

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

**РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ
СИСТЕМЫ СОСТАВЛЕНИЯ МАКЕТОВ РАБОЧИХ ПРОГРАММ
ДИСЦИПЛИН ДЛЯ НИУ «БЕЛГУ»**

Выпускная квалификационная работа
обучающегося по направлению подготовки 09.03.03 Прикладная
информатика
очной формы обучения, группы 07001405
Кривчикова Владислава Сергеевича

Научный руководитель
старший преподаватель
Болгова Е.В.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

| | |
|---|----|
| ВВЕДЕНИЕ | 4 |
| 1 Аналитическая часть | 7 |
| 1.1 Техничко-экономическая характеристика предметной области | 7 |
| 1.1.1 Характеристика НИУ «Белгородский государственный университет» | 7 |
| 1.1.2 Краткая характеристика кафедры прикладной информатики и информационных технологий | 8 |
| 1.2 Постановка задачи | 9 |
| 1.2.1 Цель и назначение автоматизированного варианта решения задачи | 9 |
| 1.2.2 Общая характеристика организации решения задачи на ЭВМ | 9 |
| 1.3 Обоснование необходимости использования вычислительной техники для решения задачи | 10 |
| 1.4 Анализ существующих разработок | 14 |
| 2 Проектная часть | 19 |
| 2.1 Обоснование проектных решений | 19 |
| 2.1.1 Обоснование проектных решений по техническому обеспечению | 19 |
| 2.1.2 Обоснование проектных решений по информационному обеспечению | 20 |
| 2.1.3 Обоснование проектных решений по программному обеспечению | 21 |
| 2.1.4 Обоснование выбора программных средств | 22 |
| 2.2 Информационное обеспечение задачи | 23 |

| | |
|---|----|
| 2.2.1 Информационная модель и ее описание..... | 23 |
| 2.2.2 Характеристика первичных документов с нормативно-справочной и входной оперативной информацией | 26 |
| 2.2.3 Характеристика базы данных | 27 |
| 2.2.3.1 Характеристика инфологической модели БД | 27 |
| 2.2.3.2 Характеристика даталогической модели БД..... | 30 |
| 2.2.4 Характеристика результатной информации..... | 32 |
| 3 Программная реализация проектных решений..... | 33 |
| 3.1 Программное обеспечение задачи..... | 33 |
| 3.1.1 Дерево функций и сценарий диалога..... | 33 |
| 3.2 Организация технологии сбора, обработки и выдачи информации | 34 |
| 3.3 Описание контрольного примера реализации проекта | 36 |
| 3.4 Обоснование экономической эффективности проекта | 51 |
| ЗАКЛЮЧЕНИЕ | 56 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 58 |
| ПРИЛОЖЕНИЯ..... | 61 |

ВВЕДЕНИЕ

В настоящее время преподавателям и сотрудникам ВУЗа значительную часть времени приходится тратить на заполнение и составление различных документов. Это могут быть личные дела студентов, экзаменационные и тестовые задания, учебные планы, зачетные ведомости и т.д.

Одним из документов, заполнение которого является весьма сложным и долгим процессом, является рабочая программа дисциплины (РПД). Сложность при заполнении состоит в том, что для составления данного документа требуется несколько дополнительных документов, например, учебный план направления подготовки, карта компетенций направления подготовки, федеральный государственный образовательный стандарт по направлению подготовки, нормативные документы ВУЗа и т.д. В настоящее время заполнение рабочей программы полностью выполняется вручную, то есть производится ручной перенос данных из указанных ранее документов в соответствующие разделы РПД, а также ручной ввод дополнительной информации, необходимой для полноты создания рабочей программы дисциплины. Данный процесс может быть упрощен за счет разработки специализированной автоматизированной информационной системы. Поэтому разработка автоматизированной системы создания макетов рабочих программ дисциплин является актуальной.[6]

В НИУ «БелГУ» рабочая программа дисциплины разрабатывается в соответствии со федеральным государственным образовательным стандартом высшего образования 3+. Настоящий стандарт высшего образования представляет собой совокупность требований, обязательных при реализации основных профессиональных образовательных программ высшего образования.[24] Данный стандарт разрабатывается отдельно для каждого направления подготовки, но требования к содержанию рабочей программы дисциплины идентичны, поэтому на основе данного стандарта в НИУ «БелГУ» был разработан единый макет рабочей программы для всех дисциплин всех направлений подготовки.

Объектом в данной работе является процесс разработки рабочих программ дисциплин в НИУ «БелГУ». Предметом – структура и алгоритмы разработки рабочих программ дисциплин.

Целью данной работы является сокращение затрачиваемого преподавателями времени на создание рабочих программ дисциплин за счет разработки и внедрения автоматизированной системы создания макетов рабочих программ дисциплин.

Для выполнения выпускной квалификационной работы были определены следующие задачи:

- анализ предметной области – процесса разработки рабочих программ дисциплин;
- анализ и моделирование существующих алгоритмов разработки рабочих программ дисциплин;
- моделирование алгоритмов разработки рабочих программ дисциплин «как должно быть»;
- выбор и обоснование выбора инструментальных средств разработки автоматизированной информационной системы;
- проектирование базы данных информационной системы;
- проектирование пользовательского интерфейса;
- разработка информационной системы;
- разработка мультимедийного руководства пользователя;
- анализ эффективности внедрения автоматизированной информационной системы;
- тестирование автоматизированной системы создания макетов рабочих программ дисциплин.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка библиографических источников и приложений.

В первой главе проведен анализ предметной области, выполняется постановка задачи, а также производится анализ существующих разработок, которые предназначены для выполнения аналогичных задач.

Во второй главе проведено обоснование проектных решений по техническому, информационному, и программному обеспечению, проведен анализ предметной области «Как должно быть», выполнено проектирование базы данных, приведены характеристики первичных и результирующих данных.

В третьей главе приведено описание разработанной системы, приводится описание контрольного примера реализации проекта, а также рассчитывается эффективность от использования автоматизированного решения.

В приложениях представлено техническое задание, пример титульного листа рабочей программы дисциплины, созданный при помощи разрабатываемой системы, а также исходный код некоторых модулей и функций системы, а именно:

- модуля обновления программы;
- модуля обращения в поддержку;
- модуля загрузки РПД;
- код утилиты обновления файлов.

Выпускная квалификационная работа изложена на 56 страницах, содержит 34 рисунка, 6 таблиц, 6 приложений и 30 библиографических источников.

1 Аналитическая часть

1.1 Технико-экономическая характеристика предметной области

1.1.1 Характеристика НИУ «Белгородский государственный университет»

Белгородский Государственный университет – крупнейшее высшее учебное заведение Белгородской области. В настоящее время здесь обучается более 25 тысяч студентов, каждый год выпускается около 5 тысяч.

В настоящее время в НИУ «БелГУ» имеется 16 совместных образовательных программ с ведущими университетами стран Европы и Азиатско-Тихоокеанского региона, 6 образовательные программы подготовки специалистов и магистров реализуются на английском языке, более 150 договоров о межвузовском сотрудничестве, 14 академиков и членов-корреспондентов РАН, более 1100 докторов и кандидатов наук, оформлены договорные отношения с более чем 600 предприятиями, являющимися базами практик, более 105 кафедр, 24 из которых базовые, 180 основных образовательных программ подготовки бакалавров, специалистов и магистров, 19 советов по защите докторских и кандидатских диссертаций и многое другое.[22]

Университет разделяется на 13 подразделений:

- Институт управления;
- Факультет журналистики;
- Институт педагогический;
- Институт юридический;
- Институт межкультурной коммуникации и международных отношений;
- Институт медицинский;
- Факультет подготовительный;

- Институт экономики;
- Институт инженерных технологий и естественных наук;
- Факультет социально-теологический;
- Факультет горного дела и природопользования;
- Факультет среднего профессионального образования;
- Старооскольский филиал.

1.1.2 Краткая характеристика кафедры прикладной информатики и информационных технологий

Кафедра прикладной информатики и информационных технологий образована в результате реструктуризации подразделений университета при образовании нового Института, в результате слияния кафедры «Прикладной математики и информатики» факультета компьютерных наук и телекоммуникаций и кафедры «Информационного менеджмента» института управления.

Кафедра прикладной информатики и информационных технологий является выпускающей кафедрой по направлениям подготовки 38.03.05 «Бизнес-информатика» (профилям «Архитектура предприятия» и «Электронный бизнес»), 09.03.03 «Прикладная информатика» (профиль «Прикладная информатика в экономике»), а также двум магистерским программам: 38.04.05 «Бизнес-информатика» (профиль «Управление жизненным циклом информационных систем»), 09.04.03 «Прикладная информатика» (профиль «Информационно-аналитическое обеспечение принятия решений»).[23]

1.2 Постановка задачи

1.2.1 Цель и назначение автоматизированного варианта решения задачи

Назначением реализации проекта «Автоматизированной информационной системы составления рабочих программ дисциплин» является решение следующих задач:

- сокращение времени, затрачиваемого на разработку рабочих программ дисциплин;
- гарантия соблюдения единых стандартов оформления рабочих программ дисциплин;
- упрощение изменения данных в ранее созданных рабочих программах дисциплин;
- автоматизированный перенос данных из внешних документов в рабочую программу.

Цель разработанной системы заключается в устранении недостатков, которые наблюдаются при ручном заполнении макетов рабочих программ.

1.2.2 Общая характеристика организации решения задачи на ЭВМ

Информационная система составления макетов рабочих программ дисциплин для НИУ «БелГУ» будет реализована в виде приложения для операционных систем семейства Windows и представлять собой многооконное[14] Windows Forms[10] приложение. Данная система предназначена исключительно для решения задач непосредственно связанных с разработкой рабочих программ дисциплин. [5]

Информационная система позволит значительно сократить время, затрачиваемое на разработку рабочих программ, за счет того, что вся необходимая информация из ФГОС, карт компетенций и учебных планов будет

хранится в базе данных системы и автоматически загружаться при выборе той или иной дисциплины.

Работа с системой будет происходить через следующие экранные формы:

- основная форма (состоит из вкладок, соответствующих разделам рабочей программы дисциплины, и позволяет заполнять информацию необходимую для формирования рабочей программы);
- форма информации о программе (позволяет перейти в руководство пользователя и обновить программу до последней версии);
- прочие вспомогательные формы (небольшие формы, необходимые для обеспечения определенного функционала, например, выбора преподавателя из списка или шкалирования оценок).

1.3 Обоснование необходимости использования вычислительной техники для решения задачи

В настоящее время в НИУ «БелГУ» составление рабочих программ дисциплин происходит ручным вводом данных в установленный правилами макет. Это создает определенные трудности при составлении и нерациональное использование рабочего времени преподавателя.

Схема данного процесса может быть представлена в виде диаграммы «КАК ЕСТЬ», разработанной при помощи CASE-средства BPWin. Диаграмма представляет собой смешанную диаграмму в нотациях IDEF0 и IDEF3.

Стандарт IDEF0 (FIPS183) предназначен для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающие эти функции.

Метод, предлагаемый стандартом IDEF0, предназначен для функционального моделирования, то есть моделирования выполнения функций объекта, путем создания описательной графической модели, показывающей что,

как и кем делается в рамках функционирования предприятия. Функциональная модель представляет собой структурированное изображение функций производственной системы или среды, информации и объектов, связывающих эти функции.[1]

Стандарт IDEF3 предназначен для документирования технологических процессов, происходящих на предприятии, и предоставляет инструментарий для наглядного исследования и моделирования их сценариев.[12]

Сценарием называется описание последовательности изменений свойств объекта, в рамках рассматриваемого процесса (например, описание последовательности этапов обработки детали в цеху и изменение её свойств после прохождения каждого этапа).[1]

Контекстная диаграмма процесса разработки рабочей программы дисциплины в нотации IDEF0 представлена на рисунке 1.1.

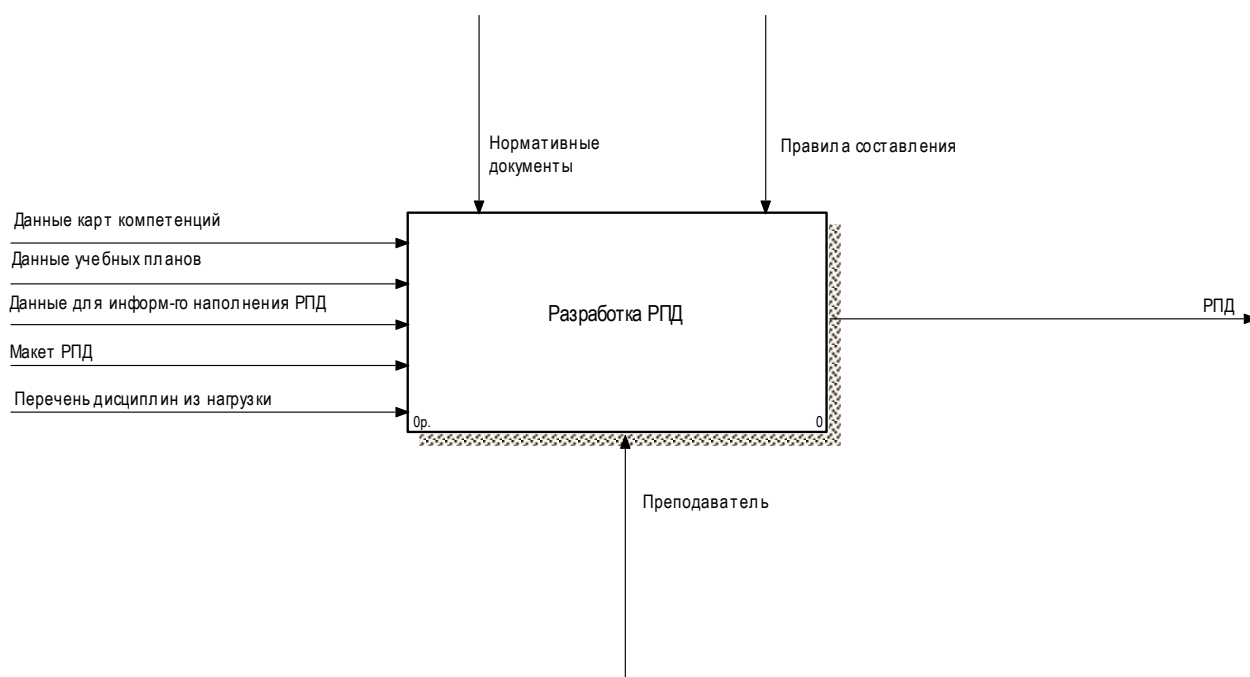


Рисунок 1.1 – Контекстная диаграмма

Из рисунка 1.1 видно, что преподаватель, руководствуясь нормативными документами и правилами составления рабочих программ, использует для разработки рабочей программы макет, данные из учебных планов, данные из

карт компетенций, данные для информационного наполнения рабочей программы (к ним относятся данные, не входящие в учебные планы и в карты компетенций, например, список литературы, список электронных ресурсов и пр.) и перечень дисциплин из нагрузки. Результатом процесса является непосредственно рабочая программа дисциплины.

На рисунке 1.2 представлена диаграмма декомпозиции данного процесса в нотации IDEF0. На рисунке видно, что в настоящее время процесс разработки рабочей программы может быть условно разделен на два процесса:

- выбор дисциплины;
- заполнение макета рабочей программы дисциплины.

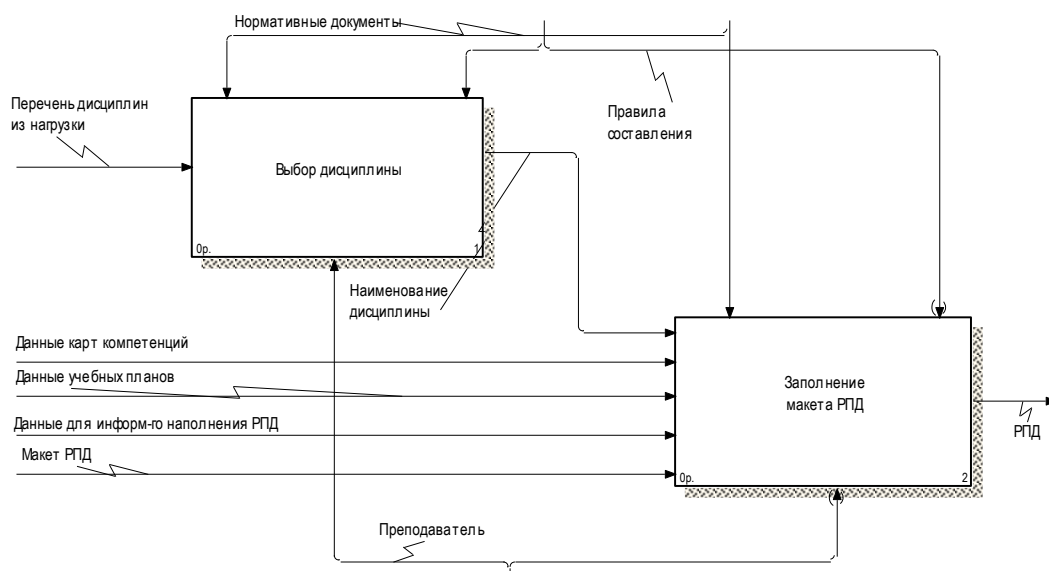


Рисунок 1.2 – Диаграмма декомпозиции «Разработка РПД»

На рисунке 1.3 представлена диаграмма декомпозиции «Заполнение макета РПД» в нотации IDEF3. Первым шагом в заполнении макета является выбор информации из документов. На данном этапе происходит выбор информации о дисциплине для всех форм обучения направления подготовки, для которого разрабатывается рабочая программа.

Далее выполняется заполнение титульной страницы, после чего происходит заполнение разделов.

На данной диаграмме порядок заполнения и связи между разделами выбраны исходя из используемых в данных разделах данных:

- в разделах 1, 9.1 и 9.2 макета рабочей программы используются данные о компетенциях и их уровнях. Заполнив раздел 1 с указанием требований к результату освоения компетенции, введенные данные переносятся в таблицы разделов 9.1 и 9.2;

- в разделах 2, 3, 4, 9.3 используются данные из учебного плана. Данные о часах из раздела 3 переносятся в поля «Всего» таблицы раздела 4, а данные о темах и разделах и часах на каждый вид деятельности из раздела 4 переносятся в раздел 9.3 для указания количества начисляемых баллов балльно-рейтинговой системы оценивания;

- остальные разделы заполняются либо используя данные иных документов (рекомендации кафедры, требования к тестовым заданиям и пр.), либо на усмотрение преподавателя, составляющего рабочую программу дисциплины.

Заполнение всех разделов должно быть обязательно (не одновременно) начато после заполнения титульной страницы и обязательно должно быть закончено для создания рабочей программы дисциплины. Данная особенность показана с помощью перекрестков нотации IDEF3.

