordApp = new Word.Application();

//Включение видимости открытоко документа

wordApp.Visible = false;

//Создание файла Word

wordDocOfWorkProgram = wordApp.Documents.Add(…);

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(0, Type.Missing);

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

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

//Открытие файла Word

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

//Создаем объект Word - равносильно запуску Word

wordApp = new Word.Application();

//Включение видимости открытоко документа

wordApp.Visible = false;

//Открытие файла Word

wordDocOfPaspComp = wordApp.Documents.Open(…);

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*37*

*Кнфу ХХХ.ХХХ.ПЗ*

//Определение области документа

wordDocRange = wordDocOfPaspComp.Range(0, wordDocOfPaspComp.Characters.Count);

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

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

//Сохранение файла Word

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

//Составление полного имени сохраняемого файла рабочей программы

savePathOfWorkPr = …;

//Сохранение пока не удачно, запись информации об этом

resOfSave = false;

//Пока файл не подготовлен к сохранению, выполнять..

while (resOfSave == false)

{

//.. если файл существует по данному пути, то..

if (File.Exists(savePathOfWorkPr + ".doc") || File.Exists(savePathOfWorkPr + ".docx"))

{

//Обработка этого события

}

else

{

//Подготовка к сохранению удачна, запись информации об этом

resOfSave = true;

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*38*

*Кнфу ХХХ.ХХХ.ПЗ*

}

}

//Попытаться сохранить файл..

try

{

//..сохранение составленного документа

((Word.\_Document)wordDocOfWorkProgram).SaveAs(…);

}

//.. и если возникло исключение, то..

catch (COMException err)

{

//..обработка исключений

}

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

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

//Закрытие файла Word

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

//Закрытие составленного документа

((Word.\_Document)wordDocOfWorkProgram).Close(…);

//Закрытие книги без сохранения

excelWorkbook.Close(Excel.XlSaveAction.xlDoNotSaveChanges, Type.Missing, Type.Missing);

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*39*

*Кнфу ХХХ.ХХХ.ПЗ*

//Закрытие приложения

((Word.\_Application)wordApp).Quit(…);

wordApp = null;

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

Эти процедуры позволяют легко управлять приложениями и документами. Реализованны запуск, закрытие приложений. Создание, открытие сохранение и закрытие документов.

Каждый документ представляет из себя объект с реализованными методами и свойствами. Весь контент документа - это несколько различных коллекций объектов: таблиц, параграфов, строк, столбцов и т.п. к каждой из которых у нас есть доступ. Оперируя ими можно анализировать документ, управлять информацией находящейся в нем, и на основе этой информации создавать новые документы.

Для хранения обработанной информации служат два массива: REZ – хранение названий учебных дисциплин и информацию о них(код предмета, код компетенции, номер семестра, адрес в таблице «Программа формирования компетенций») и KOM – хранение кодов компетенций и расшифровывающих эти коды названий. Эти массивы будут использоваться в качестве базы данных в процессе работы программы. Формат этих массивов уже обозначен в аналитической части.

Выше описанные массивы необходимы для более удобного представления информации, теперь вся интересующая нас информация будет храниться не в отдельных переменных, а в двух массивах. Это позволяет удобнее оперировать данными.

*Модуль анализа и обработки документов.*

Обработка файла учебный план заключается в следующем: после открытия файла программно запускается поиск средствами Office. По маске находятся

уникальные коды дисциплин и в соответствии с положением ячейки с кодом в таблице заполняется информация о названии учебной дисциплины и её «кода предмета» в массив REZ.

Обработка файлов паспортов компетенций заключается в следующем: после открытия очередного файла ПК также программно запускается поиск. По маске находятся коды компетенций и номера семестров, после чего они записываются в массив REZ в соответствии с названием предмета. По номеру семестра так же определяется лист книги Excel файла «Программа формирования компетенций»

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*40*

*Кнфу ХХХ.ХХХ.ПЗ*

Обработка файла компетентностной модели выпускника заключается в следующем: после открытия файла КМВ опять же запускается программно поиск. По маске ищутся коды компетенций и соответствующее название этого кода относительно положения ячейки кода в таблице. Полученные данные записываются в массив KOM.

*Модуль генерирования документов* описан двумя пунктами, которые представлены ниже (п. 2.5.1, п.2.5.2).

**2.5.1 Документ «Программа формирования компетенций»**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*41*

*Кнфу ХХХ.ХХХ.ПЗ*

Документ «Программа формирования компетенций» формируется на основе массивов REZ и KOM, используя такие значения как: название предмета, код компетенции, номер семестра, номер листа книги, адрес ячейки в книге для записи единицы на пересечении названия предмета и соответствующего кода компетенции. Формат составляемого файла – Excel. Первый лист служит для записи информации о предметах 1-4 семестров, второй лист – для записи информации о предметах 5-8 семестров.

Основные процедуры генерирования документа «Программа формирования компетенций» по пунктам:

* Создание файла "Программа формирования компетенций".
* Предварительное заполнение таблицы "Программа формирования компетенций".
* Проверка на дублирование названий предметов.
* Обработка первого листа.
* Запись заголовка "Программа формирования компетенций".
* Запись заголовка "Семестры (годы) обучения".
* Запись заголовков семестров ("1 семестр", "2 семестр" и т.д.).
* Запись заголовка "Сумма компетенций по дисциплине за 1-4 семестры".
* Запись кодов компетенций в соответствующие ячейки.
* Заполнение таблицы "Программа формирования компетенций".
* Запись формул для сумм столбца "Сумма компетенций по дисциплине за 1-4 семестры".
* Запись строки "Сумма по компетенции в семестре" и соответствующих формул для сумм.
* Обработка второго листа (аналогично обработке первого).
* Сохранение файла "Программа формирования компетенций".

**2.5.2 Документ «Рабочая программа учебной дисциплины»**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*42*

*Кнфу ХХХ.ХХХ.ПЗ*

Документы «Рабочая программа учебной дисциплины» формируются на основе массивов REZ и KOM, используя такие значения как: название предмета, код предмета, код компетенции, название компетенции. Формат составляемых файлов – Word.

Основные процедуры формирования документов «Рабочая программа учебной дисциплины» по пунктам:

* Создание файла Word.
* Заполнение титульной страницы.
* Подготовка к заполнению Раздела 1.
* Подготовка к заполнению Раздела 2.
* Заполнение Раздела 1 и Раздела 2.
* Подготовка к заполнению Раздела 3.
* Заполнение Раздела 3.
* Сохранение файла "Рабочая программа учебной дисциплины".

**3. Экспериментально – тестовая часть**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*43*

*Кнфу ХХХ.ХХХ.ПЗ*

Для оценки надёжности и эффективности программного обеспечения проведены ряд экспериментов (табл. 5), по каждому пункту проводилось не менее десяти испытаний четырьмя экспертами.

Таблица 5

**Цели тестирования**

|  |  |
| --- | --- |
| Эксперимент | Цель |
| Подача на вход программы ложных документов | Проверить реакцию программы на нестандартный случай. |
| Использование обрабатываемого документа во время работы программы, а так же сохранение результатов в используемый файл. | Исследование способности программы к обеспечению информационной безопасности. |
| Подача на вход папки с программами компетенций с дублированными документами. | Исследование реакции программы на нестандартную ситуацию. |
| Измерение времени, требуемого человеку и программе для составления документов | Исследование эффективности программы. |

Эксперименты проводились на операционной системе Windows Vista Home Premium с документами программ Microsoft Office Word 2007 и Microsoft Office Excel 2007. В четвёртом испытании принимал участие заведующий кафедрой ИВС Мясников Владимир Иванович и доцент Васяева Елена Семёновна.

После проведения описанных экспериментов получились следующие результаты:

1. *Подача на вход программы ложных документов*

В случае пустого документа «Учебный план» или «Компетентностная модель выпускника» программа, предварительно спросив пользователя, занесёт информацию о названиях предметов и кодах компетенций исходя из паспортов компетенций. В случае пустых документов «Паспорт компетенции», программа выведет сообщение о невозможности найти запись о семестрах предмета.

В случае ложных документов реакция программы схожа со случаем пустых документов, только ещё программа предупредит пользователя, что документы не соответствуют эталонным.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*44*

*Кнфу ХХХ.ХХХ.ПЗ*

В итоге программа собирает всю доступную ей информацию и предупреждает пользователя о нестандартном случае, что доказывает её устойчивость.

1. *Использование обрабатываемого документа во время работы программы, а так же сохранение результатов в используемый файл.*

Программа ни разу не сохранила документ в существующий, не спросив об этом пользователя. В случае невозможности сохранения по указанному пути, файлы сохраняются в резервный каталог, либо в тот же каталог, но уникально переименовываются (привязка к системному времени).

В итоге все открытые документы остаются в сохранности, а обработанные документы в любом случае сохраняются, если пользователь не захотел обратного. Это доказывает надёжность программы.

1. *Подача на вход папки с программами компетенций с дублированными документами.*

В этом случае пользователю предоставят выбор, что сделать: прервать обработку, переименовать предмет или сублимировать информацию из нескольких дублированных документов (в случае, если паспорта компетенций описывают разные семестры одного и того же предмета). В случае, если и названия предметов и номера семестров одинаковые, то дублированный документ не будет обрабатываться, правильным считается первый документ, поступивший на обработку.

В итоге информация максимально сохраняется, несмотря на одинаковые названия паспортов компетенций.

1. *Измерение времени, требуемого человеку и программе для составления документов.*

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*45*

*Кнфу ХХХ.ХХХ.ПЗ*

Эксперту потребовалось во много раз больше времени для составления данных документов вручную. Причём, составляя эти документы беспрерывно, растёт процент вероятности ошибок. С помощью данной программы возможно безошибочное составление этих документов. Временные затраты будут минимальны. На рис. 3 представлен график зависимости потраченного времени (в минутах) на создание документов вручную и с помощью программы от объёма ООП (точками отмечен график работы программы).

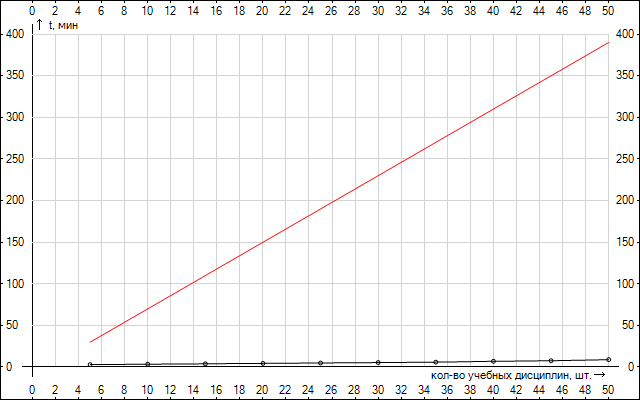


Рис. 3 График зависимости потраченного времени (в минутах) на создание документов вручную и с помощью программы от объёма ООП.

На рис. 4 показана другая сторона процесса: зависимость потраченного времени (в минутах) на создание документов одной учебной дисциплины вручную и с помощью программы от количества компетенций в данной учебной дисциплине.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*46*

*Кнфу ХХХ.ХХХ.ПЗ*

Рис. 4 Зависимость потраченного времени (в минутах) на создание документов вручную и с помощью программы от количества компетенций в учебной дисциплине.

**Заключение**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*47*

*Кнфу ХХХ.ХХХ.ПЗ*

В работе были разработаны алгоритмы и программа автоматизации процесса составления «Программы формирования компетенций» и создания таблиц описания формируемых учебной дисциплиной компетенций в «Рабочей программе учебной дисциплины». Программа написана на языке C# в среде разработки Microsoft Visual Studio 2010 на платформе .NET с использованием динамических библиотек Microsoft Office Word и Microsoft Office Excel. ПО удовлетворяет всем требованиям, которые были предъявлены. Однако функционал должен быть повышен в следующих редакциях данной разработки, так как не только два документа(«Программа формирования компетенций», «Рабочая программа учебной дисциплины») требуют автоматизации. Данная программа поможет сотрудникам кафедр в составлении документов по ФГОС-3.

**Литература**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*48*

*Кнфу ХХХ.ХХХ.ПЗ*

1. Андреев В.И. Педагогика высшей школы. М., 2006. – 287 с.

2. Федеральный государственной образовательный стандарт высшего профессионального образования: проектирование основной образовательной программы: методические рекомендации и материалы/ под общей ред. Н.Н. Старыгиной – Йошкар-Ола: МарГТУ, 2010. – 120 с.

3. А. Хортон Visual C++ 2005:базовый курс. : Пер. с англ. – М. : ООО «И.Д. Вильямс», 2007. – 1152 с. : ил.- Парал. Тит. Англ.

4. Ожегов С.И. Словарь русского языка/ Под ред. И.Ю. Шведовой.-М.:Русский язык, 1983.

5. Дипломное проектирование: Методические указания для студентов специальности 220100 «Вычислительные машины, комплексы, системы и сети» /Сост. Т.Е. Бикус, А.А. Власов, Н.С. Васяева, и др.; Под ред. С.В. Старыгина. – Йошкар-Ола: МарГТУ, 2002. – 60 с.

**Источники в Internet**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*49*

*Кнфу ХХХ.ХХХ.ПЗ*

<http://msdn.microsoft.com/ru-ru/> - MSDN справочник по C#;

<http://wladm.narod.ru/C_Sharp/index.html> - Практика программирования на С# для Windows и Web в Microsoft Visual Studio 2003/2005/2008;

<http://www.ruweb.ru/article/obektnaya-model-excel-s-tochki-zreniya-razrabotchika-pod-net.html> - Объектная модель Excel с точки зрения разработчика под .NET;

<http://www.cyberforum.ru/csharp-net/> - Вопросы использования языка программирования C# .NET. Помощь в решении задач;

<http://translate.google.com/> - Переводчик Google;

<http://www.yotx.ru/> - Сервис онлайн построения графиков;

<http://ru.wikipedia.org/> - Википедия – свободная энциклопедия;

<http://yandex.ru/> - Поисковая система;

<http://gostexpert.ru/> - ГОСТ Эксперт - база ГОСТов РФ.

*КНФУ ХХХ.ХХХ.001*

*Спецификация*

*1*

*Документация*

*КНФУ ХХХ.ХХХ.ПЗ*

*Пояснительная записка*

*1*

*КНФУ ХХХ.ХХХ.ПЗ*

*Руководство пользователя*

*1*

*КНФУ ХХХ.ХХХ.ПЗ*

*Текст программы*

*1*

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*50*

*КНФУ XXX.XXX.001*

*Реценз.*

*Н. Контр.*

*Утверд.*

*Автоматизация процесса формирования документов основной образовательной программы ФГОС ВПО Спецификация*

*Лит.*

*Листов*

*116*

*ИВТ-41 МарГТУ*

##### Прим.

##### Обозначение

*Формат*

*Зона*

*Поз.*

*Кол.*

##### Наименование

*Разраб.*

*Ягодаров А.Е.*

*Провер.*

*Васяева Е.С.*

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*51*

*Кнфу ХХХ.ХХХ.ПЗ*

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«МАРИЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информатики и вычислительной техники

ПРИЛОЖЕНИЕ 1

**Система автоматизации процесса формирования документов**

**«Автоматизация процесса формирования документов основной образовательной программы ФГОС ВПО»**

**Руководство пользователя**

Разработчик: Ягодаров А.Е.

студент гр. ИВТ-41

Консультант: доцент Васяева Е.С.

Йошкар-Ола

2011

О программе

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*52*

*Кнфу ХХХ.ХХХ.ПЗ*

Система автоматизации процесса формирования документов

«Автоматизация процесса формирования документов основной образовательной программы ФГОС ВПО» предназначена для сотрудников кафедр высших учебных заведений, работающих по ФГОС-3. Программа позволяет генерировать по ФГОС-3 документ «Программа формирования компетенций» и таблицы документа «Рабочая программа учебной дисциплины», анализируя исходные документы. Исходными документами являются УП, КМВ, ПК. Пути к этим документам задаются пользователем.

К дополнительным функциям программы относятся функции защиты информации исходных документов от стирания или замены во время работы программы, выявление несоответствия исходных документов требуемым, добавление незнакомых названий учебных дисциплин (которых нет в УП) и кодов компетенций (которых нет в КМВ).

Интерфейс программы представлен на рис.5.

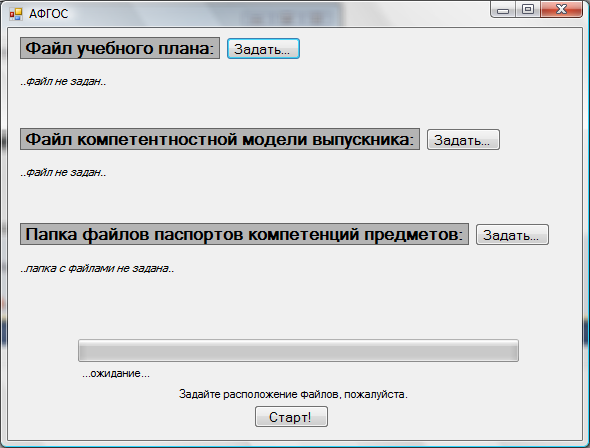


Рис.5

Диалог с пользователем ведётся с помощью диалоговых окон выбора файла, папки; сообщений информации, предупреждений, ошибок; контекстной информации.

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*53*

*Кнфу ХХХ.ХХХ.ПЗ*

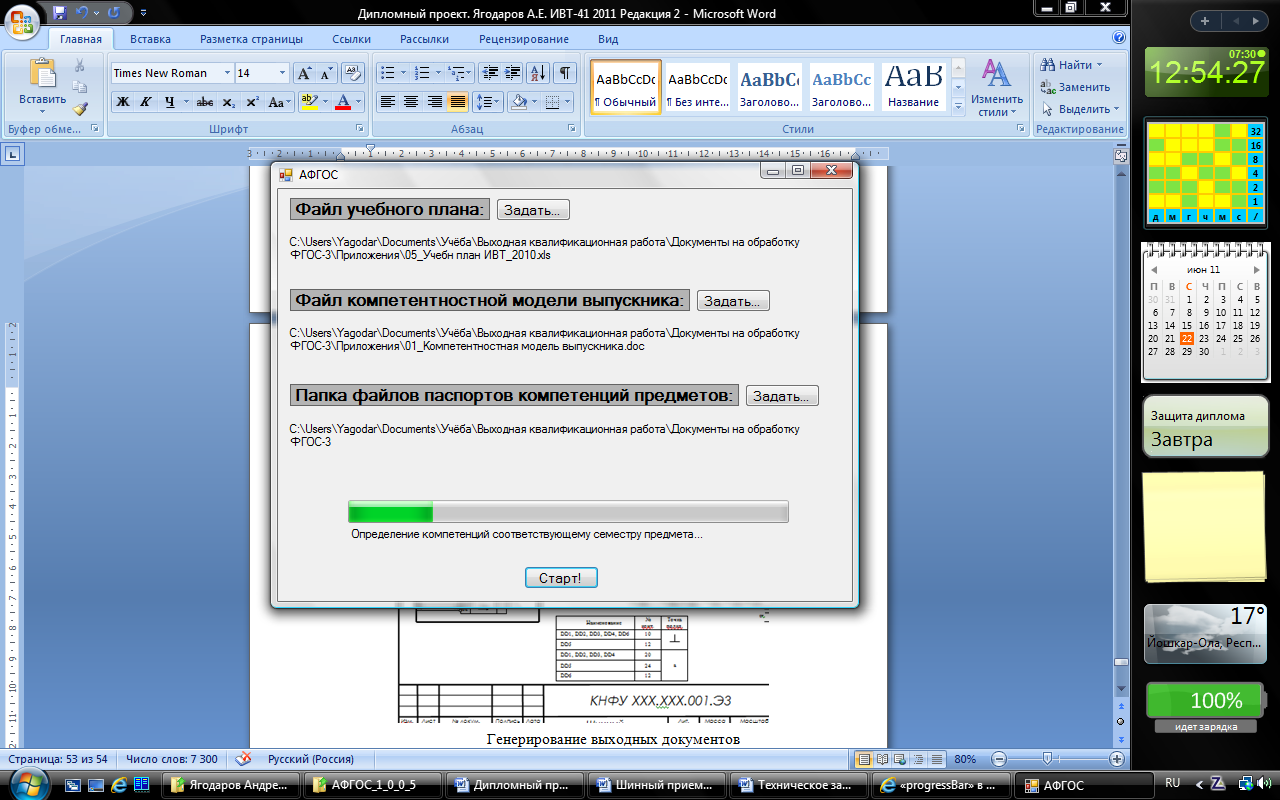
ПО рассчитано на работу в любой платформе с необходимым программным обеспечением (Microsoft Office, Framework 2.0) под управлением Windows XP, Windows Vista, Windows 7. Требования к компьютеру: Pentium Celeron, 32Мб ОЗУ, около 10 Мб на жёстком диске, клавиатура или манипулятор типа «мышь», монитор.

Задание исходных документов

Напротив каждого документа, который необходимо задать, расположена кнопка **«Задать…»**. Нажатие на неё вызовет диалоговое окно выбора файлы или папки. Чтобы начать анализирование и генерирование документов необходимо нажать на кнопку **«Старт!»**. Пока не будут заданы исходные документы, нажатие на эту кнопку вызовет появление окна с сообщением об ошибке задания файлов.

Обработка исходных документов

Если документы заданы и нажата кнопка **«Старт!»**, начнётся обработка исходных документов. Над кнопкой **«Старт!»** расположено поле состояния выполнения работы. По мере продвижения по процессу обработки документов, полоса состояния будет заполняться (рис. 6). Программа выведет все возникающие ошибки и предупреждения, а по окончанию обработки исходных документов выведет соответствующее сообщение. На этом анализ исходных документов закончится.



*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*54*

*Кнфу ХХХ.ХХХ.ПЗ*

Рис. 6

Генерирование выходных документов

Первым генерируется документ «Программа формирования компетенций». После генерации «Программы формирования компетенций» программа предложит пользователю выбрать путь сохранения файла. Если пользователь нажал «Отмена» на предложение выбрать путь сохранения файла, то программа автоматически сохранит документ в «Мои документы\АФГОС\_рез\_сохр» не прерывая работу. После сохранения «Программы формирования компетенций» начинается генерация таблиц «Рабочих программ учебных дисциплин» по приглашению пользователя выбрать папку сохранения файлов («Рабочие программы учебных дисциплин» генерируются для всех дисциплин из перечня файлов «Паспортов компетенций»). Если пользователь нажал «Отмена», то работа программы завершается. При удачном сохранении, программа возвращается в исходное состояние (рис.5).

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*55*

*Кнфу ХХХ.ХХХ.ПЗ*

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«МАРИЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информатики и вычислительной техники

ПРИЛОЖЕНИЕ 2

**Система автоматизации процесса формирования документов**

**«Автоматизация процесса формирования документов основной образовательной программы ФГОС ВПО»**

**Текст программы**

Разработчик: Ягодаров А.Е.

