

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Е.Н. ЭГОВ

**РАЗРАБОТКА ПРОГРАММЫ УЧЕТА ЛИТЕРАТУРЫ
В АБСТРАКТНОМ УНИВЕРСИТЕТЕ
практикум по дисциплине
«Технологии программирования»**

Ульяновск

УлГТУ

2021

УДК 004.9 : 338(076)

ББК 32.973.26 – 018.2я73

К69

Рекомендовано научно-методической комиссией факультета информационных систем и технологий в качестве практикума.

Эгов, Евгений Николаевич

К69 **РАЗРАБОТКА ПРОГРАММЫ УЧЕТА ЛИТЕРАТУРЫ В АБСТРАКТНОМ УНВЕРСИТЕТЕ : практикум / Е. Н. Эгов. – Ульяновск : УлГТУ, 2021. – 76 с.**

Практикум адресован студентам для выполнения и оформления курсовой работы по дисциплине «Технологии программирования». Предоставлены варианты заданий для индивидуальной и парной разработки. Рекомендации и требования к курсовой работе разработаны в соответствии с рабочей программой дисциплины. Предназначены для студентов, обучающихся по направлению 09.03.04 «Программная инженерия»

Работа подготовлена на кафедре «Информационные системы».

УДК 004.9 : 338(076)

ББК 32.973.26 – 018.2я73

© Эгов Е. Н., 2021

© Оформление. УлГТУ, 2021

СОДЕРЖАНИЕ

Содержание.....	3
Этап №0. Постановка задания	5
Этап №1. Формулировка ТЗ.....	10
Этап №2. Составление диаграмм	14
Этап №3. Составление схемы БД.....	18
Этап №4. Представление проектов	22
Этап №5. Реализация модуля с моделями	23
Этап №6. Реализация модуля бизнес-логики	24
Этап №5. Реализация модуля хранения данных	25
Этап №7. Реализация интерфейсов	26
Этап №8. Демонстрация бета-версии программы	27
Этап №9. Реализация дополнительного функционала.....	28
Этап №10. Показ готовой программы	29
Этап №11. Оформление записки и презентации	30
Этап №12. Защита курсовой работы.....	31
Темы курсрвой работы	32
Список использованных источников	48
Приложение 1 Титульный лист	49
Приложение 2 Задание на курсовую работу	50
Приложение 3 Отзыв руководителя.....	51
Приложение 4 Шаблон технического задания для роли «Работник (первая роль)»	52
Приложение 5 Шаблон технического задания для роли «Кладовщик (вторая роль)»	56

Приложение 6 Шаблон технического задания для роли «Исполнитель (третья роль)».....	60
Приложение 7 Шаблон технического задания для роли «Поручитель (четвертая роль)».....	64
Приложение 8 Шаблон технического задания для роли «Поставщик (пятая роль)»	68
Приложение 9 Шаблон технического задания для роли «Заказчик (шестая роль)»	72

ЭТАП №0.

ПОСТАНОВКА ЗАДАНИЯ

Общее задание (90 баллов):

- разработать программное обеспечение, согласно заданию по варианту;
- проект разбить на модули (архитектура DAL): бизнес-логика, хранение данных, интерфейс пользователя (WPF);
- для парного программирования реализуются отдельные модули бизнес-логики и интерфейсов пользователей для разных ролей;
- для индивидуальной работы реализуется работа одной из роли;
- для индивидуальной работы работа второй роли реализуется в виде моделирования его действий;
- в качестве хранилища данных использовать СУБД;
- предусмотреть проверку данных, вводимых пользователем;
- предусмотреть авторизацию при входе в систему;
- выставление счетов и расчётов на оплату выводить в pdf-файл;
- все этапы реализации должны выкладываться на git.

Дополнительное задание (10 баллов):

- разработать рекомендательную систему (ТОУЗ) на основе хранимых данных;

Этапы сдачи обязательного задания (после выдачи задания):

1. Формулировка ТЗ под свой вариант. Поиск аналогов.
2. Составление диаграмм (Use-case, диаграмма последовательностей и т.д.).
3. Составление БД (ER-диаграмма).
4. **Представление проекта на СНТК – конец 4 недели (30 баллов).**
5. Реализация модуля бизнес-логики.
6. Реализация модуля хранения данных.
7. Реализация интерфейса пользователя.
8. **Демонстрация бета-версии программы – конец 10 недели (30 баллов).**

9. Устранение замечаний, ошибок.

а. Дополнительное задание (если требуется, 10 баллов).

10. Повторный показ готовой программы

11. Подготовка презентации, оформление записки.

12. Защита курсовой работы – зачетная неделя (30 баллов).

При пропуске сроков сдачи этапа баллы будут снижаться. Презентация работы будет проводиться ТОЛЬКО на зачетной неделе.

Этапы 4, 8, 12 принимаются преподавателем в установленные сроки.

Выполнение этапов 5-7 проверяются при сдаче проекта через репозиторий.

Задание:

Все задания на курсовую делятся на 6 ролей. Студенты могут выполнять задания парами 1-2, 3-4 или 5-6 роли. Первая пара реализует взаимодействие ролей Работник-Кладовщик, вторая пара – Исполнитель-Поручитель, третья пара – Поставщик-Заказчик.

Для работника (первая роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 1» с выбором из списка «сущности 5» нескольких записей к ней;
- получение списка по «сущности 4» на основе выбранных записях «сущности 1» в формате doc/xls;
- формирование записей (CRUD) «сущности 2»;
- привязывание записей «сущности 2» к записям «сущности 1»;
- формирование записей (CRUD) «сущности 3» с привязкой записи «сущности 2»;
- получение отчета за период по записям из «сущности 1» с расшифровкой по записям «сущности 4», относящихся к записям «сущность 2», относящихся к записям «сущности 1» в формате pdf на почту и вывод отчета на форму.

Для кладовщика (вторая роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 5» с выбором из списка «сущности 4» нескольких записей к ней;
- получение списка по «сущности 2» на основе выбранных записях «сущности 5» в формате doc/xls;
- формирование записей (CRUD) по «сущности 4»;
- привязывание записей «сущности 4» к записям «сущности 2»;
- формирование записей (CRUD) «сущности 6» с привязкой записи «сущности 5»;
- получить отчет за период по движению записей «сущности 4» в формате pdf на почту и вывод отчета на форму;

Для исполнителя (третья роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 7»;
- формирование записи (CRUD) по «сущности 8» с выбором из списка «сущности 7» нескольких записей к ней;
- формирование записи (CRUD) по «сущности 9» с выбором из списка «сущности 7» нескольких записей к ней;
- получение списка по «сущности 12» на основе выбранных записях «сущности 7» в формате doc/xls;
- привязка к записи «сущности 8» записи «сущности 11»;
- получение отчета за период по записям «сущности 7», связанных с записями «сущности 8» и «сущности 9» в формате pdf на почту и вывод отчета на форму.

Для поручителя (четвертая роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 10»;
- формирование записи (CRUD) по «сущности 11» с привязкой записи по «сущности 10»;
- формирование записи (CRUD) по «сущности 12» с выбором из списка «сущности 10» нескольких записей к ней;

- получение списка по «сущности 8» на основе выбранных записях «сущности 10» в формате doc/xls;
- привязка к записи «сущности 12» записи «сущности 9»;
- получение отчета за период по «сущностям 11» и «сущностям 12», связанных между собой, в формате pdf на почту и вывод отчета на форму;

Для поставщика (пятая роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 13»;
- формирование записи (CRUD) по «сущности 14» с выбором из списка «сущности 13» нескольких записей к ней;
- формирование записи (CRUD) по «сущности 15» с выбором из списка «сущности 13» записи к ней;
- получение списка по «сущности 18» на основе выбранных записях «сущности 13» в формате doc/xls;
- привязка к записи «сущности 14» нескольких записей из списка «сущности 16»;
- получение отчета за период по записям «сущности 15», связанных с записями «сущности 17» в формате pdf на почту и вывод отчета на форму.

Для заказчика (шестая роль) будет требоваться реализовать следующий функционал:

- формирование записи (CRUD) по «сущности 16»;
- формирование записи (CRUD) по «сущности 17» с выбором из списка «сущности 16» нескольких записей к ней;
- формирование записи (CRUD) по «сущности 18» с выбором из списка «сущности 17» записи к ней;
- получение списка по «сущности 15» на основе выбранных записях «сущности 17» в формате doc/xls;
- привязка к записи «сущности 17» нескольких записей из списка «сущности 13»;

- получение отчета за период по «сущностям 15» и «сущностям 18», связанных между собой, в формате pdf на почту и вывод отчета на форму;

ЭТАП №1.