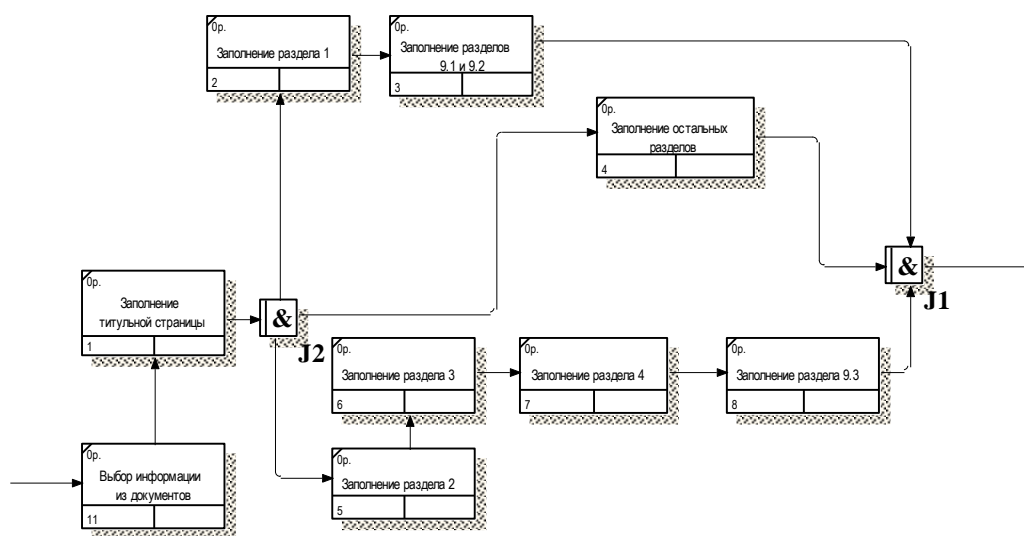


Рисунок 1.4 – Диаграмма декомпозиции «Заполнение макета РПД»

В ходе анализа структурно-функциональной диаграммы «Как есть» были выявлены следующие недостатки:

- большое количество времени, которое тратится на разработку рабочей программы;
- необходимость поиска дополнительных документов для разработки рабочей программы дисциплины;
- возможность ошибок оформления при разработке рабочих программ;
- нерациональное использование вычислительной техники.

Необходимость использования вычислительной техники заключается в:

- сокращение времени разработки рабочих программ;
- исключение необходимости поиска дополнительных документов за счёт наличия базы данных, содержащей в себе всю необходимую из них информацию;
- гарантированное соблюдение единого порядка оформления рабочих программ;
- сокращение времени работы вычислительной техники, затрачиваемого на разработку рабочих программ.

1.4 Анализ существующих разработок

В рамках анализа существующих разработок было выполнено сравнение четырех систем: ИС УУП СФУ РПД, ИМЦ: Управление университетом, Генератор рабочих программ, ПЛАНЫ. Ниже представлена характеристика каждой из них.

1) ИС Управления Учебным Процессом (УУП) СФУ Рабочие Программы Дисциплин (РПД)

Данная система разработана в Сибирском Федеральном университете и предоставляет следующие возможности:

- просмотр и редактирование учебных планов;
- просмотр и редактирование расписаний сессий;
- создание и редактирование рабочих программ дисциплин;
- работа с семестровыми графиками.

Данная система представляет собой .Net приложение, то есть оконное приложение для операционных систем семейства Windows. Пример интерфейса данной системы представлен на рисунке 1.5.

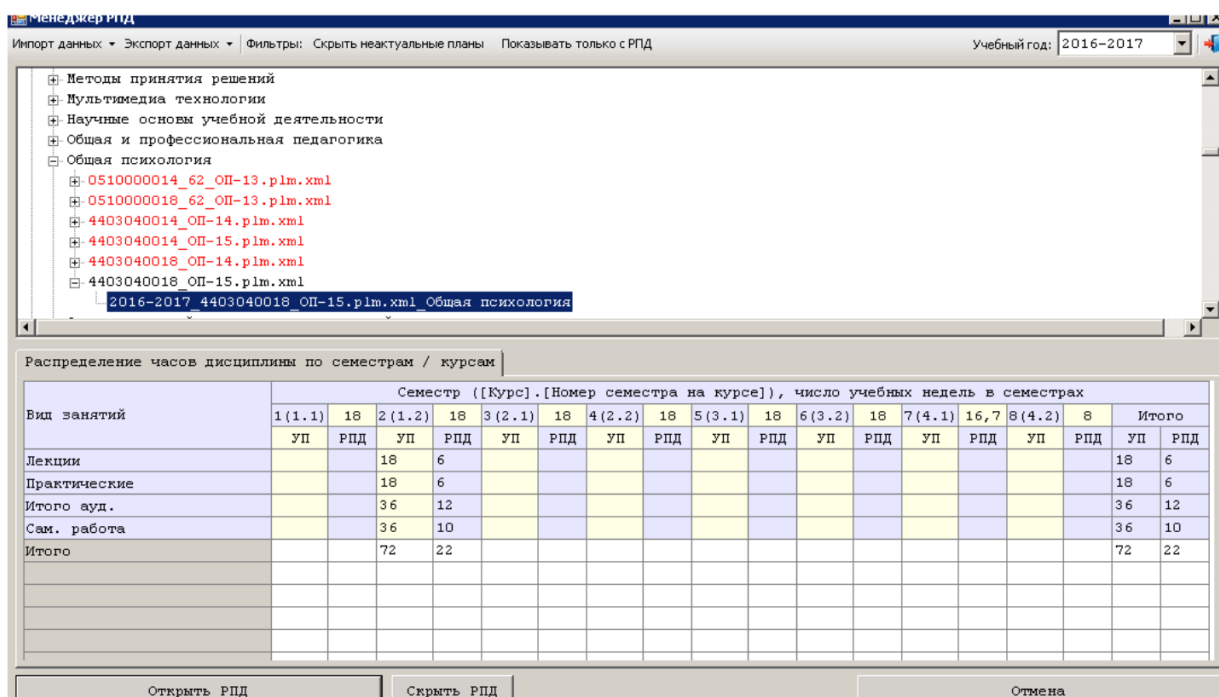


Рисунок 1.5 – Интерфейс «СФУ РПД»

2) ИМЦ: Управление Университетом

Данная система разработана в Самарском национальном исследовательском университете имени академика С.П. Королева и предоставляет следующие возможности:

- создание и редактирование рабочих программ дисциплин;
- разработка учебных планов;
- работа с зачетными ведомостями;
- разработка графиков работы подразделений и др.

Данная система разработана на платформе 1С 8.3, имеет как оконную так и веб-версию. Благодаря возможностям современных версий платформы 1С веб-версия представляет собой точную копию настольной, но может быть запущена через любой браузер с любого устройства на котором есть доступ в интернет. Пример интерфейса представлен на рисунке 1.6.

| Дисциплина | надо | есть |
|---|------------|------------|
| Logistics | | 1 |
| Strategic Management | | 1 |
| Supply chain management | | 1 |
| Антикризисное управление | 28 | 9 |
| Аутсорсинг | 4 | 1 |
| Бизнес-планирование в логистике | 4 | 3 |
| Введение в инновационное предпринимательство | 3 | 2 |
| Инвестиционный менеджмент | 14 | 3 |
| Иновационный менеджмент | 15 | 6 |
| Исследование систем управления | 10 | 7 |
| Концепции управления наукоемкими предприят... | 3 | 1 |
| Логистика | 23 | 17 |
| Логистика в торговле | 4 | 3 |
| Логистика распределения | 4 | 3 |
| Логистика складирования | 4 | 4 |
| Логистика снабжения | 4 | 3 |
| Менеджмент | 32 | 17 |
| Методы исследования в менеджменте | 3 | 1 |
| Моделирование бизнес-процессов | 4 | 4 |
| Научно-исследовательская практика | 2 | 2 |
| ИТОГО | 435 | 218 |

| Дата | Дисциплина | Документ основание |
|---------------------|-------------------------|--|
| 27.06.2013 16:15:51 | Иновационный менеджмент | Учебный план 080200.62-2011-3-ПП-5г-00м-01 от 19.04. |
| 07.11.2013 13:11:36 | Иновационный менеджмент | Учебный план 080200.2.62-2012-О-П-4г-00м (2012-2016) |
| 06.12.2013 14:32:03 | Иновационный менеджмент | Учебный план 080200.2.62-2011-О-П-4г-00м (ФМ, набор |
| 06.12.2013 14:33:02 | Иновационный менеджмент | Учебный план 080200.2.62-2011-О-П-4г-00м (ФМ, набор |
| 25.02.2015 18:05:24 | Иновационный менеджмент | Учебный план 380302.62-2014-О-ПП-4г-00м-02 от 28.05. |
| 12.03.2015 14:45:16 | Иновационный менеджмент | Учебный план 380302.62-2014-3-ПП-5г-00м-01 от 02.04. |
| 12.03.2015 14:45:23 | Иновационный менеджмент | Учебный план 380302.62-2014-3-СС-4г-00м-01 от 02.04. |

Рисунок 1.6 – Пример интерфейса «ИМЦ: Управление университетом»

3) «Генератор рабочих программ»

Данная программа разработана по заказу Нижегородского государственного педагогического университета имени К. Минина.

Данная программа предоставляет следующие возможности:

- создание рабочей программы (РП) в соответствии с данными из учебных планов: видами занятий, плановой нагрузкой, необходимыми компетенциями, формами контроля.

- автоматическая проверка соответствия фактической и плановой нагрузки по всем видам занятий и указания всех необходимых компетенций.

- использование единых справочников видов занятий и форм контроля, а также стандартизованного описания материально-технического обеспечения, в том числе с учетом обучения лиц с ограниченными возможностями здоровья.

– создание фонда оценочных средств, в т.ч. автоматическое формирование набора типовых контрольно-измерительных материалов в зависимости от содержания РП и предгенерация содержания контрольно-измерительных материалов по содержанию РП.

– поддержка генерации и повторного использования: генерация форматированного РП и фос в odt, doc или pdf форматах, а также создание РП на основе существующих с копированием всего содержания.

Данная система представляет собой WEB-приложение и может быть запущена из любого браузера. Пример интерфейса данной системы представлен на рисунке 1.7.

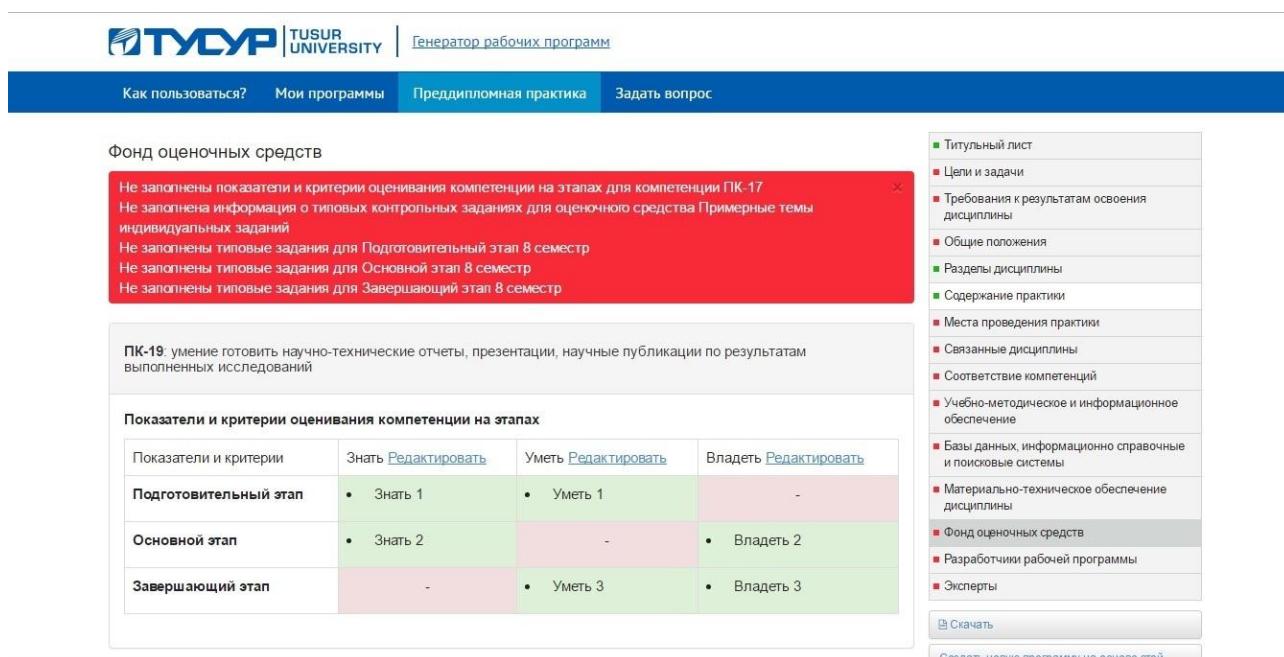


Рисунок 1.7 – Интерфейс «Генератор рабочих программ»

4) «ПЛАНЫ»

Данная система представляет собой небольшое приложение, предназначенное исключительно для разработки рабочих программ дисциплин. Программа автоматически загружает данные из учебных планов и других документов.

К недостаткам данной программы следует отнести довольно сложный и специфический интерфейс, сложную настройку заполнения и вывод результатов.

Результат сравнения и оценивания каждой системы представлен в таблице 1.1.

Таблица 1.1 – Оценка существующих разработок

| Возможности и критерии отбора | Системы | ИМЦ :Управление ВУЗом | ИС Управления Учебным Процессом | Г енератор рабочих программ | Планы |
|--------------------------------|---------|-----------------------|---------------------------------|-----------------------------|-----------|
| Возможность создания РПД | | 5 | 5 | 5 | 3 |
| Возможность редактирования РПД | | 5 | 5 | 5 | 3 |
| Интерфейс | | 4 | 4 | 4 | 1 |
| Интеграция с другими системами | | 3 | 2 | 1 | 1 |
| Кроссплатформенность | | 5 | 1 | 5 | 1 |
| Редактирование учебных планов | | 5 | 4 | 4 | 2 |
| Работа с другими документов | | 5 | 4 | 5 | 2 |
| Всего | | 32 | 25 | 29 | 15 |

Несмотря на хороший уровень выполнения своих функций, ни одна из рассмотренных систем не может быть приобретена и использоваться в НИУ «БелГУ», так как все приведенные выше системы рассчитаны исключительно на нужды и правила организаций-правообладателей, являются системами закрытого типа и не поставляются в сторонние организации, следовательно, целесообразным решением задачи является разработка собственной системы.

2 Проектная часть

2.1 Обоснование проектных решений

2.1.1 Обоснование проектных решений по техническому обеспечению

К техническому обеспечению относят комплекс технических средств, предназначенных для обеспечения работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

Разрабатываемая информационная система имеет клиент-серверную архитектуру, поэтому требования к серверной и клиентской технике различаются.

Для стабильной работы клиентской части достаточно компьютера со следующими характеристиками:

- процессор: AMD Athlon X2 340 OEM;
- материнская плата: GIGABYTE GA-F2A68HM-S1;
- видеокарта: любая с поддержкой разрешения от 1024x768;
- оперативная память (МБ): 2048;
- монитор: с разрешением от 1024x768;

Технические средства и документация:

- клавиатура;
- мышь;
- принтер для бумаги формата А4;
- бумага для печати, инструкция по эксплуатации;
- картриджи.

Для стабильного функционирования серверной части подойдет любой компьютер с не менее чем 2048 МБ оперативной памяти и доступом в Интернет.

2.1.2 Обоснование проектных решений по информационному обеспечению

Информационное обеспечение – это совокупность информационных массивов данных, единая система кодирования и классификации информации, различные схемы информационных потоков, вращающиеся в организации, а также методология построения баз данных [11].

Информационное обеспечение подразделяется

– внутримашинное информационное обеспечение – это информационная база, которая представлена на машинном носителе, реализованная в режиме реального времени (база данных);

– немашинное информационное обеспечения – это документация на бумажных носителях (приказы, распоряжения, договора и другое). [13]:

Существуют также такие понятия как входная, выходная и управленческая информация. Входная – это информация, которая используется в работе, в дальнейшем может быть преобразована, для получения результата. Выходная – это проработанная или преобразованная информация, которая является результатом работы. Управленческая – это информация, которая несет в себе полезность для принятия человеком обоснованного решения [18].

Входными данными разрабатываемой информационной системы составления макетов рабочих программ дисциплин являются сведения о направлении подготовки, о годе поступления и о дисциплине для которой разрабатывается рабочая программа.

Управленческой информацией необходимой при работе с подсистемой является: Приказ от 01.04.2016 №266-ОД «Об утверждении методических рекомендаций по разработке рабочей программы дисциплины в соответствии с ФГОС ВО».

Выходными данными разрабатываемой информационной системы составления макетов рабочих программ дисциплин являются непосредственно

рабочие программы и сведения об их составлении (дата составления, дисциплина, направление подготовки).

2.1.3 Обоснование проектных решений по программному обеспечению

Программное обеспечение – это совокупность программ, которые обеспечивают функционирование вычислительной системы, а также программ, направленных для решения конкретных задач пользователя [19].

От выбора программного обеспечения зависят конечные характеристики и особенности продукта, его производительность, интерфейс, ограничения использования и прочее.

Для выполнения данного проекта необходимо осуществить выбор операционной системы и пакета прикладных программ, необходимого для работы с макетами РПД.

Прежде всего необходимо осуществить выбор операционной системы. Для реализации данного проекта была выбрана операционная система Windows 7. По данным на март 2018 года данная система является второй по популярности операционной системой семейства Windows.

Основной причиной выбора данной операционной системы послужило то, что на большинстве компьютеров кафедры прикладной информатики и информационных технологий установлена операционная система Windows 7 Корпоративная[19].

2.1.4 Обоснование выбора программных средств

В рамках данного этапа выполнения проекта был осуществлен выбор СУБД, языка программирования и среды разработки.

В качестве СУБД была выбрана MySQL. Данный выбор был сделан в первую очередь из-за её кроссплатформенности. Также данная СУБД имеет ряд других преимуществ:

- гибкая система привилегий и паролей;
- записи имеют фиксированную и переменную длину;
- в большинстве случаев является бесплатной;
- многопоточность, поддерживается несколько одновременных запросов;
- за один проход обеспечивается оптимизация связей с присоединением многих данных;
- быстрая работа, масштабируемость;
- гибкая поддержка форматов чисел, строк переменной длины и меток времени;
- ODBC драйвер;
- интерфейс с языками C# и Perl, PHP;
- провайдеры услуг хостинга обеспечивают хорошую поддержку;
- совместимость с ANSI SQL[18];
- быстрая поддержка транзакций через механизм InnoDB.[2]

В качестве среды разработки была выбрана Microsoft Visual Studio Community 2017 и язык C#. Выбор среды разработки и языка программирования осуществлялся взаимосвязано.