студент гр. ИВТ-41

Консультант: доцент Васяева Е.С.

Йошкар-Ола

2011

**Листинг программы**

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*56*

*Кнфу ХХХ.ХХХ.ПЗ*

«Programm.cs»

using System;

using System.Collections.Generic;

using System.Linq;

using System.Windows.Forms;

//Введение алиаса пространства имен Microsoft Office Excel

using Excel = Microsoft.Office.Interop.Excel;

namespace ПФК\_в.\_1

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

try

{

Application.Run(new Form1());

}

//Если исключение вида FileNotFoundException - не найдены файлы, то выполнять..

catch (System.IO.FileNotFoundException)

{

//..вывод сообщения об ошибке - нехватке файлов

MessageBox.Show("Не найдена одна из подключаемых библиотек:\n\"Microsoft.Office.Interop.Excel.dll\"\n" +

"\"Microsoft.Office.Interop.Word.dll\"\n\"Microsoft.Vbe.Interop.dll\"\n\"office.dll\"\n\n" +

"Приложение будет закрыто!", "Ошибка О\_002!",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

«Form1.cs»

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Runtime.InteropServices;

using System.Reflection;

//Введение алиаса пространства имен Microsoft Office Word

using Word = Microsoft.Office.Interop.Word;

//Введение алиаса пространства имен Microsoft Office Excel

using Excel = Microsoft.Office.Interop.Excel;

//Введение пространства имён для обеспечения ввода-вывода

using System.IO;

namespace ПФК\_в.\_1

{

public partial class Form1 : Form

{

//Глобальное определение объектов сервера Word

private Word.Application wordApp; //Сервер(приложение) Word

private Word.Documents wordDocuments; //Экземпляр Word с ссылками на все документы Word в приложении

private Word.Document wordDocOfPaspComp; //Экземпляр Word с открытым или загруженным документом паспотра компетенци предмета

private Word.Document wordDocOfCompModGrad; //Экземпляр Word с открытым или загруженным документом компетентностной модели выпускника

private Word.Document wordDocOfWorkProgram; //Экземпляр Word с открытым или загруженным документом рабочей программы дисциплины

private Word.Find wordFind; //Объект Word для осуществления поиска в документе

private Word.Range wordDocRange; //Объект Word для определения области документа

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*57*

*Кнфу ХХХ.ХХХ.ПЗ*

private Word.Table wordTable, wordTable1; //Объект Word для определения таблиц

//Глобальное определение объектов сервера Excel

private Excel.Application excelApp; //Сервер(приложение) Excel

private Excel.Workbooks excelWorkbooks; //Ссылки на книги приложения Excel

private Excel.Workbook excelWorkbook; //Объект книги

private Excel.Sheets excelSheets; //Ссылки на страницы книги

private Excel.Worksheet excelWorksheet; //Объект страницы

private Excel.Range excelCells; //Ссылки на объекты ячейки или группы ячеек

//Глобальное определение массива REZ - массив результатов анализа

//0 столбец - название предмета

//1 столбец - номер семестра, который соответствует названию предмета

//2 столбец - код компетенции, которы соответствует семестру, который соответствует названию предмета

//3 столбец - номер листа книги Excel, в который будет внесена запись об этом предмете

//4 столбец - адрес ячейки(строки) соответствующего листа, которую нужно пометить, которая обозначит всю информацию об этом предмете

//5 столбец - адрес ячейки(столбца) соответствующего листа, которую нужно пометить, которая обозначит всю информацию об этом предмете

//6 столбец - код соответствующего предмета

string[,] REZ = new string[1000, 7];

//Количество элементов массива REZ

int lastOfREZ = 0;

//Глобальное определение массива KOM - массив кодов компетенций

//0 столбец - код компетенции

//1 столбец - название компетенции

string[,] KOM = new string[500, 2];

//Количество элементов массива KOM

int lastOfKOM = 0;

//Глобальное определение строки кода предмета Б.4

string strSpecSubj\_1 = "Б.4";

//Глобальное определение строки кода предмета Б.5

string strSpecSubj\_2 = "Б.5";

//--------ФУНКЦИЯ------------------------------------------------------------------------------------------------------------------------------------

//Анализ названия предмета для удаления ненужных символов(в основном пробелов, но иногда и перевода строки '\n')

//--------ФУНКЦИЯ------------------------------------------------------------------------------------------------------------------------------------

private StringBuilder delSpFromNameOfSubj(StringBuilder strBuilder, int iSp)

{

//Замена символа перевода строки '\n' на пробел

strBuilder.Replace('\n', ' ');

//Замена символа перевода строки '\r' на пробел

strBuilder.Replace('\r', ' ');

//Замена символа перевода строки '\a' на пробел

strBuilder.Replace('\a', ' ');

//Замена символа перевода строки '\v' на пробел

strBuilder.Replace('\v', ' ');

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

//Замена всех разновидностей тире в строке на тире одного вида: "-"

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

strBuilder.Replace('-', '-');

strBuilder.Replace('‐', '-');

strBuilder.Replace('−', '-');

strBuilder.Replace('‒', '-');

strBuilder.Replace('­', '-');

strBuilder.Replace('⁃', '-');

strBuilder.Replace('–', '-');

strBuilder.Replace('—', '-');

strBuilder.Replace('―', '-');

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

//Пока iSp - индексы символа-пробела в названии предмета в строке strBuilder не будет равен индексу конечного символа строки, выполнять..

while (iSp != strBuilder.Length)

{

//.. если значение символа строки strBuilder по индексу iSp - это пробел, то..

if (strBuilder[iSp] == ' ')

{

//.. если рассматриваемый символ не первый и не последний и значения предыдущего символа и последующего символа не равны пробелу(то есть

//это пробел между словами), то..

if (iSp != 0 && iSp != (strBuilder.Length - 1) && strBuilder[iSp - 1] != ' ' && strBuilder[iSp + 1] != ' ')

{

//Увеличение индекса iSp на один - рассмотрение следующего символа в строке

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*58*

*Кнфу ХХХ.ХХХ.ПЗ*

iSp++;

//.. продолжить сканирование, пропустив последующие преобразования

continue;

}

//.. удалить пробел

strBuilder.Remove(iSp, 1);

//Продолжить сканирование, пропустив последующие преобразования

continue;

}

//Увеличение индекса iSp на один - рассмотрение следующего символа в строке

iSp++;

}

//Возвратить преобразованную строку strBuilder

return (strBuilder);

}

//--------КОНЕЦ-ФУНКЦИИ------------------------------------------------------------------------------------------------------------------------------

//--------ФУНКЦИЯ------------------------------------------------------------------------------------------------------------------------------------

//Анализ названия предмета для удаления ненужных символов(в основном пробелов, но иногда и перевода строки '\n')

//--------ФУНКЦИЯ------------------------------------------------------------------------------------------------------------------------------------

private void showMsgBreak()

{

//Вывод сообщения о прервании обработки

MessageBox.Show(this, "Обработка документов прервана!", "Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Information);

//Обнуление результирующего массива REZ и массива KOM для следующего запроса

toZeroREZandKOM();

//Увеличение прогрес бара до минимума

progressBar1.Value = progressBar1.Minimum;

//Смена надписи label8 взависимости от состояния

label8.Text = "...ожидание...";

}

//--------КОНЕЦ-ФУНКЦИИ------------------------------------------------------------------------------------------------------------------------------

//--------ФУНКЦИЯ------------------------------------------------------------------------------------------------------------------------------------

//Обнуление результирующего массива REZ и массива кодов компетенций KOM для следующего запроса

//--------ФУНКЦИЯ------------------------------------------------------------------------------------------------------------------------------------

private void toZeroREZandKOM()

{

//Объявление счётчика

int i;

//Для всех существующих элементов массива REZ

for (i = 0; i < lastOfREZ; i++)

{

//Обнуление элементов массива REZ

REZ[i, 0] = null;

REZ[i, 1] = null;

REZ[i, 2] = null;

}

//Обнуление индекса последнего элемента массива REZ

lastOfREZ = 0;

//Для всех существующих элементов массива KOM

for (i = 0; i < lastOfKOM; i++)

{

//Обнуление элементов массива KOM

KOM[i, 0] = null;

KOM[i, 1] = null;

}

//Обнуление индекса последнего элемента массива KOM

lastOfKOM = 0;

}

//--------КОНЕЦ-ФУНКЦИИ------------------------------------------------------------------------------------------------------------------------------

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*59*

*Кнфу ХХХ.ХХХ.ПЗ*

public Form1()

{

InitializeComponent();

//Включение-выключение видимости кнопки "Заполнить" для отладки и оптимизации

button5.Visible = false;

}

//Обработка нажатия на кнопку "Задать..." рядом с надписью "Файл учебного плана:"

private void button1\_Click(object sender, EventArgs e)

{

//Отображение диалога открытия файла. Если нажата кнопка "Открыть", то..

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

//.. запись пути и имени файла в соответствующую метку

//Изменение стиля надписи

label2.Font = new Font(label2.Font, 0);

//Запись в метку

label2.Text = openFileDialog1.FileName;

//Удаление пути к папке паспортов компетенций, если он существует

if(folderBrowserDialog1.SelectedPath.Length > 0 )

folderBrowserDialog1.SelectedPath = "";

//Запись пути к папке c документами в объект открытия папки с паспортами компетенций

folderBrowserDialog1.SelectedPath = openFileDialog1.FileName.Substring(0, openFileDialog1.FileName.LastIndexOf('\\'));

//Удаление значка об ошибке, если он есть

checkForFileStPlExist.Clear();

}

}

//Обработка нажатия на кнопку "Задать..." рядом с надписью "Файл компетентностной модели выпускника:"

private void button2\_Click(object sender, EventArgs e)

{

//Отображение диалога открытия файла. Если нажата кнопка "Открыть", то..

if (openFileDialog2.ShowDialog() == DialogResult.OK)

{

//.. запись пути и имени файла в соответствующую метку

//Изменение стиля надписи

label4.Font = new Font(label4.Font, 0);

//Запись в метку

label4.Text = openFileDialog2.FileName;

//Удаление пути к папке паспортов компетенций, если он существует

if (folderBrowserDialog1.SelectedPath.Length > 0)

folderBrowserDialog1.SelectedPath = "";

//Запись пути к папке c документами в объект открытия папки с паспортами компетенций

folderBrowserDialog1.SelectedPath = openFileDialog2.FileName.Substring(0, openFileDialog2.FileName.IndexOf('\\'));

//Удаление значка об ошибке, если он есть

checkForFileCompModGradExist.Clear();

}

}

//Обработка нажатия на кнопку "Задать..." рядом с надписью "Папка файлов паспортов компетенций предметов:"

private void button3\_Click(object sender, EventArgs e)

{

//folderBrowserDialog1.SelectedPath = openFileDialog2.FileName;

//Отображение диалога выбора папки. Если нажата кнопка "Открыть", то..

if (folderBrowserDialog1.ShowDialog() == DialogResult.OK)

{

//.. запись пути папки в соответствующую метку

//Изменение стиля надписи

label6.Font = new Font(label6.Font, 0);

//Запись в метку

label6.Text = folderBrowserDialog1.SelectedPath;

//Удаление значка об ошибке, если он есть

checkForFoldOfFilesPaspCompSubjExist.Clear();

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*60*

*Кнфу ХХХ.ХХХ.ПЗ*

}

}

//Обработка нажатия на кнопку "Старт!"

private void button4\_Click(object sender, EventArgs e)

{

//Если какой-либо файл или папка не заданы, то..

if (label2.Text == "..файл не задан.." || label4.Text == "..файл не задан.." || label6.Text == "..папка с файлами не задана..")

{

//.. вывод общего сообщения об ошибке

MessageBox.Show(this, "Файлы заданы некорректно!", "Ошибка О\_001!", MessageBoxButtons.OK, MessageBoxIcon.Error);

//Раскрытие label7 с надписью "Задайте расположение файлов, пожалуйста."

label7.Visible = true;

//Если файл учебного плана не задан, то..

if (label2.Text == "..файл не задан..")

//..вывод значка об ошибке

checkForFileStPlExist.SetError(button1, "Файл учебного плана не задан!");

//Если файл компетентностной модели выпускника не задан, то..

if (label4.Text == "..файл не задан..")

//..вывод значка об ошибке

checkForFileCompModGradExist.SetError(button2, "Файл компетентностной модели выпускника не задан!");

//Если папка файлов паспортов компетенций предметов не задана, то..

if (label6.Text == "..папка с файлами не задана..")

//..вывод значка об ошибке

checkForFoldOfFilesPaspCompSubjExist.SetError(button3, "Папка файлов паспортов компетенций предметов не задана!");

}

else

{

//Вывод сообщения о просьбе закрытия файлов

MessageBox.Show(this, "Убедительная просьба сохранить все изменения и закрыть документы учебного плана, компетентностной модели " +

"выпускника, паспортов компетенций дисциплин, если они открыты, во избежание потери информации.",

"Предупреждение П\_010!", MessageBoxButtons.OK, MessageBoxIcon.Warning);

//Скрытие label7 с надписью "Задайте расположение файлов, пожалуйста."

label7.Visible = false;

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

//Составление списка предметов на основе учебного плана.

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Составление списка предметов на основе учебного плана...";

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

//Открытие файла "Учебный план"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Открытие файла 'Учебный план'...";

//Создание объекта Excel.Application - равносильно запуску Microsoft Office Excel

excelApp = new Excel.Application();

//Включение видимости открытоко документа

//excelApp.Visible = true;

//Присвоение значания excelApp.Workbooks - совокупности ссылок на открытые рабочие книги, объекту excelWorkbooks

excelWorkbooks = excelApp.Workbooks;

//Открытие книги(файла) "Учебный план"

excelWorkbook = excelApp.Workbooks.Open(label2.Text, //FileName - имя открываемого файла файла

Type.Missing, //UpdateLinks - cпособ обновления ссылок в файле

true, //ReadOnly - при значении true открытие только для чтения

Type.Missing, //Format - определение формата символа разделителя

Type.Missing, //Password - пароль доступа к файлу до 15 символов

Type.Missing, //WriteResPassword - пароль на сохранение файла

Type.Missing, //IgnoreReadOnlyRecommended - при значении true отключается вывод запроса на работу

// без внесения изменений

Type.Missing, //Origin - тип текстового файла

Type.Missing, //Delimiter - разделитель при Format = 6

Type.Missing, //Editable - используется только для надстроек Excel 4.0

Type.Missing, //Notify - при значении true имя файла добавляется в список нотификации файлов

Type.Missing, //Converter - используется для передачи индекса конвертера файла используемого для

// открытия файла

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*61*

*Кнфу ХХХ.ХХХ.ПЗ*

Type.Missing, //AddToMRU - при true имя файла добавляется в список открытых файлов

Type.Missing, //Local - принимает значение true, если требуется сохранить файлы согласно языку

// Microsoft Excel. False(по умолчанию), если требуется сохранить файлы

// согласно языку Visual Basic For Applications

Type.Missing //CorruptLoad - способ испорченной загрузки. Может принимать следующие константы:

// xlNormalLoad, xlRepairFile, xlExtractData. По умолчанию - выбор

// автоматический

);

//Присвоение значания excelWorkbook.Worksheets - массив ссылок на листы книги, объекту excelSheets

excelSheets = excelWorkbook.Worksheets;

//Получение ссылки на первый лист книги

excelWorksheet = (Excel.Worksheet)excelSheets.get\_Item(1);

//Увеличение прогрес бара

progressBar1.Value++;

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Составление списка предметов на основе учебного плана...";

//Выделение группы ячеек через задание угловых ячеек диапазона

excelCells = excelWorksheet.get\_Range("A1", "A65536");

//Объявление ссылки на найденную ячейку(поиск произведён далее)

Excel.Range excelCellFound;

//Объявление ссылки на первую найденную ячейку

Excel.Range excelCellFoundFirst = null;

//Объявление ссылки на ячейку, в которой хранится название предмета

Excel.Range excelCellSubj;

//Объявление строк для записи адреса ячейки(address), значения ячейки(value), а также проверочной строки(chkZeros) для отбрасывания ненужных

//названий.

string address = "", value = "";

//Объявление объекта StringBuilder - изменяемую строку символов

StringBuilder strBuilder = new StringBuilder("");

//Первоначальный поиск: поиск ячеек с надписями вида "\*.?.??". Например: ПД.В.03, ГСЭ.Р.00, ЕН.Р.03

excelCellFound = excelCells.Find("\*.?.??", //What - информация, которую следует искать

Type.Missing, //After - ячейка, после которой следует начать поиск

Excel.XlFindLookIn.xlValues, //LookIn - тип информации

Excel.XlLookAt.xlPart, //LookAt - может быть одной из следующих констант

// Microsoft.Office.Interop.Excel.XlLookAt: xlWhole, xlPart

Excel.XlSearchOrder.xlByRows, //SearchOrder - может быть одной из следующих констант

// Microsoft.Office.Interop.Excel.XlSearchOrder: xlByRows,

// xlByColumns

Excel.XlSearchDirection.xlNext, //SearchDirection - направление поиска. Может быть одной из следующих

// констант XlSearchDirection: xlNext, default, xlPrevious

Type.Missing, //MatchCase - принимает true, если требуется сделать поиск чувствительным,

// по умолчанию - false

Type.Missing, //MatchByte - используется, если выбрано или установлено поддержка

// double-byte language

Type.Missing //SearchFormat - формат поиска

);

//Объявление счётчика i - для записи значений в результирующий массив REZ (это номер строки)

int i = 0;

//Увеличение прогрес бара

progressBar1.Value++;

//Если при первом же поиске ничего не нашлось, то..

if (excelCellFound == null)

{

//..это не файл учебного плана

//Если пользователь ответил нет на вопрос "Продолжить?", то..

if (MessageBox.Show(this, "Похоже, что файл '" + label2.Text + "' не является учебным планом!\nПродолжить обработку?",

"Предупреждение П\_006!", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.No)

{

//..вывод сообщения о завершении обработки документов

showMsgBreak();

//Закончить работу обработки документов

return;

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*62*

*Кнфу ХХХ.ХХХ.ПЗ*

}

}

//Объявление переменной для инкремента записи кода "Б.X" предмета

int codeB1 = 0;

//Объявление переменной для записи кода "Б.X.X" предмета

int codeB2 = 0;

//Объявление переменной для записи кода "Б.X.X.X" предмета

int codeB3;

//Пока ссылка на найденную ячейку существует, выполнять..

while (excelCellFound != null)

{

//..

//Если ссылки на первую найденную ячейку нет, то присвоить её.

if (excelCellFoundFirst == null)

excelCellFoundFirst = excelCellFound;

//Иначе..

else

//..если адреса найденной ячейки и первой найденной ячейки совпадают, то..

if (excelCellFound.get\_Address(Type.Missing, Type.Missing, Excel.XlReferenceStyle.xlA1, Type.Missing, Type.Missing) ==

excelCellFoundFirst.get\_Address(Type.Missing, Type.Missing, Excel.XlReferenceStyle.xlA1, Type.Missing, Type.Missing))

//..закончить цикл

break;

//Запись адреса найденной ячейки в строку address

address = excelCellFound.get\_Address(Type.Missing, Type.Missing, Excel.XlReferenceStyle.xlA1, Type.Missing, Type.Missing);

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

//Отбрасывание ненужных названий напротив кодов "\*.?.00" . Например "Базовая часть".

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Отбрасывание ненужных названий напротив кодов '\*.?.00' . Например 'Базовая часть'...";

//Выделение ячейки по адресу

excelCellSubj = excelWorksheet.get\_Range(address, Type.Missing);

//Запись значения выделенной ячейки в строку value

value = Convert.ToString(excelCellSubj.get\_Value(Type.Missing));

//Добавление к strBuilder ("") значения выделенной ячейки

strBuilder.Append(value);

//Если предпоследний и последний символ - '0' ((strBuilder.Length - 1) - индекс последнего символа в строке)

//(то етсь это код вида "\*.?.00", например "ГСЭ.Ф.00"), то..

if (strBuilder[strBuilder.Length - 2] == '0' && strBuilder[strBuilder.Length - 1] == '0')

{

//..

//Если четвёртый с конца символ - 'Ф', то..

if (strBuilder[strBuilder.Length - 4] == 'Ф')

//Инкремент кода "Б.X" предмета

codeB1++;

//Удаление ранее записанного значения выделенной ячейки в strBuilder

strBuilder.Remove(0, strBuilder.Length);

//Поиск следующей ячейки по заданному формату поиска (excelCells.Find)

excelCellFound = excelCells.FindNext(excelCellFound);

//Продолжить, пропустив последующие преобразования

continue;

}

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

//Вычисление кода "Б.X.X" предмета

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

//Если четвёртый с конца символ - 'Ф', то..

if (strBuilder[strBuilder.Length - 4] == 'Ф')

//Запись соответствующего кода "Б.X.X" предмета

codeB2 = 1;

//Если четвёртый с конца символ - 'Р', то..

if (strBuilder[strBuilder.Length - 4] == 'Р' || strBuilder[strBuilder.Length - 4] == 'P')

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*63*

*Кнфу ХХХ.ХХХ.ПЗ*

//Запись соответствующего кода "Б.X.X" предмета

codeB2 = 2;

//Если четвёртый с конца символ - 'В', то..

if (strBuilder[strBuilder.Length - 4] == 'В' || strBuilder[strBuilder.Length - 4] == 'B')

//Запись соответствующего кода "Б.X.X" предмета

codeB2 = 3;

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

//Вычисление кода "Б.X.X.X" предмета

codeB3 = Convert.ToInt32(strBuilder.Remove(0, strBuilder.Length - 2).ToString());

//Удаление ранее записанного адреса найденной ячейки в strBuilder

strBuilder.Remove(0, strBuilder.Length);

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Составление списка предметов на основе учебного плана...";

//Добавление к strBuilder ("") адреса найденной ячейки

strBuilder.Append(address);

//Замена символа A в адресе на символ B для того, чтобы впоследствии прочитать название предмета(сдвиг курсора на столбец вправо).

//Например: $A$34 -> $B$34.

strBuilder.Replace('A', 'B', 1, 1);

//Запись преобразованного адреса в строку address

address = strBuilder.ToString();

//Удаление ранее записанного адреса найденной ячейки в strBuilder

strBuilder.Remove(0, strBuilder.Length);

//Выделение ячейки по преобразованному адресу

excelCellSubj = excelWorksheet.get\_Range(address, Type.Missing);

//Запись значения выделенной ячейки в строку value

value = Convert.ToString(excelCellSubj.get\_Value(Type.Missing));

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

//Анализ строки value для разбиения некоторых строк на подстроки(например иногда требуется разбить

//"Инженерная этика / Инженерная психология / Этика и профессиональные ценности"

//на "Инженерная этика", "Инженерная психология", "Этика и профессиональные ценности".