ФОРМУЛИРОВКА ТЗ

В качестве примера рассмотрим работу университета со студентами, сотрудниками и издательством. В качестве «сущности 1» и будут выступать обучения, формируемые студентами. «Сущность 2» – это курсы, которые относятся к разным обучениям. «Сущность 3» – это оценки, получаемые студентами на курсах. «Сущность 4» – литература, используемая при проведении занятий. «Сущность 5» – это сами проводимые занятия (проводятся в рамках обучений). «Сущность 6» – выставление баллов на занятии. Для второй пары: «сущность 7» – занятия студенты; «сущность 8» – заявки на получение литературы для занятий; «сущность 9» – записи посещения читального зала для получения информации к занятию; «сущность 10» – литература, выдаваемая для чтения студентам; «сущность 11» – записи выдачи литературы в читальном зале; «сущность 12» – записи передачи на руки литературы в библиотеке. Для третьей пары: «сущность 13» – производимая издательством литература; «сущность 14» – поставки литературы университету; «сущность 15» – акты изготовления литературы; «сущность 16» – заявки на закупку; «сущность 17» – записи поступления литературы по заявкам; и «сущность 18» – акты приема литературы в университете на основе записей поступления.

- Роль – Работник (Студент):
 - создать обучение и выбрать в него занятия;
 - получить список литературы по выбранным обучениям, в формате doc/xls;
 - создать курсы;
 - привязать выбранные курсы к выбранному обучению;
 - создать оценки на основе выставленных баллов по занятиям, проводимым за курс (с выбором курса, за который получается оценка);
 - получить отчет за период по пройденным обучениям с расшифровкой по используемой в рамках курсов литературе и полученных оценках в формате pdf на почту или на форме.

- Роль – Кладовщик (Преподаватель):
 - создать занятие и выбрать в него литературу;
 - получить список курсов по выбранным занятиям, в формате doc/xls;
 - CRUD литература;
 - привязать выбранную литературу к выбранным курсам;
 - создать отметки выставления баллов студентам, на проводимом занятии;
 - получить отчет за период по литературе с указанием по курсам и занятиям, на которых она использовалась в формате pdf на почту или на форме.
- Роль – Исполнитель (Студент):
 - создать занятия;
 - создать запись заявки на получение литературы в библиотеке и выбрать под какие занятия требуется литература;
 - создать запись посещения читального зала для чтения литературы и выбрать под какие занятия требуется литература;
 - получить список записей выдачи литературы по выбранным занятиям, в формате doc/xls;
 - привязывать запись посещения читального зала к записи передачи литературы на руки;
 - получить отчета за период по занятиям, для которых требовалось посещения читального зала и получения литературы на руки, в формате pdf на почту или на форме.
- Роль – Поручитель (Библиотекарь):
 - создать литературу;
 - создать запись выдачи литературы в читальном зале и выбрать в него литературу;
 - создать запись передачи на руки литературы в библиотеке и выбрать в него литературу;
 - получить список заявок студентов по выбранной литературе, в формате doc/xls;

- привязывать запись выдачи литературы в читальном зале к заявке студента;
- получить отчет за период по литературе, указываемой в записях выдачи литературы и передачи на руки, в формате pdf на почту или на форме.
- Роль – Поставщик (Издательство):
 - создать литературу;
 - создать запись поставки литературы, выбрать литературу для поставки;
 - создать запись изготовления литературы с указанием записи литературы, которую изготовили;
 - получить список записей актов приемки литературы по выбранным записям изготовления, в формате doc/xls;
 - привязывать запись поставки к записям заявок на поставки;
 - получить отчета за период по изготовлению литературы, поставленной университету, в формате pdf на почту или на форме.
- Роль – Заказчик (Университет):
 - создать заявки на поставки;
 - создать запись поступления литературы и выбрать по каким заявкам она поставлялась;
 - создать актов получения литературы с указанием по какой поставке она пришла;
 - получить список записей изготовления литературы по выбранным позициям поставок литературы, в формате doc/xls;
 - привязывать запись поступления литературы к литературе;
 - получить отчет за период по литературе, изготовленной и полученной (оформлены через акты), в формате pdf на почту или на форме.

Техническое задание (ТЗ) должно включать в себя следующие пункты:

1. Цель разрабатываемого программного продукта.
2. Описание всех возможных действий для роли.

3. Выбор реализации интерфейса для роли.
4. Обзор существующих программных решений.

Примеры технического задания:

Для роли работника – Приложение 4.

Для роли кладовщика – Приложение 5.

Для роли исполнителя – Приложение 6.

Для роли поручителя – Приложение 7.

Для роли поставщика – Приложение 8.

Для роли заказчика – Приложение 9.

Красным отмечено то, что требуется заменить, исходя из логики своего варианта, а также по количеству студентов, выполняющих работу.

В ТЗ рассмотреть 2-3 аналога из существующих программных решений.

Указать их достоинства и недостатки.

ЭТАП №2.

СОСТАВЛЕНИЕ ДИАГРАММ

В рамках разрабатываемой системы необходимо составить ряд UML диаграмм:

- диаграмма вариантов использования (use-case);
- диаграмма последовательности (sequence);
- диаграмма состояний (state-machine);
- диаграмма развертывания (deployment).

Рассмотрим для чего нужны эти диаграммы и что на них должно быть.

Диаграмма вариантов использования

Диаграмма вариантов использования отражает отношения между актёрами и прецедентами и является составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Основная цель построения диаграммы вариантов использования для нашего проекта – определить действующих лиц (актёров), их взаимодействие с системой и ожидаемую функциональность системы;

Работа над диаграммой может начаться с текстового описания, полученного при работе с заказчиком. При этом нефункциональные требования (например, конкретный язык или система программирования) при составлении модели прецедентов опускаются.

Так как в плане интерфейсов пользователей будут разрабатываться две различные программы (для парного программирования), то для каждой роли строится своя, отдельная, диаграмма вариантов использования.

Рассмотрим, как может выглядеть диаграмма (рисунок 2.1).

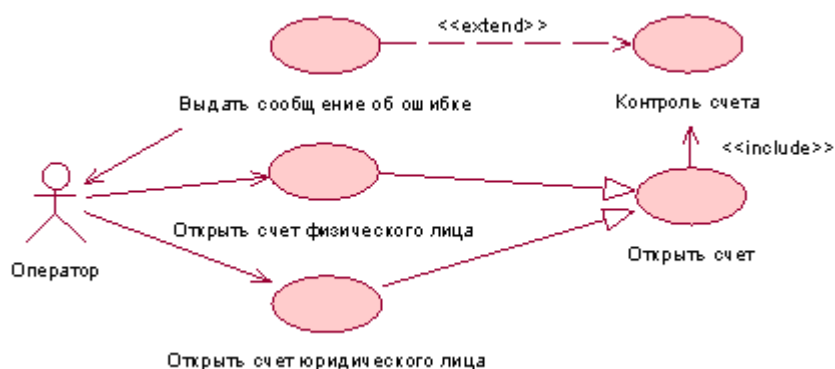


Рисунок 2.1 – Пример диаграммы вариантов использования

На данной диаграмме представлен один актер – оператор. Он имеет 2 возможных действия в системе: открыть счет физического лица или открыть счет юридического лица. Оба этих действия имеют общую часть функционала – действие «Открыть счет». При выполнении действия «Открыть счет» также требуется выполнить действие «Контроль счета». У этого действия имеется необязательное действие «Выдать сообщение об ошибке».

ПРИМЕР ДЕМОНСТРИРУЕТ КАК МОЖЕТ ВЫГЛЯДЕТЬ ДИАГРАММА ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ. ОНА НЕ СООТВЕТСТВУЕТ ТОМУ, ЧТО ТРЕБУЕТСЯ СДЕЛАТЬ ПО КУРСОВОЙ РАБОТЕ. В ДИАГРАММЕ ПО КУРСОВОЙ РАБОТЕ ТРЕБУЕТСЯ ОТРАЗИТЬ ВСЕ ВОЗМОЖНЫЕ ДЕЙСТВИЯ ПОЛЬЗОВАТЕЛЯ, ОПИСАННЫЕ В ТЗ.

Для разрабатываемого проекта в качестве возможных действий в системе могут выступать:

- Авторизация
- Регистрация
- Получение списка. К ней дополнительный функционал в виде добавления, редактирования и удаления.
- Формирование отчетов. К ним дополнительный функционал в виде отправки их на почту.
- И т.д.