C# – объектно-ориентированный [9], строго типизированный, управляемый язык, разработанный компанией Microsoft разработанный в 2000 году. В настоящее время актуальная версия языка 7.2, выпущенная 15 ноября 2017 года. Данный язык был выбран благодаря целому ряду преимуществ, среди которых:

- возможность расширения системы (использование сторонних библиотек, сторонних программ и пр.);
- кроссплатформенность;
- невысокая сложность разработки, поддержки и расширения;
- высокая степень доступности исходных текстов библиотек;
- легкая читаемость кода;
- встроенные возможности распределения процессов и увеличения быстродействия.

Также для языка C# создана специальная библиотека, позволяющая значительно упростить работу с COM сервером[23] Word и осуществлять программное формирование документов используя объектную модель продуктов Microsoft Office.

Visual Studio 2017 Community является самой популярной средой разработки .Net приложений (к которым относятся и приложения на языке C#). Данная среда программирования имеет обширный функционал для разработки, отладки, сборки и публикации приложений.

Данная среда программирования была выбрана благодаря своим широким возможностям для разработчиков и благодаря тому, что данная версия является бесплатной для учебных учреждений.

2.2 Информационное обеспечение задачи

2.2.1 Информационная модель и ее описание

На данном этапе проектирования информационной системы была построена информационная модель, которая представляет собой смешанную модель в нотациях IDEF0, IDEF3, DFD[7] и показывает процесс формирования

макета рабочей программы дисциплины с использованием проектируемой системы.

Выбор нотаций осуществлялся исходя из возможности более наглядно продемонстрировать каждый конкретный этап создания макета рабочей программы дисциплины.

С использованием разрабатываемой системы процесс создания макета рабочей программы может быть разделен на 3 этапа: выбор дисциплины; заполнение формы программы; формирование РПД. Данные этапы были отражены в нотации IDEF0 на диаграмме декомпозиции первого уровня. Входными данными в данном случае является перечень дисциплин из нагрузки преподавателя, управляющими данными являются нормативные документы и руководство пользователя информационной системы, механизмом исполнения являются преподаватель, разрабатывающий рабочую программу, и информационная система. Диаграмма декомпозиции первого уровня представлена на рисунке 2.1.

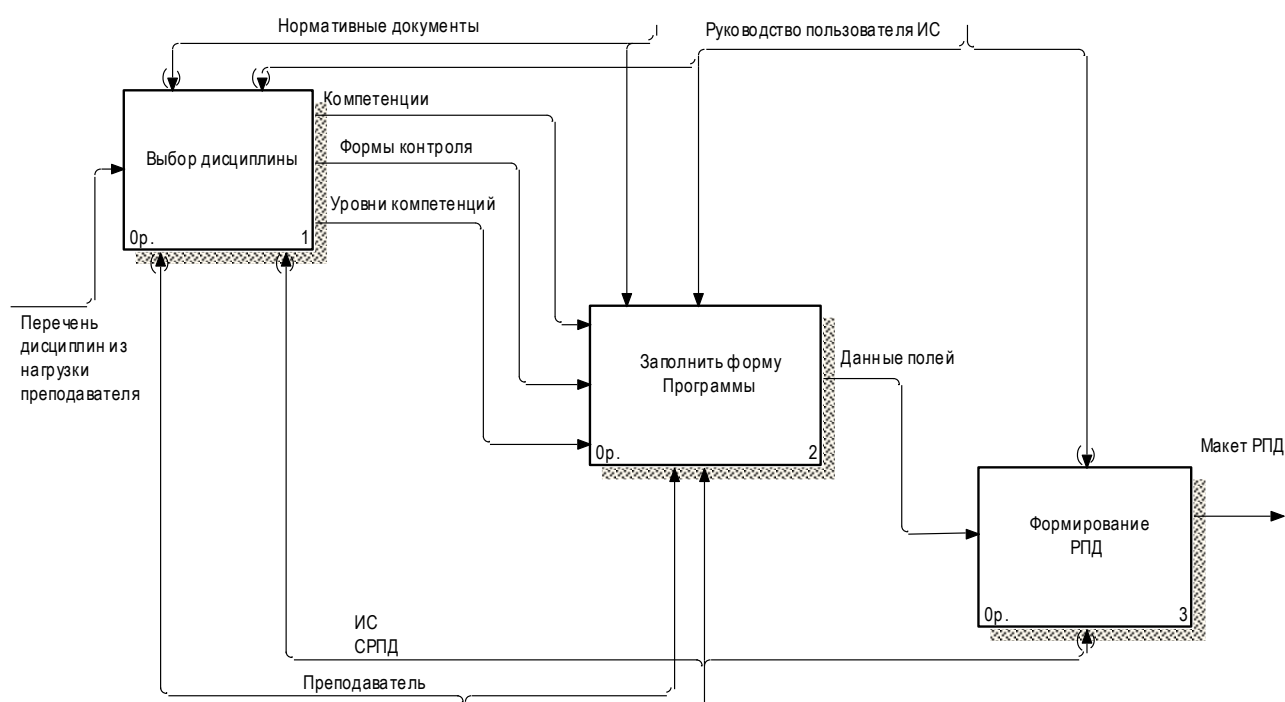


Рисунок 2.1 –Диаграмма декомпозиции «Составление РПД»

В свою очередь этап «Выбор дисциплины» можно разделить на три этапа: выбрать год поступления; выбрать образовательную программу; выбрать дисциплину. Входными данными является перечень дисциплин из нагрузки преподавателя, выходными – компетенции, формы контроля и уровни компетенций. Также на данном этапе отмечены хранилище данных – база данных и внешняя сущность – администратор базы данных, который осуществляет наполнение базы данных. Диаграмма декомпозиции «Выбор дисциплины» представлена на рисунке 2.2.

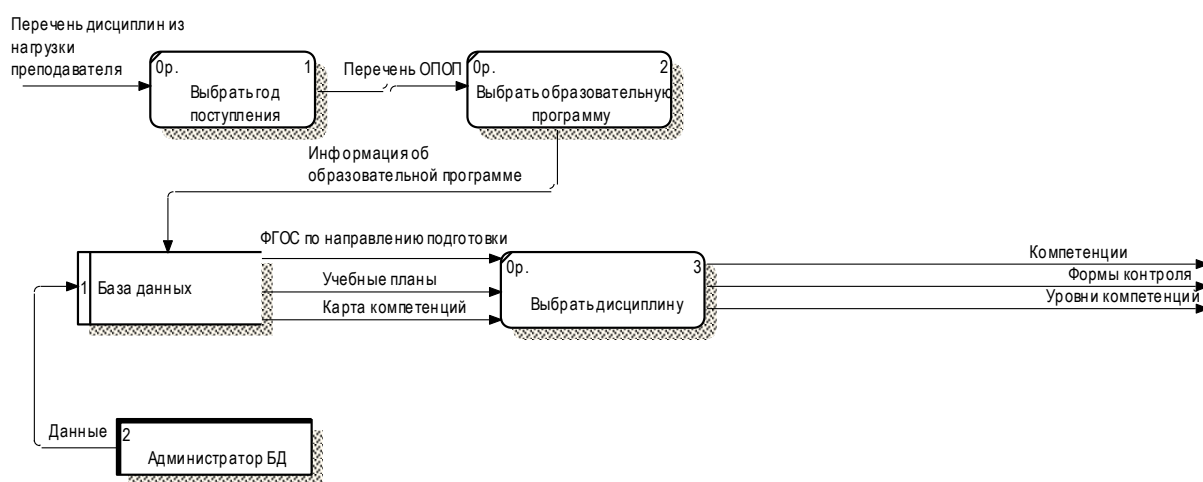


Рисунок 2.2 – Диаграмма декомпозиции «Выбор дисциплины»

После выбора дисциплины необходимо заполнить форму программы. Данный процесс частично автоматизирован за счет автоматического подбора данных из базы, согласно выбранной информации о дисциплине. Первым заполняется титульная страница, далее происходит заполнение разделов по порядку. После того, как завершено заполнение всех разделов, выполняется проверка правильности и исправление ошибок в случае их обнаружения. Данный процесс наиболее наглядно может быть отображен в нотации IDEF3. Диаграмма декомпозиции процесса «Заполнить форму программы» представлена на рисунке 2.3.

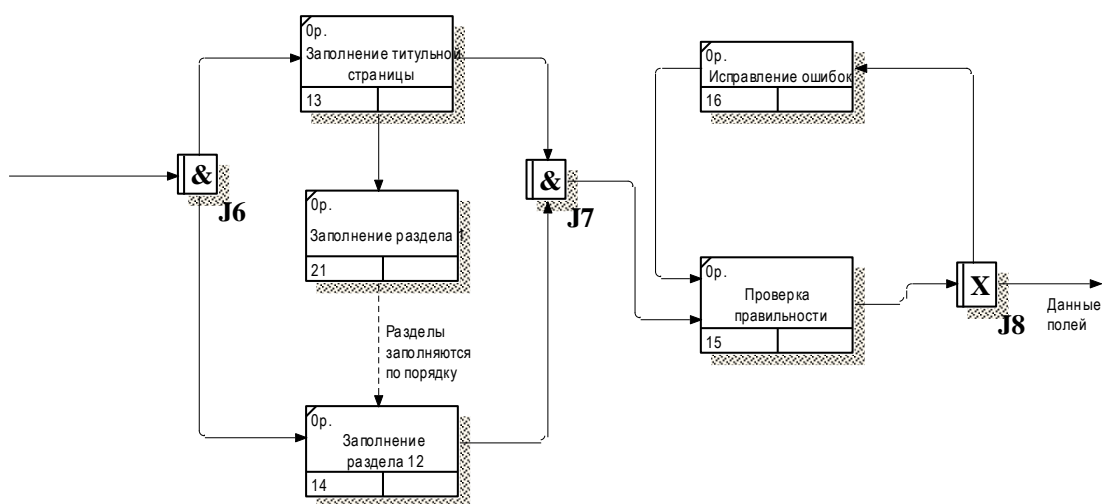


Рисунок 2.3 – Диаграмма декомпозиции «Заполнить форму программы»

Процесс «Формирование РПД» в декомпозиции не нуждается, так как он выполняется системой без участия преподавателя. Преподавателю необходимо лишь запустить данный процесс нажатием соответствующей кнопки на форме программы.

2.2.2 Характеристика первичных документов с нормативно-справочной и входной оперативной информацией

Для создания рабочей программы дисциплины необходим утвержденный шаблон[25] (правила оформления) и следующие документы:

- учебные планы направления подготовки всех реализуемых форм обучения (из которых берётся информация о часах, о семестрах, о количестве зачетных единиц, о формах контроля и др.);
- карты компетенций направления подготовки (из которых берётся информация о результатах освоения дисциплины);
- ФГОС по направлению подготовки.

Таким образом структура документов методического отдела в рамках деятельности по разработке рабочих программ дисциплин имеет 4 группы: карты

компетенций, учебные планы, ФГОС и непосредственно рабочие программы дисциплин.

Структура документов методического отдела в рамках деятельности по разработке рабочих программ дисциплин представлена на рисунке 2.4

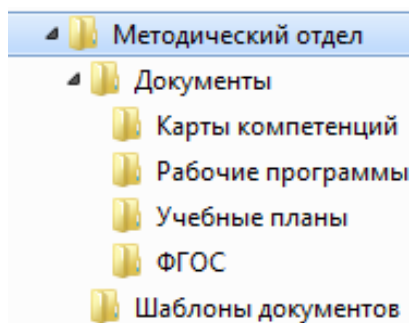


Рисунок 2.4 – Структура документов

2.2.3 Характеристика базы данных

2.2.3.1 Характеристика инфологической модели БД

Инфологическое проектирование – процесс создания логической модели информационной системы. Логической моделью называется модель информационной системы без привязки к конкретной СУБД. Это означает, что данная модель отражает характеристики всех сущностей и связей в информационной системе, но не указывает конкретных типов полей, названий полей и других ограничений накладываемых СУБД.

По результатам анализа предметной области было принято решение о создании следующего набора сущностей:

– компетенции – содержат информацию о всех возможных компетенциях;

– карта компетенций – сопоставляет информацию о компетенции, её уровне и дисциплине во время изучения которой данная компетенция достигается;

– факультеты_институты – содержит информацию о структурных подразделениях университета;

– часы – содержит информацию о часах на каждый вид деятельности по семестрам;

– формы обучения – содержит информацию о возможных формах обучения;

– направления подготовки – содержит информацию о всех доступных направлениях подготовки;

– должности – содержит информацию о должностях преподавателей;

– учебный план – содержит информацию о дисциплине для направления подготовки, о формах контроля и пр.;

– преподаватели – содержит информацию о преподавателях;

– предметы – содержит информацию о возможных образовательных дисциплинах;

– степени – содержит список возможных ученых степеней;

– отрасли науки – содержит список возможных отраслей науки (для ученых степеней);

– звание;

– кафедры – содержит информацию о кафедрах университета;

– РПД – содержит информацию о созданных в системе рабочих программах.

Каждой сущности были установлены необходимые атрибуты, а именно:

– для сущности «Компетенции»: код компетенции, требования к результату;

– для сущности «Предметы»: ID предмета, название;

- для сущности «Степени»: ID степени, название;
 - для сущности «Отрасли науки»: ID науки, наука;
 - для сущности «Звание»: ID звания, название;
 - для сущности «Кафедры»: ID кафедры, название, сокр. название;
 - для сущности «Направления подготовки»: Код НП, название, профиль;
 - для сущности «Формы обучения»: ид формы обучения, название;
 - для сущности «Должности»: ид должности, название, сокр. название;
 - для сущности «Карта компетенций»: ID КП, уровень, код компетенции, ID УП;
 - для сущности «Факультеты_институты»: ID факультета, название, Сокр. название, Декан_Директор;
 - для сущности «Часы»: ID, ID УП, семестр, лекции, лаб. р., пр. раб, сам. раб, контроль, ЗЕТ;
 - для сущности «Учебный план»: ID УП, код НП, год поступления, ID формы обучения, ID предмета, ID кафедры, экзамены, зачёты, диф. зачеты, курсовые работы, курсовые проекты;
 - для сущности «Преподаватели»: ID преподавателя, Фамилия, Имя, Отчество, ID должности, ID степени, ID науки, ID звания, ID кафедры;
 - для сущности «РПД»: ID РПД, ID УП, ID Предмета, дата, XML, DOCX.
- После создания сущностей между ними были установлены необходимые связи.

В разрабатываемой модели все связи являются неидентифицирующими, то есть атрибуты родительской сущности, не определяют атрибуты дочерней сущности, а лишь дополняют её (внешний ключ не является первичным или частью составного первичного ключа в дочерней сущности).

Результат создания модели представлен на рисунке 2.5.

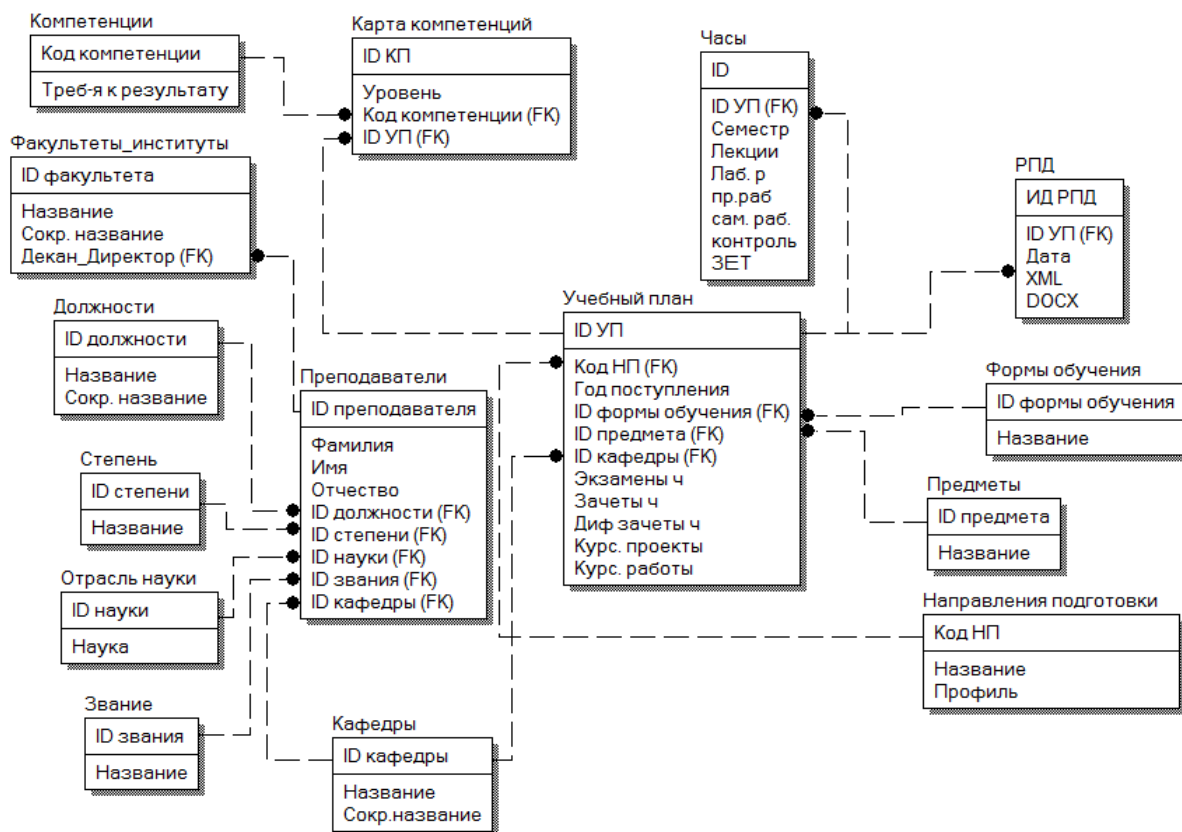


Рисунок 2.5 – Логическая модель базы данных

Разработанная модель базы данных отвечает требованиям третьей нормальной формы[8], а именно: все атрибуты атомарны; каждый неключевой атрибут функционально полно определяется первичным ключом; все неключевые атрибуты взаимно независимы и полностью зависят от первичного ключа.

2.2.3.2 Характеристика даталогической модели БД

Даталогическое проектирование – процесс создания физической модели информационной системы. Физическая модель отличается от логической тем, что имеет привязку к конкретной СУБД с указанием всех присущих ей ограничений, типов полей и т.д. Данная модель была получена с помощью утилиты Reverse Engineer, входящей в состав программного

пакета MySQL Workbench[3]. Данная утилита позволяет построить визуальную модель базы данных, на основе которой в последствии непосредственно создать базу данных на выбранном сервере.