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Анализ строки названия предмета для разбиения на несколько предметов...";

//Объявление счётчика j - для анализа строки value (это индекса символа, по который заканчивается название предмета в строке с

//несколькими названиями предметов)

int j = 0;

//Объявление счётчика k - для анализа строки value (это длина названия предмета)

int k = 0;

//Объявление индекса indx - для анализа строки value (это индекс символа, с которого начинается название предмета в строке с

//несколькими названиями предметов)

int indx = 0;

//Добавление к strBuilder ("") строки value - значения выделенной ячейки

strBuilder.Append(value);

//Пока j - индекса символа, по который заканчивается название предмета в строке strBuilder не будет равен инжексу конечного символа

//строки

while (j != strBuilder.Length)

{

//..выполнять

//Если символ strBuilder[j] это '/'..

if (strBuilder[j] == '/')

{

//то..

//Удаление символов до символа с индексом indx - перед названием предмета

strBuilder.Remove(0, indx);

//Удаление символов после символа с индексом k - после названия предмета

strBuilder.Remove(k, strBuilder.Length - k);

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*64*

*Кнфу ХХХ.ХХХ.ПЗ*

//Удаление ненужных пробелов из названия предмета

strBuilder = delSpFromNameOfSubj(strBuilder, 0);

//Запись названия предмета в первый столбец массива результатов анализа REZ

REZ[i, 0] = strBuilder.ToString(0, strBuilder.Length);

//Запись в массив REZ кода "Б.X.X.X" предмета

REZ[i, 6] = "Б." + codeB1 + "." + codeB2 + "." + codeB3;

//Удаление ранее записанной строки strBuilder

strBuilder.Remove(0, strBuilder.Length);

//Добавление к strBuilder ("") строки value - значения выделенной ячейки

strBuilder.Append(value);

//Увеличение i на один - увеличение номера строки массива REZ на один

i++;

//Замена значения индекса символа, с которого начинается название предмета в строке strBuilder. Теперь - уже после '/'

indx = j + 1;

//Обнуление длины названия предмета. -1 это компенсация перемещения indx на один вперёд.

k = -1;

}

//Увеличение k на один - увеличение длины названия предмета на один

k++;

//Увеличение j на один - увеличение номера индекса символа, по который заканчивается название предмета в строке strBuilder на

//один

j++;

}

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Составление списка предметов на основе учебного плана...";

//Удаление символов до символа с индексом indx - перед названием предмета

strBuilder.Remove(0, indx);

//Удаление ненужных пробелов из названия предмета

strBuilder = delSpFromNameOfSubj(strBuilder, 0);

//Запись названия предмета в первый столбец массива результатов анализа REZ

REZ[i, 0] = strBuilder.ToString(0, strBuilder.Length);

//Запись в массив REZ кода "Б.X.X.X" предмета

REZ[i, 6] = "Б." + codeB1 + "." + codeB2 + "." + codeB3;

//Увеличение i на один - увеличение номера строки массива REZ на один

i++;

//Замена значения индекса символа, с которого начинается название предмета в строке strBuilder. Теперь - начальная позиция

indx = 0;

//Замена значения индекса символа, по который заканчивается название предмета в строке strBuilder. Теперь - начальная позиция

j = 0;

//Обнуление длины названия предмета

k = 0;

//Удаление ранее записанной строки value в strBuilder

strBuilder.Remove(0, strBuilder.Length);

//Поиск следующей ячейки по заданному формату поиска (excelCells.Find)

excelCellFound = excelCells.FindNext(excelCellFound);

}

//Сохранение индекса строчки, с которой массив REZ пуст

lastOfREZ = i;

//Увеличение прогрес бара

progressBar1.Value++;

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

//Запись в результирующий массив REZ предмета с специальным кодом №1 (строка strSpecSubj\_1)

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

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*65*

*Кнфу ХХХ.ХХХ.ПЗ*

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись в результирующий массив REZ предмета с специальным кодом №1 (строка strSpecSubj\_1)...";

//Поиск ячейки со значением кода специального предмета №1 (строка strSpecSubj\_1)

excelCellFound = excelCells.Find(strSpecSubj\_1, Type.Missing, Excel.XlFindLookIn.xlValues, Excel.XlLookAt.xlPart,

Excel.XlSearchOrder.xlByRows, Excel.XlSearchDirection.xlNext, Type.Missing, Type.Missing, Type.Missing);

if (excelCellFound != null)

{

//Запись адреса найденной ячейки в строку address

address = excelCellFound.get\_Address(Type.Missing, Type.Missing, Excel.XlReferenceStyle.xlA1, Type.Missing, Type.Missing);

//Добавление к strBuilder ("") адреса найденной ячейки

strBuilder.Append(address);

//Замена символа A в адресе на символ B для того, чтобы впоследствии прочитать название предмета(сдвиг курсора на столбец вправо).

//Например: $A$34 -> $B$34.

strBuilder.Replace('A', 'B', 1, 1);

//Запись преобразованного адреса в строку address

address = strBuilder.ToString();

//Удаление ранее записанного адреса найденной ячейки в strBuilder

strBuilder.Remove(0, strBuilder.Length);

//Выделение ячейки по преобразованному адресу

excelCellSubj = excelWorksheet.get\_Range(address, Type.Missing);

//Запись значения выделенной ячейки в строку value

value = Convert.ToString(excelCellSubj.get\_Value(Type.Missing));

//Добавление к strBuilder ("") строки value - значения выделенной ячейки

strBuilder.Append(value);

//Удаление ненужных пробелов из названия предмета

strBuilder = delSpFromNameOfSubj(strBuilder, 0);

//Запись названия предмета в первый столбец массива результатов анализа REZ

REZ[lastOfREZ, 0] = strBuilder.ToString(0, strBuilder.Length);

//Запись в массив REZ кода "Б.X" предмета

REZ[lastOfREZ, 6] = strSpecSubj\_1;

//Увеличение индекса элемента массива REZ, который по прядку сразу за последним

lastOfREZ++;

//Удаление ранее записанной строки value в strBuilder

strBuilder.Remove(0, strBuilder.Length);

}

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Запись в результирующий массив REZ предмета с специальным кодом №2 (строка strSpecSubj\_2)

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись в результирующий массив REZ предмета с специальным кодом №1 (строка strSpecSubj\_2)...";

//Поиск ячейки со значением кода специального предмета №2 (строка strSpecSubj\_2)

excelCellFound = excelCells.Find(strSpecSubj\_2, Type.Missing, Excel.XlFindLookIn.xlValues, Excel.XlLookAt.xlPart,

Excel.XlSearchOrder.xlByRows, Excel.XlSearchDirection.xlNext, Type.Missing, Type.Missing, Type.Missing);

if (excelCellFound != null)

{

//Запись адреса найденной ячейки в строку address

address = excelCellFound.get\_Address(Type.Missing, Type.Missing, Excel.XlReferenceStyle.xlA1, Type.Missing, Type.Missing);

//Добавление к strBuilder ("") адреса найденной ячейки

strBuilder.Append(address);

//Замена символа A в адресе на символ B для того, чтобы впоследствии прочитать название предмета(сдвиг курсора на столбец вправо).

//Например: $A$34 -> $B$34.

strBuilder.Replace('A', 'B', 1, 1);

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*66*

*Кнфу ХХХ.ХХХ.ПЗ*

//Запись преобразованного адреса в строку address

address = strBuilder.ToString();

//Удаление ранее записанного адреса найденной ячейки в strBuilder

strBuilder.Remove(0, strBuilder.Length);

//Выделение ячейки по преобразованному адресу

excelCellSubj = excelWorksheet.get\_Range(address, Type.Missing);

//Запись значения выделенной ячейки в строку value

value = Convert.ToString(excelCellSubj.get\_Value(Type.Missing));

//Добавление к strBuilder ("") строки value - значения выделенной ячейки

strBuilder.Append(value);

//Удаление ненужных пробелов из названия предмета

strBuilder = delSpFromNameOfSubj(strBuilder, 0);

//Запись названия предмета в первый столбец массива результатов анализа REZ

REZ[lastOfREZ, 0] = strBuilder.ToString(0, strBuilder.Length);

//Запись в массив REZ кода "Б.X" предмета

REZ[lastOfREZ, 6] = strSpecSubj\_2;

//Увеличение индекса элемента массива REZ, который по прядку сразу за последним

lastOfREZ++;

//Удаление ранее записанной строки value в strBuilder

strBuilder.Remove(0, strBuilder.Length);

}

//Увеличение прогрес бара

progressBar1.Value++;

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Составление списка предметов на основе учебного плана...";

//Удаление какой-либо записанной строки в strBuilder

strBuilder.Remove(0, strBuilder.Length);

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Закрытие файла "Учебный план"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Закрытие файла 'Учебный план'...";

//Отсечение запроса на сохранение для книги, так как никаких изменений программой не делается(книга открыта только на чтение)

//(документ может измениться самим приложением Excel - например, пересчёт формул)

excelWorkbook.Saved = true;

//Закрытие книги без сохранения

excelWorkbook.Close(Excel.XlSaveAction.xlDoNotSaveChanges, Type.Missing, Type.Missing);

//Закрытие приложения

excelApp.Quit();

//Обнуление объекта Excel.Application

excelApp = null;

//Увеличение прогрес бара

progressBar1.Value++;

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

//Если режим отдаки и оптимизации, то..

if (button5.Visible == true)

//..вывести окно, информирующее выполнение составления списка названий предметов

MessageBox.Show(this, "Файл учебного плана обработан!", "Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Information);

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*67*

*Кнфу ХХХ.ХХХ.ПЗ*

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

//Составление массива из названий файлов "Паспорт компетенции ..."

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Составление массива из названий файлов 'Паспорт компетенции ...' ...";

//Объявление массива строк для записи названий файлов паспортов компетенций

string[] NKOM = new string[0];

//Выполнять, пока не возникло исключение

try

{

//Запись в массив строк-названий файлов паспортов компетенций названия этих файлов в указанной папке и подпапках указанной папки

NKOM = Directory.GetFiles(label6.Text, "Паспорт\*компетенции\*.doc\*", SearchOption.AllDirectories);

}

//Если возникло исключение, то..

catch

{

//.. папка не может быть прочитана

//.. если пользователь ответил нет на вопрос "Продолжить?", то..

if (MessageBox.Show(this, "Папка " + label6.Text + " не может быть прочитана!\nПродолжить?", "Предупреждение П\_001!",

MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.No)

{

//..вывод сообщения о завершении обработки документов

showMsgBreak();

//Закончить работу обработки документов

return;

}

}

//Объявление флага FD - flagOfDecision - флаг того, сообщалось ли пользователю, что не найдены паспорты компетенций

//true когда пользователю уже сообщалось о том, что паспорта компетенций не найдены

bool FD = false;

//Если файлы паспортов компетенцийпо маске "Паспорт\*компетенции\*.doc\*" не найдены, то..

if (NKOM.Length == 0)

{

//.. если пользователь ответил нет на вопрос "Продолжить?", то..

if (MessageBox.Show(this, "Файлы паспортов компетенций не найдены!\nПродолжить?", "Предупреждение П\_002!",

MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.No)

{

//..обнуление результирующего массива REZ и массива KOM

toZeroREZandKOM();

//Вывод сообщения о завершении обработки документов

showMsgBreak();

//Завершение обработки

return;

}

else

//Смена флага на true - пользователь осведомлён о том, что паспорта компетенций не найдены

FD = true;

}

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Определение компетенций соответствующему семестру предмета

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Определение компетенций соответствующему семестру предмета...";

if (FD == true)

//Пользователь ответил "Да" на вопрос "Файлы паспортов компетенций не найдены! Продолжить?"

MessageBox.Show(this, "Работа продолжена!\rПаспорта компетенций не будут обработаны!", "Выполнение...",

MessageBoxButtons.OK, MessageBoxIcon.Information);

//Создаем объект Word - равносильно запуску Word

wordApp = new Word.Application();

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*68*

*Кнфу ХХХ.ХХХ.ПЗ*

//Включение видимости открытоко документа

wordApp.Visible = false;

//Оределение индекса массива NKOM(массива названий фалов паспортов кометенций) и nameOfSubj(массив соответствующих названий

//предметов)

i = 0;

//Оределение индекса символа строк strOfNumSem

int h = 0;

//Объявление массива строк nameOfSubj, куда будет записываться название предмета в соответствии с обрабатываемым файлом

string[] nameOfSubj = new string[1000];

//Объявление массива номеров семестров предмета

string[] numbOfSem = new string[8];

//Объявление строки strOfNumSem, куда будет записываться строка с номерами семестров

string strOfNumSem;

//Объявление индекса массива REZ

int g = 0;

//Объявление индекса для проверки массива nameOfSubj на дублирование названий предметов

int m = 0;

//Объявление булевой переменной для записи информации о закрытии документа при коллизиях названий файлов

bool closed = true;

//Увеличение прогрес бара

progressBar1.Value++;

while (i != NKOM.Length)

{

//Включение видимости открытоко документа

wordApp.Visible = false;

//Открытие файла "Паспорт компетенции ..."

wordDocOfPaspComp = wordApp.Documents.Open(NKOM[i], //FileName - имя открываемого файла файла

Type.Missing, //ConfirmConversions - true, чтобы открыть диалог конвертации, если файл не

//в формате Microsoft Office Word

true, //ReadOnly - при значении true открытие только для чтения

false, //AddToRecentFiles - при true имя файла добавляется в список открытых файлов

Type.Missing, //PasswordDocument - пароль доступа к файлу

Type.Missing, //PasswordTemplate - пароль доступа к шаблону

false, //Revert - определяет, что делать, если открываемый документ уже открыт.

//True - переоткрыть документ без сохранения, false - активировать открытый

//документ

Type.Missing, //WritePasswordDocument - пароль для сохранения файла

Type.Missing, //WritePasswordTemplate - пароль для сохранения шаблона

Type.Missing, //Format - определяет, какой конвертировщик будет использован при открытии

//файла. Одна из следующих Word.WdOpenFormat констант: wdOpenFormatAllWord,

//wdOpenFormatAuto, wdOpenFormatDocument, wdOpenFormatEncodedText,

//wdOpenFormatRTF, wdOpenFormatTemplate, wdOpenFormatText,

//wdOpenFormatUnicodeText или wdOpenFormatWebPages. По умолчанию

//wdOpenFormatAuto.

Type.Missing, //Encoding - определяет кодировку открываемого файла

Type.Missing, //Visible - видимость открытого документа

Type.Missing, //OpenAndRepair - открытие с восстановлением

Type.Missing, //DocumentDirection - направление текста - одна из Word.WdDocumentDirection

//констант: WdLeftToRight, WdRightToLeft.

false, //NoEncodingDialog - при true подавляется показ диалогового окна Encoding,

//которое отображается если кодировка не распознана.

Type.Missing //XMLTransform - определяет тип XML данных при XML преобразованиях

);

//Определение области документа для поиска(выделяется область в виде всего документа)

wordDocRange = wordDocOfPaspComp.Range(0, wordDocOfPaspComp.Characters.Count);

//Определение объекта для поиска в определённой ранее области wordDocRange

wordFind = wordDocRange.Find;

//Очистака параметров предыдущего поиска во избежание ошибок, так как формат поиска сохраняется

wordFind.ClearFormatting();

//Запуск поиска..Компетентностная модель выпускника

//Если документ начинается с комбинации символов "\*Паспорт\*компетенции\*Учебная\*дисциплина\*«\*»", то..

if (wordFind.Execute("Паспорт\*компетенции\*Учебная\*дисциплина\*«\*»", //FindText - текст для поиска

Type.Missing, //MatchCase - чувствительность к регистру при true

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*69*

*Кнфу ХХХ.ХХХ.ПЗ*

Type.Missing, //MatchWholeWord - только слово целиком

true, //MatchWildcards - использовать или нет подстановочные знаки

Type.Missing, //MatchSoundsLike - использовать настройку "произносится как"

Type.Missing, //MatchAllWordForms - все словоформы

true, //Forward - направление поиска вперёд

Type.Missing, //Wrap - управляет таким случаем, когда поиск начинается не с

//начала документа, а конец документа уже достигнут

Type.Missing, //Format - поиск форматирования

Type.Missing, //ReplaceWith - заменяемый текст

Type.Missing, //Replace - сколько замен: одна, две или нисколько

Type.Missing, //MatchKashida - специальный поиск в Арабских документах

Type.Missing, //MatchDiacritics - специальный поиск в документак, написанных

//справа-налево

Type.Missing, //MatchAlefHamza - специальный поиск в Арабских документах

Type.Missing //MatchControl - специальный поиск в документак, написанных

//справа-налево

) == true)

{

//..значит открыт документ паспорта компетенции некоего предмета

//Очистака параметров предыдущего поиска во избежание ошибок, так как формат поиска сохраняется

wordFind.ClearFormatting();

//Переопределение области документа для поиска(выделяется область в виде всего документа)(переопределение из-за того, что во

//время удачного поиска, область изменяется на найденную)

wordDocRange.Start = 0;

wordDocRange.End = wordDocOfPaspComp.Characters.Count;

//Поиск названия предмета в переопределённой области

wordFind.Execute("«\*»", Type.Missing, Type.Missing, true, Type.Missing, Type.Missing, true, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing);

//Добавление к strBuilder ("") строки результата поиска(это название предмета в кавычках)

strBuilder.Append(wordDocRange.Text);

//Удаление из названия предмета кавычки спереди

strBuilder.Remove(0, 1);

//Удаление из названия предмета кавычки вконце

strBuilder.Remove(strBuilder.Length - 1, 1);

//Удаление ненужных символов из названия предмета

strBuilder = delSpFromNameOfSubj(strBuilder, 0);

//Запись преобразованного названия предмета в массив nameOfSubj

nameOfSubj[i] = strBuilder.ToString(0, strBuilder.Length);

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

//Проверка на дублирование названий предметов

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Проверка на дублирование названий предметов...";

//Обнуление проверочного индекса массива nameOfSubj

m = 0;

//Объявление переменной result типа DialogResult - туда будет записываться ответ пользователя на диалоговое окно

DialogResult result = DialogResult.None;

//Пока проверочный индекс не перевалил за пределы массива nameOfSubj

while (m != nameOfSubj.Length)

{

//Если проверочный индекс совпадает с индексом рассматриваемого файла(чтобы не сработало дублирование на

//один и тот же эелемент)

if (m == i)

{

//Перейти на следующий элемент, увеличив индекс

m++;

//Продолжить сравнение элементов

continue;

}

//Если элемент с проверочным индексом совпадает с названием предмета обрабатываемого файла, то..

if(nameOfSubj[m] == nameOfSubj[i])

{

//Закрытие обрабатываемого документа во избежания конфликтов во время показа следующего сообщения(файлы могут открываться и закрыться)

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*70*

*Кнфу ХХХ.ХХХ.ПЗ*

try

{

((Word.\_Document)wordDocOfPaspComp).Close(Word.WdSaveOptions.wdDoNotSaveChanges, Type.Missing, Type.Missing);

}

//Если возникает исключение закрытия файла, то..

catch (COMException)

{

//Документ не закрыт, запись информации обэтом

closed = false;

}

//Закрытие приложения

((Word.\_Application)wordApp).Quit(Word.WdSaveOptions.wdDoNotSaveChanges, Word.WdOriginalFormat.wdOriginalDocumentFormat,

Type.Missing);

wordApp = null;

//..спросить у пользователя что делать: Прервать обработку, переименовать предмет, продолжить обработку?

//Запись ответа в переменную result

result = MessageBox.Show(this, "Название предмета `" + nameOfSubj[i] +

"` совпадает с уже обработанным!\n\nПохоже, что обрабатываемый файл\n\n" + NKOM[i] + "\n\n" +

"описывают один и тот же предмет, что и обработанный файл\n\n" + NKOM[m] +

"\n\nПрервать обработку, переименовать предмет " +

"(`Повтор`), продолжить обработку(`Пропустить`)?\n\n\\при переименовании к названию предмета добавляется \"(копия)\"\\",

"Предупреждение П\_004!", MessageBoxButtons.AbortRetryIgnore, MessageBoxIcon.Warning);

//Если документ закрыт, то..

if (closed == true)

{

//..открытие повторно

//Создаем объект Word - равносильно запуску Word

wordApp = new Word.Application();

//Включение видимости открытоко документа

wordApp.Visible = false;

//Повторное открытие файла "Паспорт компетенции ..."

wordDocOfPaspComp = wordApp.Documents.Open(NKOM[i], Type.Missing, true, false, Type.Missing, Type.Missing, false, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, false, Type.Missing);

//Определение области документа для поиска(выделяется область в виде всего документа)

wordDocRange = wordDocOfPaspComp.Range(0, wordDocOfPaspComp.Characters.Count);

//Определение объекта для поиска в определённой ранее области wordDocRange

wordFind = wordDocRange.Find;

}

//Обнуление булевой переменно closed

closed = true;

//Если ответ пользователя - прервать обработку, то..

if (result == DialogResult.Abort)

{

//..закрытие приложения

((Word.\_Application)wordApp).Quit(Word.WdSaveOptions.wdSaveChanges,

Word.WdOriginalFormat.wdOriginalDocumentFormat, Type.Missing);

wordApp = null;

//Обнуление результирующего массива REZ и массива KOM

toZeroREZandKOM();

//Вывод сообщения о завершении обработки документов

showMsgBreak();

//Завершение обработки

return;

}

//Если ответ пользователя - переименовать предмет, то..

if (result == DialogResult.Retry)

{

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*71*

*Кнфу ХХХ.ХХХ.ПЗ*

//..добавление к совпавшему названию "(копия)"

nameOfSubj[i] = nameOfSubj[i] + "(копия)";

//Запись нового названия предмета в массив REZ

REZ[lastOfREZ, 0] = nameOfSubj[i];

//Увеличение индекса элемента массива REZ, который по прядку сразу за последним

lastOfREZ++;

//Выход из цикла сравнения

break;

}

//Если ответ пользователя - продолжить обработку, то..

if (result == DialogResult.Ignore)

//Выход из цикла сравнения

break;

}

//Перейти на следующий элемент, увеличив индекс

m++;

}

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Определение компетенций соответствующему семестру предмета...";

//Очистака параметров предыдущего поиска во избежание ошибок, так как формат поиска сохраняется

wordFind.ClearFormatting();

//Переопределение области документа для поиска(выделяется область в виде всего документа)(переопределение из-за того, что во

//время удачного поиска, область изменяется на найденную)

wordDocRange.Start = 0;

wordDocRange.End = wordDocOfPaspComp.Characters.Count;

//Если поиск номеров семестров предмета в переопределённой области удачен, то..

if (wordFind.Execute("Семестр\*\r", Type.Missing, Type.Missing, true, Type.Missing, Type.Missing,

true, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing) == true)

{

//Удаление из строки strBuilder предыдущего значения

strBuilder.Remove(0, strBuilder.Length);

//..добавление к strBuilder ("") строки результата поиска(это строчка "Семестр\*\r". Например: "Семестр – 7,8")

strBuilder.Append(wordDocRange.Text);

//Удаление из строки результата поиска слова "Семестр"

strBuilder.Remove(0, 7);

//Удаление из строки результата поиска пробелов

strBuilder.Replace(" ","");

//Обнуление индекса символа строки strOfNumSem

h = 0;

//Запись в строку strOfNumSem обработанной строки strBuilder. Например: "–7,8"

strOfNumSem = strBuilder.ToString();

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

//Поиск и запись в массив всех названий компетенций в обрабатываемом файле, то есть всех названий компетенций которые

//относятся к данному предмету

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Поиск и запись в массив всех названий компетенций в обрабатываемом файле...";

//Удаление из строки strBuilder предыдущего значения

strBuilder.Remove(0, strBuilder.Length);

//Объявление массива строк nameOfComp, куда будет записываться название компетенции предмета в соответствии с обрабатываемым

//файлом

string[] nameOfComp = new string[1000];

//Оределение индекса массива строк nameOfComp

int c = 0;

//Оределение индекса символа строк strBuilder для записи компетенции в nameOfComp

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*72*

*Кнфу ХХХ.ХХХ.ПЗ*

int v = 0;

//Очистака параметров предыдущего поиска во избежание ошибок, так как формат поиска сохраняется

wordFind.ClearFormatting();

//Переопределение области документа для поиска(выделяется область в виде всего документа)(переопределение из-за того, что во

//время удачного поиска, область изменяется на найденную)

wordDocRange.Start = 0;

wordDocRange.End = wordDocOfPaspComp.Characters.Count;

//Объявление булевой переменной для записи удачности результата поиска кодов компетенций

bool resOfSearch = false;

//Пока поиск кодов компетенций предмета в переопределённой области удачен

//[(] - символ "("

//\* - комбинация любых символов

//[А-ЯЁ]@ - комбинация символов прописных русских букв

//[-‐−‒­⁃–—―] - все разновидности символа дефис(их 9, включая символ переноса('­')(код 173), различные тире, минуса и т.д.)