Диаграмма последовательности

Это диаграмма, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актёров (действующих лиц) ИС в рамках какого-либо определённого прецедента (отправка запросов и получение ответов). Рассмотрим пример (рисунок 2.2)

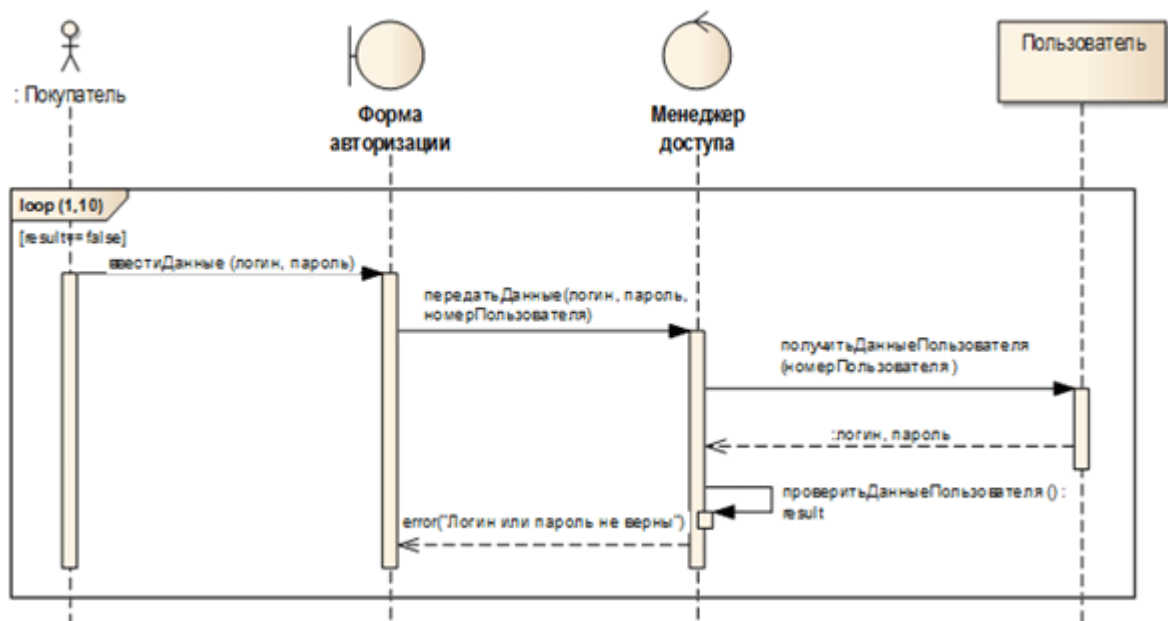


Рисунок 2.2 – Пример диаграммы последовательности

На данной диаграмме есть один актер – Покупатель, одна форма – форма авторизации, класс-контроллер (менеджер доступа) и сущность (пользователь). В диаграмме отображена последовательность действий при неудачной авторизации. Пользователь (покупатель) вводит на форму авторизации данные. Форма передает данные контролу. Контрол на основе переданных данных запрашивает у сущности (например, в БД) данные по пользователю и сверяет их. Если они не совпадают, то контрол возвращает на форму ошибку. Форма отображает эти данные пользователю (на диаграмме этого участка не представлено).

ПРИМЕР ДЕМОНИСТРИРУЕТ КАК МОЖЕТ ВЫГЛЯДЕТЬ ДИАГРАММА ПОСЛЕДОВАТЕЛЬНОСТИ. ОНА НЕ СООТВЕТСТВУЕТ ТОМУ, ЧТО ТРЕБУЕТСЯ СДЕЛАТЬ ПО КУРСОВОЙ РАБОТЕ. В ДИАГРАММЕ ПО КУРСОВОЙ РАБОТЕ ТРЕБУЕТСЯ ОТРАЗИТЬ ПРОЦЕСС ВЗАИМОДЕЙСТВИЯ ЭЛЕМЕНТОВ СИСТЕМЫ, ОТ РАБОТЫ ПОЛЬЗОВАТЕЛЯ С ФОРМОЙ, ДО РАБОТЫ С БД.

В рамках курсовой работы для каждой роли следует реализовать последовательность:

- Работник
 - процесс формирования записи «сущности 1» с выбором записей «сущности 2»;

- Кладовщик
 - процесс формирования записи «сущности 5» с выбором записей «сущности 4»;
- Исполнитель
 - процесс формирования записи «сущности 8» с выбором записей «сущности 7»;
- Поручитель
 - процесс формирования записи «сущности 11» с выбором записей «сущности 10»;
- Поставщик
 - процесс формирования записи «сущности 14» с выбором записей «сущности 13»;
- Заказчик
 - процесс формирования записи «сущности 17» с выбором записей «сущности 16»;

Диаграмма развертывания

Диаграмма развёртывания в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать веб-сайт диаграмма развертывания должна показывать, какие аппаратные компоненты («узлы») существуют (например, веб-сервер, сервер базы данных, сервер приложения), какие программные компоненты («артефакты») работают на каждом узле (например, веб-приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, REST, RMI).

В рамках курсовой работы будет формироваться диаграмма развертывания с указанием проектов системы. Для парного программирования будут проекты с моделями, проекты под DAL-архитектуру, сервер БД и 2 проекта-приложения. Для индивидуального программирования будет только 1 проект-приложение.

ЭТАП №3.

СОСТАВЛЕНИЕ СХЕМЫ БД

Основная цель – составление схемы базы данных, какие таблицы будут в БД, какие в них будут поля и как таблицы будут связаны между друг другом. Для этих целей создают ER-диаграмму.

Рассмотрим пример ER-диаграммы на рисунке 3.1.



Рисунок 3.1 – Пример ER-диаграммы

На данной схеме представлены 5 сущностей (таблицы БД):

- Студент;
- Успеваемость;
- Предмет;
- Специальность;
- Факультет.

Под названием каждой сущности указываются атрибуты – данные сущности, которые мы хотим хранить. Линиями обозначаются связанные между собой сущности. Цифры или символы на концах линий обозначают тип связи:

- Один к одному. На концах линий стоят цифры 1. Для записи из одной сущности может быть (или должна быть) одна и только одна запись в другой сущности.
- Один ко многим. На одном конце стоит цифра 1, на другом знак бесконечности. Для записи из одной сущности (рядом с которой цифра 1) может быть несколько записей в другой сущности (рядом с которой стоит знак бесконечности).

- Многие ко многим. На обоих концах линии стоят знаки бесконечности. Означает, что для записи из одной сущности может быть несколько записей в другой сущности, и для записи из другой сущности может быть несколько записей из первой сущности.

Рассмотрим на примере схемы с рисунка 3.1, как это выглядит.

На данной схеме нет примера связи **один к одному**, так как эта связь крайне редко применяется. По сути, это означает что описание одной сущности разнесено в разные таблицы. Если бы, например, для сущности «Факультет» была отдельная сущность «История создания факультета», где хранилась бы информация о дате создания, кем был создан и т.п. В таком случае, для любого факультета из таблицы «Факультет» была бы одна запись в таблице «История создания факультета».

Один ко многим. В примере у нас есть сразу 2 записи такого типа:

- Специальность-Студент;
- Факультет-Студент.

Связь говорит о том, что к одному факультету может относиться сразу несколько студентов. Но каждый студент может относиться только к **ОДНОМУ** факультету. Для специальностей аналогично. Представленная на рисунке схема не совсем верна, правильнее будет, чтобы специальности относились к факультетам (один ко многим, на один факультет несколько специальностей), а студенты относились бы только к специальностям и через специальность относились бы уже к факультету (рисунок 3.2).



Рисунок 3.2 – Обновлённая ER-диаграмма

Связь многие ко многим, по сути, представляет собой связь между 3 сущностями (одна из сущностей является связующей). На схеме такая связь представлена в виде связи Студент-Предмет. Много студентов сдают много предметов. Для этого создана отдельная сущность «Успеваемость». В сущности-связи, помимо ссылок на записи связываемых сущностей могут быть дополнительные атрибуты, как представлено на схеме в примере. Минимально в сущности-связи должно быть 2 поля: ссылка на сущность 1 и ссылка на сущность 2.

В базе данных для связи между таблицами требуется, чтобы в каждой таблице был атрибут (колонка), в которой для каждой записи было бы уникальное значение в рамках таблицы. Рассмотрим на примере студента. Можно ли в качестве такого атрибута рассматривать, например, ФИО? Ответ: нет. Мы не можем исключать случай, что в университете могут одновременно учиться полные тезки. В данном случае уникальным будет атрибут «Номер зачетки», так как нет двух студентов с одинаковым номером зачетной книжки. В общем случае для таблицы создают атрибут «Идентификатор» (сокращенно Id) и каждой добавляемой записи в таблицу в этот атрибут проставляют значение, которого нет ни у какой другой записи в этой таблице (уникальность требуется только в рамках таблицы, в других таблицах могут также быть поля Id и в них могут быть такие же значения, что и в поле Id у данной таблицы). При создании связи (один ко многим) в таблице (рядом с ней будет знак ∞), к которой будет относиться «другая таблица» (у нее цифра 1) требуется создать поле-значение уникального поля из «другой таблицы» (рисунок 3.3).



Рисунок 3.3 – ER-диаграмма с указанием полей-связей

ПРИМЕР ДЕМОНСТРИРУЕТ КАК МОЖЕТ ВЫГЛЯДЕТЬ ER-ДИАГРАММА. ОНА НЕ СООТВЕТСТВУЕТ ТОМУ, ЧТО ТРЕБУЕТСЯ СДЕЛАТЬ ПО КУРСОВОЙ РАБОТЕ. В ДИАГРАММЕ ПО КУРСОВОЙ РАБОТЕ ТРЕБУЕТСЯ ОТРАЗИТЬ СУЩНОСТИ, КОТОРЫЕ СЛЕДУЕТ ХРАНИТЬ, КАК ОНИ СВЯЗАНЫ МЕЖДУ СОБОЙ, И КАКИЕ В НИХ БУДУТ ПОЛЯ.