Все названия сущностей и полей были заменены на латинские названия, а полям установлены соответствующие типы значений. В данной информационной системе используются следующие типы полей:

- int – целочисленный тип;
- varchar – символьный тип
- double – числа с фиксированной точкой

Физическая модель информационной системы представлена на рисунке 2.6.

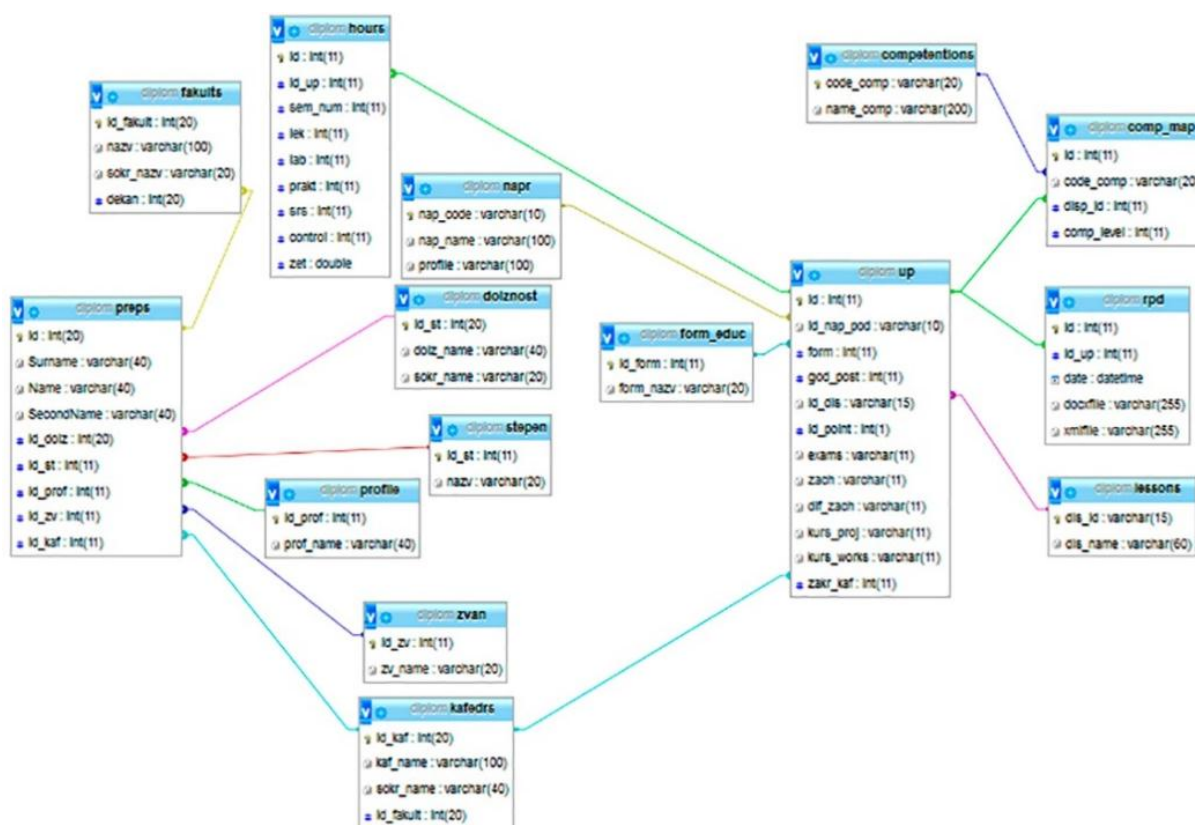


Рисунок 2.6 – Дatalogическая модель

2.2.4 Характеристика результатной информации

Результатной информацией в разрабатываемой системе является:

- рабочая программа в .docx формате;
- файл с данными рабочей программы в .xml формате;
- запись в таблице grd базы данных о созданной рабочей программе с указанием пути к xml файлу и к .docx файлу рабочей программы grd.

Рабочая программа в .docx формате создается как результат формирования документа системой и сохраняется на компьютер пользователя и на центральный сервер.

Файл с данными рабочей программы в .xml формате представляет собой файл специально разработанной структуры в котором содержится информация всех элементов формы. Данный файл необходим для продолжения работы с создаваемой ранее рабочей программой. Данный файл создается автоматически после завершения формирования РПД и загружается на центральный сервер. Помимо этого, данный файл может быть создан пользователем на любом этапе работы с программой, чтобы сохранить текущий сеанс работы.

Запись в таблице содержит информацию о дисциплине для которой разрабатывалась рабочая программа, о направлении подготовки, о времени публикации и о пути к xml и docx файлам на сервере. Данная информация позволяет вести историю изменений, а также легко найти необходимую рабочую программу чтобы использовать её любым необходимым образом.

3 Программная реализация проектных решений

3.1 Программное обеспечение задачи

3.1.1 Дерево функций и сценарий диалога

Управление программой происходит через выбор пункта меню (или подменю), либо нажатием какой-либо кнопки на экранной форме. Каждый пункт меню выполняет свою определенную функцию.

Дерево функций системы состоит из двух частей – основные и служебные. К основным относятся функции, непосредственно относящиеся к процессу работы с рабочими программами дисциплин. К служебным относятся функции, обеспечивающие работу с программой. Дерево функций разрабатываемой системы представлено на рисунке 3.1.

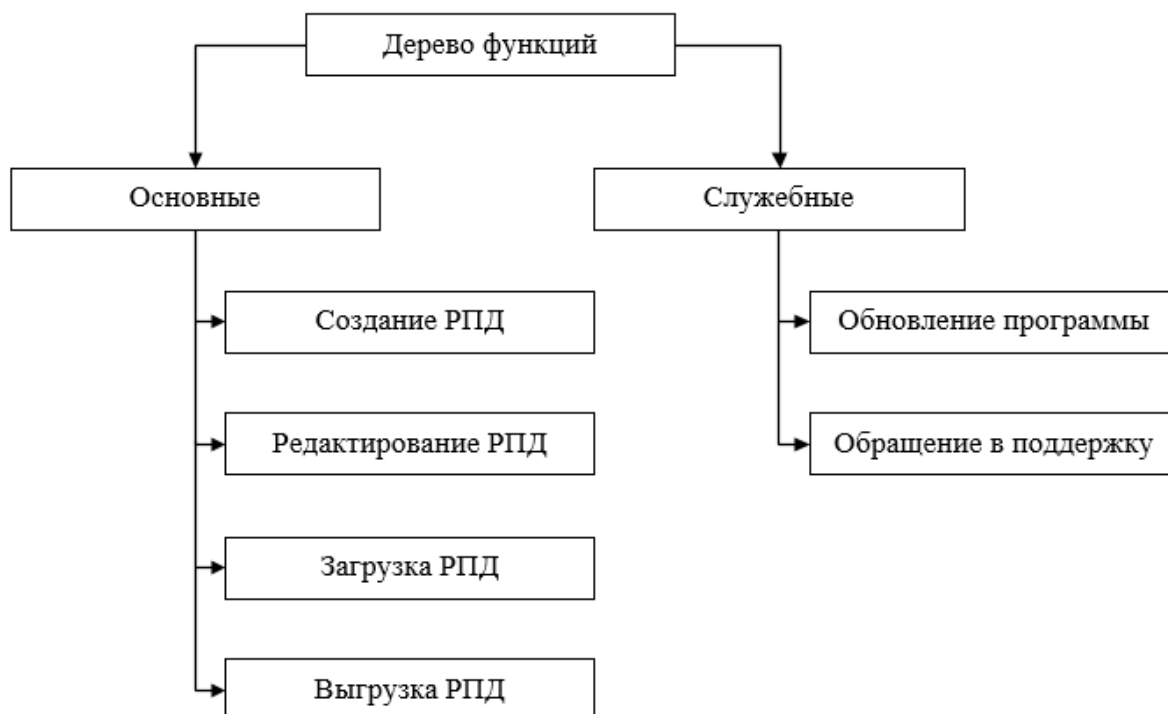


Рисунок 3.1 – Дерево функций системы

Сценарий диалога представлен в виде структуры форм программы и выполняемых ими функций. Диалог пользователя с системой может быть разделен на 5 групп:

- главное меню – позволяет выбрать необходимое действие;
- создание РПД – позволяет выполнить создание новой рабочей программы дисциплины;
- редактирование РПД – позволяет открыть созданную ранее рабочую программу дисциплины и изменить введенные данные.
- обновление программы – позволяет проверить наличие новых версий и установить их;
- поддержка – позволяет отправить сообщение в службу технической поддержки программы;

Сценарий диалога представлен на рисунке 3.2.



Рисунок 3.2 – Сценарий диалога

3.2 Организация технологии сбора, обработки и выдачи информации

Источниками нормативно-справочной информации (НСИ)[4] в разрабатываемой системе являются документы по учебной деятельности, такие как: учебный план, карта компетенций, ФГОС по направлению подготовки.

Помимо данных документов источниками НСИ для системы также являются следующие документы: штатное расписание сотрудников и нормативные документы по учебной деятельности.

Источниками оперативной информации в разрабатываемой системе являются любые, используемые преподавателем, источники информации, необходимые для разработки рабочей программы дисциплины.

Сбор информации в системе заключается в том, чтобы перевести информацию из документов в информацию базы данных. В текущей реализации автоматизированного преобразования не предусмотрено, поэтому необходимо вручную осуществлять ввод данных в базу.

В настоящее время планируется интеграция с информационной системой ИнфоБелГУ. В данной системе содержится вся необходимая информация для заполнения базы данных. При использовании данного источника технология сбора информации заключается в следующем: выполняется выборка данных из базы информационной системы, после чего выбранные данные преобразовываются в соответствии со структурой базы данных разрабатываемой системы и выполняется вставка (или обновление) данных в базу системы.

Обработка и выдача информации происходит во время работы с системой за счет выполнения запросов с установленными пользователем параметрами к базе данных. Параметры устанавливаются программно, вся структура запросов и процесс их выполнения скрыты от пользователя, ему доступны лишь необходимые для работы конечные данные.

Схему технологического процесса можно представить графически в виде графа, объектами которого являются субъекты обмена информацией, а связями – потоки данных. Схема процесса сборки, обработки и выдачи информации представлена на рисунке 3.3.

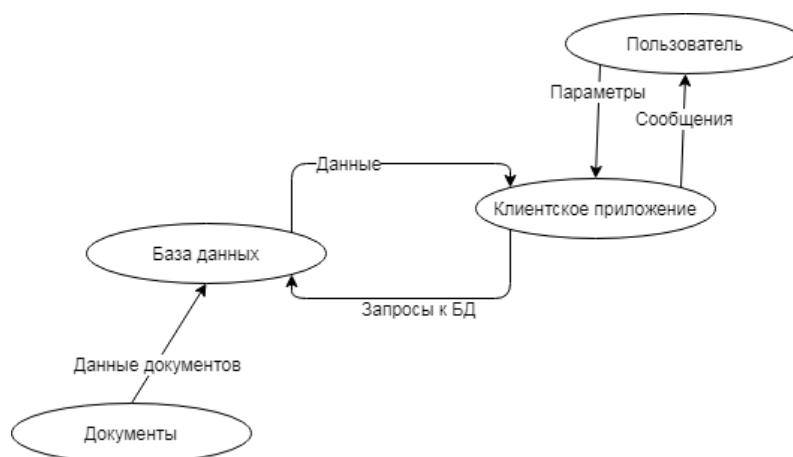


Рисунок 3.3 – Схема обработки данных

3.3 Описание контрольного примера реализации проекта

После запуска программы перед пользователем появляется главная форма. Главная форма разделена на две области: область меню и область заполнения макета рабочей программы дисциплины. Общий вид главной формы представлен на рисунке 3.4.

Панель меню содержит три раздела – «Файл», «О программе», «Поддержка».

В разделе «Файл» представлены возможности для работы с файлами рабочих программ дисциплин, создаваемых в системе. Данный раздел включает три кнопки:

- «Создать» – позволяет создать новую рабочую программу дисциплины (очистить форму, если в ней есть данные);
- «Открыть» – позволяет открыть сохраненный ранее файл (все данные сохраненного сеанса будут перенесены в форму);
- «Выбрать из ранее опубликованных» – позволяет загрузить в форму данные файла созданной ранее рабочей программы с сервера.

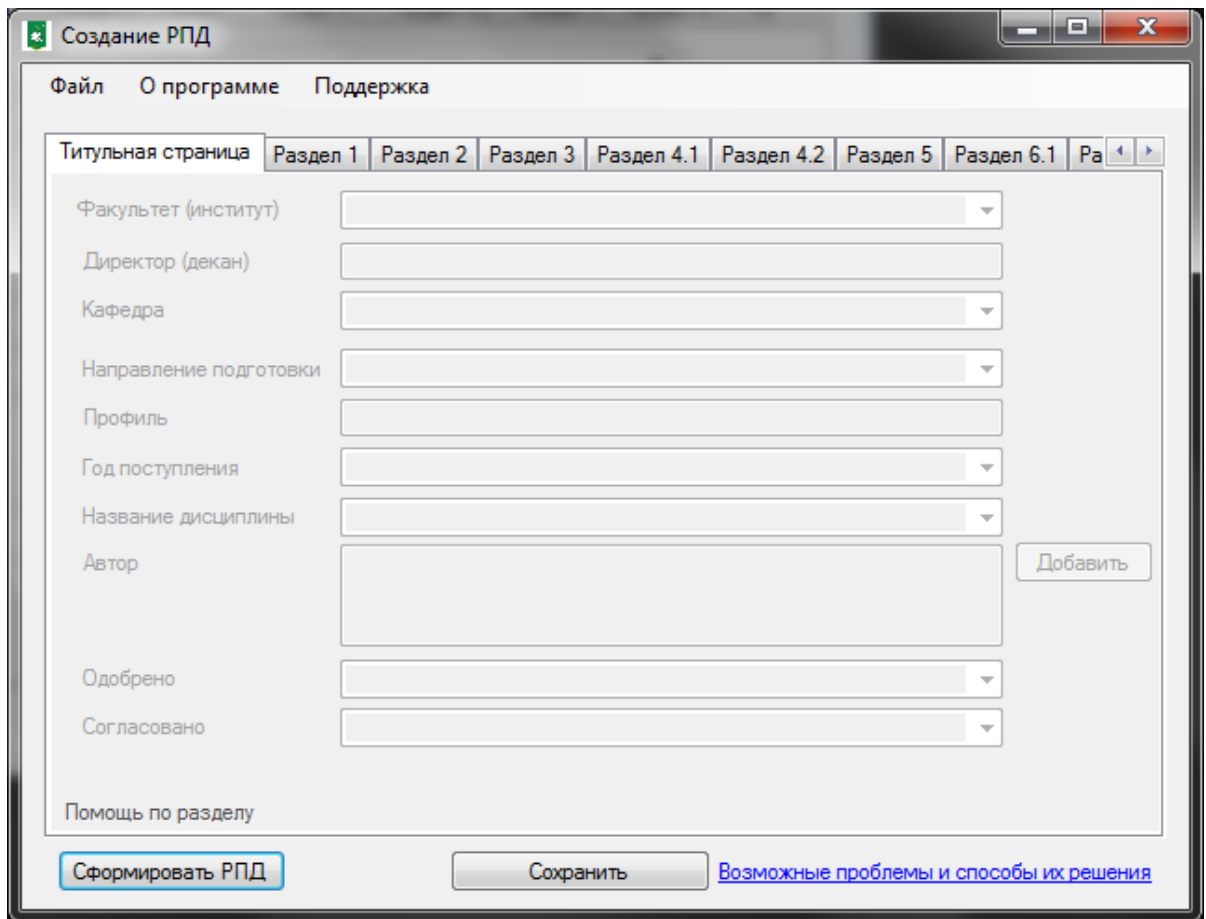


Рисунок 3.4 – Главная форма программы

При нажатии кнопки «Открыть» происходит вызов стандартного диалогового окна выбора файла с установленным фильтром по формату .xml.

При нажатии кнопки «Выбрать из ранее опубликованных» открывается специальная форма, в которой содержится список всех ранее создаваемых рабочих программ. Общий вид данной формы представлен на рисунке 3.5.

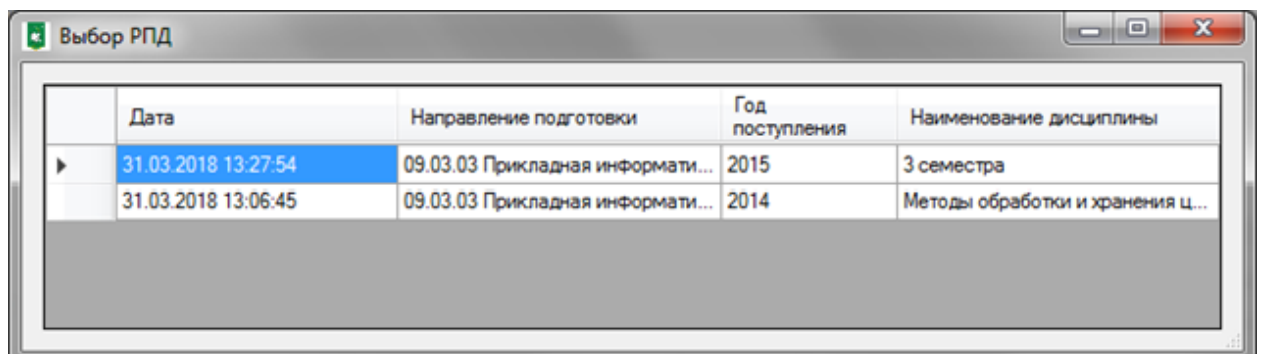


Рисунок 3.5 – Форма выбора РПД

Для выбора какой-либо рабочей программы дисциплины необходимо дважды нажать на соответствующую строку.

Раздел «О программе» вызывает форму информации о программе, о версии и о возможном обновлении. Общий вид данной формы представлен на рисунке 3.6.

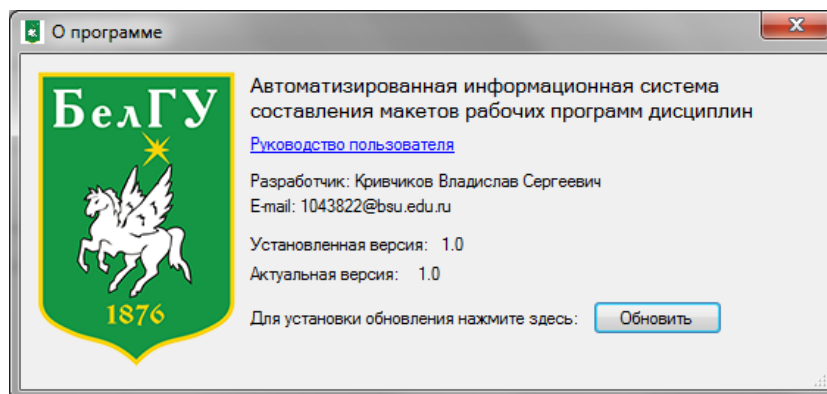


Рисунок 3.6 – Окно «О программе»

Область заполнения рабочей программы состоит из вкладок, которые разделены по разделам макета рабочей программы.