//[0-9]@ - последовательность цифр

//[)] - символ ")"

//Под это поисковое слово как раз подходят коды компетенций,заключённые в скобки. Например: (ОК–10), (ПК-9), (ДПК-1)

resOfSearch = wordFind.Execute("[(]\*[А-ЯЁ]@\*[-‐−‒­⁃–—―]\*[0-9]@\*[)]", Type.Missing, Type.Missing, true, Type.Missing,

Type.Missing, true, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing);

//Пока результаты поиска удачны, выполнять

while(resOfSearch == true)

{

//Добавление к strBuilder ("") строки результата поиска(это строчка "[(]\*[-‐−‒­⁃–—―]\*[0-9]@\*[)]". Например: "ОК-10")

strBuilder.Append(wordDocRange.Text);

//Удаление из строки результата поиска пробелов

strBuilder.Replace(" ", "");

//Запись кода компетенции в массив nameOfComp уже в едином, стандартизованном виде

//(то етсь 'буквенный\_код''цифры')

//Пока индекс символа строки strBuilder не выходит за рамки элементов strBuilder

while (v < strBuilder.Length)

{

//Если рассматриваемый символ - не буква и не цифра, то..

if (strBuilder[v] == '(' || strBuilder[v] == ')' ||

strBuilder[v] == '‐' || strBuilder[v] == '−' || strBuilder[v] == '‒' || strBuilder[v] == 173 ||

strBuilder[v] == '⁃' || strBuilder[v] == '–' || strBuilder[v] == '—' || strBuilder[v] == '―' ||

strBuilder[v] == '-')

{

//..удаление этого символа из строки

strBuilder.Remove(v,1);

//Продолжить сканирование, без последующих команд

continue;

}

//Увеличение индекса символа строки strBuilder

v++;

}

//Запись в массив названий компетенций обработанное название компетенции

nameOfComp[c] = strBuilder.ToString();

//Увеличение индекса массива строк названий компетенций соответствующего предмета

c++;

//Поиск оставшихся кодов компетенций в документе и запись удачности поиска в resOfSearch(true или false)

resOfSearch = wordFind.Execute();

//Удаление из строки strBuilder предыдущего значения

strBuilder.Remove(0, strBuilder.Length);

//Обнуление индекса символа строки strBuilder

v = 0;

}

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

//Смена надписи label8 взависимости от состояния

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*73*

*Кнфу ХХХ.ХХХ.ПЗ*

label8.Text = "Определение компетенций соответствующему семестру предмета...";

//Обнуление индекса массива строк nameOfComp

c = 0;

//Пока символ в строке номера семестра существует, выполнять

while (h != strOfNumSem.Length)

{

//Если рассматриваемый символ - цифра от 1 до 8, то..

if (strOfNumSem[h] == '1' || strOfNumSem[h] == '2' || strOfNumSem[h] == '3' || strOfNumSem[h] == '4' ||

strOfNumSem[h] == '5' || strOfNumSem[h] == '6' || strOfNumSem[h] == '7' || strOfNumSem[h] == '8')

{

//.. это и есть номер семестра

//Пока не будет найдено совпадение названия предмета в результирующем массиве с обрабатываемым названием,

//либо пока индекс не выходит за рамки REZ массива

while (nameOfSubj[i] != REZ[g, 0] && g < lastOfREZ)

//Инкремент индекса массива REZ

g++;

//Если индекс REZ массива вышел за рамки(а значит совпадения с таким названием предмета не найдено), то..

if (g == lastOfREZ)

{

//Обнуление переменной- ответа пользователя на диалоговое окно

result = DialogResult.None;

//Закрытие обрабатываемого документа во избежания конфликтов во время показа следующего сообщения(файлы могут открываться и закрыться)

try

{

((Word.\_Document)wordDocOfPaspComp).Close(Word.WdSaveOptions.wdDoNotSaveChanges, Type.Missing, Type.Missing);

}

//Если возникает исключение закрытия файла, то..

catch (COMException)

{

//Документ не закрыт, запись информации обэтом

closed = false;

}

//Закрытие приложения

((Word.\_Application)wordApp).Quit(Word.WdSaveOptions.wdDoNotSaveChanges, Word.WdOriginalFormat.wdOriginalDocumentFormat,

Type.Missing);

wordApp = null;

//Вывод сообщения с предложением добавить название данного предмета в REZ массив

result = MessageBox.Show(this, "В файле\n\n`" + NKOM[i] + "`\n\nнеизвестное название предмета:\n\n`" +

nameOfSubj[i] + "`\n\nДобавить предмет?", "Предупреждение П\_005!",

MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

//Если документ закрыт успешно, то..

if (closed == true)

{

//..открытие повторно

//Создаем объект Word - равносильно запуску Word

wordApp = new Word.Application();

//Включение видимости открытоко документа

wordApp.Visible = false;

//Повторное открытие файла "Паспорт компетенции ..."

wordDocOfPaspComp = wordApp.Documents.Open(NKOM[i], Type.Missing, true, false, Type.Missing, Type.Missing, false, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, false, Type.Missing);

}

//Обнуление булевой переменно closed

closed = true;

//..если пользователь ответит "Да" на сообщение с предложением добавить название данного предмета в

//REZ массив, то..

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*74*

*Кнфу ХХХ.ХХХ.ПЗ*

if (result == DialogResult.Yes)

{

//.. пока код компетенции по индексу c существует,

//выполнять

while (nameOfComp[c] != null)

{

//Запись нового названия предмета в массив REZ

REZ[lastOfREZ, 0] = nameOfSubj[i];

//Запись напротив нового названия предмета в массиве REZ номер соответствующего семестра

REZ[lastOfREZ, 1] = strOfNumSem[h].ToString();

//Запись напротив номера соответствующего семестра в массиве REZ, код компетенции

REZ[lastOfREZ, 2] = nameOfComp[c];

//Увеличение индекса массива строк названий компетенций соответствующего предмета

c++;

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

//Вычисление листа, на котором будет выводится информация о предмете

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Вычисление листа, на котором будет выводится информация о предмете...";

//Если семестр - с первого по четвёртый, то..

if (strOfNumSem[h] == '1' || strOfNumSem[h] == '2' || strOfNumSem[h] == '3' || strOfNumSem[h] == '4')

{

//..это первый лист

//Запись соответствующей записи

REZ[lastOfREZ, 3] = Convert.ToString("1");

}

//Если семестр - с пятого по восьмой, то..

if (strOfNumSem[h] == '5' || strOfNumSem[h] == '6' || strOfNumSem[h] == '7' || strOfNumSem[h] == '8')

{

//..это второй лист

//Запись соответствующей записи

REZ[lastOfREZ, 3] = Convert.ToString("2");

}

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

//Увеличение индекса элемента массива REZ, который по прядку сразу за последним

lastOfREZ++;

}

//Обнуление индекса массива строк nameOfComp

c = 0;

//Увеличение индекса символа строк strBuilder на один

h++;

//Обнуление индекса миссива REZ

g = 0;

//Переход на следующий шаг цикла

continue;

}

}

//Если ячейка для номера семестра рядом с найденным названием предмета в массиве REZ не занята, то..

if (REZ[g, 1] == "" || REZ[g, 1] == null)

{

//Запись напротив найденного названия предмета в массиве REZ номер соответствующего семестра

REZ[g, 1] = strOfNumSem[h].ToString();

//Запись напротив номера соответствующего семестра в массиве REZ, код компетенции

REZ[g, 2] = nameOfComp[c];

//Увеличение индекса массива строк названий компетенций соответствующего предмета

c++;

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

//Вычисление листа, на котором будет выводится информация о предмете

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

//Смена надписи label8 взависимости от состояния

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*75*

*Кнфу ХХХ.ХХХ.ПЗ*

label8.Text = "Вычисление листа, на котором будет выводится информация о предмете...";

//Если семестр - с первого по четвёртый, то..

if (strOfNumSem[h] == '1' || strOfNumSem[h] == '2' || strOfNumSem[h] == '3' || strOfNumSem[h] == '4')

{

//..это первый лист

//Запись соответствующей записи

REZ[g, 3] = Convert.ToString("1");

}

//Если семестр - с пятого по восьмой, то..

if (strOfNumSem[h] == '5' || strOfNumSem[h] == '6' || strOfNumSem[h] == '7' || strOfNumSem[h] == '8')

{

//..это второй лист

//Запись соответствующей записи

REZ[g, 3] = Convert.ToString("2");

}

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

//.. пока код компетенции по индексу c существует,

//выполнять

while (nameOfComp[c] != null)

{

//Запись нового названия предмета в массив REZ

REZ[lastOfREZ, 0] = nameOfSubj[i];

//Запись напротив нового названия предмета в массиве REZ номер соответствующего семестра

REZ[lastOfREZ, 1] = strOfNumSem[h].ToString();

//Запись напротив номера соответствующего семестра в массиве REZ, код компетенции

REZ[lastOfREZ, 2] = nameOfComp[c];

//Увеличение индекса массива строк названий компетенций соответствующего предмета

c++;

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

//Вычисление листа, на котором будет выводится информация о предмете

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Вычисление листа, на котором будет выводится информация о предмете...";

//Если семестр - с первого по четвёртый, то..

if (strOfNumSem[h] == '1' || strOfNumSem[h] == '2' || strOfNumSem[h] == '3' || strOfNumSem[h] == '4')

{

//..это первый лист

//Запись соответствующей записи

REZ[lastOfREZ, 3] = Convert.ToString("1");

}

//Если семестр - с пятого по восьмой, то..

if (strOfNumSem[h] == '5' || strOfNumSem[h] == '6' || strOfNumSem[h] == '7' || strOfNumSem[h] == '8')

{

//..это второй лист

//Запись соответствующей записи

REZ[lastOfREZ, 3] = Convert.ToString("2");

}

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

//Увеличение индекса элемента массива REZ, который по прядку сразу за последним

lastOfREZ++;

}

//Обнуление индекса массива строк nameOfComp

c = 0;

}

//Иначе

else

{

//Проверка на повторную обработку одного и того же премета(повторно обрабатывается паспорт

//компетенции одного и того же предмета)

//Если поле номера семестра напротив найденого названия предмета совпадает и с номером семестра, то..

if (REZ[g, 1] == strOfNumSem[h].ToString())

{

//..эта ситуация - дублирование паспорта компетенции

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*76*

*Кнфу ХХХ.ХХХ.ПЗ*

//Вывод сообщения об этом.

MessageBox.Show(this, "Данные паспорта компетенции предмета `" + nameOfSubj[i] + "` файла `"

+ NKOM[i] + "` совпадает с уже обработанным и обрабатываться повторно не " +

"будут. Сверьте правильность названий и дублирование документов!",

"Предупреждение П\_003!", MessageBoxButtons.OK,

MessageBoxIcon.Warning);

//Обнуление индекса миссива REZ

g = 0;

//Переход к обработке следующего файла

break;

}

//.. пока код компетенции по индексу c существует,

//выполнять

while (nameOfComp[c] != null)

{

//Запись названия предмета в новую строку (незанятую) массива REZ

REZ[lastOfREZ, 0] = nameOfSubj[i];

//Запись напротив нового названия предмета в массиве REZ номер соответствующего семестра

REZ[lastOfREZ, 1] = strOfNumSem[h].ToString();

//Запись напротив номера соответствующего семестра в массиве REZ, код компетенции

REZ[lastOfREZ, 2] = nameOfComp[c];

//Увеличение индекса массива строк названий компетенций соответствующего предмета

c++;

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

//Вычисление листа, на котором будет выводится информация о предмете

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Вычисление листа, на котором будет выводится информация о предмете...";

//Если семестр - с первого по четвёртый, то..

if (strOfNumSem[h] == '1' || strOfNumSem[h] == '2' || strOfNumSem[h] == '3' || strOfNumSem[h] == '4')

{

//..это первый лист

//Запись соответствующей записи

REZ[lastOfREZ, 3] = Convert.ToString("1");

}

//Если семестр - с пятого по восьмой, то..

if (strOfNumSem[h] == '5' || strOfNumSem[h] == '6' || strOfNumSem[h] == '7' || strOfNumSem[h] == '8')

{

//..это второй лист

//Запись соответствующей записи

REZ[lastOfREZ, 3] = Convert.ToString("2");

}

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

//Увеличение индекса элемента массива REZ, который по прядку сразу за последним

lastOfREZ++;

}

//Обнуление индекса массива строк nameOfComp

c = 0;

}

//Обнуление индекса миссива REZ

g = 0;

}

//Увеличение индекса символа строк strBuilder на один

h++;

}

}

//Иначе

else

//Вывод сообщения об ошибке

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*77*

*Кнфу ХХХ.ХХХ.ПЗ*

MessageBox.Show(this, "В файле `" + NKOM[i] + "` невозможно найти запись о номерах семестров", "Ошибка О\_004!",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

//Иначе этот документ не является паспортом компетенции некой дисциплины

else

//Вывод сообщения об ошибке

MessageBox.Show(this, "Файл `" + NKOM[i] + "` не является паспортом компетенции!", "Ошибка О\_003!", MessageBoxButtons.OK,

MessageBoxIcon.Error);

//Удаление из строки strBuilder предыдущего значения

strBuilder.Remove(0, strBuilder.Length);

//Закрытие обработанного документа

try

{

((Word.\_Document)wordDocOfPaspComp).Close(Word.WdSaveOptions.wdDoNotSaveChanges, Type.Missing, Type.Missing);

}

//Если возникает исключение закрытия файла, то..

catch (COMException)

{

//..вывод сообщения об ошибке закрытия файла компетентностной модели выпускника

MessageBox.Show("Файл паспорта копетенций \"" + NKOM[i] + "\" не может быть закрыт. Возможно он был " +

"открыт до запуска программы или во время работы программы! При необходимости сохраните все " +

"изменения в нём и перезапустите цикл обработки документов. Работа программы продолжится без " +

"закрытия данного документа!", "Ошибка О\_006!",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

//Увеличение индекса массива NKOM - массива строк названий файлов паспортов компетенций

i++;

}

//Увеличение прогрес бара

progressBar1.Value++;

//Смена надписи label8 взависимости от состояния

label8.Text = "Определение компетенций соответствующему семестру предмета...";

//Если режим отдаки и оптимизации, то..

if (button5.Visible == true)

//..вывести окно, информирующее выполнение обработки паспортов компетенций предметов

MessageBox.Show(this, "Паспорта компетенций обработаны!", "Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Information);

//Удаление какой-либо записанной строки в strBuilder

strBuilder.Remove(0, strBuilder.Length);

//Открытие файла "Компетентностная модель выпускника"

wordDocOfCompModGrad = wordApp.Documents.Open(label4.Text, Type.Missing, true, false, Type.Missing, Type.Missing, false,

Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, false, Type.Missing);

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

//Проверка на достоверность файла компетентностной моедели выпускника

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Проверка на достоверность файла компетентностной моедели выпускника...";

//Определение области документа для поиска(выделяется область в виде всего документа)

wordDocRange = wordDocOfCompModGrad.Range(0, wordDocOfCompModGrad.Characters.Count);

//Определение объекта для поиска в определённой ранее области wordDocRange

wordFind = wordDocRange.Find;

//Очистака параметров предыдущего поиска во избежание ошибок, так как формат поиска сохраняется

wordFind.ClearFormatting();

//Запуск поиска..

//Если документ не начинается с комбинации символов "Компетентностная[ ]@модель[ ]@выпускника", то..

if (wordFind.Execute("Компетентностная\*модель\*выпускника", Type.Missing, Type.Missing, true, Type.Missing, Type.Missing,

true, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing) == false)

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*78*

*Кнфу ХХХ.ХХХ.ПЗ*

{

//..значит открыт документ далеко не компетентностной модели выпускника

//Если пользователь ответил нет на вопрос "Продолжить?", то..

if (MessageBox.Show(this, "Похоже, что файл '" + label4.Text + "' не является компетентностной моделью выпускника!\n" +

"Продолжить обработку?", "Предупреждение П\_007!", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.No)

{

//..вывод сообщения о завершении обработки документов

showMsgBreak();

//Закончить работу обработки документов

return;

}

}

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

//Увеличение прогрес бара

progressBar1.Value++;

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

//Составление списка из названий компетенций на основе документа компетентностной модели выпускника

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Составление списка из названий компетенций на основе документа компетентностной модели выпускника...";

//Определение области документа для поиска(выделяется область в виде всего документа)

wordDocRange = wordDocOfCompModGrad.Range(0, wordDocOfCompModGrad.Characters.Count);

//Определение объекта для поиска в определённой ранее области wordDocRange

wordFind = wordDocRange.Find;

//Очистака параметров предыдущего поиска во избежание ошибок, так как формат поиска сохраняется

wordFind.ClearFormatting();

//Объявление булевой переменной для записи удачности результата поиска кодов компетенций

bool resOfSearchKMV = false;

//Объявление индекса символа строк strBuilder для записи компетенции в KOM

int t = 0;

//Объявление индекса строк KOM для проверки на дублирование

int p = 0;

//Объявление булевой переменной для записи результата совпадения кодов компетенций

bool repeatKOM = false;

//Оределение индекса массива строк nameOfComp

lastOfKOM = 0;

//Запуск поиска и запись результата(удачно или нет) в перемнную resOfSearchKMV

resOfSearchKMV = wordFind.Execute("<[А-ЯЁ]@[-‐−‒­⁃–—―]\*[0-9]@>", Type.Missing, Type.Missing, true, Type.Missing, Type.Missing,

true, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing);

//Увеличение прогрес бара

progressBar1.Value++;

//Объявление переменной для проверки достоверности документа

int forCheck = 0;

//Объявление строки для записи полного названия компетенции

string fullNameOfComp = "";

//Объявление StringBuilder строки для преобразования полного названия компетенции

StringBuilder strBuilderForFullNameComp = new StringBuilder("");

//Пока в документе найдена комбинация символов "<[А-ЯЁ]@[-‐−‒­⁃–—―]\*[0-9]@>" (равнозначно коду компентенции "ОК-2" или

//"ОК- 2"), выполнять

while (resOfSearchKMV == true)

{

//Обнуление переменной для проверки достоверности документа

forCheck = 0;

//Обнуление строки для записи полного названия компетенции

fullNameOfComp = "";

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*79*

*Кнфу ХХХ.ХХХ.ПЗ*

//Добавление к strBuilder ("") строки результата поиска(это строчка "[(]\*[-‐−‒­⁃–—―]\*[0-9]@\*[)]". Например: "ОК-10")

strBuilder.Append(wordDocRange.Text);

//Выделение как мышью диапазона

wordDocRange.Select();

//Попытаться посчитать количество ячеек в выделенном диапазое, чтобы уточниться в достоверности документа

try

{

//Запись в forCheck количества ячеек в выделенном диапазоне

forCheck = wordApp.Selection.Cells.Count;

//Если ячеек - одна, то..

if (forCheck == 1)

{

//..записать в forCheck количество таблиц в документе

forCheck = wordDocOfCompModGrad.Tables.Count;

//Если количество таблиц - одна штука, то..

if (forCheck == 1)

{

//..определние этой таблицы в объект

wordTable = wordDocOfCompModGrad.Tables[1];

//Если заголовок третъего столбца этой таблицы начинается с "Название", то..

if (wordTable.Cell(1, 3).Range.Text.IndexOf("Название") == 0)

{

//..запись полного названия соответствующего кода компетенции

fullNameOfComp = wordTable.Cell(wordApp.Selection.Cells[1].RowIndex, 3).Range.Text;

//Добавление к strBuilderForFullNameComp ("") строки с полным названием компетенции

strBuilderForFullNameComp.Append(fullNameOfComp);

//Запись обработанного названия компетенции(с удалением пробелов и переносов строки)

fullNameOfComp = delSpFromNameOfSubj(strBuilderForFullNameComp, 0).ToString();

//Удаление из строки strBuilderForFullNameComp предыдущего значения

strBuilderForFullNameComp.Remove(0, strBuilderForFullNameComp.Length);

}

else

{

//Иначе запись в полное название компетенции - "!!! Неизвестное название компетенции !!!"

fullNameOfComp = "!!! Неизвестное название компетенции !!!";

}

}

else

{

//Иначе запись в полное название компетенции - "!!! Неизвестное название компетенции !!!"

fullNameOfComp = "!!! Неизвестное название компетенции !!!";

}

}

else

{

//Иначе запись в полное название компетенции - "!!! Неизвестное название компетенции !!!"

fullNameOfComp = "!!! Неизвестное название компетенции !!!";

}

}

//Если исключение, то..

catch (COMException err)

{

//..если исключение вызвано тем, что не существует в данном месте ячейки таблицы, то..