В рамках курсовой работы требуется составить ER-диаграмму сущностей, которые будут использоваться в разрабатываемом проекте. Требуется дать им названия, указать названия и **типы** атрибутов, а также проставить связи между сущностями.

ЭТАП №4.

ПРЕДСТАВЛЕНИЕ ПРОЕКТОВ

В рамках недели студенческой науки каждая группа разработчиков должна представить модель разрабатываемого программного продукта. В рамках доклада следует озвучить следующие пункты:

- описание предметной области;
- рассмотрение аналогов;
- диаграммы, описывающие поведение системы;
- диаграмма базы данных;
- применяемые технологии;
- распределение обязанностей (для парных работ, кто за что отвечает).

Лучше работы будут выбраны для публикации в сборнике.

ЭТАП №5.

РЕАЛИЗАЦИЯ МОДУЛЯ С МОДЕЛЯМИ

При разработке современных проектов часто используют концепцию Data Access Layer для обработки данных. В основе концепции лежит принцип управляемого специализированного доступа к данным, который вводит унифицированный и контролируемый способ доступа к различным данным для приложений. Одной из особенностей применения данной концепции заключается в возможности быстро заменить способ работы и хранения данных (например, перейти от хранения в файлах на хранение в СУБД, или сменить СУБД, или способ работы с ней). Проект разбивается на ряд модулей:

- Модуль с моделями (классами, сущностями);
- Модуль с интерфейсами, которые описывают бизнес-логику проекта;
- Модуль с реализацией интерфейса.

На данном этапе требуется реализовать модуль с моделями. В качестве моделей будут выступать все сущности, описанные в ER-диаграмме, также, необходимые вспомогательные типы, например, перечисления и т.п.

В разработке будет применяться EntityFramework, подход Code-First для создания БД, поэтому требуется создать проект с моделями, на основе которых будет создана БД с указанием через атрибуты связанность между моделями.

Исходя из распределенных обязанностей, команда разработчиков на основе ER-диаграммы создают классы (каждый разработчик обязан создать классы, требуемые ему, а также часть общих классов).

Результат выкладывается в git. Несоблюдение сроков штрафуются «-» баллами.

ЭТАП №6.

РЕАЛИЗАЦИЯ МОДУЛЯ БИЗНЕС-ЛОГИКИ

При разработке современных проектов часто используют концепцию Data Access Layer для обработки данных. В основе концепции лежит принцип управляемого специализированного доступа к данным, который вводит унифицированный и контролируемый способ доступа к различным данным для приложений. Одной из особенностей применения данной концепции заключается в возможности быстро заменить способа работы и хранения данных (например, перейти от хранения в файлах на хранение в СУБД, или сменить СУБД, или способ работы с ней). Проект разбивается на ряд модулей:

- Модуль с бизнес-логикой и с интерфейсами, которые описывают логику работы с данными;
- Модуль хранения данных;
- Модуль с реализацией интерфейса.

На данном этапе требуется реализовать модуль бизнес-логики: интерфейсы и классы, которые их используют в работе.

Для реализации модуля требуется создать проект, в котором будут описаны интерфейсы для работы с моделями и классы с общей бизнес-логикой проекта. Также, для уменьшения связанности проектов требуется создать классы-модели для передачи данных в методы интерфейсов и классов (binding models) и классы-модели для передачи данных из методов интерфейсов и классов (view models).

Исходя из распределенных обязанностей, команда разработчиков разрабатывает интерфейсы и классы с бизнес-логикой (каждый разработчик обязан создать элементы, требуемые ему, а также часть общих элементов).

ЭТАП №5.

РЕАЛИЗАЦИЯ МОДУЛЯ ХРАНЕНИЯ ДАННЫХ

При разработке современных проектов часто используют концепцию Data Access Layer для обработки данных. В основе концепции лежит принцип управляемого специализированного доступа к данным, который вводит унифицированный и контролируемый способ доступа к различным данным для приложений. Одной из особенностей применения данной концепции заключается в возможности быстро заменить способа работы и хранения данных (например, перейти от хранения в файлах на хранение в СУБД, или сменить СУБД, или способ работы с ней). Проект разбивается на ряд модулей:

- Модуль с бизнес-логикой и с интерфейсами, которые описывают логику работы с данными;
- Модуль хранения данных;
- Модуль с реализацией интерфейса.

На данном этапе требуется реализовать модуль с реализацией интерфейсов под конкретный вариант хранения данных.

В разработке будет применяться EntityFramework, подход Code-First для создания БД, поэтому требуется создать проект с моделями, на основе которых будет создана БД с указанием через атрибуты связанность между моделями. В качестве моделей будут выступать все сущности, описанные в ER-диаграмме.

Исходя из распределенных обязанностей, команда разработчиков на основе ER-диаграммы создают классы (каждый разработчик обязан создать классы, требуемые ему, а также часть общих классов).

ЭТАП №7.

РЕАЛИЗАЦИЯ ИНТЕРФЕЙСОВ

На данном этапе разработки каждый из разработчиков должен разработать свой модуль взаимодействия с пользователем. Для этого необходимо создать проект для работы с формами и реализовать в нем функционал, описанный в ТЗ.

При реализации необходимо выполнять проверку вводимой от пользователя информации на корректность и заполненность (валидация данных). Программа должна работать стабильно, не падать. Необходимо обрабатывать возникающие ошибки и сохранять информацию по ним.

ЭТАП №8.

ДЕМОНСТРАЦИЯ БЕТА-ВЕРСИИ ПРОГРАММЫ

В рамках демонстрации каждая группа разработчиков должна представить разрабатываемый программный продукт. В рамках демонстрации следует озвучить показать все реализованные пункты, согласно ТЗ и use-case диаграммы. Допустимо присутствие небольшого числа «заглушек» в программе в нереализованном функционале (например, отправке на почту) по причине изучения способов реализации требуемого функционала, либо ожидания его разъяснения в рамках лабораторных работ.

По результатам демонстрации будет составлен список замечаний, который следует устранить к финальной сдаче.

ЭТАП №9.

РЕАЛИЗАЦИЯ ДОПОЛНИТЕЛЬНОГО ФУНКЦИОНАЛА

Для реализации функционала рекомендательной системы (ТОУЗ) необходимо формировать дополнительные графики и рекомендации на основе статистики, накапливаемой по сущностям.

Для приводимых в первом этапе примерах, это могут быть графики оплат курсов студентами, рекомендации к назначению заданий на занятия (например, исходя из уже назначенных по другим заданиям этого курса) и т.п.

ЭТАП №10.

ПОКАЗ ГОТОВОЙ ПРОГРАММЫ

В рамках демонстрации каждая группа разработчиков должна представить итоговый программный продукт. В рамках демонстрации следует озвучить показать все реализованные пункты, согласно ТЗ и use-case диаграммы, устраненные ошибки, доработанный функционал.

ЭТАП №11.

ОФОРМЛЕНИЕ ЗАПИСКИ И ПРЕЗЕНТАЦИИ

На основе разработанного программного продукта, оформить пояснительную записку и подготовить презентацию. Пояснительная записка оформляется ИНДИВИДУАЛЬНО каждым студентом на основе того функционала, который он реализовывал.

Основной текст записки: Times New Roman, 14 шрифт, полуторный интервал, отступ первой строк 1.27.

Пояснительная записка должна содержать следующие пункты:

- Титульный лист (приложение 1).
- Задание на курсовую работу (приложение 2).
- Отзыв руководителя (приложение 3).
- Введение. Описывается актуальность задачи.
- Первая глава. Теоретическая. Описывается предметная область, аналоги, ТЗ.
- Вторая глава. Проектирование. Приводятся диаграммы с их описанием, согласно ТЗ и варианту.
- Третья глава. Руководства. Приводится руководство пользователя и программиста для разработанного проекта.
- Четвертая глава. Тестирование. Приводится пример работы всего функционала, озвученного в ТЗ и use-case диаграмме.
- Заключение. Выводы по проделанной работе, какие технологии были применены и освоены.
- Список литературы. Не менее **10** источников с ссылками в тексте.
- Приложение. Листинг кода (8 шрифт, 3 колонки).

Презентация оформляется на основе подготовленной записки. Первые слайды – описание предметной области, диаграммы. Далее слайды со скриншотами программы, как она выглядит, как работает (за основу можно взять руководство пользователя). В конце слайд с заключением.

Записка предоставляется в электронном виде преподавателю для проверки.

ЭТАП №12.

ЗАЩИТА КУРСОВОЙ РАБОТЫ

К защите курсовой работы допускаются студенты, успешно выполнившие и сдавшие все предыдущие этапы. В рамках защиты каждая группа разработчиков представляет презентацию с информацией о проделанной работе и разработанном программном продукте.

Сдается распечатанная и подписанная пояснительная записка к курсовой работе.