Раздел «Поддержка» позволяет открыть форму отправки сообщения в поддержку. В данной форме необходимо заполнить все поля, после чего нажатием кнопки «Отправить» введенное сообщение будет отправлено на электронный адрес, указанный в настройках системы. Общий вид формы с примером заполнения данных представлен на рисунке 3.7.

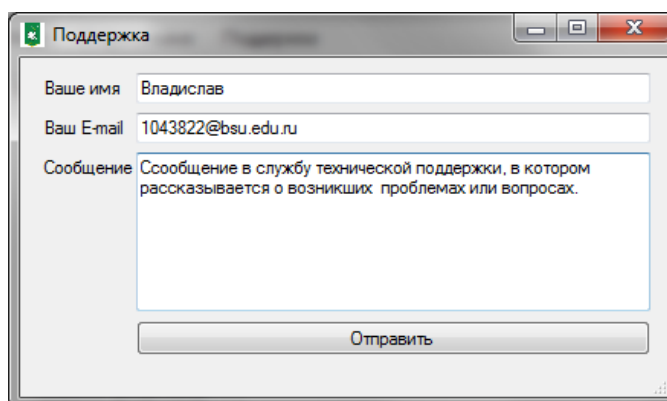


Рисунок 3.7 – Форма обращения в поддержку

На вкладке «Титульная страница» изначально данные доступны только в блоке выбора значения «Факультет (институт)», далее при заполнении данных блоков по порядку их расположения данные автоматически загрузятся из базы данных в соответствии с выбранными значениями. Например, при выборе института, в форме будет автоматически заполнено значение «Директор (декан)», а также будет загружен список кафедр данного факультета (института). Пример представлен на рисунке 3.8.

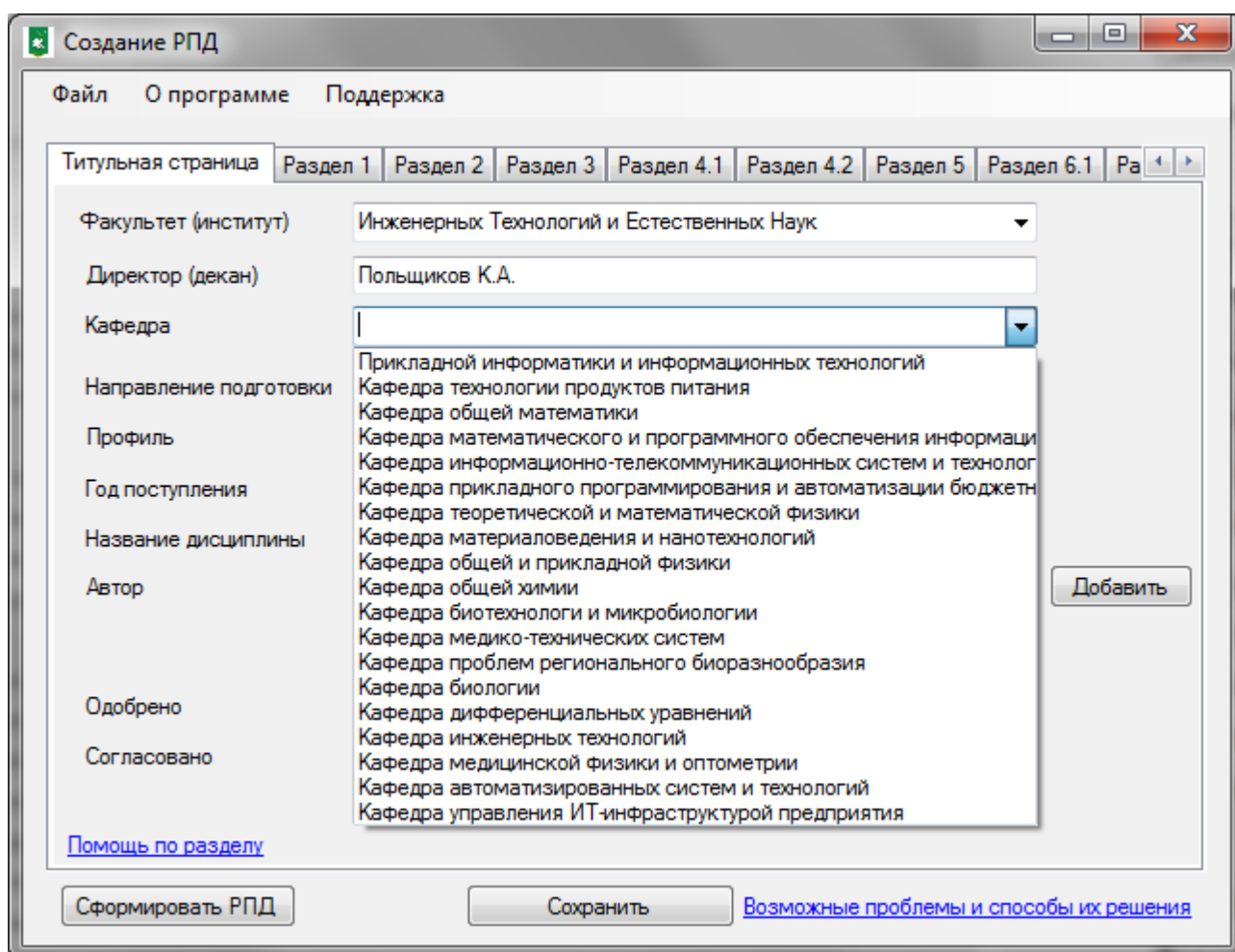


Рисунок 3.8 – Пример загрузки данных

Аналогично при выборе кафедры, загружаются направления подготовки; при выборе направления подготовки загружается профиль, доступные года поступления и доступные дисциплины.

Для автоматизированной подставки информации об авторе рабочей программы необходимо нажать кнопку «Добавить» рядом с соответствующим полем. После чего откроется форма выбора автора, в которой содержится список преподавателей. Также в данной форме реализован поиск по фамилии, при вводе значения в строку поиска данные будут фильтроваться в соответствии с вводимым параметром. Общий вид данной формы представлен на рисунке 3.9.

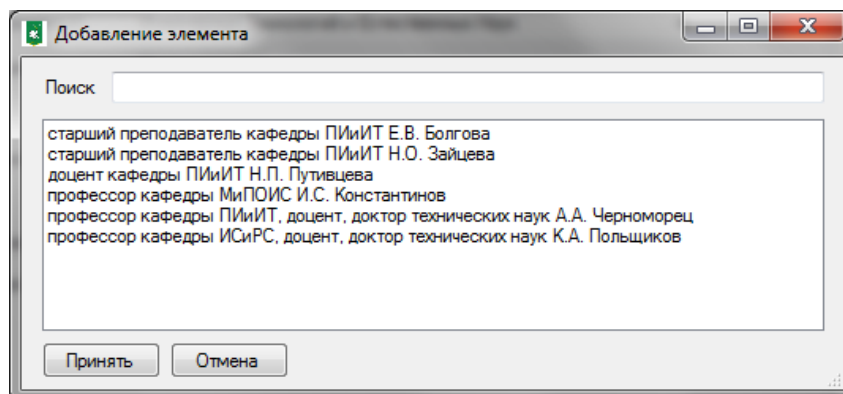


Рисунок 3.9 – Добавление автора

Пример заполненной вкладки «Титульная страница» представлен на рисунке 3.10.

| | | |
|------------------------|--|----------|
| Факультет (институт) | Инженерных Технологий и Естественных Наук | ▼ |
| Директор (декан) | Польщиков К.А. | |
| Кафедра | Прикладной информатики и информационных технологий | ▼ |
| Направление подготовки | 09.03.03 Прикладная информатика | ▼ |
| Профиль | Администрирование информационных систем в экономике | |
| Год поступления | 2014 | ▼ |
| Название дисциплины | Методы обработки и хранения цифровых изображений | ▼ |
| Автор | старший преподаватель кафедры ПИИИТ Е.В. Болгова, профессор кафедры ПИИИТ, доцент, доктор технических наук А.А. Черноморец | Добавить |
| Одобрено | Прикладной информатики и информационных технологий | ▼ |
| Согласовано | Прикладной информатики и информационных технологий | ▼ |

Рисунок 3.10 – Результат заполнения титульной страницы

Раздел 1 макета рабочей программы представляет собой таблицу, содержащую информацию об осваиваемых компетенциях в рамках дисциплины и о требованиях к результату освоения. В разрабатываемой программе заполнение данного раздела также реализовано с помощью таблиц, общий вид вкладки для заполнения раздела 1 представлен на рисунке 3.11.

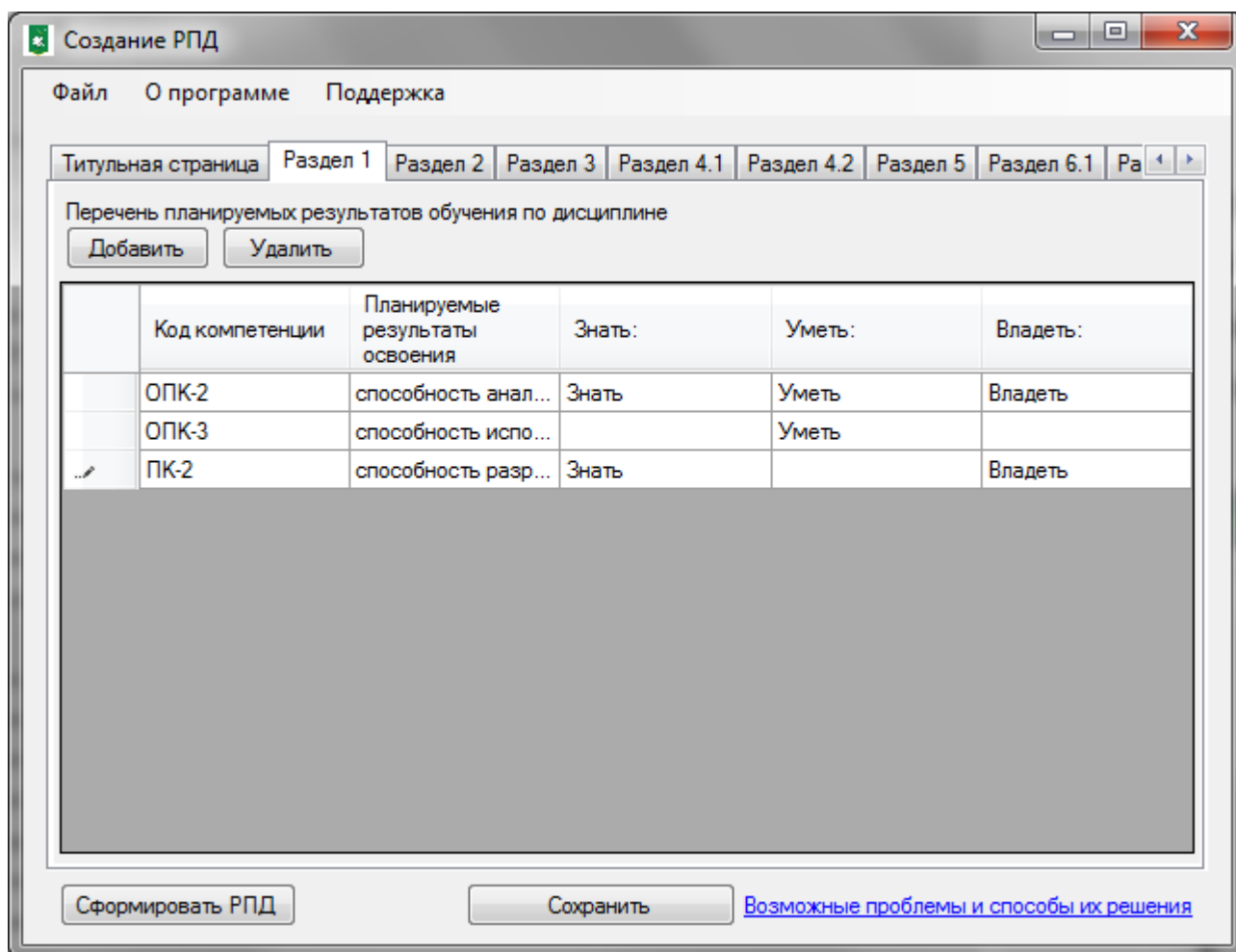


Рисунок 3.11 – Интерфейс заполнения раздела 1

Для большей наглядности информации реализована форма для укрупненного представления информации о каждой компетенции в которую необходимо вводить данные согласно правилам разработки рабочих программ дисциплин. Данная форма вызывается двойным нажатием на строчку, соответствующую заполняемой компетенции. Общий вид данной формы представлен на рисунке 3.12.

| | |
|------------------------|--|
| Код компетенции | ОПК-2 |
| Планируемые результаты | способность анализировать социально-экономические задачи и процессы с применением методов системного анализа и математического моделирования |
| Знать | Знать |
| Уметь | Уметь |
| Владеть | Владеть |

Сохранить Отмена

Рисунок 3.12 – Форма описания компетенций

Раздел 2 макета рабочей программы разделен на 3 подраздела: место дисциплины в структуре образовательной программы; требования к предварительной подготовке; дисциплины для которых необходимо освоение данной дисциплины. Заполнение данного раздела в программе реализовано с помощью текстовых полей. Место дисциплины в структуре образовательной программы заполняется автоматически при выборе дисциплины на титульной странице, остальные поля необходимо заполнить вручную. Общий вид данной вкладки представлен на рисунке 3.13.

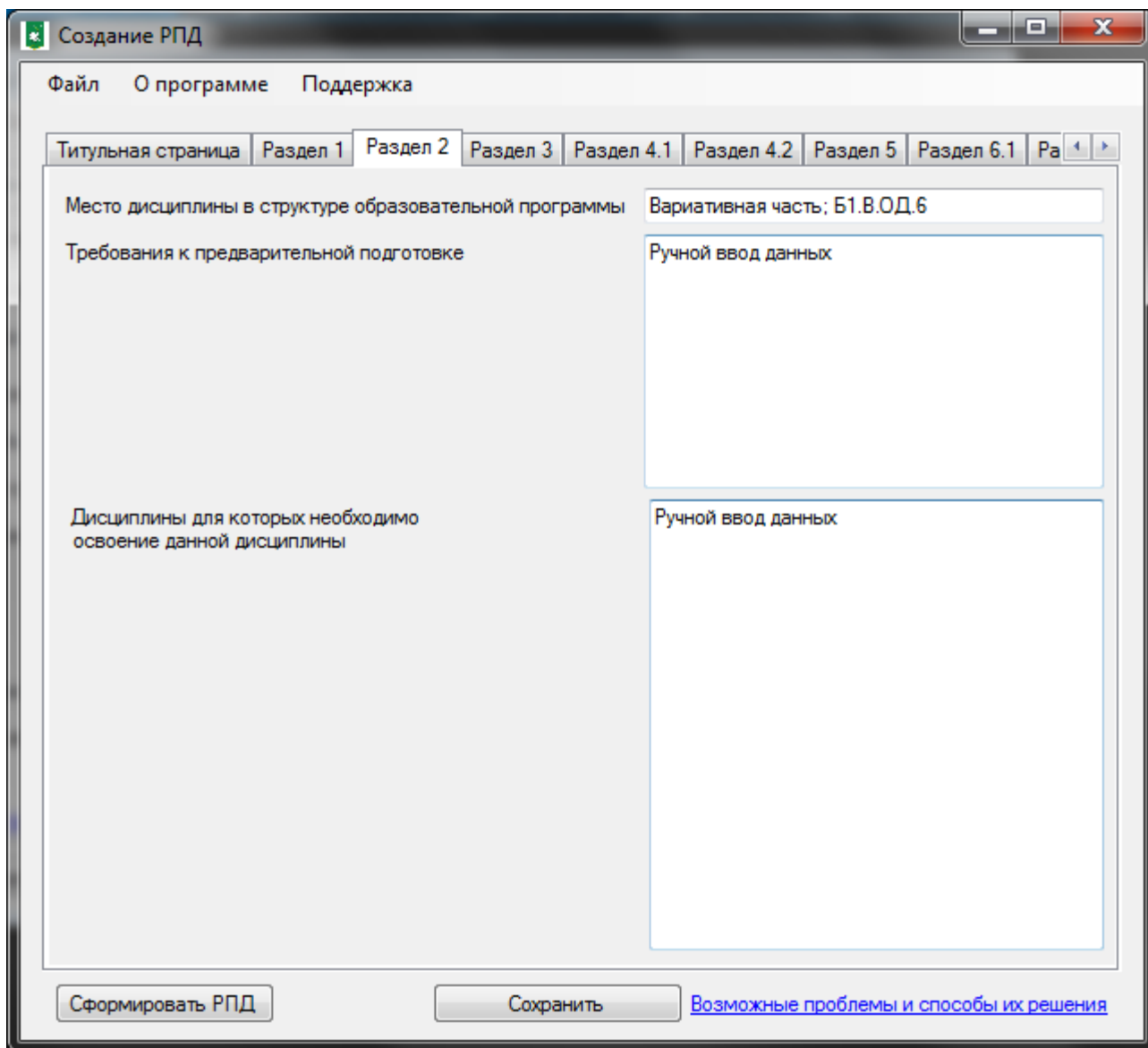


Рисунок 3.13 – Заполнение раздела 2

Раздел 3 макета рабочей программы дисциплины представляет собой таблицу с информацией о часах на каждый вид деятельности (лекции, лабораторные занятия, самостоятельная работа и т.д.) по каждой форме обучения. В программе заполнение данного раздела реализовано с помощью таблиц. Информация о каждой форме обучения заполняется в отдельной таблице на отдельной вкладке. Данные о часах загружаются из базы данных при выборе дисциплины на титульной странице. Общий вид интерфейса представлен на рисунке 3.14.