if (err.Message.IndexOf("Запрашиваемый номер семейства не существует.") == 0)

{

//...запись в полное название компетенции - "!!! Неизвестное название компетенции !!!"

fullNameOfComp = "!!! Неизвестное название компетенции !!!";

}

}

//Удаление из строки результата поиска пробелов

strBuilder.Replace(" ", "");

//Запись кода компетенции в массив nameOfComp уже в едином, стандартизованном виде

//(то етсь 'буквенный\_код''цифры')

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*80*

*Кнфу ХХХ.ХХХ.ПЗ*

//Пока индекс символа строки strBuilder не выходит за рамки элементов strBuilder

while (t < strBuilder.Length)

{

//Если рассматриваемый символ - не буква и не цифра, то..

if (strBuilder[t] == '‐' || strBuilder[t] == '−' || strBuilder[t] == '‒' || strBuilder[t] == 173 ||

strBuilder[t] == '⁃' || strBuilder[t] == '–' || strBuilder[t] == '—' || strBuilder[t] == '―' ||

strBuilder[t] == '-')

{

//..удаление этого символа из строки

strBuilder.Remove(t, 1);

//Продолжить сканирование, без последующих команд

continue;

}

//Увеличение индекса символа строки strBuilder

t++;

}

//Переключение repeatKOM в начальное состояние

repeatKOM = false;

//Проверка на дублирование названия компетенции

for (p = 0; p < lastOfKOM; p++)

{

//Если рассматриваемый код компетенции уже записан в массив KOM, то..

if (KOM[p, 0] == strBuilder.ToString())

{

//Переключение repeatKOM в true - то есть дублирование

repeatKOM = true;

//Завершить цикл

break;

}

}

//Если дублирований нет, то..

if (repeatKOM == false)

{

//Запись в массив названий компетенций обработанное название кода компетенции

KOM[lastOfKOM, 0] = strBuilder.ToString();

//Запись в массив названий компетенций обработанное название компетенции

KOM[lastOfKOM, 1] = fullNameOfComp;

//Увеличение индекса массива строк названий компетенций соответствующего предмета

lastOfKOM++;

}

//Поиск оставшихся кодов компетенций в документе и запись удачности поиска в resOfSearch(true или false)

resOfSearchKMV = wordFind.Execute();

//Удаление из строки strBuilder предыдущего значения

strBuilder.Remove(0, strBuilder.Length);

//Обнуление индекса символа строки strBuilder

t = 0;

}

//Увеличение прогрес бара

progressBar1.Value++;

//Определение области документа для поиска(выделяется область в виде всего документа)

wordDocRange = wordDocOfCompModGrad.Range(0, wordDocOfCompModGrad.Characters.Count);

//Определение объекта для поиска в определённой ранее области wordDocRange

wordFind = wordDocRange.Find;

//Очистака параметров предыдущего поиска во избежание ошибок, так как формат поиска сохраняется

wordFind.ClearFormatting();

//Запуск поиска и запись результата(удачно или нет) в перемнную resOfSearchKMV

resOfSearchKMV = wordFind.Execute("<[А-ЯЁ]@[ ]@[-‐−‒­⁃–—―]\*[0-9]@>", Type.Missing, Type.Missing, true, Type.Missing,

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*81*

*Кнфу ХХХ.ХХХ.ПЗ*

Type.Missing, true, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing);

//Увеличение прогрес бара

progressBar1.Value++;

//Обнуление переменной для проверки достоверности документа

forCheck = 0;

//Обнуление строки для записи полного названия компетенции

fullNameOfComp = "";

//Пока в документе найдена комбинация символов "<[А-ЯЁ]@[ ]@\*[-‐−‒­⁃–—―]\*[0-9]@>" (равнозначно коду компентенции "ДПК -2" или

//"ДПК - 2"), выполнять

while (resOfSearchKMV == true)

{

//Обнуление переменной для проверки достоверности документа

forCheck = 0;

//Обнуление строки для записи полного названия компетенции

fullNameOfComp = "";

//Добавление к strBuilder ("") строки результата поиска(это строчка "[(]\*[-‐−‒­⁃–—―]\*[0-9]@\*[)]". Например: "ОК-10")

strBuilder.Append(wordDocRange.Text);

//Выделение как мышью диапазона

wordDocRange.Select();

//Попытаться посчитать количество ячеек в выделенном диапазое, чтобы уточниться в достоверности документа

try

{

//Запись в forCheck количества ячеек в выделенном диапазоне

forCheck = wordApp.Selection.Cells.Count;

//Если ячеек - одна, то..

if (forCheck == 1)

{

//..записать в forCheck количество таблиц в документе

forCheck = wordDocOfCompModGrad.Tables.Count;

//Если количество таблиц - одна штука, то..

if (forCheck == 1)

{

//..определние этой таблицы в объект

wordTable = wordDocOfCompModGrad.Tables[1];

//Если заголовок третъего столбца этой таблицы начинается с "Название", то..

if (wordTable.Cell(1, 3).Range.Text.IndexOf("Название") == 0)

{

//..запись полного названия соответствующего кода компетенции

fullNameOfComp = wordTable.Cell(wordApp.Selection.Cells[1].RowIndex, 3).Range.Text;

//Добавление к strBuilderForFullNameComp ("") строки с полным названием компетенции

strBuilderForFullNameComp.Append(fullNameOfComp);

//Запись обработанного названия компетенции(с удалением пробелов и переносов строки)

fullNameOfComp = delSpFromNameOfSubj(strBuilderForFullNameComp, 0).ToString();

//Удаление из строки strBuilderForFullNameComp предыдущего значения

strBuilderForFullNameComp.Remove(0, strBuilderForFullNameComp.Length);

}

else

{

//Иначе запись в полное название компетенции - "!!! Неизвестное название компетенции !!!"

fullNameOfComp = "!!! Неизвестное название компетенции !!!";

}

}

else

{

//Иначе запись в полное название компетенции - "!!! Неизвестное название компетенции !!!"

fullNameOfComp = "!!! Неизвестное название компетенции !!!";

}

}

else

{

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*82*

*Кнфу ХХХ.ХХХ.ПЗ*

//Иначе запись в полное название компетенции - "!!! Неизвестное название компетенции !!!"

fullNameOfComp = "!!! Неизвестное название компетенции !!!";

}

}

//Если исключение, то..

catch (COMException err)

{

//..если исключение вызвано тем, что не существует в данном месте ячейки таблицы, то..

if (err.Message.IndexOf("Запрашиваемый номер семейства не существует.") == 0)

{

//...запись в полное название компетенции - "!!! Неизвестное название компетенции !!!"

fullNameOfComp = "!!! Неизвестное название компетенции !!!";

}

}

//Удаление из строки результата поиска пробелов

strBuilder.Replace(" ", "");

//Запись кода компетенции в массив nameOfComp уже в едином, стандартизованном виде

//(то етсь 'буквенный\_код''цифры')

//Пока индекс символа строки strBuilder не выходит за рамки элементов strBuilder

while (t < strBuilder.Length)

{

//Если рассматриваемый символ - не буква и не цифра, то..

if (strBuilder[t] == '‐' || strBuilder[t] == '−' || strBuilder[t] == '‒' || strBuilder[t] == 173 ||

strBuilder[t] == '⁃' || strBuilder[t] == '–' || strBuilder[t] == '—' || strBuilder[t] == '―' ||

strBuilder[t] == '-')

{

//..удаление этого символа из строки

strBuilder.Remove(t, 1);

//Продолжить сканирование, без последующих команд

continue;

}

//Увеличение индекса символа строки strBuilder

t++;

}

//Переключение repeatKOM в начальное состояние

repeatKOM = false;

//Проверка на дублирование названия компетенции

for (p = 0; p < lastOfKOM; p++)

{

//Если рассматриваемый код компетенции уже записан в массив KOM, то..

if (KOM[p, 0] == strBuilder.ToString())

{

//Переключение repeatKOM в true - то есть дублирование

repeatKOM = true;

//Завершить цикл

break;

}

}

//Если дублирований нет, то..

if (repeatKOM == false)

{

//Запись в массив названий компетенций обработанное название компетенции

KOM[lastOfKOM, 0] = strBuilder.ToString();

//Запись в массив названий компетенций обработанное название компетенции

KOM[lastOfKOM, 1] = fullNameOfComp;

//Увеличение индекса массива строк названий компетенций соответствующего предмета

lastOfKOM++;

}

//Поиск оставшихся кодов компетенций в документе и запись удачности поиска в resOfSearch(true или false)

resOfSearchKMV = wordFind.Execute();

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*83*

*Кнфу ХХХ.ХХХ.ПЗ*

//Удаление из строки strBuilder предыдущего значения

strBuilder.Remove(0, strBuilder.Length);

//Обнуление индекса символа строки strBuilder

t = 0;

}

//Увеличение прогрес бара

progressBar1.Value++;

//Закрытие обработанного документа

try

{

((Word.\_Document)wordDocOfCompModGrad).Close(Word.WdSaveOptions.wdDoNotSaveChanges, Type.Missing, Type.Missing);

}

//Если возникает исключение закрытия файла, то..

catch (COMException)

{

//..вывод сообщения об ошибке закрытия файла компетентностной модели выпускника

MessageBox.Show("Файл копметентностной модели выпускника \"" + label4.Text + "\" не может быть закрыт. Возможно он был " +

"открыт до запуска программы или во время работы программы! При необходимости сохраните все " +

"изменения в нём и перезапустите цикл обработки документов. Работа программы продолжится без " +

"закрытия данного документа!", "Ошибка О\_005!",

MessageBoxButtons.OK, MessageBoxIcon.Error);

}

//Закрытие приложения

((Word.\_Application)wordApp).Quit(Word.WdSaveOptions.wdSaveChanges, Word.WdOriginalFormat.wdOriginalDocumentFormat,

Type.Missing);

wordApp = null;

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

//Если режим отдаки и оптимизации, то..

if (button5.Visible == true)

//..вывести окно, информирующее выполнение составления списка названий кодов компетенций предметов

MessageBox.Show(this, "Список названий кодов компетенций предметов составлен!", "Выполнение...", MessageBoxButtons.OK,

MessageBoxIcon.Information);

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, "Документы 'Учебный план', все 'Паспорт компетенции ...', 'Компетентностная модель выпускника' " +

"обработаны. По полученным данным будет составлен документ 'Программа формирования компетенций'.",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Information);

//Увеличение прогрес бара

progressBar1.Value++;

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

//Создание файла "Программа формирования компетенций"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Создание файла 'Программа формирования компетенций'...";

//Создание объекта Excel.Application - равносильно запуску Microsoft Office Excel

excelApp = new Excel.Application();

//Включение видимости открытоко документа

//excelApp.Visible = true;

//Определение числа листов в создаваемой книге(файле)

excelApp.SheetsInNewWorkbook = 1;

//Добавление листов к книге

excelApp.Workbooks.Add(Type.Missing);

//Присвоение значания excelApp.Workbooks - совокупности ссылок на открытые рабочие книги, объекту excelWorkbooks

excelWorkbooks = excelApp.Workbooks;

//Определение, что работа производится с первой книгой

excelWorkbook = excelWorkbooks[1];

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*84*

*Кнфу ХХХ.ХХХ.ПЗ*

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Составление таблицы "Программа формирования компетенций"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Создание файла 'Программа формирования компетенций'...";

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

//Предварительное заполнение таблицы "Программа формирования компетенций"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Предварительное заполнение таблицы 'Программа формирования компетенций'...";

//Присвоение значания excelWorkbook.Worksheets - массив ссылок на листы книги, объекту excelSheets

excelSheets = excelWorkbook.Worksheets;

//Получение ссылки на первый лист книги

excelWorksheet = (Excel.Worksheet)excelSheets.get\_Item(1);

//Выделение ячейки для порядкового номера предмета(по сути изменение параметров производится для столбца)

excelCells = (Excel.Range)excelWorksheet.Cells[5, 1];

//Выравнивание надписи по горизонтали - по правому краю ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignRight;

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

//Ширина ячейки = 3.57

excelCells.ColumnWidth = 3.57;

//Выделение ячейки для названия предмета(по сути изменение параметров производится для столбца)

excelCells = (Excel.Range)excelWorksheet.Cells[5, 2];

//Выравнивание надписи по горизонтали - по левому краю ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignLeft;

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

//Ширина ячейки = 70.71

excelCells.ColumnWidth = 70.71;

//Увеличение прогрес бара

progressBar1.Value++;

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

//Проверка на дублирование названий предметов

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Заполнение таблицы 'Программа формирования компетенций'...";

//Обнуление проверочного индекса расммотренных названий предметов массива REZ

m = 0;

//Обнуление проверочного индекса расмматриваемого названия предмета массива REZ

i = 0;

//Объявление переменной - индекс и одновременно порядковый номер каода компетенции

int ki = 0;

//Объявление переменной - порядкового номера предмета

int numOfSubj = 1;

//Булевая переменная, где хранится результат совпадения рассматриваемого названия(i) с уже рассмотренными(m)

bool resOfCol = false;

//Увеличение прогрес бара

progressBar1.Value++;

//Объявление булевой переменной для хранения результата проверки на совпадение кодов компетенций массивов REZ и KOM

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*85*

*Кнфу ХХХ.ХХХ.ПЗ*

bool newCom = true;

//Пока название предмета не null и не "", выполнять

while (REZ[i, 0] != null && REZ[i, 0] != "")

{

//Обнуление проверочного индекса расммотренных название предметов массива REZ

m = 0;

//Пока индекс рассмотренных предметов меньше, чем индекс рассматриваемого, выполнять

while (m < i)

{

//Если название рассмотренного предмета совпало с уже рассмотренным, то..

if (REZ[m, 0] == REZ[i, 0])

{

//..запись параметров

//Запись адреса строки только что рассмотренного предмета

REZ[i, 4] = REZ[m, 4];

//Обнуление переменной - индекс и одновременно порядковый номер каода компетенции

ki = 0;

//Переключение newCom в true, в начальное состояние

newCom = true;

//Пока значение массива названий KOM по индексу ki существует, выполнять

while (KOM[ki, 0] != null)

{

//Если код компетенции рассматриваемого названия предмета совпал с кодом компетенции в массиве всех компетенций, то..

if (REZ[i, 2] == KOM[ki, 0])

{

//Переключение newCom в false, так как совпадение найдено

newCom = false;

//Закончить цикл вычисления порядкового номера кода компетенции

break;

}

//Увеличение порядкового номера

ki++;

}

//Если совпадения кодов компетенций не найдено, то..

if (newCom == true && REZ[i, 2] != "" && REZ[i, 2] != null)

{

//.. это новый код компетенции

//Если пользователь ответит "Да" на сообщение с предложением добавить название данного кода в KOM массив, то..

if (MessageBox.Show(this, "У предмета `" + REZ[i, 0] + "` неизвестное название кода компетенции: `" +

REZ[i, 2] + "`. Добавить код компетенции в существующие?", "Предупреждение П\_008!",

MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)

{

//..добавление нового названия кода компетенции в массив KOM

KOM[lastOfKOM, 0] = REZ[i, 2];

//Увеличение индекса массива строк названий компетенций соответствующего предмета

lastOfKOM++;

//Вычисление адреса столбца для определения пересеченияс кодом компетенции соответствующего семестра предмета

REZ[i, 5] = Convert.ToString((Convert.ToInt32(REZ[i, 1]) - 1) \* lastOfKOM + 2 + (ki + 1));

//Переключение newCom в обратное состояние

newCom = false;

//Обнуление индекса рассматриваемого предмета i

i = 0;

//Возврат к начальной установке переменной - порядкового номера предмета

numOfSubj = 1;

//Возврат к начальной установке булевой переменной, где хранится результат совпадения рассматриваемого названия(i)

//с уже рассмотренными(m)

resOfCol = false;

//Продолжение цикла с обнуленными переменными (начать цикл заново)

continue;

}

}

//Если лист для записи - первый(с первого по четвёртый семестр), то..

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*86*

*Кнфу ХХХ.ХХХ.ПЗ*

if (Convert.ToInt32(REZ[i, 3]) == 1)

//Вычисление адреса столбца для определения пересеченияс кодом компетенции соответствующего семестра предмета

REZ[i, 5] = Convert.ToString((Convert.ToInt32(REZ[i, 1]) - 1) \* lastOfKOM + 2 + (ki + 1));

else

//Если лист для записи - второй(с пятого по восьмой семестр), то..

if (Convert.ToInt32(REZ[i, 3]) == 2)

//Вычисление адреса столбца для определения пересеченияс кодом компетенции соответствующего семестра предмета

REZ[i, 5] = Convert.ToString((Convert.ToInt32(REZ[i, 1]) - 5) \* lastOfKOM + 2 + (ki + 1));

//Значение resOfCol переключается на true, то есть название предмета совпало с рассмотренным

resOfCol = true;

//Завершение цикла

break;

}

//Увеличение проверочного индекса расммотренных названий предметов массива REZ

m++;

}

//Если результат совпадение - false, то есть название рассматриваемого предмета не рассматривалось ранее, то..

if (resOfCol == false)

{

//..запись параметров

//Выделение ячейки для записи в неё порядкового номера названия предмета

excelCells = (Excel.Range)excelWorksheet.Cells[4 + numOfSubj, 1];

//Запись значения порядкового номера премета

excelCells.Value2 = numOfSubj;

//Выделение ячейки для записи в неё названия предмета

excelCells = (Excel.Range)excelWorksheet.Cells[4 + numOfSubj, 2];

//Запись значения названия премета

excelCells.Value2 = REZ[i, 0];

//Запись адреса строки только что рассмотренного предмета

REZ[i, 4] = Convert.ToString(4 + numOfSubj);

//Увеличение порядкового номера предмета для дальнейших записей

numOfSubj++;

//Обнуление переменной - индекс и одновременно порядковый номер кода компетенции

ki = 0;

//Переключение newCom в true, в начальное состояние

newCom = true;

//Пока значение массива названий KOM по индексу ki существует, выполнять

while (KOM[ki, 0] != null)

{

//Если код компетенции рассматриваемого названия предмета совпал с кодом компетенции в массиве всех компетенций, то..

if (REZ[i, 2] == KOM[ki, 0])

{

//Переключение newCom в false, так как совпадение найдено

newCom = false;

//Закончить цикл вычисления порядкового номера кода компетенции

break;

}

//Увеличение порядкового номера

ki++;

}

//Если совпадения кодов компетенций не найдено, то..

if (newCom == true && REZ[i, 2] != "" && REZ[i, 2] != null)

{

//.. это новый код компетенции

//Если пользователь ответит "Да" на сообщение с предложением добавить название данного кода в KOM массив, то..

if (MessageBox.Show(this, "У предмета `" + REZ[i, 0] + "` неизвестное название кода компетенции: `" +

REZ[i, 2] + "`. Добавить код компетенции в существующие?", "Предупреждение П\_009!",

MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)

{

//..добавление нового названия кода компетенции в массив KOM

KOM[lastOfKOM, 0] = REZ[i, 2];

//Увеличение индекса массива строк названий компетенций соответствующего предмета

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*87*

*Кнфу ХХХ.ХХХ.ПЗ*

lastOfKOM++;

//Вычисление адреса столбца для определения пересеченияс кодом компетенции соответствующего семестра предмета

REZ[i, 5] = Convert.ToString((Convert.ToInt32(REZ[i, 1]) - 1) \* lastOfKOM + 2 + (ki + 1));

//Переключение newCom в обратное состояние

newCom = false;

//Обнуление индекса рассматриваемого предмета i

i = 0;

//Продолжение цикла с обнулением индекса рассматриваемого предмета i (начать цикл заново)

continue;

}

}

//Если лист для записи - первый(с первого по четвёртый семестр), то..

if (Convert.ToInt32(REZ[i, 3]) == 1)

//Вычисление адреса столбца для определения пересеченияс кодом компетенции соответствующего семестра предмета

REZ[i, 5] = Convert.ToString((Convert.ToInt32(REZ[i, 1]) - 1) \* lastOfKOM + 2 + (ki + 1));

else

//Если лист для записи - второй(с пятого по восьмой семестр), то..

if (Convert.ToInt32(REZ[i, 3]) == 2)

//Вычисление адреса столбца для определения пересеченияс кодом компетенции соответствующего семестра предмета

REZ[i, 5] = Convert.ToString((Convert.ToInt32(REZ[i, 1]) - 5) \* lastOfKOM + 2 + (ki + 1));

}

//Значение resOfCol переключается на false, то есть на начальное

resOfCol = false;

//Увеличение проверочного индекса расмматриваемого названия предмета массива REZ

i++;

}

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Обработка первого листа

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Обработка первого листа...";

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

//Запись заголовка "Программа формирования компетенций"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись заголовка 'Программа формирования компетенций'...";

//Выделение группы ячеек через задание угловых ячеек диапазона(выделение ячейки C5 первого и второго листа)

excelCells = excelWorksheet.get\_Range("C5", Type.Missing);

//Явное (как мышью) выделение заданного выделения(явное выделение ячейки C5 первого листа)

excelCells.Select();

//Применение свойства заморозить ячейку(замораживание ячейки C5 первого листа)

excelApp.Application.ActiveWindow.FreezePanes = true;

//Выделение группы ячеек через задание угловых ячеек диапазона(выделение ячеек A1 - до ячейки, требуемой, чтобы в строчку вместилось 4

//суммы всех существующих компетенций(для четырёх семестров) + 1 ячейка для подсчёта сумм первого листа

excelCells = excelWorksheet.Range[excelWorksheet.Cells[1, 1], excelWorksheet.Cells[1, 2 + 4 \* lastOfKOM + 1]];

//Объединение выделенного диапазона

excelCells.Merge(Type.Missing);

//Запись значения объединённой ячейки

excelCells.Value2 = "Программа формирования компетенций";

//Тип надписи - жирный

excelCells.Font.Bold = true;

//Выравнивание надписи по горизонтали - по центру ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*88*

*Кнфу ХХХ.ХХХ.ПЗ*

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Запись заголовка "Семестры (годы) обучения"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись заголовка 'Семестры (годы) обучения'...";

//Выделение группы ячеек через задание угловых ячеек диапазона(с ячейки С2 до ячейки, требуемой, чтобы в строчку вместилось 4 суммы всех

//существующих компетенций(для четырёх семестров))

excelCells = excelWorksheet.Range[excelWorksheet.Cells[2, 3], excelWorksheet.Cells[2, 2 + 4 \* lastOfKOM]];

//Объединение выделенного диапазона

excelCells.Merge(Type.Missing);

//Запись значения объединённой ячейки

excelCells.Value2 = "Семестры (годы) обучения";

//Выравнивание надписи по горизонтали - по центру ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Запись заголовков семестров ("1 семестр", "2 семестр" и т.д.)