ТЕМЫ КУРСРВОЙ РАБОТЫ

В темах курсовых работ представлены краткие описания того, что требуется сделать. Развернутые описания представлены в приложениях для примера, рассматриваемого в методичке (обучение студентов). В 0 этапе представлено описание заданий для каждой роли с указанием сущностей, с которыми ей необходимо будет работать. В описании к теме курсовой есть подчеркнутые слова – это и есть сущности для конкретного варианта. В приложении к выбранной роли красным цветом выделены сущности по рассматриваемому примеру. Вам вместо этих сущностей нужно будет подставить свои из описания к варианту. **Синим** цветом помечены элементы, которые нужно будет дописать **самостоятельно**.

Например, если выбран первый вариант и первая роль (вариант 1.1 и, соответственно, роль Клиент), то здесь в качестве «сущности 1», описываемой в задании для Клиента будет выступать «визит». Второй сущностью будет «оплата» (вторая сущность практически у всех ролей и во всех вариантах одинаковая). В приложении 4 описывается ТЗ к роли «Клиент». Следует взять за шаблон это ТЗ и заменить все выделенные красным слова «обучение» на «визит». Также, потребуется переработать цель программы (первый абзац), скорректировать по логике варианта общие фразы и дополнить то, что выделено синим цветом.

№	Тема	Описание
1. Салон красоты «Вы ужасны»		
1.1	Салон красоты «Вы ужасны». Работник	Ведение учета оказываемых услуг в салоне красоты. Клиенты могут создавать <u>заказы (1)</u> , <u>процедуры (2)</u> , которые они будут получать в рамках заказа, и проставлять <u>оценки (3)</u> за процедуры. Программа предоставляет возможность получения отчетов по заказам и услугам к ним.
1.2	Салон красоты «Вы ужасны». Кладовщик	Ведение учета оказываемых услуг в салоне красоты. Сотрудники могут формировать список предлагаемых <u>услуг (4)</u> и списки <u>косметики (5)</u> к ним, а также <u>трудоzатраты (6)</u> на оказания услуг клиентам. Программа предоставляет возможность получения отчетов по оказываемым услугам.
1.3	Салон красоты «Вы ужасны». Исполнитель	Ведение учета выдаваемой клиентам косметики. Клиенты указывают <u>процедуры (7)</u> , под которые они могут <u>покупать (8)</u> косметику и <u>посещения (9)</u> процедур. Программа предоставляет возможность получения отчетов по выполненным процедурам.
1.4	Салон красоты «Вы ужасны». Поручитель	Ведение учета выдаваемой клиентам косметики. Сотрудники пробивают <u>чеки (11)</u> на покупаемую <u>косметику (10)</u> и <u>выдачу (12)</u> косметики специалистам при посещении клиентами салона. Программа предоставляет возможность получения отчетов по чекам и выдачам.
1.5	Салон красоты «Вы ужасны». Поставщик	Ведение учета поставляемой салону косметики. Поставщик может фиксировать <u>производство (15)</u> и <u>поставку (14)</u> <u>косметики (13)</u> . Программа предоставляет возможность получения отчетов по производству.
1.6	Салон красоты «Вы ужасны». Заказчик	Ведение учета поставляемой салону косметики. Салоны могут формировать <u>заявки (16)</u> на косметику, записи их <u>поступления (17)</u> и <u>списания (18)</u> . Программа предоставляет возможность получения отчетов по списаниям.

№	Тема	Описание
---	------	----------

2. Школа «Опять учиться»

2.1	Школа «Опять учиться». Работник	Ведение учета кружков к школе. Школьники указывают свои <u>интересы (1)</u> , <u>занятия (2)</u> , на которых они хотят их осваивать, и проставляют свои <u>достижения (3)</u> за занятия. Программа предоставляет возможность получения отчетов по интересам школьников.
2.2	Школа «Опять учиться». Кладовщик	Ведение учета кружков к школе. Учителя могут формировать список проводимых <u>кружков (4)</u> и списки <u>материалов (5)</u> , которые требуются на них, а также <u>медальки (6)</u> школьникам за успешную работу на кружках. Программа предоставляет возможность получения отчетов по проводимым кружкам.
2.3	Школа «Опять учиться». Исполнитель	Ведение учета изготовления поделок школьниками. Школьники указывают <u>интересы (7)</u> , изготавливают дома <u>поделки (8)</u> и <u>изделия (9)</u> на занятиях. Программа предоставляет возможность получения отчетов по интересам школьников.
2.4	Школа «Опять учиться». Поручитель	Ведение учета изготовления поделок школьниками. Учителя выдают <u>задания на поделки (11)</u> и проводят <u>занятия (12)</u> , выдавая школьникам <u>материалы (10)</u> под это. Программа предоставляет возможность получения отчетов по заданиям и занятиям.
2.5	Школа «Опять учиться». Поставщик	Ведение учета конкурсов поделок школьников. Учитель может фиксировать <u>поступление (15)</u> и <u>списание (14)</u> <u>материалов (13)</u> для изготовления поделок школьниками. Программа предоставляет возможность получения отчетов по поступлениям.
2.6	Школа «Опять учиться». Заказчик	Ведение учета конкурсов поделок школьников. Школьники могут делать <u>поделки (16)</u> из материалов учителя, выставлять их на <u>конкурсы (17)</u> и получать <u>награды (18)</u> . Программа предоставляет возможность получения отчетов по полученным наградам.

№	Тема	Описание
---	------	----------

3. Столовая «Рога и копыта»

3.1	Столовая «Рога и копыта». Работник	Ведение учета обедов в столовой. Посетители указывают свои <u>обеда (1)</u> , <u>заказы (2)</u> , которые они совершали, и какими <u>приборами (3)</u> пользовались. Программа предоставляет возможность получения отчетов по обедам.
3.2	Столовая «Рога и копыта». Кладовщик	Ведение учета обедов в столовой. Управляющие могут формировать список <u>поваров (4)</u> и списки <u>продуктов (5)</u> , которые им требуются, а также <u>блюда (6)</u> которые они готовят. Программа предоставляет возможность получения отчетов по поварам.
3.3	Столовая «Рога и копыта». Исполнитель	Ведение учета заказов. Управляющие указывают <u>официантов (7)</u> , которые обслуживают заказы, <u>заказы (8)</u> и <u>алкогольные карты (9)</u> для клиентов. Программа предоставляет возможность получения отчетов по официантам.
3.4	Столовая «Рога и копыта». Поручитель	Ведение учета заказов. Повара, используя <u>продукты (11)</u> , делают <u>блюда (12)</u> к заказам и <u>напитки (10)</u> для карт. Программа предоставляет возможность получения отчетов по продуктам и блюдам.
3.5	Столовая «Рога и копыта». Поставщик	Ведение учета продуктов. Хранители могут <u>поставлять (15)</u> и <u>хранить (14)</u> <u>продукты (13)</u> , требуемые для изготовления блюд. Программа предоставляет возможность получения отчетов по поставкам.
3.6	Столовая «Рога и копыта». Заказчик	Ведение учета продуктов. Повара могут делать <u>блюда (16)</u> из продуктов с хранилищ, добавлять их к <u>заказам (17)</u> и указывать к заказам <u>комплектации (18)</u> . Программа предоставляет возможность получения отчетов по комплектациям.

№	Тема	Описание
---	------	----------

4. Турфирма «Иван Сусанин»

4.1	Турфирма «Иван Сусанин». Работник	Ведение учета экскурсий в путешествиях. Туристы создают свои <u>путешествия (1)</u> , <u>экскурсии (2)</u> в них и какие <u>места (3)</u> они посетили. Программа предоставляет возможность получения отчетов по путешествиям.
4.2	Турфирма «Иван Сусанин». Кладовщик	Ведение учета экскурсий в путешествиях. Операторы указывают <u>гидов (4)</u> , проводящих экскурсии и <u>туры (5)</u> , а также <u>остановки (6)</u> клиентов в турах. Программа предоставляет возможность получения отчетов по задействованным гидам.
4.3	Турфирма «Иван Сусанин». Исполнитель	Ведение учета экскурсий. Турфирмы указывают <u>туры (7)</u> , для <u>экскурсий (8)</u> и больших <u>экскурсионных групп (9)</u> . Программа предоставляет возможность получения отчетов по турам.
4.4	Турфирма «Иван Сусанин». Поручитель	Ведение учета экскурсий. Организаторы, нанимают <u>гидов (11)</u> , указывают <u>места для посещения (12)</u> к экскурсиям и <u>поездки (10)</u> для больших групп. Программа предоставляет возможность получения отчетов по гидам и местам.
4.5	Турфирма «Иван Сусанин». Поставщик	Ведение учета туров. Организаторы могут набирать <u>гидов (13)</u> , указывать их <u>накопительный опыт (15)</u> и указывать <u>места (14)</u> под экскурсии. Программа предоставляет возможность получения отчетов по накопительному опыту.
4.6	Турфирма «Иван Сусанин». Заказчик	Ведение учета туров. Турфирмы могут создавать <u>туры (17)</u> , <u>экскурсии (16)</u> к ним и указывать <u>заезды (18)</u> в рамках туров. Программа предоставляет возможность получения отчетов по заездам.