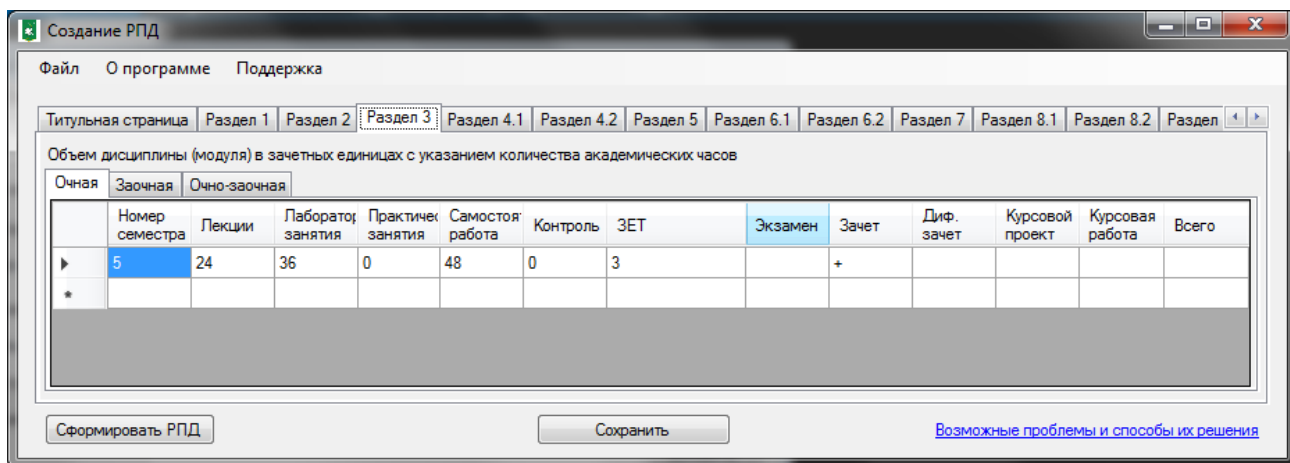


Рисунок 3.14 – Заполнение раздела 3

Раздел 4.1 макета рабочей программы представляет собой таблицу с информацией о разделах (темах) осваиваемой дисциплины и о часах на каждый вид деятельности по данному разделу (теме). Для заполнения информации необходимо ввести количество разделов (с учётом тем (если есть)) и нажать кнопку «Создать», после чего заполнить данные о разделах для очной формы обучения (на вкладке «Очная») и нажать кнопку «Применить». Далее необходимо заполнить информацию о часах для каждой формы обучения и нажать «Рассчитать». Пример заполненной информации представлен на рисунке 3.15.

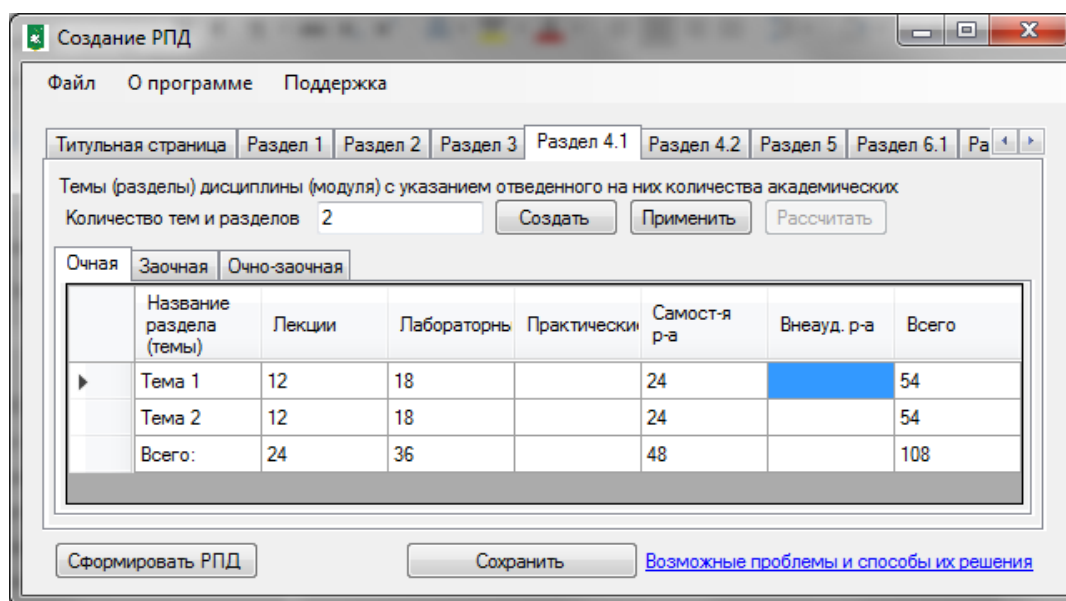


Рисунок 3.15 – Заполнение раздела 4.1

В случае несовпадения данных о часах в данном разделе с информацией в разделе 3 несовпадающие ячейки будут выделены красным цветом и в скобках будет указано верное значение. Пример представлен на рисунке 3.16.

| Очная | Заочная | Очно-заочная |
|-------------------------|---------|--------------|
| Название раздела (темы) | Лекции | Лабораторные |
| Тема 1 | 32 | 18 |
| Тема 2 | 12 | 18 |
| Всего: | 44 (24) | 36 |

Рисунок 3.16 – Пример информации при несовпадении данных

Раздел 4.2 макета рабочей программы представляет собой таблицу с информацией о содержании тем и разделов осваиваемой дисциплины, а именно: информация о содержании каждой темы и информация о лабораторных и практических занятиях с указанием часов для каждой формы обучения. В программе заполнение данного раздела реализовано с помощью таблицы. После заполнения раздела 4.1 информация будет перенесена в данный раздел, а именно: будет перенесен список тем и разделов, а также если заполнялась информация о часах для лабораторных и(или) практических занятий будут автоматически созданы соответствующих строки в таблице. Далее при помощи кнопок «Вставить перед» и «Вставить после» можно добавить строку относительно выделенной ячейки, а при помощи кнопки «Удалить» удалить выделенную строку. Пример заполнения интерфейса данного раздела представлен на рисунке 3.17.

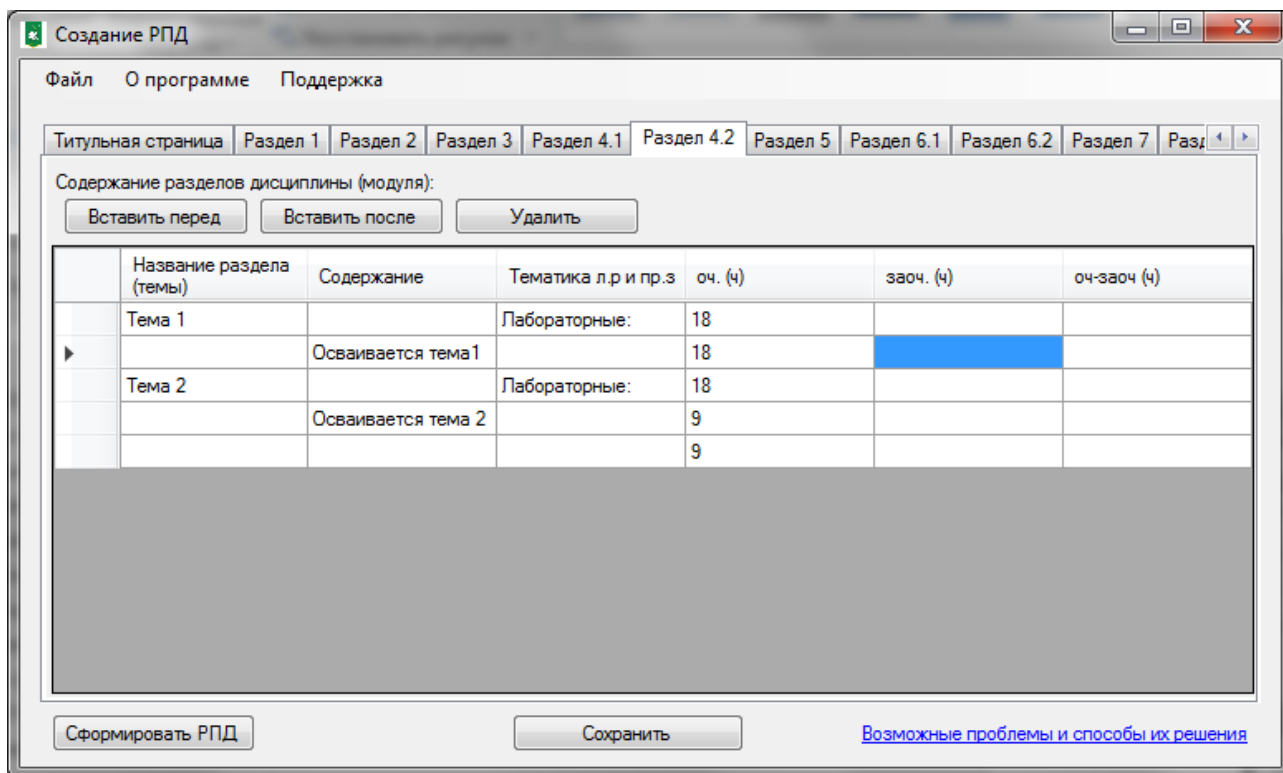


Рисунок 3.17 – Пример заполнения раздела 4.2

Разделы 5-8.3, 9.3 (за исключением 9.3.1), 9.4-12 представляют собой текстовые разделы для ручного заполнения. Заполнение разделов данного типа реализовано в программе с помощью текстовых полей, а также предусмотрена возможность выбора файла с данными одного из следующих форматов: .doc, .docx, .rtf. Для выбора файла необходимо нажать кнопку «Обзор» на вкладке соответствующего раздела, после чего откроется стандартное окно выбора файла. После выбора файла его содержимое (без форматирования) будет перенесено в текстовое поле. В макет же содержимое переносится со всеми элементами (таблицы, маркированные и нумерованные списки, рисунки и пр.) и объединяется с установленными макетом правилами форматирования. Пример заполнения такого раздела при помощи выбора файла представлен на рисунке 3.18.

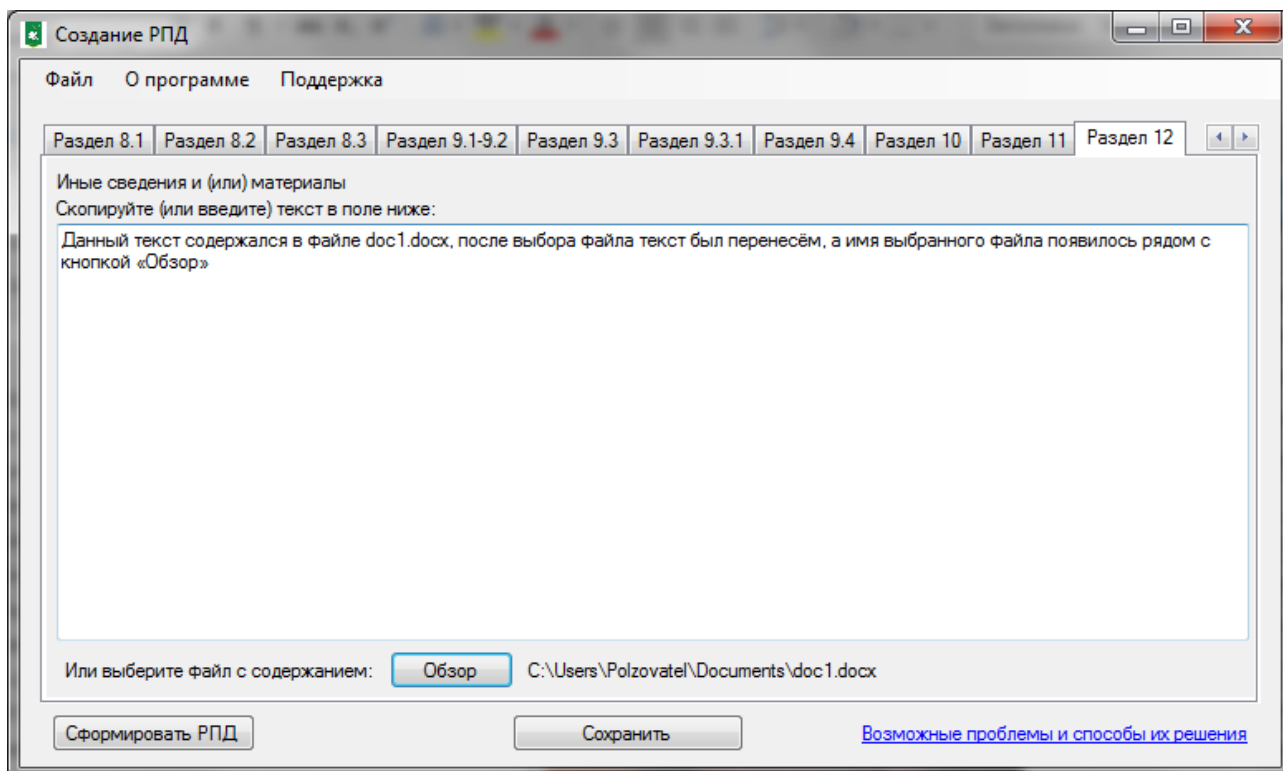


Рисунок 3.18 – Заполнение текстового раздела

Заполнение разделов 9.1-9.2 макета РПД в программе было объединено, так как раздел 9.1 представляет собой таблицу, содержащую информацию об осваиваемых компетенциях и их уровнях, а раздел 9.2 приставляет собой таблицу, в которой расписаны требования к результатам освоения данных компетенций на различную оценку (в случае экзамена или дифференцированного зачёта) или на зачёт. Начальные данные (названия компетенций и их уровни) выгружаются из базы данных при выборе дисциплины на титульной странице. Если перечень компетенций был изменен в разделе 1 данные также будут обновлены и здесь, но уровень компетенции в данном случае необходимо указать вручную. Общий вид формы заполнения данного раздела представлен на рисунке 3.19.

| | Код комп-и | ТН | неуд./незач | уд./зач | хор./зач | отл./зач |
|---|------------|----------|-------------|-------------|-------------|-------------|
| ▶ | ОПК-2 | Знать: | Ручной ввод | Ручной ввод | Ручной ввод | Ручной ввод |
| | 2 | Уметь: | Ручной ввод | Ручной ввод | Ручной ввод | Ручной ввод |
| | | Владеть: | Ручной ввод | Ручной ввод | Ручной ввод | Ручной ввод |
| | ОПК-3 | Знать: | | | | |
| | 2 | Уметь: | | | | |
| | | Владеть: | | | | |
| | ПК-2 | Знать: | | | | |
| | 2 | Уметь: | | | | |
| | | Владеть: | | | | |

Рисунок 3.19 – Форма заполнения разделов 9.1-9.2

Для более понятного и наглядного заполнения таблицы была создана форма, обеспечивающая отдельное заполнение информации о каждой компетенции. На данной форме информацию о каждом результате на каждую оценку необходимо ввести в соответствующее поле, после заполнения всех полей необходимо нажать «Сохранить» и введенная информация будет перенесена в основную таблицу. Общий вид данной формы представлен на рисунке 3.20.

| Код компетенции | ОПК-2 | | | |
|-----------------|------------------|--------------|----------------|-----------------|
| | Неудовл./незачт. | Удовл./зачт. | Хорошо/зачтено | Отлично/зачтено |
| Знать: | Ручной ввод | Ручной ввод | Ручной ввод | Ручной ввод |
| Уметь: | Ручной ввод | Ручной ввод | Ручной ввод | Ручной ввод |
| Владеть: | Ручной ввод | Ручной ввод | Ручной ввод | Ручной ввод |

Рисунок 3.20 – Общий вид формы

Раздел 9.3.1 макета рабочей программы представляет собой две таблицы, в которых содержится информация о начисляемых баллах за определенные виды деятельности в рамках дисциплины и о необходимых суммах баллов на каждую из возможных оценок. Заполнение данного раздела в программе реализовано с помощью таблицы для указания баллов за виды деятельности и с помощью полей для указания диапазонов сумм баллов на каждую оценку.

При заполнении баллов по видам деятельности производится контроль суммы и в случае, если сумма не равна 100, ячейка подсвечивается красным. Общий вид данной таблицы представлен на рисунке 3.21.

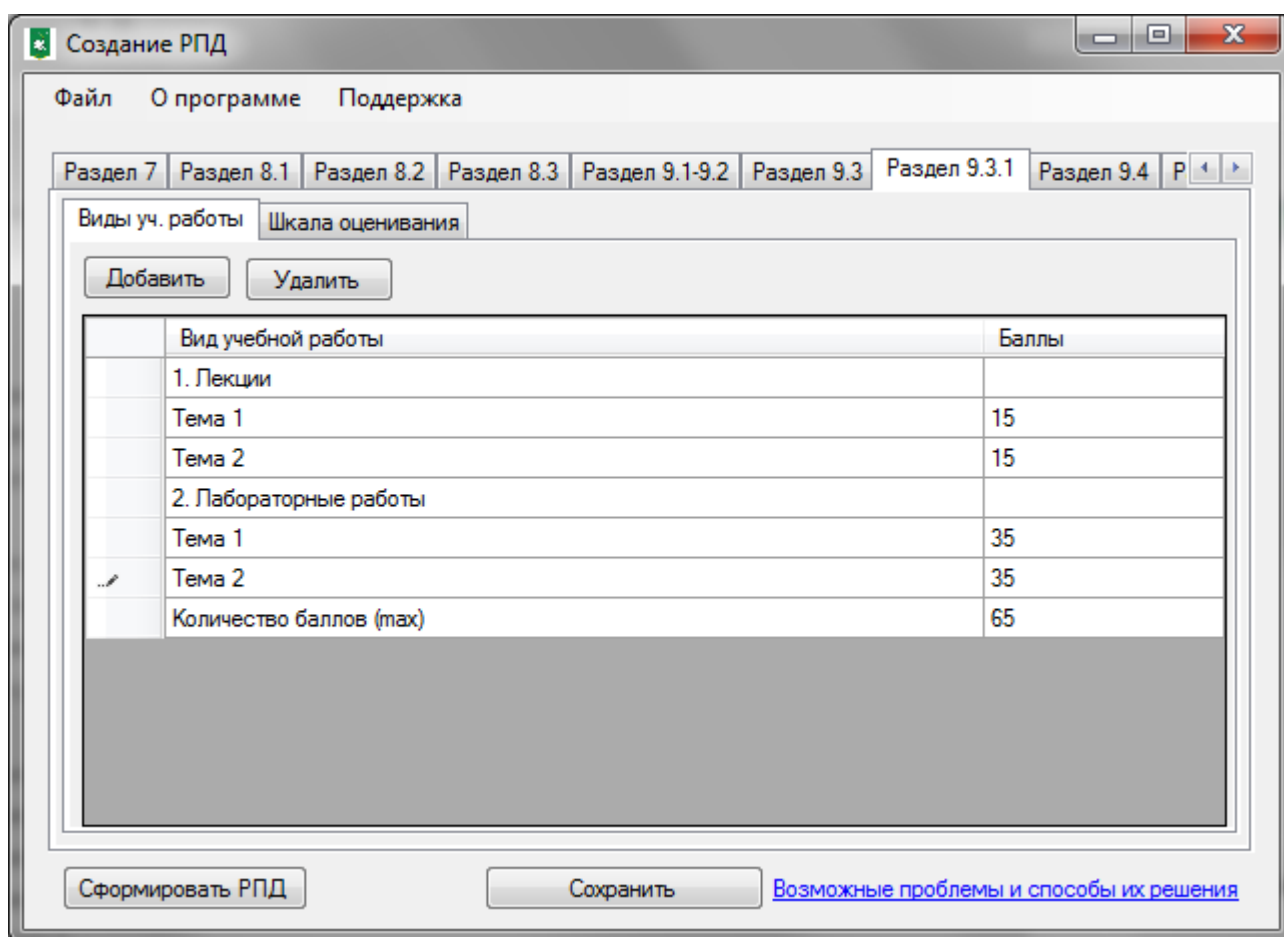


Рисунок 3.21 – Пример заполнения раздела

Шкала оценивания изначально заполнена по умолчанию в соответствии с правилами университета, но она может быть отредактирована. При изменении шкалы и нарушении правильности указания диапазонов, ошибочные ячейки

подсвечиваются красным. Пример заполнения шкалы представлен на рисунке 3.22.

| Неуд. | Удовл. | Хорошо | Отлично |
|--------|---------|---------|----------|
| 0 - 50 | 51 - 70 | 71 - 89 | 90 - 100 |

Рисунок 3.22 – Пример заполнения шкалы оценивания

После заполнения всех разделов следует нажать кнопку «Сформировать РПД». Будет запущен процесс формирования рабочей программы, о прогрессе которого будет сообщать окно со шкалой. После завершения процесса формирования откроется программа Microsoft Office Word с заполненным макетом, а в программе появится сообщение с предложением сохранить созданную РПД на сервер. В случае, если все разделы заполнены верно, можно нажать в данном окне «Да» и произойдет загрузка созданных файлов в центральное хранилище, в случае же возникновения ошибок, следует нажать «Нет» и исправить недоработки.