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись заголовков семестров ('1 семестр', '2 семестр' и т.д.)...";

//Объявлние переменной порядкового номера семестра

int s;

//Объявление переменной, куда будет записываться столбец стартовой ячейки для объединения ячеек для записи семестра

int stCellForSem = 3;

//Увеличение прогрес бара

progressBar1.Value++;

//Цикл записи семестров

//Пока семестр от 1 до 4

for (s = 1; s <= 4; s++)

{

//Выделение группы ячеек через задание угловых ячеек диапазона(с ячейки С3 или с конца предыдущего диапазона до ячейки, требуемой,

//чтобы в строчку вместилась сумма всех существующих компетенций)

excelCells = excelWorksheet.Range[excelWorksheet.Cells[3, stCellForSem], excelWorksheet.Cells[3, 2 + s \* lastOfKOM]];

//Изменение стартовой ячейки для следующих объединений(прибавление суммы кодов компетенций)

stCellForSem += lastOfKOM;

//Объединение выделенного диапазона

excelCells.Merge(Type.Missing);

//Запись значения объединённой ячейки

excelCells.Value2 = Convert.ToString(s) + " семестр";

//Выравнивание надписи по горизонтали - по центру ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

}

//Увеличение прогрес бара

progressBar1.Value++;

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*89*

*Кнфу ХХХ.ХХХ.ПЗ*

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

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

//Запись заголовка "Сумма компетенций по дисциплине за 1-4 семестры"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись заголовка 'Сумма компетенций по дисциплине за 1-4 семестры'...";

//Выделение группы ячеек через задание угловых ячеек диапазона(выделение ячейки первого листа)

excelCells = excelWorksheet.Range[excelWorksheet.Cells[2, 2 + 4 \* lastOfKOM + 1], excelWorksheet.Cells[4, 2 + 4 \* lastOfKOM + 1]];

//Объединение выделенного диапазона

excelCells.Merge(Type.Missing);

//Запись значения объединённой ячейки

excelCells.Value2 = "Сумма компетенций по дисциплине за 1-4 семестры";

//Ширина ячейки = 14

excelCells.ColumnWidth = 14;

//Включение параматра "Переносить по словам"

excelCells.WrapText = true;

//Выравнивание надписи по горизонтали - по центру ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Запись кодов компетенций в соответствующие ячейки

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись кодов компетенций в соответствующие ячейки...";

//Объявление переменной, куда будет записываться столбец стартовой ячейки для объединения ячеек для записи семестра

int stCellForCom;

//Увеличение прогрес бара

progressBar1.Value++;

//Цикл записи кодов компетенций

//Пока семестр от 1 до 4

for (s = 0; s <= 3; s++)

{

//Первоначальное положение стартового столбца

stCellForCom = 3;

//Пока стартовый столбец не больше элементов массива KOM(+3 - так как начальное значение 3)

while (stCellForCom < lastOfKOM + 3)

{

//Выделение очередной ячейки для записи кода компетенции

excelCells = (Excel.Range)excelWorksheet.Cells[4, s \* lastOfKOM + stCellForCom];

//Ориентация надписи - вертикальная

excelCells.Orientation = Excel.XlOrientation.xlVertical;

//Выравнивание надписи по горизонтали - по центру ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

//Выравнивание надписи по вертикали - вверху ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignTop;

//Ширина ячейки = 2.14 - подгонка под 20 пикселов - высота строки

excelCells.ColumnWidth = 2.14;

//Запись значения объединённой ячейки

excelCells.Value2 = KOM[stCellForCom - 3, 0];

//Увеличение на один стартового столбца

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*90*

*Кнфу ХХХ.ХХХ.ПЗ*

stCellForCom++;

}

}

//Увеличение прогрес бара

progressBar1.Value++;

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

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

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

//Заполнение таблицы "Программа формирования компетенций"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Заполнение таблицы 'Программа формирования компетенций'...";

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

//Запись формул для сумм столбца "Сумма компетенций по дисциплине за 1-4 семестры"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись формул для сумм столбца 'Сумма компетенций по дисциплине за 1-4 семестры'...";

//Объявление переменной - отсчёт порядкового номера предмета

int f;

//Увеличение прогрес бара

progressBar1.Value++;

//Пока порядковый номер семестра не перевалил за их количество

for (f = 1; f < numOfSubj; f++)

{

//Выделение очередной ячейки для записи формулы

excelCells = (Excel.Range)excelWorksheet.Cells[4 + f, 2 + 4 \* lastOfKOM + 1];

//Запись значения формулы ячейки

excelCells.Value2 = "=СУММ(RC[-" + Convert.ToString(4 \* lastOfKOM) + "]:RC[-1])";

//Выравнивание надписи по горизонтали - по правому краю ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignRight;

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

}

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Запись строки "Сумма по компетенции в семестре" и соответствующих формул для сумм

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись строки 'Сумма по компетенции в семестре' и соответствующих формул для сумм ...";

//Выделение ячейки для записи строки "Сумма по компетенции в семестре"

excelCells = (Excel.Range)excelWorksheet.Cells[4 + numOfSubj, 2];

//Запись строки в выделенную область

excelCells.Value2 = "Сумма по компетенции в семестре";

//Увеличение прогрес бара

progressBar1.Value++;

//Пока номер столбца ячейки для записи формулы не перевалил за их количество

for (f = 3; f <= 2 + 4 \* lastOfKOM; f++)

{

//Выделение очередной ячейки для записи формулы

excelCells = (Excel.Range)excelWorksheet.Cells[4 + numOfSubj, f];

//Запись значения формулы ячейки

excelCells.Value2 = "=СУММ(R[-" + (numOfSubj - 1) + "]C:R[-1]C)";

//Выравнивание надписи по горизонтали - по правому краю ячейки

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*91*

*Кнфу ХХХ.ХХХ.ПЗ*

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignRight;

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

}

//Увеличение прогрес бара

progressBar1.Value++;

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

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

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

//Обработка второго листа

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Обработка второго листа...";

//Копирование первого листа на новый лист

excelWorksheet.Copy(Type.Missing, excelWorksheet);

//Переопределение ссылок. Теперь ссылки на второй лист

excelWorksheet = (Excel.Worksheet)excelSheets.get\_Item(2);

//Смена названия второго листа с "Лист 1 (2)" на "Лист 2"

excelWorksheet.Name = "Лист 2";

//Увеличение прогрес бара

progressBar1.Value++;

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

//Переопределение объединения ячеек для записи "Программа формирования компетенций"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Переопределение объединения ячеек для записи 'Программа формирования компетенций'...";

//Выделение группы ячеек через задание угловых ячеек диапазона(выделение ячеек A1 - до ячейки, требуемой, чтобы в строчку вместилось

//4 суммы всех существующих компетенций(для четырёх семестров) + 3 ячейки для подсчёта сумм второго листа

excelCells = excelWorksheet.Range[excelWorksheet.Cells[1, 1], excelWorksheet.Cells[1, 2 + 4 \* lastOfKOM + 3]];

//Объединение выделенного диапазона

excelCells.Merge(Type.Missing);

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Переименование ячеек с номером семестра

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Переименование ячеек с номером семестра...";

//Определение начальной позиции столбца изменяемой ячейки

stCellForSem = 3;

//Увеличение прогрес бара

progressBar1.Value++;

//Цикл записи семестров

//Пока семестр от 5 до 8

for (s = 5; s <= 8; s++)

{

//Выделение ячееки для записи номера семестра

excelCells = (Excel.Range)excelWorksheet.Cells[3, stCellForSem];

//Изменение стартовой ячейки для следующих объединений(прибавление суммы кодов компетенций)

stCellForSem += lastOfKOM;

//Запись значения ячейки

excelCells.Value2 = Convert.ToString(s) + " семестр";

}

//Увеличение прогрес бара

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*92*

*Кнфу ХХХ.ХХХ.ПЗ*

progressBar1.Value++;

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

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

//Переопределение ячеек-формул для записи "Сумма компетенций по дисциплине за 1-4 семестры"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Переопределение ячеек-формул для записи 'Сумма компетенций по дисциплине за 1-4 семестры'...";

//Пока порядковый номер семестра не перевалил за их количество

for (f = 1; f < numOfSubj; f++)

{

//Выделение очередной ячейки для записи формулы

excelCells = (Excel.Range)excelWorksheet.Cells[4 + f, 2 + 4 \* lastOfKOM + 1];

//Запись значения формулы ячейки

excelCells.Value2 = "=СУММ(Лист1!RC[-" + Convert.ToString(4 \* lastOfKOM) + "]:Лист1!RC[-1])";

}

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Запись заголовка "Сумма компетенций по дисциплине за 5-8 семестры"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись заголовка 'Сумма компетенций по дисциплине за 5-8 семестры'...";

//Выделение группы ячеек через задание угловых ячеек диапазона(выделение ячейки второго листа)

excelCells = excelWorksheet.Range[excelWorksheet.Cells[2, 2 + 4 \* lastOfKOM + 2], excelWorksheet.Cells[4, 2 + 4 \* lastOfKOM + 2]];

//Объединение выделенного диапазона

excelCells.Merge(Type.Missing);

//Запись значения объединённой ячейки

excelCells.Value2 = "Сумма компетенций по дисциплине за 5-8 семестры";

//Ширина ячейки = 14

excelCells.ColumnWidth = 14;

//Включение параматра "Переносить по словам"

excelCells.WrapText = true;

//Выравнивание надписи по горизонтали - по центру ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Запись формул для сумм столбца "Сумма компетенций по дисциплине за 5-8 семестры"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись формул для сумм столбца 'Сумма компетенций по дисциплине за 5-8 семестры'...";

//Пока порядковый номер семестра не перевалил за их количество

for (f = 1; f < numOfSubj; f++)

{

//Выделение очередной ячейки для записи формулы

excelCells = (Excel.Range)excelWorksheet.Cells[4 + f, 2 + 4 \* lastOfKOM + 2];

//Запись значения формулы ячейки

excelCells.Value2 = "=СУММ(RC[-" + Convert.ToString(4 \* lastOfKOM) + "]:RC[-2])";

//Выравнивание надписи по горизонтали - по правому краю ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignRight;

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*93*

*Кнфу ХХХ.ХХХ.ПЗ*

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

}

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Запись заголовка "Сумма компетенций по дисциплине суммарная"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись заголовка 'Сумма компетенций по дисциплине суммарная'...";

//Выделение группы ячеек через задание угловых ячеек диапазона(выделение ячейки второго листа)

excelCells = excelWorksheet.Range[excelWorksheet.Cells[2, 2 + 4 \* lastOfKOM + 3], excelWorksheet.Cells[4, 2 + 4 \* lastOfKOM + 3]];

//Объединение выделенного диапазона

excelCells.Merge(Type.Missing);

//Запись значения объединённой ячейки

excelCells.Value2 = "Сумма компетенций по дисциплине суммарная";

//Ширина ячейки = 14

excelCells.ColumnWidth = 14;

//Включение параматра "Переносить по словам"

excelCells.WrapText = true;

//Выравнивание надписи по горизонтали - по центру ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

//Увеличение прогрес бара

progressBar1.Value++;

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

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

//Запись формул для сумм столбца "Сумма компетенций по дисциплине суммарная"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Запись формул для сумм столбца 'Сумма компетенций по дисциплине суммарная'...";

//Пока порядковый номер семестра не перевалил за их количество

for (f = 1; f < numOfSubj; f++)

{

//Выделение очередной ячейки для записи формулы

excelCells = (Excel.Range)excelWorksheet.Cells[4 + f, 2 + 4 \* lastOfKOM + 3];

//Запись значения формулы ячейки

excelCells.Value2 = "=СУММ(RC[-2]:RC[-1])";

//Выравнивание надписи по горизонтали - по правому краю ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignRight;

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

}

//Увеличение прогрес бара

progressBar1.Value++;

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

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

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

//Расставление единичек на пересечении названия предмета и кода соответствующей компетенции

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Расставление единичек на пересечении названия предмета и кода соответствующей компетенции...";

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*94*

*Кнфу ХХХ.ХХХ.ПЗ*

//Для всех элементов массива REZ

for (i = 0; i < lastOfREZ; i++)

{

//Если адрес столбца предмета есть, то..

if (REZ[i, 5] != null && REZ[i, 5] != "")

{

//.. значит этот предмет обрабатывался

//Если номер семестра от 1 до 4, то..

if (Convert.ToInt32(REZ[i, 1]) >= 1 && Convert.ToInt32(REZ[i, 1]) <= 4)

{

//Переопределение ссылок. Теперь ссылки на первый лист

excelWorksheet = (Excel.Worksheet)excelSheets.get\_Item(1);

}

else

//Иначе, если номер семестра от 5 до 8, то..

if (Convert.ToInt32(REZ[i, 1]) >= 5 && Convert.ToInt32(REZ[i, 1]) <= 8)

{

//Переопределение ссылок. Теперь ссылки на второй лист

excelWorksheet = (Excel.Worksheet)excelSheets.get\_Item(2);

}

else

{

//Иначе вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, "У предмета `" + REZ[i, 0] + "` неизвестный номер семестра: [" + Convert.ToInt32(REZ[i, 1]) +

"]. Эта запись обработана не будет. Проверьте правильность номера семестра!",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

//Продолжить сканирование массива REZ

continue;

}

//Выделение очередной ячейки для записи единички

excelCells = (Excel.Range)excelWorksheet.Cells[Convert.ToInt32(REZ[i,4]), Convert.ToInt32(REZ[i,5])];

//Запись единички

excelCells.Value2 = "1";

//Выравнивание надписи по горизонтали - по центру ячейки

excelCells.HorizontalAlignment = Excel.XlHAlign.xlHAlignCenter;

//Выравнивание надписи по вертикали - по центру ячейки

excelCells.VerticalAlignment = Excel.XlVAlign.xlVAlignCenter;

}

}

//Увеличение прогрес бара

progressBar1.Value++;

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

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

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

//Сохранение файла "Программа формирования компетенций"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Сохранение файла 'Программа формирования компетенций'...";

//Строка, содержащая имя сохраняемого файла "Программы формирования компетенций"

string savePath = "";

//Объявление булевой переменной для записи решения, спрашивать ли пользователя вопрос: "Открыть сгенерированный файл?"

bool resOfAnsw = false;

//Объявление булевой переменной для записи состояния сохранённости файла

bool resOfSave = false;

//Пока файл не сохранён, выполнять..

while (resOfSave == false)

{

//..отображение диалога сохранения файла. Если нажата кнопка "Сохранить", то..

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

//.. запись пути и имени файла в соответствующую строку

//Запись в строку

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*95*

*Кнфу ХХХ.ХХХ.ПЗ*

savePath = saveFileDialog1.FileName;

//Так как путь к файлу известен, то спросить пользователя в будущем об открытии файла

resOfAnsw = true;

}

else

{

//Создание папки АФГОС\_резервное\_сохранение в папке "Мои документы"

Directory.CreateDirectory(Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments) + "\\АФГОС\_резервное\_сохранение");

savePath = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments) +

"\\АФГОС\_резервное\_сохранение\\РезСохр Программа формирования компетенций " +

DateTime.Now.ToString("yyyyг-MMмес-ddд-HHч-mmмин-ssсек-fffмилисек");

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, "Документ 'Программа формирования компетенций' будет сохранён в \"" +

savePath + "\" в формате Microsoft Oficce Excel!",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

//Попытаться сделать..

try

{

//..сохранение файла "Программа формирования компетенций"..

excelWorkbook.SaveAs(savePath, //Filename - Имя сохраняемого файла

Type.Missing, //FileFormat - Формат сохраняемого файла

Type.Missing, //Password - Пароль доступа к файлу до 15 символов

Type.Missing, //WriteResPassword - Пароль на доступ на запись

Type.Missing, //ReadOnlyRecommended - При true режим только для чтения

Type.Missing, //CreateBackup - Создать резервную копию файла при true

Excel.XlSaveAsAccessMode.xlNoChange, //AccessMode - Режим доступа к рабочей книге

Type.Missing, //ConflictResolution - Способ разрешения конфликтов

false, //AddToMru - При true сохраненный документ добавляется

//в список ранее открытых файлов

Type.Missing, //TextCodePage - Кодовая страница

Type.Missing, //TextVisualLayout - Направление размещения текста

Type.Missing //Local - Идентификатор ExcelApplication

);

}

//.. и если возникло исключение, то..

catch (COMException err)

{

//..если исключение вызвано нажатием "Отмена" или "Нет" на вопрос Excel`а заменить или нет сохраняемый файл

if (err.Message == "Исключение из HRESULT: 0x800A03EC")

{

//..отсечение запроса на сохранение для книги, так как пользователь нажал "Отмена" на вопрос сохранения файла

excelWorkbook.Saved = true;

//Закрытие файла "Программа формирования компетенций", то есть закрытие приложения

excelApp.Quit();

//Обнуление результирующего массива REZ и массива KOM для следующего запроса

toZeroREZandKOM();

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, "Документ 'Программа формирования компетенций' не сохранён по требованию пользователя!",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

//Вывод сообщения о завершении обработки документов

showMsgBreak();

//Завершение обработки

return;

}

else

{

//Если исключение вызвано тем, что нет доступа к резервно-сохраняемому файлу

if (err.Message.IndexOf("Нет доступа к 'РезСохр ") == 0)

{

//Объявление счётчика для копий файла

int copyNom = 0;

//Изменение имени резервно-сохраняемого файла

savePath = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments) +

"\\АФГОС\_резервное\_сохранение\\РезСохр Программа формирования компетенций " +

DateTime.Now.ToString("yyyyг-MMмес-ddд-HHч-mmмин-ssсек-fffмилисек") + " копия " + Convert.ToString(copyNom);

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*96*

*Кнфу ХХХ.ХХХ.ПЗ*

//Увеличение счётчика копий файла

copyNom++;

//Пока файл не сохранён, выполнять..

while (resOfSave == false)

{

//.. если файл существует по данному пути, то..

if (File.Exists(savePath + ".xls") || File.Exists(savePath + ".xlsx"))

{

savePath = savePath.Substring(0, savePath.LastIndexOf(' '));

//.. прибавить к названию файла номер счётчика копии файла

savePath += " " + Convert.ToString(copyNom);

//Увеличение счётчика копий файла

copyNom++;

}

else

//Сохранение удачно, запись информации об этом

resOfSave = true;

}

//..сохранение файла "Программа формирования компетенций"..

excelWorkbook.SaveAs(savePath, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Excel.XlSaveAsAccessMode.xlNoChange, Type.Missing, false, Type.Missing, Type.Missing,

Type.Missing);

//Продолжить цикл сохранения без выполнения последующего кода

continue;

}

else

{

//Если исключение вызвано тем, что нет доступа к сохраняемому файлу

if (err.Message.IndexOf("Нет доступа к '") == 0)

{

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, err.Message + "\n\nПо выбранному пути невозможно сохранение документа \"Программа формирования " +

"компетенций\"! Возможные решения проблемы:\n1) Если данный документ уже открыт, то закройте его и " +

"повторите процедуру выбора файла сохранения после нажатия \"ОК\".\n" +

"2)Выберите другой путь сохранения документа после нажатия \"ОК\"!",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

//Продолжить цикл сохранения без выполнения последующего кода

continue;

}

//Иначе, если исключение неизвестно

else

{

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, "Неизвестное исключение:\n" + err.Message,

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

//Вывод сообщения о завершении обработки документов

showMsgBreak();

//Завершение обработки

return;

}

}

}

}

//Сохранение удачно, запись информации об этом

resOfSave = true;

}

//Закрытие файла "Программа формирования компетенций", то есть закрытие приложения

excelApp.Quit();

//Увеличение прогрес бара

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*97*

*Кнфу ХХХ.ХХХ.ПЗ*

progressBar1.Value++;

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

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, "Документ 'Программа формирования компетенций' составлен и сохранён! Далее будет составляться документы \"Рабочая " +

"программа дисциплины\".",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Information);

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

//Составление документов "Рабочая программа дисциплин"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Составление документов \"Рабочая программа дисциплин\"...";

//Создаем объект Word - равносильно запуску Word

wordApp = new Word.Application();

//Включение видимости открытоко документа

wordApp.Visible = false;

//Объявление строки для записи пути к папке, куда будут сохраняться рабочие программы дисциплин

string selectedFolderForWorkPrs;

//Запись отображаемой первоначально папки при выборе папки для сохранения - отображение выбранной папки паспортов компетенций

folderBrowserDialogForWorkPr.SelectedPath = folderBrowserDialog1.SelectedPath;

//Отображение диалога выбора папки для сохранения рабочих программ. Если нажата кнопка "ОК", то..

if (folderBrowserDialogForWorkPr.ShowDialog() == DialogResult.OK)

{

//.. запись пути папки для сохранения рабочих программ

//Запись в метку

selectedFolderForWorkPrs = folderBrowserDialogForWorkPr.SelectedPath;

//Обнуление счётчика для элементов массива nameOfSubj

m = 0;

//Объявление строки для записи полного имени сохраняемого файла

string savePathOfWorkPr;

//Увеличение прогрес бара

progressBar1.Value++;

//Пока существует название предмета, выполнять

while (nameOfSubj[m] != null && nameOfSubj[m] != "")

{

//Обнуление счётчика для элементов массива REZ

i = 0;

//Пока название предмета в массиве nameOfSubj существует, выполнять

while (REZ[i, 0] != null && REZ[i, 0] != "")

{

//Если название прдмета в массиве REZ совпал с названием предмета в массиве nameOfSubj - массив названий предметов паспортов компетенций

if (REZ[i, 0] == nameOfSubj[m])

{

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

//Заполнение титульной страницы

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Заполнение титульной страницы \"Раб\_прогр\_" + REZ[i, 0] + "\"...";

//Если приложение Word по каким-то причинам закрылось, то..

if (wordApp == null)

{

//Создаем объект Word - равносильно запуску Word

wordApp = new Word.Application();

//Включение видимости открытоко документа

wordApp.Visible = false;

}

//Начать формирование очередного документа

try

{

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*98*

*Кнфу ХХХ.ХХХ.ПЗ*

//Создание файла "Рабочая программа дисциплины"

wordDocOfWorkProgram = wordApp.Documents.Add(Type.Missing, //Template - имя шаблона, по которому создается

//новый документ. Если значение не указано, то

//используется шаблон Normal.dot.

Type.Missing, //NewTemplate - при true новый документ

//открывается как шаблон. Значение по умолчанию -

//False.

Word.WdNewDocumentType.wdNewBlankDocument, //DocumentType - тип документа, может принимать

//одно из следующих значений констант типа

//word.WdNewDocumentType: wdNewBlankDocument -

//документ Word (по умолчанию);

//wdNewEmailMessage - электронное сообщение;

//wdNewWebPage - Web-страница;

//wdNewXMLDocument - XML документ.

false //Visible - видимость документа. При true

//(по умолчанию) документ отображается.