№	Тема	Описание
---	------	----------

5. Гостиница «Принцесса на горошине»

5.1	Гостиница «Принцесса на горошине». Работник	Ведение учета конференций в гостинице. Организаторы создают <u>конференции (1)</u> , <u>семинары (2)</u> в них и <u>участников (3)</u> семинаров. Программа предоставляет возможность получения отчетов по конференциям.
5.2	Гостиница «Принцесса на горошине». Кладовщик	Ведение учета конференций в гостинице. Метрдотели указывают <u>обеда (4)</u> , подаваемые на семинары и в номера, <u>номера (5)</u> под конференции, а также <u>постояльцев (6)</u> номеров. Программа предоставляет возможность получения отчетов по предоставляемым обедам.
5.3	Гостиница «Принцесса на горошине». Исполнитель	Ведение учета питания и проживания участников конференций. Организаторы указывают <u>участников (7)</u> , <u>планы питания для них (8)</u> и <u>конференции (9)</u> . Программа предоставляет возможность получения отчетов по участникам.
5.4	Гостиница «Принцесса на горошине». Поручитель	Ведение учета питания и проживания участников конференций. Метрдотели, указывают <u>номера (11)</u> , <u>брони по конференциям (12)</u> и <u>обеда (10)</u> , подаваемые в номера. Программа предоставляет возможность получения отчетов по номерам и конференциям.
5.5	Гостиница «Принцесса на горошине». Поставщик	Ведение учета уборок номеров. Уборщики указывают <u>комплекты для подготовки номеров (13)</u> , из чего они <u>состоят (15)</u> и <u>записи по уборкам номеров (14)</u> . Программа предоставляет возможность получения отчетов по используемым элементам комплектов.
5.6	Гостиница «Принцесса на горошине». Заказчик	Ведение учета уборок номеров. Метрдотели могут создавать <u>брони (17)</u> , к <u>номерам (16)</u> и <u>постояльцев (18)</u> , вселяемых по броням. Программа предоставляет возможность получения отчетов по постояльцам.

№	Тема	Описание
---	------	----------

6. Поликлиника «Будьте больны»

6.1	Поликлиника «Будьте больны». Работник	Ведение учета лечения пациентов. Врачи указывают <u>пациентов (1)</u> , выписывают <u>рецепты (2)</u> им и под какие <u>болезни (3)</u> . Программа предоставляет возможность получения отчетов по пациентам.
6.2	Поликлиника «Будьте больны». Кладовщик	Ведение учета лечения пациентов. Медсестры указывают используемые <u>лекарства (4)</u> , в каких <u>процедурах (5)</u> они используются, а также <u>описание процедур для врачей (6)</u> . Программа предоставляет возможность получения отчетов по предоставляемым лекарствам.
6.3	Поликлиника «Будьте больны». Исполнитель	Ведение учета лечения болезней. Врачи описывают <u>болезни (7)</u> , <u>курсы приема препаратов (8)</u> и <u>симптоматику (9)</u> . Программа предоставляет возможность получения отчетов по болезням.
6.4	Поликлиника «Будьте больны». Поручитель	Ведение учета лечения болезней. Медсестры, указывают <u>лекарства (11)</u> , <u>рецепты по ним (12)</u> и <u>процедуры (10)</u> , для уменьшения симптоматики болезней. Программа предоставляет возможность получения отчетов по лекарствам и рецептам.
6.5	Поликлиника «Будьте больны». Поставщик	Ведение учета лечения пациентов. Аптекари указывают <u>лекарства (13)</u> , фиксируют их <u>поступления (15)</u> и <u>рецепты (14)</u> , в которых они используются. Программа предоставляет возможность получения отчетов по поступлениям.
6.6	Поликлиника «Будьте больны». Заказчик	Ведение учета лечения пациентов. Врачи могут создавать <u>процедуры (17)</u> , <u>лечения (16)</u> , в которых они применяются, и <u>пациентов (18)</u> , которым назначают лечения. Программа предоставляет возможность получения отчетов по пациентам.

№	Тема	Описание
---	------	----------

7. Университет «Все отчислены»

7.1	Университет «Все отчислены». Работник	Ведение учета успеваемости студентов. Деканаты указывают <u>планы обучения (1)</u> , прописывают <u>студентов (2)</u> и <u>аттестацию (3)</u> по ним. Программа предоставляет возможность получения отчетов по планам обучения.
7.2	Университет «Все отчислены». Кладовщик	Ведение учета успеваемости студентов. Кафедры указывают проводимые <u>дисциплины (4)</u> , какие <u>преподаватели (5)</u> их ведут, а также <u>промежуточные ведомости (6)</u> . Программа предоставляет возможность получения отчетов по проводимым дисциплинам.
7.3	Университет «Все отчислены». Исполнитель	Ведение учета успеваемости студентов. Преподаватели указывают <u>студентов (7)</u> , <u>результаты испытаний (8)</u> и <u>ведомости (9)</u> . Программа предоставляет возможность получения отчетов по студентам.
7.4	Университет «Все отчислены». Поручитель	Ведение учета успеваемости студентов. Кафедры, указывают <u>типы отчетности (10)</u> , как их применять на <u>занятиях (11)</u> и в <u>дисциплинах (12)</u> . Программа предоставляет возможность получения отчетов по занятиям и дисциплинам.
7.5	Университет «Все отчислены». Поставщик	Ведение учета распределения студентов. Деканаты вводят <u>студентов (13)</u> , через <u>приказы (14)</u> , и отслеживают их <u>статусы обучения (15)</u> . Программа предоставляет возможность получения отчетов по статусам обучения.
7.6	Университет «Все отчислены». Заказчик	Ведение учета распределения студентов. Университеты задают <u>группы обучения (16)</u> , объединяют их в <u>потоки (17)</u> для преподавания <u>дисциплин (18)</u> . Программа предоставляет возможность получения отчетов по дисциплинам.

№	Тема	Описание
---	------	----------

8. Автоцентр «Корыто»

8.1	Автоцентр «Корыто». Работник	Ведение учета продажи машин. Покупатели создают <u>покупки (1)</u> , прописывают необходимые <u>предпродажные работы (2)</u> и <u>пожелания (3)</u> к ним. Программа предоставляет возможность получения отчетов по покупкам.
8.2	Автоцентр «Корыто». Кладовщик	Ведение учета продажи машин. Дилеры указывают продаваемые <u>машины (5)</u> , какими <u>комплектациями (4)</u> их можно оснащать, а также <u>особенности (6)</u> конкретных машин. Программа предоставляет возможность получения отчетов по используемым комплектациям.
8.3	Автоцентр «Корыто». Исполнитель	Ведение учета предпродажной подготовки машин. Дилеры создают <u>продажи (7)</u> , <u>предпродажные работы (8)</u> и <u>сотрудников (9)</u> , которые выполняют подготовку машин. Программа предоставляет возможность получения отчетов по продажам.
8.4	Автоцентр «Корыто». Поручитель	Ведение учета предпродажной подготовки машин. Руководители, указывают продаваемые <u>машины (10)</u> , <u>комплектации (11)</u> к ним и проводимые <u>осмотры (12)</u> . Программа предоставляет возможность получения отчетов по комплектациям и осмотрам.
8.5	Автоцентр «Корыто». Поставщик	Ведение учета комплектаций машин. Начальники вводят <u>комплектации (13)</u> , на какие <u>виды работ (14)</u> они распространяются, и их <u>поступления (15)</u> . Программа предоставляет возможность получения отчетов по поступлениям.
8.6	Автоцентр «Корыто». Заказчик	Ведение учета распределения студентов. Работники задают <u>предпродажные работы (16)</u> , <u>машины (17)</u> для которых они выполняются и их <u>продажи (18)</u> . Программа предоставляет возможность получения отчетов по продажам.

№	Тема	Описание
---	------	----------

9. Магазин компьютерной техники «Ты ж программист»

9.1	Магазин компьютерной техники «Ты ж программист». Работник	Ведение учета продажи техники. Покупатели создают <u>покупки (1)</u> , прописывают необходимые <u>сборки (2)</u> и <u>комментарии (3)</u> к сборкам. Программа предоставляет возможность получения отчетов по покупкам.
9.2	Магазин компьютерной техники «Ты ж программист». Кладовщик	Ведение учета продажи машин. Продавцы указывают продаваемые <u>товары (5)</u> , <u>комплектующие (4)</u> , а также <u>заказы на товары (6)</u> . Программа предоставляет возможность получения отчетов по комплектующим.
9.3	Магазин компьютерной техники «Ты ж программист». Исполнитель	Ведение учета заказов техники. Продавцы формируют <u>заказы (7)</u> , из <u>партий товаров (8)</u> и <u>заявок (9)</u> . Программа предоставляет возможность получения отчетов по заказам.
9.4	Магазин компьютерной техники «Ты ж программист». Поручитель	Ведение учета заказов техники. Сотрудники, указывают из имеющихся <u>комплектующих (10)</u> , делают готовые <u>товары (11)</u> и персональные <u>сборки (12)</u> по заявкам. Программа предоставляет возможность получения отчетов по товарам и сборкам.
9.5	Магазин компьютерной техники «Ты ж программист». Поставщик	Ведение учета поставок техники. Сборщики вводят <u>комплектующие (13)</u> , в какие <u>сборки (14)</u> они входят, и их <u>закупки (15)</u> . Программа предоставляет возможность получения отчетов по закупкам.
9.6	Магазин компьютерной техники «Ты ж программист». Заказчик	Ведение учета поставок техники. Сотрудники создают <u>заказы (16)</u> , <u>поставки (17)</u> по ним и <u>получения техники (18)</u> по поставкам. Программа предоставляет возможность получения отчетов по получению техники.