Также в программе предусмотрена функция сохранения сеанса работы. Для того чтобы продолжить начатый сеанс в дальнейшем, необходимо нажать кнопку «Сохранить» внизу формы. Будет открыт стандартный диалог сохранения файла, в котором необходимо выбрать путь и указать наименования файла. После выполнения данных действий все данные формы будут сохранены в формате .xml и в дальнейшем с ними можно будет продолжить работу (при помощи меню Файл -> Открыть).

Пример титульного листа, созданного с помощью разрабатываемой системы представлен в приложении Е.

3.4 Обоснование экономической эффективности проекта

В данном разделе выпускной квалификационной работы рассматриваются экономические аспекты проектирования и внедрения информационной системы составления макетов рабочих программ дисциплин в НИУ «БелГУ», производится анализ этапов разработки информационной системы, оцениваются затраты на разработку.

При разработке любой информационной системы следует придерживаться заранее созданного календарного плана работ. В данном случае разработка системы состоит из следующих этапов:

- формирование пользовательских требований;
- разработка и утверждение технического задания;
- разработка проектных решений;
- разработка документации на АИС;
- разработка документации на поставку комплектующих изделия;
- адаптация АИС и её частей;
- подготовка объекта автоматизации;
- подготовка персонала;
- комплектация АИС;
- пусконаладочные работы;
- проведение предварительных испытаний;
- проведение приемочных испытаний.

По экспертным оценкам данный проект может быть выполнен одним исполнителем за 166 дней. Начало проекта было установлено на 01.11.2017, завершение 21.06.2018, таким образом срок выполнения проекта 191 день без учета выходных дней. В таблице 3.1 представлен календарный план проекта.

Таблица 3.1 – Календарный план проекта

| Название задачи | Начало | Окончание |
|---|-------------|-------------|
| Обследование объекта и обоснование необходимости создания АИС | Ср 01.11.17 | Вт 21.11.17 |
| Формирования пользовательских требований | Вт 21.11.17 | Пт 01.12.17 |
| Разработка и утверждение технического задания | Пн 04.12.17 | Пт 15.12.17 |
| Разработка проектных решений | Сб 16.12.17 | Ср 31.01.18 |
| Разработка документации на АИС | Ср 31.01.18 | Вт 27.02.18 |
| Разработка документации на поставку комплектующих изделия | Вт 27.02.18 | Пн 12.03.18 |
| Адаптация АИС и её частей | Пн 12.03.18 | Чт 29.03.18 |
| Подготовка объекта автоматизации | Чт 29.03.18 | Сб 14.04.18 |
| Подготовка персонала | Сб 14.04.18 | Вт 08.05.18 |
| Комплектация АИС | Вт 08.05.18 | Ср 16.05.18 |
| Пусконаладочные работы | Ср 16.05.18 | Пн 28.05.18 |
| Проведение предварительных испытаний | Вт 29.05.18 | Сб 09.06.18 |
| Проведение приемочных испытаний | Сб 09.06.18 | Чт 21.06.18 |
| ИТОГО | Ср 01.11.17 | Чт 21.06.18 |

На любом этапе выполнения проекта необходимы определенные ресурсы. Ресурсы разделяют на две группы: трудовые и материальные.

Трудовые ресурсы – это возобновляемые ресурсы компании, которые включают людей, машины и оборудование, необходимые для исполнения проекта.

Материальные ресурсы используются для моделирования материалов, необходимых для создания результата проекта.

Трудовые ресурсы также отличаются от материальных тем, что затраты на трудовые ресурсы прямо пропорциональны времени их использования, а затраты на материальные ресурсы осуществляются единоразово при их приобретении.

В таблице 3.2 представлен лист ресурсов данного проекта. В материальных ресурсах стандартная ставка указана за одну стандартную единицу.

Таблица 3.2 – Лист ресурсов проекта

| Название ресурса | Тип ресурса | Стандартная ставка |
|-------------------------|--------------|--------------------|
| Разработчики | Трудовой | 40,00р./ч |
| Компьютеры | | 8,00р./ч |
| Данные учебных планов | | 2,00р./ч |
| Данные карт компетенций | | 2,00р./ч |
| Данные ФГОС | | 2,00р./ч |
| Проект интерфейса | | 10,00р./ч |
| База данных | | 8,00р./ч |
| Преподаватели | | 80,00р./ч |
| CD диск | Материальный | 45,00р. |
| Переплёт | | 150,00р. |
| Печать | | 2,00р. |

Чтобы избежать излишних экономических затрат необходимо заранее оценить распределение ресурсов по этапам проекта. В таблице 3.3 представлен результат распределения ресурсов. Все трудовые ресурсы на каждом этапе использовались с коэффициентом 1, что означает затраты полной фиксированной ставки в единицу времени. На этапе комплектация АИС использовались CD диск 1 шт, переплет 1 шт. и печать в количестве 27 штук (по объему руководства пользователя системы).

Таблица 3.3 – Распределение ресурсов

| Название задачи | Ресурсы |
|---|--|
| 1 | 2 |
| Обследование объекта и обоснование необходимости создания АИС | Преподаватели, разработчики |
| Формирования пользовательских требований | |
| Разработка и утверждение технического задания | |
| Разработка проектных решений | Разработчики, база данных, данные карт компетенций, данные учебных планов, данные ФГОС, компьютеры проект интерфейса |
| Разработка документации на АИС | Разработчики, компьютеры |
| Разработка документации на поставку комплектующих изделия | Разработчики, компьютеры |

Продолжение таблицы 3.3

| 1 | 2 |
|--------------------------------------|--|
| Адаптация АИС и её частей | Разработчики, база данных, компьютеры |
| Подготовка объекта автоматизации | Разработчики, компьютеры |
| Подготовка персонала | Разработчики, компьютеры, преподаватели, проект интерфейса |
| Комплектация АИС | Разработчики, CD диск, печать, переплет |
| Пусконаладочные работы | База данных, разработчики, компьютеры, проект интерфейса |
| Проведение предварительных испытаний | База данных, компьютеры, преподаватели |
| Проведение приемочных испытаний | |

В таблице 3.4 представлен результат стоимостной оценки проекта. Общие затраты составили 108 533 рубля. Данные затраты включают в себя как затраты на материальные ресурсы, так и на трудовые.

Таблица 3.4 – Результат оценки затрат

| Название задачи | Общие затраты |
|---|---------------------|
| Обследование объекта и обоснование необходимости создания АИС | 11 256,00р. |
| Формирование пользовательских требований | 2 592,00р. |
| Разработка и утверждение технического задания | 3 200,00р. |
| Разработка проектных решений | 24 888,00р. |
| Разработка документации на АИС | 8 448,00р. |
| Разработка документации на поставку комплектующих изделия | 3 840,00р. |
| Адаптация АИС и её частей | 7 220,00р. |
| Подготовка объекта автоматизации | 4 992,00р. |
| Подготовка персонала | 18 992,00р. |
| Пусконаладочные работы | 5 780,00р. |
| Комплектация АИС | 1 965,00р. |
| Проведение предварительных испытаний | 7 680,00р. |
| Проведение приемочных испытаний | 7 680,00р. |
| ИТОГО: | 108 533,00р. |

Таким образом общая длительность проекта составляет 191 рабочий день, а общие затраты на его реализацию 108 533 рубля.

Для оценки временной эффективности системы было проведено сравнение затрачиваемого времени на заполнения разделов макета рабочей программы дисциплины без использования разработанной системы и с ее использованием. Для сравнения были выбраны те разделы, заполнение которых в системе автоматизировано, а именно: титульный лист, разделы 1-4.2, 9.1-9.3.1. Результаты сравнения представлены в таблице 3.5.

Таблица 3.5 – Результаты сравнения времени

| Раздел | Без использования системы (мин:сек) | Время с использованием системы (мин:сек) |
|----------------|-------------------------------------|--|
| Титульный лист | 3:11 | 1:47 |
| Раздел 1 | 6:33 | 0:00 (был заполнен автоматически) |
| Раздел 2 | 4:12 | 2:44 |
| Раздел 3 | 8:48 | 3:31 |
| Раздел 4.1 | 9:24 | 4:47 |
| Раздел 9.1 | 1:12 | 0:00 (был заполнен автоматически) |
| Раздел 9.2 | 5:33 | 3:12 |
| Раздел 9.3.1 | 4:17 | 2:21 |

Таким образом время, затраченное на заполнение указанных ранее разделов рабочей программы дисциплины без использования системы, составило 42 минуты 53 секунды, а с использованием информационной системы 18 минут 22 секунды на заполнение формы программы и дополнительно 19 секунд на формирование рабочей программы дисциплины, то есть в совокупности время составило 18 минут 41 секунду. При оценке эффективности от внедрения данного программного средства трудно говорить о каких-либо экономических показателях, поскольку разработка предназначена для внутреннего пользования. Ожидаемую прибыль трудно вычислить ввиду того, что внедрение программы служит исключительно для упрощения процесса формирования рабочих программ дисциплин.

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы были сформированы требования к информационной системе составления рабочих программ дисциплин. Подробно изучив процесс разработки рабочих программ в НИУ «БелГУ» для разработки информационной системы был выбран следующий ряд программных продуктов: СУБД MySQL, среда программирования Visual Studio 2017 и высокоуровневый язык программирования C#.

Исходя из выявленных требований, была создана информационная система составления макетов рабочих программ дисциплин для НИУ «БелГУ».

Данный программный продукт разработан непосредственно под организационные особенности НИУ «БелГУ». Данная система имеет простой пользовательский интерфейс и не требует какой-либо специфической подготовки и знаний в области работы с базами данных или глубоких знаний в работе с офисными программами.

Реализованный проект позволяет значительно сократить время, необходимое на разработку рабочих программ дисциплин за счет автоматизированного заполнения многих разделов, что было подтверждено в результате проведения исследования, которое показало, что использование полученной системы позволяет сократить время, затрачиваемое на заполнение некоторых разделов более чем в два раза. Также данный проект позволит избавиться от ошибок при заполнении макета рабочей программы, за счет того, что все форматирование текста выполняется программно по строго заданному алгоритму.

Разработанная информационная система внедрена в практическую деятельность НИУ «БелГУ», о чем имеется акт о внедрении.

В ходе выполнения данной работы были решены следующие задачи:

– проведен анализ и моделирование существующего процесса разработки рабочих программ дисциплин;

- проведено моделирование алгоритмов разработки рабочих программ дисциплин «как должно быть»;
- осуществлен выбор инструментальных средств разработки автоматизированной информационной системы;
- спроектирована и разработана база данных информационной системы;
- разработана информационная система составления макетов рабочих программ дисциплин;
- разработано мультимедийное руководство пользователя;
- проведен анализ эффективности внедрения автоматизированной информационной системы;
- выполнено тестирование автоматизированной системы создания макетов рабочих программ дисциплин.

Все поставленные задачи в рамках выполнения выпускной квалификационной работы выполнены, поставленная цель достигнута в полном объеме.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Маторин, С.И. Теория систем и системный анализ. Учебное пособие [Текст] / С.И. Маторин, О.А. Зимовец – Белгород: Белгородский государственный университет, 2012 –288 с.

2 MySQL (Введение) [Электронный ресурс] / Режим доступа: http://www.compdoc.ru/bd/mysql/mysql_introduction/ (дата обращения 21.10.2017 г.)

3 СУБД MySQL [Электронный ресурс] / Режим доступа: <https://www.methodlab.ru/technology/mysql.shtml> (дата обращения 21.10.2017 г.)

4 Технологические процессы обработки информации в информационных технологиях [Электронный ресурс] / Режим доступа: <http://www.intuit.ru/studies/courses/3609/851/lecture/31650?page=7> (дата обращения 21.10.2017 г.)

5 Кривчиков, В.С. О разработке автоматизированной информационной системы составления рабочих программ дисциплин для НИУ «БелГУ» [Текст] / В.С. Кривчиков // Естественнонаучные, инженерные и экономические исследования в технике, промышленности, медицине и сельском хозяйстве: материалы I Молодежной научно-практической конференции с международным участием / Белгород, 2017. – 693 с.

6 Кривчиков, В.С. О необходимости разработки автоматизированной информационной системы составления макетов рабочих программ дисциплин для НИУ «БелГУ» [Текст] // Вестник СНО. – 2017. – № 7. – с. 32-34.

7 Методология DFD [Электронный ресурс] / Режим доступа: https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3 (дата обращения 21.10.2017 г.)

8 Кузнецов, С.Д. Базы данных / С.Д. Кузнецов– Москва: Academia, 2012 –496 с.

9 Албахари, Д. С# 6.0. Справочник. Полное описание языка / Д. Албахари – Москва: Вильямс, 2017 – 1040 с.

10 Windows Forms | msdn.microsoft.com [Электронный ресурс] / Режим доступа: <https://msdn.microsoft.com/en-us/library/dd30h2yb%28v=vs.110%29> (дата обращения 21.10.2017 г.)

11 Чистов, Д.В. Проектирование информационных систем. Учебник и практикум. [Текст] / Д.В. Чистов – Москва: Юрайт, 2016. – 258 с.

12 Основы IDEF3 [Электронный ресурс] / Режим доступа: <http://www.cfin.ru/vernikov/idef/idef3.shtml> (дата обращения 21.10.2017 г.)

13 Коваленко, В.В. Проектирование информационных систем [Текст] / В.В. Коваленко – Москва: Форум, 2014 – 320 с.

14 Мандел, Т. Разработка пользовательского интерфейса [Текст] / Т. Мандел – Москва: ДМК, 2010 – 209 с.

15 Исаев, Г.Н. Проектирование информационных систем. Учебное пособие [Текст] / Исаев Г.Н. – Москва: Омега-Л, 2012 – 432 с.

16 Нормальные формы [Электронный ресурс] / Режим доступа: <http://www.studfiles.ru/preview/5999054/page:11/> (дата обращения 21.10.2017 г.)

17 Гольцман, В. MySQL 5.0 [Текст] / В. Гольцман – СПб: Питер, 2010 – 2530 с.

18 Самоучитель по языку SQL (SQLDML) [Электронный ресурс]. – Режим доступа: <http://www.sql-ex.ru/help> (дата обращения 21.10.2017 г.)

19 Windows 7 – отличия версий, различия версий Windows 7 [Электронный ресурс] – Режим доступа: <https://petrenco.com/windows.php?txt=108> (дата обращения 21.10.2017 г.)

20 Энциклопедия языков программирования. C# [Электронный ресурс] / Режим доступа: <http://progopedia.ru/language/csharp/> (дата обращения 21.10.2017 г.)

21 Белгородский государственный национальный исследовательский университет. Университет сегодня [Электронный ресурс] – Режим доступа: <https://www.bsu.edu.ru/bsu/info/today/index.php> (дата обращения 21.10.2017 г.)

22 Институт инженерных технологий и естественных наук – Кафедра прикладной информатики и информационных технологий [Электронный ресурс]

– Режим доступа: <http://iten.bsu.edu.ru/iten/info/dir-mi/kaf-piit/> (дата обращения 21.10.2017 г.)

23 Практическое руководство. Программная вставка текста в документы Word [Электронный ресурс] – Режим доступа: <https://msdn.microsoft.com/ru-ru/library/6b9478cs.aspx> (дата обращения 21.10.2017 г.)

24 Федеральные государственные образовательные стандарты [Электронный ресурс] – Режим доступа: <https://минобрнауки.рф/документы/336> (дата обращения 21.10.2017 г.)

25 Белгородский государственный национальный исследовательский университет. Документы по организации учебной работы [Электронный ресурс] – Режим доступа: <https://www.bsu.edu.ru/bsu/resource/officialdocs> (дата обращения 21.10.2017 г.)

26 Формы организации учебного процесса [Электронный ресурс] – Режим доступа: <https://studfiles.net/preview/2957337/page:51> (дата обращения 21.10.2017)

27 Справочная информация: Федеральные государственные образовательные стандарты [Электронный ресурс] – Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_142304/ (дата обращения 21.10.2017)

28 СУБД MySQL [Электронный ресурс] – Режим доступа: <http://www.internet-technologies.ru/articles/mysql-eto-prosto.html> (дата обращения 21.10.2017)

29 Архитектура клиент-сервер [Электронный ресурс] – Режим доступа <https://studfiles.net/preview/1882615/> (дата обращения 21.10.2017)

30 Клиент-серверная архитектура [Электронный ресурс] – Режим доступа https://studopedia.ru/3_194267_arhitektura-klient--server.html (дата обращения 21.10.2017)

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Техническое задание на разработку системы

Техническое задание на разработку информационной системы составления макетов рабочих программ дисциплин для НИУ «БелГУ»

1. Введение

В настоящее время преподавателям и сотрудникам ВУЗа значительную часть времени приходится тратить на заполнение и составление различных документов. Это могут быть личные дела студентов, экзаменационные и тестовые задания, учебные планы, зачетные ведомости и т.д.

Одним из документов, заполнение которого является весьма сложным и долгим процессом, является рабочая программа дисциплины (РПД).

Рабочая программа дисциплины (РПД) – это документ, определяющий на основе Федерального государственного образовательного стандарта (ФГОС) высшего профессионального образования (ВПО) по направлению (специальности) содержание дисциплины, вырабатываемые компетенции, составные части учебного процесса по дисциплине, учебно-методические приемы, используемые при преподавании, взаимосвязь данной дисциплины и других дисциплин учебного плана, формы и методы контроля знаний обучающихся, рекомендуемую литературу.