);

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(0, Type.Missing);

//Задание типа шрифта

wordDocRange.Font.Name = "Times New Roman";

//Задание размера шрифта

wordDocRange.Font.Size = 14;

//Вывод текста

wordDocRange.Text = "\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 10;

//Применение свойства "Полужирный"

wordDocRange.Font.Bold = 1;

//Применение свойства "По центру"

wordDocRange.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

//Вывод текста

wordDocRange.Text = "МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 12;

//Применение свойства "Полужирный"

wordDocRange.Font.Bold = 0;

//Вывод текста

wordDocRange.Text = "ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ\nМАРИЙСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание типа шрифта

wordDocRange.Font.Name = "Pragmatica";

//Применение свойства "По левому краю"

wordDocRange.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

//Вывод текста

wordDocRange.Text = "\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание типа шрифта

wordDocRange.Font.Name = "Times New Roman";

//Задание отступа первой строки

wordDocRange.ParagraphFormat.FirstLineIndent = wordDocOfWorkProgram.Application.CentimetersToPoints((float)11.5);

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*99*

*Кнфу ХХХ.ХХХ.ПЗ*

//Вывод текста

wordDocRange.Text = "УТВЕРЖДАЮ\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание отступа первой строки

wordDocRange.ParagraphFormat.FirstLineIndent = wordDocOfWorkProgram.Application.CentimetersToPoints((float)11);

//Вывод текста

wordDocRange.Text = "Декан\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 8;

//Задание отступа первой строки

wordDocRange.ParagraphFormat.FirstLineIndent = wordDocOfWorkProgram.Application.CentimetersToPoints((float)12.6);

//Вывод текста

wordDocRange.Text = "(назв. факультета)\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 12;

//Задание отступа первой строки

wordDocRange.ParagraphFormat.FirstLineIndent = wordDocOfWorkProgram.Application.CentimetersToPoints((float)13.75);

//Вывод текста

wordDocRange.Text = "\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 8;

//Применение свойства "По центру"

wordDocRange.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

//Задание отступа первой строки

wordDocRange.ParagraphFormat.FirstLineIndent = wordDocOfWorkProgram.Application.CentimetersToPoints((float)11);

//Вывод текста

wordDocRange.Text = "(подпись, Ф.И.О. декана)\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 12;

//Вывод текста

wordDocRange.Text = "« » " + Convert.ToString(DateTime.Now.Year) + "г.\n\n\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 16;

//Применение свойства "Полужирный"

wordDocRange.Font.Bold = 1;

//Задание отступа первой строки

wordDocRange.ParagraphFormat.FirstLineIndent = wordDocOfWorkProgram.Application.CentimetersToPoints((float)0);

//Вывод текста

wordDocRange.Text = "РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ\n\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*100*

*Кнфу ХХХ.ХХХ.ПЗ*

wordDocRange.Font.Size = 14;

//Вывод текста

wordDocRange.Text = REZ[i, 6] + " " + REZ[i, 0] + "\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 12;

//Применение свойства "Полужирный"

wordDocRange.Font.Bold = 0;

//Применение свойства "Курсив"

wordDocRange.Font.Italic = 1;

//Вывод текста

wordDocRange.Text = "(указывается шифр и наименование дисциплины по учебному плану)\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Применение свойства "Полужирный"

wordDocRange.Font.Bold = 1;

//Применение свойства "Курсив"

wordDocRange.Font.Italic = 0;

//Вывод текста

wordDocRange.Text = "\n\n\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(0, wordDocOfWorkProgram.Characters.Count);

//Задание свойства "Интервал перед"

wordDocRange.ParagraphFormat.SpaceBefore = 0;

//Задание свойства "Интервал после"

wordDocRange.ParagraphFormat.SpaceAfter = 0;

//Задание свойства "Интервал междустрочный" - одинарный

wordDocRange.ParagraphFormat.LineSpacingRule = Word.WdLineSpacing.wdLineSpaceSingle;

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

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

//Подготовка к заполнению Раздела 1

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Подготовка к заполнению Раздела 1 \"Раб\_прогр\_" + REZ[i, 0] + "\"...";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 14;

//Применение свойства "Полужирный"

wordDocRange.Font.Bold = 0;

//Применение свойства "По левому краю"

wordDocRange.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

//Вывод текста

wordDocRange.Text = "Раздел 1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 12;

//Вывод текста

wordDocRange.Text = "\n\n\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*101*

*Кнфу ХХХ.ХХХ.ПЗ*

//Применение свойства "Полужирный"

wordDocRange.Font.Bold = 1;

//Задание свойства "Интервал перед"

wordDocRange.ParagraphFormat.SpaceBefore = 3;

//Задание свойства "Интервал после"

wordDocRange.ParagraphFormat.SpaceAfter = 6;

//Вывод текста

wordDocRange.Text = "общекультурных:\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Добавляем таблицу

wordTable = wordDocOfWorkProgram.Tables.Add(wordDocRange, //Range место формирования таблицы

1, //NumRows Число строк

2, //NumColumns Число столбцов

Word.WdDefaultTableBehavior.wdWord9TableBehavior, //DefaultTableBehavior Определяет, изменяет ли Word

// автоматически размеры ячеек в

// таблицах, чтобы они соответствовали

// содержанию ячеек. Может быть

// одна из Word.WdDefaultTableBehavior

// констант: wdWord8TableBehavior (нет)

// или wdWord9TableBehavior (да).

// По умолчанию - wdWord8TableBehavior.

Word.WdAutoFitBehavior.wdAutoFitWindow //AutoFitBehavior Автоподбор ширины столбцов

// одна из следующих Word.WdAutoFitBehavior

// констант: wdAutoFitContent - по

// содержимому, wdAutoFitFixed -

// фиксированная или wdAutoFitWindow

// по ширине окна. Если DefaultTableBehavior

// установлен в wdWord8TableBehavior,

// этот параметр игнорируется.

);

//Выделение области документа - точнее таблицы

wordDocRange = wordDocOfWorkProgram.Range(wordTable.Cell(1, 1).Range.Start, wordTable.Cell(1, 2).Range.End);

//Выделение области документа как мышью

wordDocRange.Select();

//Применение выравнивания ячейки по центру вертикально

wordApp.Selection.Cells.VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

//Применение свойства "По центру"

wordApp.Selection.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

//Выделение области документа - точнее таблицы

wordDocRange = wordTable.Cell(1, 1).Range;

//Применение свойства ячейки переносить по словам

wordTable.Cell(1, 1).WordWrap = true;

//Задание ширины ячейки

wordTable.Cell(1, 1).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)2.75);

//Задание высоты ячейки

wordTable.Cell(1, 1).Height = wordDocOfWorkProgram.Application.CentimetersToPoints((float)0.6);

//Вывод текста

wordDocRange.Text = "Код компетенции";

//Выделение области документа - точнее таблицы

wordDocRange = wordTable.Cell(1, 2).Range;

//Применение свойства ячейки переносить по словам

wordTable.Cell(1, 2).WordWrap = true;

//Задание ширины ячейки

wordTable.Cell(1, 2).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)14.03);

//Вывод текста

wordDocRange.Text = "Название компетенции";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*102*

*Кнфу ХХХ.ХХХ.ПЗ*

//Задание свойства "Интервал перед"

wordDocRange.ParagraphFormat.SpaceBefore = 3;

//Задание свойства "Интервал после"

wordDocRange.ParagraphFormat.SpaceAfter = 0;

//Вывод текста

wordDocRange.Text = "\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание свойства "Интервал перед"

wordDocRange.ParagraphFormat.SpaceBefore = 6;

//Задание свойства "Интервал после"

wordDocRange.ParagraphFormat.SpaceAfter = 6;

//Вывод текста

wordDocRange.Text = "профессиональных:\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Добавляем таблицу

wordTable = wordDocOfWorkProgram.Tables.Add(wordDocRange, 1, 2, Word.WdDefaultTableBehavior.wdWord9TableBehavior, Word.WdAutoFitBehavior.wdAutoFitWindow);

//Выделение области документа - точнее таблицы

wordDocRange = wordDocOfWorkProgram.Range(wordTable.Cell(1, 1).Range.Start, wordTable.Cell(1, 2).Range.End);

//Выделение области документа как мышью

wordDocRange.Select();

//Применение выравнивания ячейки по центру вертикально

wordApp.Selection.Cells.VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

//Применение свойства "По центру"

wordApp.Selection.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

//Выделение области документа - точнее таблицы

wordDocRange = wordTable.Cell(1, 1).Range;

//Применение свойства ячейки переносить по словам

wordTable.Cell(1, 1).WordWrap = true;

//Задание ширины ячейки

wordTable.Cell(1, 1).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)2.75);

//Задание высоты ячейки

wordTable.Cell(1, 1).Height = wordDocOfWorkProgram.Application.CentimetersToPoints((float)0.6);

//Вывод текста

wordDocRange.Text = "Код компетенции";

//Выделение области документа - точнее таблицы

wordDocRange = wordTable.Cell(1, 2).Range;

//Применение свойства "Полужирный"

wordDocRange.Font.Bold = 1;

//Применение свойства ячейки переносить по словам

wordTable.Cell(1, 2).WordWrap = true;

//Задание ширины ячейки

wordTable.Cell(1, 2).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)14.03);

//Вывод текста

wordDocRange.Text = "Название компетенции";

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

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

//Подготовка к заполнению Раздела 2

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Подготовка к заполнению Раздела 2 \"Раб\_прогр\_" + REZ[i, 0] + "\"...";

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*103*

*Кнфу ХХХ.ХХХ.ПЗ*

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание свойства "Интервал перед"

wordDocRange.ParagraphFormat.SpaceBefore = 6;

//Задание свойства "Интервал после"

wordDocRange.ParagraphFormat.SpaceAfter = 0;

//Вывод текста

wordDocRange.Text = "\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 14;

//Отмена свойства "Полужирный"

wordDocRange.Font.Bold = 0;

//Задание свойства "Интервал перед"

wordDocRange.ParagraphFormat.SpaceBefore = 0;

//Задание свойства "Интервал после"

wordDocRange.ParagraphFormat.SpaceAfter = 0;

//Вывод текста

wordDocRange.Text = "Раздел 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП.\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 12;

//Вывод текста

wordDocRange.Text = "\n\n\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Применение свойства "Полужирный"

wordDocRange.Font.Bold = 1;

//Задание свойства "Интервал перед"

wordDocRange.ParagraphFormat.SpaceBefore = 2;

//Задание свойства "Интервал после"

wordDocRange.ParagraphFormat.SpaceAfter = 0;

//Вывод текста

wordDocRange.Text = "\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Добавляем таблицу для раздела 2

wordTable = wordDocOfWorkProgram.Tables.Add(wordDocRange, 1, 4, Word.WdDefaultTableBehavior.wdWord9TableBehavior, Word.WdAutoFitBehavior.wdAutoFitWindow);

//Выделение области документа - точнее таблицы

wordDocRange = wordDocOfWorkProgram.Range(wordTable.Cell(1, 1).Range.Start, wordTable.Cell(1, 4).Range.End);

//Выделение области документа как мышью

wordDocRange.Select();

//Применение выравнивания ячейки по центру вертикально

wordApp.Selection.Cells.VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

//Применение свойства "По центру"

wordApp.Selection.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

//Выделение области документа - точнее таблицы

wordDocRange = wordTable.Cell(1, 1).Range;

//Применение свойства ячейки переносить по словам

wordTable.Cell(1, 1).WordWrap = true;

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*104*

*Кнфу ХХХ.ХХХ.ПЗ*

//Задание ширины ячейки

wordTable.Cell(1, 1).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)1.71);

//Задание высоты ячейки

wordTable.Cell(1, 1).Height = wordDocOfWorkProgram.Application.CentimetersToPoints((float)0.6);

//Вывод текста

wordDocRange.Text = "№ п/п";

//Выделение области документа - точнее таблицы

wordDocRange = wordTable.Cell(1, 2).Range;

//Применение свойства ячейки переносить по словам

wordTable.Cell(1, 2).WordWrap = true;

//Задание ширины ячейки

wordTable.Cell(1, 2).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)5.8);

//Вывод текста

wordDocRange.Text = "Название компетенции";

//Выделение области документа - точнее таблицы

wordDocRange = wordTable.Cell(1, 3).Range;

//Применение свойства ячейки переносить по словам

wordTable.Cell(1, 3).WordWrap = true;

//Задание ширины ячейки

wordTable.Cell(1, 3).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)6.5);

//Вывод текста

wordDocRange.Text = "Предшествующие дисциплины";

//Выделение области документа - точнее таблицы

wordDocRange = wordTable.Cell(1, 4).Range;

//Применение свойства ячейки переносить по словам

wordTable.Cell(1, 4).WordWrap = true;

//Задание ширины ячейки

wordTable.Cell(1, 4).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)3);

//Вывод текста

wordDocRange.Text = "Последующие дисциплины";

//Добавление строки вниз таблицы

wordTable.Rows.Add();

//Добавление строки вниз таблицы

wordTable.Rows.Add();

//Добавление строки вниз таблицы

wordTable.Rows.Add();

//Добавление строки вниз таблицы

wordTable.Rows.Add();

//Выделение области документа - точнее таблицы

wordDocRange = wordDocOfWorkProgram.Range(wordTable.Cell(2, 1).Range.Start, wordTable.Cell(2, 4).Range.End);

//Выделение области документа как мышью

wordDocRange.Select();

//Применение объединения ячеек

wordApp.Selection.Cells.Merge();

//Выделение области документа - точнее таблицы

wordDocRange = wordTable.Cell(2, 1).Range;

//Отмена свойства "Полужирный"

wordDocRange.Font.Bold = 0;

//Применение свойства "Курсив"

wordDocRange.Font.Italic = 1;

//Применение свойства "По левому краю"

wordApp.Selection.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*105*

*Кнфу ХХХ.ХХХ.ПЗ*

//Вывод текста

wordDocRange.Text = "Общекультурные компетенции";

//Выделение области документа - точнее таблицы

wordDocRange = wordDocOfWorkProgram.Range(wordTable.Cell(4, 1).Range.Start, wordTable.Cell(4, 4).Range.End);

//Выделение области документа как мышью

wordDocRange.Select();

//Применение объединения ячеек

wordApp.Selection.Cells.Merge();

//Выделение области документа - точнее таблицы

wordDocRange = wordTable.Cell(4, 1).Range;

//Отмена свойства "Полужирный"

wordDocRange.Font.Bold = 0;

//Применение свойства "Курсив"

wordDocRange.Font.Italic = 1;

//Применение свойства "По левому краю"

wordApp.Selection.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

//Вывод текста

wordDocRange.Text = "Профессиональные компетенции";

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

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

//Заполнение Раздела 1 и Раздела 2

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Заполнение Раздела 1 и Раздела 2 \"Раб\_прогр\_" + REZ[i, 0] + "\"...";

//Объявление строки для записи рассматриваемого названия предмета

string nameOfSubjForWorkPr = REZ[i, 0];

//Объявление строки для записи номера семестра рассматриваемого названия предмета

string nameOfSemForWorkPr = REZ[i, 1];

//Объявление счётччика индекса массива REZ

int ir = 0;

//Объявление счётчика для массива KOM

int inc = 0;

//Определение, что wordTable1 - это таблица второго раздела

wordTable1 = wordDocOfWorkProgram.Tables[3];

//Объявление переменной для записи счётчика строк wordTable1

int wordTable1Row = 0;

//Объявление счётчика строки wordTable1 общекультурной компетенции

int wordTable1RowO = 3;

//Объявление счётчика строки wordTable1 профессиональной компетенции

int wordTable1RowP = 5;

//Пока существует название предмета REZ[ir, 0]

while (REZ[ir, 0] != null && REZ[ir, 0] != "")

{

//Если название предмета REZ[ir, 0] совпало с рассматриваемым, то..

if (REZ[ir, 0] == REZ[i, 0])

{

//..если номер семестра REZ[ir, 1] совпал с рассматриваемым, то..

if (REZ[ir, 1] == REZ[i, 1])

{

//..если код компетенции REZ[ir, 2] начинается с "ОК", то..

if (REZ[ir, 2].IndexOf("ОК") == 0)

{

//..определение, что работа с таблицей общекультурных компетенций 1 раздела

wordTable = wordDocOfWorkProgram.Tables[1];

//Определение, что работа с общекультурными компетенциями

wordTable1Row = wordTable1RowO;

//Инкремент счётчика строки wordTable1 общекультурной компетенции

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*106*

*Кнфу ХХХ.ХХХ.ПЗ*

wordTable1RowO++;

//Инкремент счётчика строки wordTable1 профессиональной компетенции

wordTable1RowP++;

//Добавление строки вниз таблицы

wordTable1.Rows.Add(wordTable1.Rows[wordTable1RowO - 1]);

}

else

//..если код компетенции REZ[ir, 2] начинается с "ДПК" или с "ПК", то..

if (REZ[ir, 2].IndexOf("ДПК") == 0 || REZ[ir, 2].IndexOf("ПК") == 0)

{

//..определение, что работа с таблицей проффесиональных компетенций 1 раздела

wordTable = wordDocOfWorkProgram.Tables[2];

//Определение, что работа с профессиональными компетенциями

wordTable1Row = wordTable1RowP;

//Инкремент счётчика строки wordTable1 профессиональной компетенции

wordTable1RowP++;

//Добавление строки вниз таблицы

wordTable1.Rows.Add();

}

else

{

//Увеличение индекса, переход к следующему элементу сравниваемого массива

ir++;

//Продолжить выполнение цикла без последующего кода

continue;

}

//Отностится к Разделу 1:

//Добавление строки вниз таблицы

wordTable.Rows.Add();

//Применение свойства "По ширине"

wordTable.Cell(wordTable.Rows.Count, 2).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphJustify;

//Отмена свойства "Полужирный"

wordTable.Cell(wordTable.Rows.Count, 1).Range.Font.Bold = 0;

//Печать кода компетенции в соответствующий столбец

wordTable.Cell(wordTable.Rows.Count, 1).Range.Text = REZ[ir, 2];

//Обнуление счётчика для массива KOM

inc = 0;

//Пока код компетенции в массиве KOM существует

while (KOM[inc, 0] != null && KOM[inc, 0] != "")

{

//Если код компетенции REZ[ir, 2] совпадает с кодом компетенции KOM[inc, 0], то..

if (REZ[ir, 2] == KOM[inc, 0])

{

//..отмена свойства "Полужирный"

wordTable.Cell(wordTable.Rows.Count, 2).Range.Font.Bold = 0;

//Печать полного названия кода компетенции в соответствующий столбец

wordTable.Cell(wordTable.Rows.Count, 2).Range.Text = KOM[inc, 1];

//Прервать выполнение цикла

break;

}

//Инкремент счётчика для массива KOM

inc++;

}

//Отностится к Разделу 2:

//Применение свойства "По ширине"

wordTable1.Cell(wordTable1Row, 2).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphJustify;

//Применение свойства "По ширине"

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*107*

*Кнфу ХХХ.ХХХ.ПЗ*

wordTable1.Cell(wordTable1Row, 3).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphJustify;

//Применение свойства "По левому краю"

wordTable1.Cell(wordTable1Row, 4).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

//Отмена свойства "Полужирный"

wordTable1.Cell(wordTable1Row, 1).Range.Font.Bold = 0;

//Отмена свойства "Полужирный"

wordTable1.Cell(wordTable1Row, 2).Range.Font.Bold = 0;

//Отмена свойства "Полужирный"

wordTable1.Cell(wordTable1Row, 3).Range.Font.Bold = 0;

//Отмена свойства "Полужирный"

wordTable1.Cell(wordTable1Row, 4).Range.Font.Bold = 0;

//Печать кода компетенции в соответствующий столбец

wordTable1.Cell(wordTable1Row, 1).Range.Text = REZ[ir, 2];

//Обнуление счётчика для массива KOM

inc = 0;

//Пока код компетенции в массиве KOM существует

while (KOM[inc, 0] != null && KOM[inc, 0] != "")

{

//Если код компетенции REZ[ir, 2] совпадает с кодом компетенции KOM[inc, 0], то..

if (REZ[ir, 2] == KOM[inc, 0])

{

//..печать полного названия кода компетенции в соответствующий столбец

wordTable1.Cell(wordTable1Row, 2).Range.Text = KOM[inc, 1];

//Прервать выполнение цикла

break;

}

//Инкремент счётчика для массива KOM

inc++;

}

}

}

//Увеличение индекса, переход к следующему элементу сравниваемого массива

ir++;

}

//Удаление пустой строки общекультурных компетенций раздела 2

wordTable1.Rows[wordTable1RowO].Delete();

//Удаление пустой строки профессиональных компетенций раздела 2

wordTable1.Rows[wordTable1RowP - 1].Delete();

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

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

//Подготовка к заполнению Раздела 3

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Подготовка к заполнению Раздела 3 \"Раб\_прогр\_" + REZ[i, 0] + "\"...";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 14;

//Отмена свойства "Полужирный"

wordDocRange.Font.Bold = 0;

//Задание свойства "Интервал перед"

wordDocRange.ParagraphFormat.SpaceBefore = 0;

//Задание свойства "Интервал после"

wordDocRange.ParagraphFormat.SpaceAfter = 0;

//Применение свойства "По центру"

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*108*

*Кнфу ХХХ.ХХХ.ПЗ*

wordDocRange.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

//Вывод текста

wordDocRange.Text = "\nРаздел 3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Задание размера шрифта

wordDocRange.Font.Size = 12;

//Вывод текста

wordDocRange.Text = "(матрица распределения компетенций по разделам и темам дисциплины)\n\n";

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Добавляем таблицу для раздела 3

wordTable = wordDocOfWorkProgram.Tables.Add(wordDocRange, 1, 2 + wordDocOfWorkProgram.Tables[3].Rows.Count - 3 + 1,

Word.WdDefaultTableBehavior.wdWord9TableBehavior, Word.WdAutoFitBehavior.wdAutoFitWindow);

//Задание ширины ячейки

wordTable.Cell(1, 1).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)5.69);

//Задание ширины ячейки

wordTable.Cell(1, 2).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)2.56);

//Цикл задания ширины ячейки для кодов компетенций

for (int r3 = 1; r3 <= wordDocOfWorkProgram.Tables[1].Rows.Count - 1; r3++)

{

//Задание ширины ячейки

wordTable.Cell(1, r3 + 2).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)0.6);

}

//Цикл задания ширины ячейки для кодов компетенций

for (int r3 = 1; r3 <= wordDocOfWorkProgram.Tables[2].Rows.Count - 1; r3++)

{

//Задание ширины ячейки

wordTable.Cell(1, r3 + 2 + wordDocOfWorkProgram.Tables[1].Rows.Count - 1).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)0.6);

}

//Задание ширины ячейки

wordTable.Cell(1, 2 + wordDocOfWorkProgram.Tables[3].Rows.Count - 3 + 1).Width = wordDocOfWorkProgram.Application.CentimetersToPoints((float)3);