№	Тема	Описание
---	------	----------

10. Завод «Иди работать»

10.1	Завод «Иди работать». Работник	Ведение учета изготовления изделий. Инженеры указывают <u>заготовки (1)</u> , из которых будут делаться изделия, прописывают <u>планы производства (2)</u> и <u>этапы выполнения (3)</u> . Программа предоставляет возможность получения отчетов по заготовкам.
10.2	Завод «Иди работать». Кладовщик	Ведение учета изготовления изделий. Мастера указывают имеющиеся <u>станки (4)</u> , производимые <u>изделия (5)</u> , а также <u>загруженность станков (6)</u> . Программа предоставляет возможность получения отчетов по используемым станкам.
10.3	Завод «Иди работать». Исполнитель	Ведение учета изготовления изделий. Инженеры формируют <u>изделия (7)</u> , из каких <u>деталей (8)</u> они состоят и какие <u>производства (9)</u> будут задействованы в изготовлении. Программа предоставляет возможность получения отчетов по изделиям.
10.4	Завод «Иди работать». Поручитель	Ведение учета изготовления изделий. Начальники, указывают <u>работников (10)</u> , <u>станки (11)</u> , на которых они работают и <u>цеха (12)</u> , в которых они работают. Программа предоставляет возможность получения отчетов по станкам и цехам.
10.5	Завод «Иди работать». Поставщик	Ведение учета изготовления изделий. Начальники указывают <u>работников (13)</u> , какие <u>станки (14)</u> они используют в работе, и их <u>смены (15)</u> . Программа предоставляет возможность получения отчетов по сменам.
10.6	Завод «Иди работать». Заказчик	Ведение учета изготовления изделий. Инженеры описывают <u>изделия (17)</u> , <u>детали (16)</u> , из которых они состоят, и <u>акты приемки (18)</u> изделий. Программа предоставляет возможность получения отчетов по актам приемки.

№	Тема	Описание
---	------	----------

11. Ветклиника «Айболит»

11.1	Ветклиника «Айболит». Работник	Ведение учета оказания услуг. Клиенты создают записи <u>визитов (1)</u> , <u>животных (2)</u> , которые в рамках визита будут обследоваться, и <u>прививки (3)</u> животного. Программа предоставляет возможность получения отчетов по визитам.
11.2	Ветклиника «Айболит». Кладовщик	Ведение учета оказания услуг. Врач указывают имеющиеся <u>медикаменты (4)</u> , предлагаемые <u>услуги (5)</u> , а также <u>рекомендации (6)</u> по ним. Программа предоставляет возможность получения отчетов по используемым медикаментам.
11.3	Ветклиника «Айболит». Исполнитель	Ведение учета медикаментов. Питомники указывают <u>животных (7)</u> , какие лекарства для них <u>покупаются (8)</u> и их <u>посещения (9)</u> врачей. Программа предоставляет возможность получения отчетов по животным.
11.4	Ветклиника «Айболит». Поручитель	Ведение учета медикаментов. Врачи, указывают имеющиеся <u>медикаменты (10)</u> , <u>лекарства (11)</u> на их основе и оказываемые <u>услуги (12)</u> . Программа предоставляет возможность получения отчетов по лекарствам и услугам.
11.5	Ветклиника «Айболит». Поставщик	Ведение учета изготовления изделий. Аптекари указывают <u>медикаменты (13)</u> , какие из них делаются <u>лекарства (14)</u> и их <u>закупки (15)</u> . Программа предоставляет возможность получения отчетов по закупкам.
11.6	Ветклиника «Айболит». Заказчик	Ведение учета изготовления изделий. Врачи указывают <u>животных (17)</u> , <u>лечения (16)</u> , которые они проходят, и <u>записи из прививок (18)</u> . Программа предоставляет возможность получения отчетов по записям прививок.

№	Тема	Описание
---	------	----------

12. Банк «Вы банкрот»

12.1	Банк «Вы банкрот». Работник	Ведение учета вкладов и кредитов. Клерки указывают <u>клиентов (1)</u> , их <u>вклады (2)</u> и <u>пополнения (3)</u> этих вкладов. Программа предоставляет возможность получения отчетов по клиентам.
12.2	Банк «Вы банкрот». Кладовщик	Ведение учета вкладов и кредитов. Руководители указывают с какими <u>валютами (4)</u> возможна работа, предлагаемые <u>кредитные программы (5)</u> , а также <u>сроки (6)</u> по ним. Программа предоставляет возможность получения отчетов по используемым валютам.
12.3	Банк «Вы банкрот». Исполнитель	Ведение учета перевода денег. Клиенты указывают <u>карты (7)</u> , <u>заявки (8)</u> на получение с них наличных и <u>операции (9)</u> на пополнения. Программа предоставляет возможность получения отчетов по картам.
12.4	Банк «Вы банкрот». Поручитель	Ведение учета перевода денег. Кассиры, указывают имеющиеся <u>счета (10)</u> , <u>выдачу наличных (11)</u> на их основе и <u>переводы денег (12)</u> . Программа предоставляет возможность получения отчетов по выдачам и переводам.
12.5	Банк «Вы банкрот». Поставщик	Ведение учета выдаваемых кредитов. Банки указывают <u>валюты (13)</u> , <u>кредитные программы (14)</u> и <u>закупки (15)</u> валют. Программа предоставляет возможность получения отчетов по закупкам.
12.6	Банк «Вы банкрот». Заказчик	Ведение учета выдаваемых кредитов. Оператор указывают <u>сделки с клиентами (16)</u> , <u>платы по сделкам (17)</u> и <u>зачисления средств к платам (18)</u> . Программа предоставляет возможность получения отчетов по зачислениям средств.

№	Тема	Описание
---	------	----------

13. Юридическая фирма «Вас обманут»

13.1	Юридическая фирма «Вас обманут». Работник	Ведение учета оказания юридических услуг. Клиент указывают свои <u>дела (1)</u> , <u>заседания (2)</u> по ним и <u>постановления (3)</u> . Программа предоставляет возможность получения отчетов по делам.
13.2	Юридическая фирма «Вас обманут». Кладовщик	Ведение учета оказания юридических услуг. Фирмы указывают своих <u>юристов (4)</u> , оказываемые <u>услуги (5)</u> , а также <u>соглашения (6)</u> по ним. Программа предоставляет возможность получения отчетов по используемым юристам.
13.3	Юридическая фирма «Вас обманут». Исполнитель	Ведение учета работы юристов. Сотрудники указывают <u>клиентов (7)</u> , их <u>визиты (8)</u> и <u>дела (9)</u> . Программа предоставляет возможность получения отчетов по клиентам.
13.4	Юридическая фирма «Вас обманут». Поручитель	Ведение учета работы юристов. Руководители, указывают имеющихся <u>юристов (10)</u> , оказываемые ими <u>консультации (11)</u> и <u>слушания (12)</u> на которых они присутствуют. Программа предоставляет возможность получения отчетов по консультациям и слушаниям.
13.5	Юридическая фирма «Вас обманут». Поставщик	Ведение учета ведения дел. Клиенты указывают <u>дела (13)</u> , <u>договора (14)</u> к ним и <u>слушания (15)</u> по ним. Программа предоставляет возможность получения отчетов по слушаниям.
13.6	Юридическая фирма «Вас обманут». Заказчик	Ведение учета ведения дел. Фирмы указывают <u>заключаемые контракты (16)</u> , <u>юристов (17)</u> , которые по ним работают и их <u>специализации (18)</u> . Программа предоставляет возможность получения отчетов по специализациям.