Настоящее техническое задание распространяется на разработку программного обеспечения информационной системы составления макетов рабочих программ дисциплин для НИУ «БелГУ».

2. Основание для разработки

Основанием для разработки является задание в рамках курса "Программная инженерия".

3. Назначение разработки

Информационная система составления макетов РПД предназначена для упрощения, ускорения, а также стандартизации процесса разработки рабочих программ дисциплин.

4. Требования к программному изделию

4.1. Требования к функциональным характеристикам

Система должна обеспечивать создание и изменение информации о рабочих программах дисциплин, а именно вносить или изменять в центральном хранилище данных (организованном в виде удаленной базы данных) следующую информацию: наименование дисциплины, год поступления, направление подготовки для которого создавалась РПД.

Система должна обеспечивать создание непосредственно рабочих программ дисциплин. Создание РПД должно производиться с помощью специальной формы, которая должна быть построена в соответствии с расположением разделов в макете РПД.

При создании РПД система должна в автоматическом режиме загружать из базы данных информацию из учебного плана, карты компетенций и ФГОС направления подготовки для которого создаётся рабочая программа дисциплины, а именно должно быть организовано автоматическое заполнение следующих значений:

- количество часов на каждый вид деятельности;
- место дисциплины в структуре образовательной программы;
- коды компетенций и их содержание.

Также система должна обеспечивать заполнение разделов РПД, не связанных с внешними документами. Заполнение таких разделов должно обеспечиваться ручным вводом данных в форму, либо при помощи импорта файлов txt, docx, doc.

Помимо указанного выше *система* при создании непосредственно РПД должна производить автоматический расчет и контроль вводимых

пользователем значений при заполнении табличных частей макета РПД, а именно:

- контролировать соответствие количества часов в таблице 4.1 макета РПД количеству часов в учебном плане;
- контролировать соответствие количества часов в таблице 4.2 макета РПД количеству часов в учебном плане;
- контролировать соответствие суммы баллов в таблице 9.3.1 максимально возможному баллу;
- автоматически рассчитывать необходимые суммы в таблицах.

В системе должна быть реализована возможность сохранения пользователем введенной информации для продолжения работы с ней. Следовательно, должно быть реализовано автоматическое заполнение полей формы данными загружаемого пользователем сеанса.

При создании РПД *системой* должен использоваться макет РПД утвержденный Приказом от 05.05.2015 №310-ОД «Об утверждении унифицированных форм учебно-методической документации».

Выходные данные должны быть организованы в виде заполненного макета РПД в формате .docx с предоставлением пользователю выбора пути сохранения полученного файла.

Также должна быть обеспечена возможность редактирования ранее созданных РПД средствами *системы*. Для этого должна быть реализована возможность выбора ранее созданной РПД и автоматическое заполнение полей формы данными из выбранного файла.

Система должна синхронизировать данные с существующими в НИУ «БелГУ» информационными системами, а именно синхронизировать данные об учебных планах и картах компетенций.

Администрирование базы данных системы должно осуществляться пользователем обладающим административными правами при помощи утилиты phpMyAdmin.

Редактирование информации из ФГОС направлений подготовки должно осуществляться через утилиту phpMyAdmin.

4.2. Требования к надежности

Система должна:

- проводить контроль вводимой информации;
- блокировать некорректные действия пользователя;
- обеспечивать целостность данных.

4.3. Условия эксплуатации

Использовать систему будут пользователи средней квалификации. Интерфейс системы должен быть максимально приближен к интерфейсам подобных систем. Ввод информации должен осуществляться в наиболее унифицированных формах.

4.4. Требования к составу и параметрам технических средств

Система подразумевает наличие двух частей: клиентской и серверной. В соответствии с этим требования были разделены на две группы.

На сервере:

- СУБД MySQL и утилита phpMyAdmin;

На клиенте:

- компьютер с оперативной памятью не менее 512МБ;
- операционная система семейства Windows не старше Windows XP SP3;
- установленный пакет Microsoft Office 2010 и последующих;
- доступ в интернет;

4.5. Требования к маркировке и упаковке

Готовое программное изделие и документация поставляется на компакт-дисках в стандартной упаковке. Один комплект программной документации должен быть распечатан с помощью лазерного принтера на листах формата А4 и иметь типографский переплет.

4.6. Требования к транспортированию и хранению

Требования к транспортированию и хранению программного изделия совпадают с аналогичными требованиями, предъявляемыми к компакт-дискам.

5. Требования к программной документации

Программная документация должна содержать следующие документы (см. ГОСТ 19.101-77):

1. Программные документы:

- Спецификация (ГОСТ 19.202-78);
- Текст программы (ГОСТ 19.401-78);
- Описание программы (ГОСТ 19.402-78);
- Пояснительная записка (ГОСТ 19.404-79);
- Программа и методика испытаний (ГОСТ 19.301-79);

2. Эксплуатационные документы:

- Ведомость эксплуатационных документов (ГОСТ 19.507-79);
- Формуляр (ГОСТ 19.501-78);
- Описание применения (ГОСТ 19.502-78);
- Руководство системного программиста (ГОСТ 19.503-79);
- Руководство программиста (ГОСТ 19.504-79);
- Руководство оператора (ГОСТ 19.505-79);

Требования к перечисленным документам не отличаются от требований, определенных в ЕСПД.

6. Порядок контроля и приемки

Приемка работы осуществляется по результатам приемо-сдаточных испытаний проводимых в присутствии представителей заказчика в соответствие с программой и методикой испытаний на тестовых данных, подготовленных заказчиком.

ПРИЛОЖЕНИЕ Б

Исходный код модуля обновления программы

```
namespace rpdSystem
{
    public partial class UpdateForm : Form
    {
        public UpdateForm()
        {
            InitializeComponent();
        }
        BackgroundWorker bg1;
        XmlDocument localDoc, serverDoc;
        public delegate void ct(Label l, string text);
        public delegate void cb(Control c, bool a);
        string Host = "http://localhost/updates/";

        private void UpdateForm_Load(object sender, EventArgs e)
        {
            if (File.Exists("tempVer.xml"))
            {
                File.Delete("tempVer.xml");
            }
            if (!File.Exists(Application.StartupPath + "\\version.xml"))
            {
                Invoke(new ct(changeText), new object[] { label4, "Не установлена"
});
                MessageBox.Show("Рекомендуется      выполнить      полное
обновление!");
            }
            return;
        }
    }
}
```

```

    }
    else
    {
        localDoc = new XmlDocument();
        localDoc.Load("version.xml");
        label4.Text
localDoc.SelectSingleNode("files/programm").InnerText;
    }
    bg1 = new BackgroundWorker();
    bg1.DoWork += zapusk;
    bg1.RunWorkerAsync();
}
public bool ConnectionAvailable(string strServer)
{
    try
    {
        HttpWebRequest reqFP
(HttpWebRequest)HttpWebRequest.Create (strServer);
        HttpWebResponse
rspFP=(HttpWebResponse)reqFP.GetResponse();
        if (HttpStatusCode.OK == rspFP.StatusCode)
        {
            rspFP.Close();
            return true;
        }
        else
        {
            rspFP.Close();
            return false;
        }
    }
}

```

```

    }
    catch (WebException)
    {
        return false;
    }
}

void zapusk(object sender, DoWorkEventArgs e)
{
    string http = "http://localhost/index2.html";
    string ver=label4.Text;
    if (!ConnectionAvailable(http))
    {
        Invoke(new ct(changeText), new object[] { label5, "Проверка
файлов...." });
    }
    string st = Application.StartupPath;
    WebClient wl = new WebClient();
    wl.DownloadProgressChanged += Wl_DownloadProgressChanged;
    wl.DownloadFile(new Uri(Host + "version.xml"), "tempVer.xml");
    serverDoc = new XmlDocument();
    serverDoc.Load("tempVer.xml");
    string s = serverDoc.SelectSingleNode("files/programm").InnerText;
    Invoke(new ct(changeText), new object[] { label5, s });
    if (s != ver)
    {
        Invoke(new ct(changeText), new object[] { label5, s + "(требуется
обновление)" });
    }
    return;
}

```

```

        if(serverDoc.SelectNodes("files/file").Count==
localDoc.SelectNodes("files/file").Count)
        {
            foreach (XmlNode n in serverDoc.SelectNodes("files/file"))
            {
                foreach (XmlNode p in localDoc.SelectNodes("files/file"))
                {
                    if (n.Attributes["name"].Value == p.Attributes["name"].Value
&& n.InnerText != p.InnerText)
                    {
                        Invoke(new ct(changeText), new object[] { label5,
s+"(требуется обновление файлов)" });
                        break;
                    }
                }
            }
        }
        else
        {
            Invoke(new ct(changeText), new object[] { label5, s + "(требуется
обновление файлов)" });
        }
    }
    private void Wl_DownloadProgressChanged(object sender,
DownloadProgressChangedEventArgs e)
    {
        progressBar1.Value = e.ProgressPercentage;
    }
    void changeText(Label l, string text)
    {

```

```

        l.Text = text;
    }
    void Enable(Control c, bool a)
    {
        c.Visible = a;
    }
    private void button1_Click(object sender, EventArgs e)
    {
        progressBar1.Visible = true;
        BackgroundWorker patcher = new BackgroundWorker();
        patcher.DoWork += Patcher_DoWork;
        patcher.RunWorkerAsync();
    }
    private void Patcher_DoWork(object sender, DoWorkEventArgs e)
    {
        List<string> fileList = new List<string>();
        fileList.Add("rpdSystem.exe");
        XmlDocument xmldoc = new XmlDocument();
        xmldoc.Load("tempVer.xml");
        foreach (XmlNode n in xmldoc.SelectNodes("/files/file"))
        {
            fileList.Add(n.Attributes["name"].Value);
        }
        WebClient wl = new WebClient();
        foreach (string s in fileList)
        {
            wl.DownloadFile(Host + s, "new" + s);
        }
        string param = "rpdSystem.exe";
        for (int i=1; i<fileList.Count; i++)

```

```

        {
            param += " " + fileList[i];
        }
        if (File.Exists("version.xml")) File.Delete("version.xml");
        File.Move("tempVer.xml", "version.xml");
        MessageBox.Show("Скачивание файлов завершено! Программа
будет перезапущена автоматически.", "Инфо", MessageBoxButtons.OK,
MessageBoxIcon.Information);
        Process.Start("updater.exe", param);
        Process.GetCurrentProcess().Kill();
    }
    private void UpdateForm_FormClosing(object sender,
FormClosingEventArgs e)
    {
        if (bg1.IsBusy)
        {
            e.Cancel = true;
            return;
        }
        if (File.Exists("tempVer.xml"))
        {
            File.Delete("tempVer.xml");
        }
    }
}
}

```


ПРИЛОЖЕНИЕ В

Исходный код модуля обращения в поддержку

```
namespace rpdSystem
{
    public partial class Support : Form
    {
        private const string messageFormat= @"
E-mail:{0}
Имя:{1}
Сообщение:{2}";

        public Support()
        {
            InitializeComponent();
        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {
            Regex reg = new Regex("^([a-z0-9_-]+\.\.)*[a-z0-9_-]+@[a-z0-9_-
]+(\.\.[a-z0-9_-]+)*\.[a-z]{2,6}$");

            if (!string.IsNullOrEmpty(textBox2.Text) &&
!reg.IsMatch(textBox2.Text))
            {
                label4.Text = "Введите корректный E-mail";
            }
            else
            {
                label4.Text = string.Empty;
            }
        }

        private void button1_Click(object sender, EventArgs e)
```

```

        {
            if (!string.IsNullOrEmpty(textBox1.Text) &&
                !string.IsNullOrEmpty(textBox2.Text) && !string.IsNullOrEmpty(textBox3.Text)
                && string.IsNullOrEmpty(label4.Text))
            {
                try
                {
                    Guid guid = Guid.NewGuid();
                    using (MailMessage mm = new MailMessage($"Support RPD
BSU <{Configuration.supportEmail}>", Configuration.supportEmail))
                    {
                        mm.Subject = $"[{guid}] Запрос в поддержку";
                        mm.Body = string.Format(messageFormat, textBox1.Text,
textBox2.Text, textBox3.Text);
                        mm.IsBodyHtml = false;
                        using (SmtpClient sc = new
SmtpClient(Configuration.supportServerAddress, 25))
                        {
                            sc.EnableSsl = true;
                            sc.DeliveryMethod = SmtpDeliveryMethod.Network;
                            sc.UseDefaultCredentials = false;
                            sc.Credentials = new
NetworkCredential(Configuration.supportEmail, Configuration.supportPassword);
                            sc.Send(mm);
                        }
                    }
                }
                MessageBox.Show("Отправлено!\nНомер вашей заявки: " +
guid.ToString(), string.Empty, MessageBoxButtons.OK,
MessageBoxIcon.Information);
            }
        }

```

```
        catch
        {
            MessageBox.Show("Ошибка отправки. Повторите попытку
позже!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
else
{
    MessageBox.Show("Заполните корректно все поля!", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
}
```

ПРИЛОЖЕНИЕ Г

Исходный код модуля загрузки РПД

```
private void loadFormFromXml(string filename)
{
    XmlDocument xmlDoc = new XmlDocument();
    xmlDoc.Load(filename);
    try
    {
        foreach (XmlNode n in xmlDoc.SelectNodes("/elements/TextBox"))
        {
            TextBox a =
            (TextBox)this.Controls.Find(n.Attributes["name"].Value, true)[0];
            a.Text = n.SelectSingleNode("Text").InnerText;
        }
        ComboBox[] cb = { comboBox1, comboBox2, comboNapr,
godPostBox, comboDist, comboBox3, comboBox4 };
        for (int i=0; i<cb.Length; i++)
        {
            foreach (XmlNode n in
xmlDoc.SelectNodes("/elements/ComboBox"))
            {
                if (n.Attributes["name"].Value == cb[i].Name)
                {
                    if (cb[i].Items.Count<
Convert.ToInt32(n.SelectSingleNode("SelectedIndex").InnerText) ||
cb[i].Items[Convert.ToInt32(n.SelectSingleNode("SelectedIndex").InnerText)].ToStr
ing() != n.SelectSingleNode("Text").InnerText)
                    {
```

```

        if (MessageBox.Show("Данный файл является
устаревшим, его использование может привести к ошибкам, продолжить?!",
"Предупреждение", MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.No)
    {
        totalClear();
        return;
    }
    cb[i].SelectedIndex =
Convert.ToInt32(n.SelectSingleNode("SelectedIndex").InnerText);
    //cb[i].Text = n.SelectSingleNode("Text").InnerText;
    }
    }
    }
    foreach (XmlNode n in
xmlDoc.SelectNodes("/elements/DataGridView"))
    {
        DataGridView tb =
(DataGridView)this.Controls.Find(n.Attributes["name"].Value, true)[0];
        tb.Rows.Clear();
        for (int i = 0; i <
Convert.ToInt16(n.SelectSingleNode("rowCount").InnerText); i++)
        {
            tb.Rows.Add();
            for (int j = 0; j < tb.ColumnCount; j++)
            {
                XmlNode p = n.SelectSingleNode("Cell[@rowIndex=" +
i.ToString() + " and @columnIndex=" + j.ToString() + "]);
                tb[j, i].Value = p.SelectSingleNode("Value").InnerText;

```

```

        }
    }
}
foreach (XmlNode n in xmlDoc.SelectNodes("/elements/label"))
{
    Label tb = (Label)this.Controls.Find(n.Attributes["name"].Value,
true)[0];

    tb.Text = n.SelectSingleNode("Text").InnerText;
}
}
catch
{
    MessageBox.Show("Ошибка чтения файла", "Ошибка!",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
newLoad = true;
}
void totalClear()
{
    foreach (TabPage n in tabControl1.TabPages)
    {
        if (n.Controls.OfType<TabControl>().Count<TabControl>() > 0)
        {
            foreach (TabControl tc2 in n.Controls.OfType<TabControl>())
            {
                foreach (TabPage n2 in tc2.TabPages)
                {
                    localClear(n2);
                }
            }
        }
    }
}

```

```

        }
    }
    localClear(n);
}
Label[] labels = { label21, label44, label45, label46, label47, label48,
label49, label50, label51, label52, label53, label54 };
foreach (Label tb in labels)
{
    tb.Text = string.Empty;
}
}
void localClear(TabPage n)
{
    foreach (TextBox tb in n.Controls.OfType<TextBox>())
    {
        if (tb.Text != null && tb.Text != "") tb.Text = null;
    }
    foreach (ComboBox tb in n.Controls.OfType<ComboBox>())
    {
        if (tb.Text != null && tb.Text != "")
        {
            tb.Text = null;
            tb.SelectedIndex = -1;
        }
    }
    foreach (DataGridView tb in n.Controls.OfType<DataGridView>())
    {
        tb.Rows.Clear();
    }
}
}

```

ПРИЛОЖЕНИЕ Д.

Исходный код утилиты обновления файлов программы

```
namespace updater
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                string process = args[0].Replace(".exe", "");
                Console.WriteLine("Terminate process!");
                while (Process.GetProcessesByName(process).Length > 0)
                {
                    Process[] myProcesses2 =
Process.GetProcessesByName(process);
                    for (int i = 1; i < myProcesses2.Length; i++) {
myProcesses2[i].Kill(); }
                    Thread.Sleep(300);
                }
                for (int i = 0; i < args.Length; i++)
                {
                    if (File.Exists(args[i])) { File.Delete(args[i]); }
                    File.Move("new" + args[i], args[i]);
                }
                Console.WriteLine("Starting " + args[0]);
                Process.Start(args[0]);
            }
            catch (Exception ex){Console.WriteLine("Aborting! " + ex.Message);
}
}
}
```


ПРИЛОЖЕНИЕ Е

Пример титульной страницы РПД разработанной с помощью информационной системы

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»
(НИУ «БелГУ»)

УТВЕРЖДАЮ

Директор института Инженерных Технологий
и Естественных Наук

Польшиков К.А.

_____ .20__

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Методы обработки и хранения цифровых изображений

наименование дисциплины (модуля)

Программа составлена в соответствии с требованиями ФГОС ВО по направлению
подготовки/специальности

Направление 09.03.03 Прикладная информатика

подготовки/
специальность

Профиль подготовки/ Администрирование информационных систем в экономике

специализация/

магистерская программа

Автор: старший преподаватель кафедры ПИИИТ Е.В. Болгова, профессор кафедры ПИИИТ, доцент, доктор технических наук А.А. Черноморец

Программа одобрена кафедрой Прикладной информатики и информационных технологий

Протокол заседания кафедры от _____ № _____
дата

Программа согласована кафедрой Прикладной информатики и информационных технологий

Протокол заседания кафедры от _____ № _____
дата