//Добавление строки вниз таблицы

wordTable.Rows.Add();

//Выделение области документа - точнее таблицы

wordDocRange = wordDocOfWorkProgram.Range(wordTable.Cell(1, 1).Range.Start, wordTable.Cell(2, 1).Range.End);

//Выделение области документа как мышью

wordDocRange.Select();

//Применение объединения ячеек

wordApp.Selection.Cells.Merge();

//Применение выравнивания ячейки по центру вертикально

wordApp.Selection.Cells.VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

//Применение свойства "По центру"

wordApp.Selection.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

//Применение свойства ячейки переносить по словам

wordTable.Cell(1, 1).WordWrap = true;

//Вывод текста

wordTable.Cell(1, 1).Range.Text = "ТЕМЫ,\nРАЗДЕЛЫ\nДИСЦИПЛИНЫ";

//Выделение области документа - точнее таблицы

wordDocRange = wordDocOfWorkProgram.Range(wordTable.Cell(1, 2).Range.Start, wordTable.Cell(2, 2).Range.End);

//Выделение области документа как мышью

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*109*

*Кнфу ХХХ.ХХХ.ПЗ*

wordDocRange.Select();

//Применение объединения ячеек

wordApp.Selection.Cells.Merge();

//Применение выравнивания ячейки по центру вертикально

wordApp.Selection.Cells.VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

//Применение свойства "По центру"

wordApp.Selection.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

//Применение свойства ячейки переносить по словам

wordTable.Cell(1, 2).WordWrap = true;

//Вывод текста

wordTable.Cell(1, 2).Range.Text = "КОЛИЧЕСТВО ЧАСОВ\n(АУД/СРС)";

//Применение выравнивания ячейки по центру вертикально

wordTable.Cell(2, 2 + wordDocOfWorkProgram.Tables[3].Rows.Count - 3 + 1).VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

//Применение свойства "По центру"

wordTable.Cell(2, 2 + wordDocOfWorkProgram.Tables[3].Rows.Count - 3 + 1).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

//Применение свойства ячейки переносить по словам

wordTable.Cell(2, 2 + wordDocOfWorkProgram.Tables[3].Rows.Count - 3 + 1).WordWrap = true;

//Вывод текста

wordTable.Cell(2, 2 + wordDocOfWorkProgram.Tables[3].Rows.Count - 3 + 1).Range.Text = "Σ\nОБЩЕЕ КОЛИЧЕСТВО КОМПЕТЕНЦИЙ";

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

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

//Заполнение Раздела 3

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Заполнение Раздела 3 \"Раб\_прогр\_" + REZ[i, 0] + "\"...";

//Заполнение общекультурных кодов компетенций

for (int r3 = 1; r3 <= wordDocOfWorkProgram.Tables[1].Rows.Count - 1; r3++)

{

//Изменение ориентации надписи на вертикальную

wordTable.Cell(2, r3 + 2).Range.Orientation = Word.WdTextOrientation.wdTextOrientationUpward;

//Вывод текста

wordTable.Cell(2, r3 + 2).Range.Text = wordDocOfWorkProgram.Tables[1].Cell(r3 + 1, 1).Range.Text;

}

//Заполнение профессиональных кодов компетенций

for (int r3 = 1; r3 <= wordDocOfWorkProgram.Tables[2].Rows.Count - 1; r3++)

{

//Изменение ориентации надписи на вертикальную

wordTable.Cell(2, r3 + 2 + wordDocOfWorkProgram.Tables[1].Rows.Count - 1).Range.Orientation = Word.WdTextOrientation.wdTextOrientationUpward;

//Вывод текста

wordTable.Cell(2, r3 + 2 + wordDocOfWorkProgram.Tables[1].Rows.Count - 1).Range.Text = wordDocOfWorkProgram.Tables[2].Cell(r3 + 1, 1).Range.Text;

}

//Выделение области документа - точнее таблицы

wordDocRange = wordDocOfWorkProgram.Range(wordTable.Cell(1, 3).Range.Start, wordTable.Cell(1, wordTable.Columns.Count).Range.End);

//Выделение области документа как мышью

wordDocRange.Select();

//Применение объединения ячеек

wordApp.Selection.Cells.Merge();

//Применение выравнивания ячейки по центру вертикально

wordApp.Selection.Cells.VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

//Применение свойства "По центру"

wordApp.Selection.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

//Применение свойства ячейки переносить по словам

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*110*

*Кнфу ХХХ.ХХХ.ПЗ*

wordTable.Cell(1, 3).WordWrap = true;

//Вывод текста

wordTable.Cell(1, 3).Range.Text = "КОМПЕТЕНЦИИ";

//Добавление строки вниз таблицы

wordTable.Rows.Add();

//Применение свойства "По ширине"

wordTable.Cell(3, 1).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphJustify;

//Применение выравнивания ячейки по центру вертикально

wordTable.Cell(3, 1).VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

//Вычисление применения свойства "По центру" и ещё

for (int r3 = 2; r3 <= wordDocOfWorkProgram.Tables[4].Columns.Count - 1; r3++)

{

//Применение выравнивания ячейки по центру вертикально

wordTable.Cell(3, r3).VerticalAlignment = Word.WdCellVerticalAlignment.wdCellAlignVerticalCenter;

//Изменение ориентации надписи на горизонтальную

wordTable.Cell(3, r3).Range.Orientation = Word.WdTextOrientation.wdTextOrientationHorizontal;

//Применение свойства "По центру"

wordTable.Cell(3, r3).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphCenter;

}

//Добавление строки вниз таблицы

wordTable.Rows.Add();

//Добавление строки вниз таблицы

wordTable.Rows.Add();

//Добавление строки вниз таблицы

wordTable.Rows.Add();

//Применение свойства "По правому краю"

wordTable.Cell(6, 1).Range.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphRight;

//Вывод текста

wordTable.Cell(6, 1).Range.Text = "ИТОГО";

//Задание ширины ячейки

wordTable.Cell(6, 1).Height = wordDocOfWorkProgram.Application.CentimetersToPoints((float)0.6);

//Вычисление применения свойства "Полужирный"

for (int r3 = 2; r3 <= wordDocOfWorkProgram.Tables[4].Columns.Count; r3++)

{

//Применение свойства "Полужирный"

wordTable.Cell(6, r3).Range.Font.Bold = 1;

}

//Выделение области документа

wordDocRange = wordDocOfWorkProgram.Range(wordDocOfWorkProgram.Characters.Count - 1, Type.Missing);

//Применение свойства "Полужирный"

wordDocRange.Font.Bold = 0;

//Задание размера шрифта

wordDocRange.Font.Size = 14;

//Применение свойства "По левому краю"

wordDocRange.ParagraphFormat.Alignment = Word.WdParagraphAlignment.wdAlignParagraphLeft;

//Задание свойства "Интервал перед"

wordDocRange.ParagraphFormat.SpaceBefore = 0;

//Задание свойства "Интервал после"

wordDocRange.ParagraphFormat.SpaceAfter = 0;

//Вывод текста

wordDocRange.Text = "\n";

//Выделение области документа - точнее таблицы

wordDocRange = wordDocOfWorkProgram.Range(0, 1);

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*111*

*Кнфу ХХХ.ХХХ.ПЗ*

//Выделение области документа как мышью

wordDocRange.Select();

//Рисование линии под надписью "Декан"

wordDocOfWorkProgram.Shapes.AddLine(310, 97, 465, 97, Type.Missing);

//Рисование линии над надписью "(подпись, Ф.И.О. декана)"

wordDocOfWorkProgram.Shapes.AddLine(310, 120, 465, 120, Type.Missing);

//Рисование линии для числа

wordDocOfWorkProgram.Shapes.AddLine(323, 142, 350, 142, Type.Missing);

//Рисование линии для месяца

wordDocOfWorkProgram.Shapes.AddLine(360, 142, 426, 142, Type.Missing);

//Рисование линии под надписью кода и названия дисциплины

wordDocOfWorkProgram.Shapes.AddLine(1, 224, 465, 224, Type.Missing);

}

//Если во время формирования документа возникло каккое-либо исключение, то..

catch

{

//..повторить цикл заново

i = 0;

continue;

}

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

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

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

//Сохранение файла "Рабочая программа дисциплины"

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

//Смена надписи label8 взависимости от состояния

label8.Text = "Сохранение файла \"Раб\_прогр\_" + REZ[i, 0] + "\"...";

//Составление полного имени сохраняемого файла рабочей программы

savePathOfWorkPr = selectedFolderForWorkPrs + "\\Раб\_прогр\_" + REZ[i, 0];

//Объявление счётчика для копий файла

int copyNom = 0;

//Увеличение счётчика копий файла

copyNom++;

//Сохранение пока не удачно, запись информации об этом

resOfSave = false;

//Объявление строки для записи старого имени

string oldName = savePathOfWorkPr;

//Пока файл не подготовлен к сохранению, выполнять..

while (resOfSave == false)

{

//.. если файл существует по данному пути, то..

if (File.Exists(savePathOfWorkPr + ".doc") || File.Exists(savePathOfWorkPr + ".docx"))

{

//Если последний символ названия существующего файла - цифра, то..

if (savePathOfWorkPr[savePathOfWorkPr.Length - 1] == '0' || savePathOfWorkPr[savePathOfWorkPr.Length - 1] == '1' ||

savePathOfWorkPr[savePathOfWorkPr.Length - 1] == '2' || savePathOfWorkPr[savePathOfWorkPr.Length - 1] == '3' ||

savePathOfWorkPr[savePathOfWorkPr.Length - 1] == '4' || savePathOfWorkPr[savePathOfWorkPr.Length - 1] == '5' ||

savePathOfWorkPr[savePathOfWorkPr.Length - 1] == '6' || savePathOfWorkPr[savePathOfWorkPr.Length - 1] == '7' ||

savePathOfWorkPr[savePathOfWorkPr.Length - 1] == '8' || savePathOfWorkPr[savePathOfWorkPr.Length - 1] == '9')

{

//..выделение имени сохраняемого файла без инкремента копии

savePathOfWorkPr = savePathOfWorkPr.Substring(0, savePathOfWorkPr.LastIndexOf(' '));

//.. прибавить к названию файла номер счётчика копии файла

savePathOfWorkPr += " " + Convert.ToString(copyNom);

//Увеличение счётчика копий файла

copyNom++;

}

else

{

//.. иначе прибавить к названию файла номер счётчика копии файла

savePathOfWorkPr += " копия " + Convert.ToString(copyNom);

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*112*

*Кнфу ХХХ.ХХХ.ПЗ*

//Увеличение счётчика копий файла

copyNom++;

}

//Объявление resAnswSave для записи ответа пользователя

DialogResult resAnswSave = DialogResult.No;

//Вывод окна, информирующее выполнение обрабтки документов

resAnswSave = MessageBox.Show(this, "Сохраняемый файл\n\n" + oldName + "\n\n уже существует!\n\n Заменить?",

"Предупреждение П\_011", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

//Если ответ на замену файла - Да

if (resAnswSave == DialogResult.Yes)

{

//Запись для сохранения файла старого имени

savePathOfWorkPr = oldName;

//Подготовка к сохранению удачна, запись информации об этом

resOfSave = true;

}

else

{

//Иначе вывод окна, информирующее выполнение обрабтки документов

resAnswSave = MessageBox.Show(this, "Файл будет сохранён под именем:\n\n" + savePathOfWorkPr + "!",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

else

{

//Подготовка к сохранению удачна, запись информации об этом

resOfSave = true;

}

}

//Попытаться сохранить файл..

try

{

//..сохранение составленного документа

((Word.\_Document)wordDocOfWorkProgram).SaveAs(savePathOfWorkPr, //FileName Имя файла

Type.Missing, //FileFormat Формат сохраняемого файла, одна из

// Word.WdSaveFormat констант:

// wdFormatDocument, wdFormatWebArchive,

// wdFormatUnicodeText,

// wdFormatTextLineBreaks, wdFormatRTF,

// wdFormatText, wdFormatTemplate,

// wdFormatHTML, wdFormatFilteredHTML,

// wdFormatEncodedText, wdFormatDOSText

// wdFormatDOSTextLineBreaks,

false, //LockComments При true блокируется содержимое

// поля Заметки, находящегося на

// вкладке Документ в диалоговом окне

// Свойства меню Файл.

Type.Missing, //Password Пароль доступа к документу при открытии

false, //AddToRecentFiles При true имя сохраняемого файла

// добавляется в список недавно

// открытых файлов в меню Файл.

Type.Missing, //WritePassword Пароль для внесения изменений в

// документ

false, //ReadOnlyRecommended При true - при открытии

// документа будет отображаться

// диалоговое окно с

// рекомендацией открывать

// документ только для чтения

false, //EmbedTrueTypeFonts При true - TrueType шрифты

// сохраняются вместе с

// документом

false, //SaveNativePictureFormat При true сохраняет

// только родную

// графику (Windows)

false, //SaveFormsData При true сохраняет только данные,

// введенные пользователем во вводные

// формы.

Type.Missing, //SaveAsAOCELetter Если в документе используется

// attached mailer (программа

// доставки электронной почты

// адресату), то true для того

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*113*

*Кнфу ХХХ.ХХХ.ПЗ*

// чтобы документ был сохранен.

Type.Missing, //Encoding Кодовая страница, или набор символов,

// для документов, сохраненных как

// кодируемые текстовые файлы. Значение по

// умолчанию - системная кодовая страница.

// Задается как

// Microsoft.Office.Core.MsoEncoding

// .msoEncodingUSASCII;

Type.Missing, //InsertLineBreaks Если документ сохраняется как

// текстовый файл, то при true

// разрешается вставка разрывов

// строк.

Type.Missing, //AllowSubstitutions Если документ сохраняется как

// текстовый файл, то при true,

// Word заменяет некоторые

// символы текстом. Например,

// символ авторского права как

// (c).

Type.Missing, //LineEnding Если документ сохраняется как

// текстовый файл, то одна из

// Word.WdLineEndingType констант

// (wdCRLF, wdLSPS,wdCROnly, wdLFCR,

// wdLFOnly), определяющих какие символы

// (перевод строки, возврат каретки)

// используются для отделения строк друг

// от друга.

Type.Missing //AddBiDiMarks При true, Word добавляет к файлу

// символы управления вывода, чтобы

// сохранить двунаправленное размещение

// текста в оригинале документа.

);

}

//.. и если возникло исключение, то..

catch (COMException err)

{

//..если исключение вызвано нажатием "Отмена" или "Нет" на вопрос Word`а заменить или нет сохраняемый файл

if (err.Message == "Исключение из HRESULT: 0x800A03EC")

{

//..отсечение запроса на сохранение для книги, так как пользователь нажал "Отмена" на вопрос сохранения файла

wordDocOfWorkProgram.Saved = true;

//Закрытие приложения

((Word.\_Application)wordApp).Quit(Word.WdSaveOptions.wdDoNotSaveChanges, Word.WdOriginalFormat.wdOriginalDocumentFormat,

Type.Missing);

//Обнуление результирующего массива REZ и массива KOM для следующего запроса

toZeroREZandKOM();

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, "Документ\n\n\"" + savePathOfWorkPr + "\"\n\nне сохранён по требованию пользователя!",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

{

//Если исключение вызвано тем, что нет доступа к резервно-сохраняемому файлу

if (err.Message.IndexOf("Нет доступа к") == 0)

{

//Определение строки для записи старого имени

oldName = savePathOfWorkPr;

//Объявление счётчика для копий файла

copyNom = 1;

//Изменение имени файла

savePathOfWorkPr += " " + DateTime.Now.ToString("yyyyг-MMмес-ddд-HHч-mmмин-ssсек-fffмилисек") + " копия " +

Convert.ToString(copyNom);

//Увеличение счётчика копий файла

copyNom++;

//Сохранение не удачно, запись информации об этом

resOfSave = false;

//Пока файл не сохранён, выполнять..

while (resOfSave == false)

{

//.. если файл существует по данному пути, то..

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*114*

*Кнфу ХХХ.ХХХ.ПЗ*

if (File.Exists(savePathOfWorkPr + ".doc") || File.Exists(savePathOfWorkPr + ".docx"))

{

//Выделение имени сохраняемого файла без инкремента копии

savePathOfWorkPr = savePathOfWorkPr.Substring(0, savePathOfWorkPr.LastIndexOf(' '));

//.. прибавить к названию файла номер счётчика копии файла

savePathOfWorkPr += " " + Convert.ToString(copyNom);

//Увеличение счётчика копий файла

copyNom++;

}

else

//Сохранение удачно, запись информации об этом

resOfSave = true;

}

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, err.Message + "\n\n Файл\n\n" + oldName + "\n\nбудет сохранён под именем:\n\n" + savePathOfWorkPr + "!",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

//Сохранение составленного документа

((Word.\_Document)wordDocOfWorkProgram).SaveAs(savePathOfWorkPr, Type.Missing, false, Type.Missing, false, Type.Missing, false,

false, false, false, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing);

}

else

{

//Если исключение вызвано тем, что файл используется другим процессом

if (err.Message.IndexOf("Приложению Word не удалось сохранить данный файл, так как он используется другим процессом") == 0)

{

//Определение строки для записи старого имени

oldName = savePathOfWorkPr;

//Объявление счётчика для копий файла

copyNom = 1;

//Изменение имени файла

savePathOfWorkPr += " " + DateTime.Now.ToString("yyyyг-MMмес-ddд-HHч-mmмин-ssсек-fffмилисек") + " копия " +

Convert.ToString(copyNom);

//Увеличение счётчика копий файла

copyNom++;

//Сохранение не удачно, запись информации об этом

resOfSave = false;

//Пока файл не сохранён, выполнять..

while (resOfSave == false)

{

//.. если файл существует по данному пути, то..

if (File.Exists(savePathOfWorkPr + ".doc") || File.Exists(savePathOfWorkPr + ".docx"))

{

//Выделение имени сохраняемого файла без инкремента копии

savePathOfWorkPr = savePathOfWorkPr.Substring(0, savePathOfWorkPr.LastIndexOf(' '));

//.. прибавить к названию файла номер счётчика копии файла

savePathOfWorkPr += " " + Convert.ToString(copyNom);

//Увеличение счётчика копий файла

copyNom++;

}

else

//Сохранение удачно, запись информации об этом

resOfSave = true;

}

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, err.Message + "\n\n Файл\n\n" + oldName + "\n\nбудет сохранён под именем:\n\n" + savePathOfWorkPr + "!",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

//Сохранение составленного документа

((Word.\_Document)wordDocOfWorkProgram).SaveAs(savePathOfWorkPr, Type.Missing, false, Type.Missing, false, Type.Missing, false,

false, false, false, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing);

}

else

{

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*115*

*Кнфу ХХХ.ХХХ.ПЗ*

//Иначе, если исключение неизвестно

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, "Неизвестное исключение:\n" + err.Message + "\n\n Файл\n\n" + savePathOfWorkPr + "\n\nне будет сохранён!",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

}

}

//Закрытие составленного документа

((Word.\_Document)wordDocOfWorkProgram).Close(Word.WdSaveOptions.wdDoNotSaveChanges, Type.Missing, Type.Missing);

//Закончить цикл

break;

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

}

//Инкремент счётчика для элементов массива REZ

i++;

}

//Инкремент счётчика для элементов массива nameOfSubj

m++;

}

//Увеличение прогрес бара

progressBar1.Value++;

//Закрытие приложения

((Word.\_Application)wordApp).Quit(Word.WdSaveOptions.wdSaveChanges, Word.WdOriginalFormat.wdOriginalDocumentFormat,

Type.Missing);

wordApp = null;

//Увеличение прогрес бара

progressBar1.Value++;

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

}

else

{

//Закрытие приложения

((Word.\_Application)wordApp).Quit(Word.WdSaveOptions.wdDoNotSaveChanges, Word.WdOriginalFormat.wdOriginalDocumentFormat,

Type.Missing);

wordApp = null;

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, "Документы \"Рабочая программа дисциплины\" не сохранёны по требованию пользователя!",

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Warning);

//Обнуление результирующего массива REZ и массива KOM для следующего запроса

toZeroREZandKOM();

//Вывод сообщения о завершении обработки документов

showMsgBreak();

//Завершение обработки

return;

}

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

//Увеличение прогрес бара домаксимума

progressBar1.Value = progressBar1.Maximum;

//Вывод окна, информирующее выполнение обрабтки документов

MessageBox.Show(this, "Документы 'Рабочая программа дисциплины' составлены и сохранёны!",

*Изм.*

*Лист*

*№ докум.*

*Подпись*

*Дата*

*Лист*

*116*

*Кнфу ХХХ.ХХХ.ПЗ*

"Выполнение...", MessageBoxButtons.OK, MessageBoxIcon.Information);

//Увеличение прогрес бара до минимума

progressBar1.Value = progressBar1.Minimum;

//Смена надписи label8 взависимости от состояния

label8.Text = "...ожидание...";

//Если решение о спрашивании открытия документа - положительное, то..

if (resOfAnsw == true)

{

//..если пользователь ответил да на вопрос "Открыть сгенерированный файл?", то..

if (MessageBox.Show(this, "Открыть сохранённый файл `" + savePath + "`?", "Выполнение...",

MessageBoxButtons.YesNo, MessageBoxIcon.Information) == DialogResult.Yes)

{

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

//Открытие итогового документа "Программа формирования компетенций"

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

//Создание объекта Excel.Application - равносильно запуску Microsoft Office Excel

excelApp = new Excel.Application();

//Включение видимости открытоко документа

excelApp.Visible = true;

//Присвоение значания excelApp.Workbooks - совокупности ссылок на открытые рабочие книги, объекту excelWorkbooks

excelWorkbooks = excelApp.Workbooks;

//Открытие сохранённой книги(файла) "Программа формирования компетенций"

excelWorkbook = excelApp.Workbooks.Open(savePath, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing, Type.Missing,

Type.Missing, Type.Missing, Type.Missing, Type.Missing);

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

}

}

//Обнуление результирующего массива REZ и массива KOM для следующего запроса

toZeroREZandKOM();

}

}

//Автоматическое заполнение путей к файлам. (Используется при отладке и оптимизации)

private void button5\_Click(object sender, EventArgs e)

{

label2.Text = @"C:\Users\Yagodar\Documents\Учёба\Выходная квалификационная работа\Документы на обработку ФГОС-3\Приложения\05\_Учебн план ИВТ\_2010.xls";

label4.Text = @"C:\Users\Yagodar\Documents\Учёба\Выходная квалификационная работа\Документы на обработку ФГОС-3\Приложения\01\_Компетентностная модель выпускника.doc";

label6.Text = @"C:\Users\Yagodar\Documents\Учёба\Выходная квалификационная работа\Документы на обработку ФГОС-3";

}

}

}