№	Тема	Описание
---	------	----------

14. СТО «Руки-крюки»

14.1	СТО «Руки-крюки». Работник	Ведение учета обслуживания машин. Работники указывают доступные <u>ТО (1)</u> , <u>машины (2)</u> , которые их проходили, и <u>записи сервисов (3)</u> машин. Программа предоставляет возможность получения отчетов по ТО.
14.2	СТО «Руки-крюки». Кладовщик	Ведение учета обслуживания машин. Начальники указывают <u>работы (5)</u> , которые выполняются в сервисе, <u>запчасти (4)</u> , необходимые для работ и <u>продолжительность работ (6)</u> по ним. Программа предоставляет возможность получения отчетов по используемым запчастям.
14.3	СТО «Руки-крюки». Исполнитель	Ведение учета ремонтных работ. Приемщики указывают <u>машины (7)</u> , их <u>неисправности (8)</u> и проводимые <u>ТО (9)</u> . Программа предоставляет возможность получения отчетов по машинам.
14.4	СТО «Руки-крюки». Поручитель	Ведение учета ремонтных работ. Мастера, указывают имеющиеся <u>запчасти (10)</u> , выполняемые <u>ремонт (11)</u> и оказываемые <u>работы (12)</u> . Программа предоставляет возможность получения отчетов по ремонтам и работам.
14.5	СТО «Руки-крюки». Поставщик	Ведение учета ремонта машин. Старшие мастера указывают <u>запчасти (13)</u> , <u>работы (14)</u> , на которые они списываются, и их <u>получение (15)</u> . Программа предоставляет возможность получения отчетов по получению запчастей.
14.6	СТО «Руки-крюки». Заказчик	Ведение учета ремонта машин. Диагносты указывают <u>машины (17)</u> , необходимый для них <u>ремонт (16)</u> , и их <u>диагностику (18)</u> . Программа предоставляет возможность получения отчетов по диагностикам.

№	Тема	Описание
---	------	----------

15. Фабрика мебели «Мягкое место»

15.1	Фабрика мебели «Мягкое место». Работник	Ведение учета производства мебели. Менеджеры на основе <u>мебельных модулей (1)</u> , формируют <u>гарнитуры (2)</u> и их <u>продажи (3)</u> . Программа предоставляет возможность получения отчетов по мебельным модулям.
15.2	Фабрика мебели «Мягкое место». Кладовщик	Ведение учета производства мебели. Плотники указывают <u>мебель (5)</u> , <u>материалы (4)</u> , необходимые для их создания, и <u>области применения (6)</u> . Программа предоставляет возможность получения отчетов по используемым материалам.
15.3	Фабрика мебели «Мягкое место». Исполнитель	Ведение учета поставок мебели. Менеджеры указывают <u>салоны продаж (7)</u> , <u>заказы (8)</u> на мебель и продаваемые <u>гарнитуры (9)</u> . Программа предоставляет возможность получения отчетов по продажам.
15.4	Фабрика мебели «Мягкое место». Поручитель	Ведение учета поставок мебели. Мастера, указывают имеющиеся <u>материалы (10)</u> , производимую из них <u>мебель (11)</u> и <u>модули для гарнитуров (12)</u> . Программа предоставляет возможность получения отчетов по мебели и модулям.
15.5	Фабрика мебели «Мягкое место». Поставщик	Ведение учета изготовления гарнитуров. Мебельщики указывают <u>материалы (13)</u> , производимые из них <u>модули (14)</u> и их <u>поставки (15)</u> . Программа предоставляет возможность получения отчетов поставкам.
15.6	Фабрика мебели «Мягкое место». Заказчик	Ведение учета изготовления гарнитуров. Сборщики указывают собираемую <u>мебель (16)</u> , в какие <u>гарнитуры (16)</u> она входит, и их <u>отгрузку (18)</u> . Программа предоставляет возможность получения отчетов по отгрузкам.

Список использованных источников

1. Pro Git. 2nd Edition [Электронный ресурс] / Режим доступа: <https://git-scm.com/book/ru/v2>.
2. METANIT.COM. Сайт о программировании [Электронный ресурс] / Режим доступа: <https://metanit.com/sharp/>. – Загл. с экрана.
3. ProfessorWeb. .Net & Web Programming [Электронный ресурс] / Режим доступа: <https://professorweb.ru/>. – Загл. с экрана.
4. Tiberiu Covaci, Rod Stephens, Vincent Varallo, Gerry O'Brien. MCSD Certification Toolkit (Exam 70-483) // Published by John Wiley & Sons, Inc. – 2013. – 656p.
5. MCTS Self-Paced Training Kit (Exam 70-536): Microsoft .NET Framework–Application Development Foundation, Second Edition eBook // Published by Microsoft Press. – 2009. – 829 p.
6. Баби́ч А.В. Введение в UML [Электронный ресурс]/ Баби́ч А.В.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 198 с.— Режим доступа: https://www.intuit.ru/goods_store/ebooks/8359.
7. Informicus. Диаграммы вариантов использования [Электронный ресурс] / Режим доступа: <http://www.informicus.ru/default.aspx?SECTION=6&id=73&subdivisionid=4>. – Загл. с экрана.
8. Теория и практика UML. Диаграмма последовательности[Электронный ресурс] / Режим доступа: http://it-gost.ru/articles/view_articles/94. – Загл. с экрана.

ПРИЛОЖЕНИЕ 1

Титульный лист

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет ИСТ
Кафедра «Информационные системы»
Дисциплина «Технологии программирования»

КУРСОВАЯ РАБОТА

Тема указать свою тему по приказу

Выполнил студент

подпись

/Фамилия И.О./
инициалы, фамилия

Курс второй

Группа ПИбд-21

Направление/специальность 09.03.04 «Программная инженерия»

Руководитель

ст. преподаватель
должность, ученая степень, ученое звание

Эгов Е.Н.

фамилия, имя, отчество

Дата сдачи:

«___» _____ 2020 г.

Дата защиты:

«___» _____ 2020 г.

Оценка: _____

Ульяновск
2020 г.

ПРИЛОЖЕНИЕ 2

Задание на курсовую работу

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет ИСТ
Кафедра «Информационные системы»
Дисциплина «Технологии программирования»

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студенту ПИБД-21 Фамилия И.О.
группа фамилия, инициалы

Тема работы указать свою тему по приказу

Срок сдачи законченной работы «__» _____ 2020 г.

Исходные данные к работе: описание задания по теме, утвержденной распоряжением деканата ФИСТ

Рекомендуемая литература: курс лекций по дисциплине «Технологии программирования», методические указания к лабораторным работам по дисциплине «Технологии программирования», интернет-источники.

Содержание пояснительной записки (перечень подлежащих разработке вопросов)

Введение. Описание актуальности задачи.

Первая глава. Описание предметной области, поиск аналогов, ТЗ.

Вторая глава. Представление диаграмм с их описанием.

Третья глава. Представление руководств пользователя и программиста для разработанного проекта

Четвертая глава. Тестирование. Приводится пример работы всего функционала, озвученного в ТЗ и use-case диаграмме.

Перечень графического материала (с точным указанием обязательных чертежей)

Диаграммы UML: диаграмма вариантов использования (use-case), диаграмма последовательности (sequence), диаграмма развертывания (deployment).

ER-диаграмма.

Скриншоты разработанного программного продукта

Руководитель ст. преподаватель _____ /Эгов Е.Н./
должность подпись инициалы, фамилия

«__» _____ 2020 г.

Студент _____ /Фамилия И.О./
подпись инициалы, фамилия

«__» _____ 2020 г.

Отзыв руководителя

ОТЗЫВ

руководителя на курсовую работу

Отмечаются следующие моменты: актуальность темы исследования; соответствие содержания и структуры курсовой работы ее теме; степень разработанности проблемы, наиболее интересно исследованные вопросы. Оценивается степень самостоятельности и инициативы студента; умение пользоваться различными источниками информации; уровень его теоретической подготовки; умение анализировать научные материалы, делать практические выводы; знание основных концепций, научной и специальной литературы по избранной теме. Содержится оценка проекта (работы) руководителем.

[illegible]

_____ /Эгов Е.Н./
подпись инициалы, фамилия

« » 2020 г.

Шаблон технического задания для роли «Работник (первая роль)»

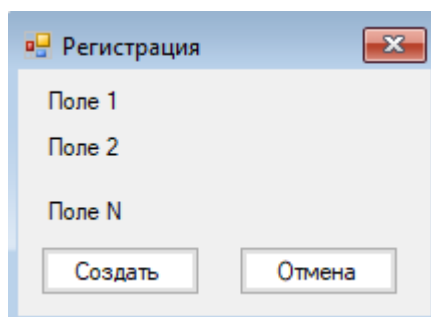
Техническое задание на разработку программного продукта «Учет литературы. Работник»

Программа «Учет литературы. Работник» предназначена для ведения учета выдаваемой студентам литературы для изучения на занятиях. Студенты могут создавать обучения и курсы, на основе которых им будут выдаваться литература и проставлять оценки за курс. Программа предоставляет возможность получения отчетов по выдаваемой литературе для студентов.

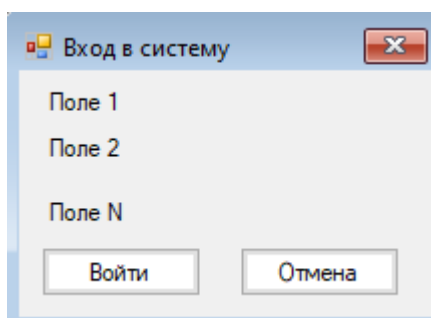
Основной функционал программы:

- Регистрация. Для регистрации пользователь должен заполнить следующие данные:

Данные от пользователя. Указать требования к каждому полю. Пример: логин. Логин должен быть уникальным, иметь длину не более 50 символов. Обязателен к заполнению.

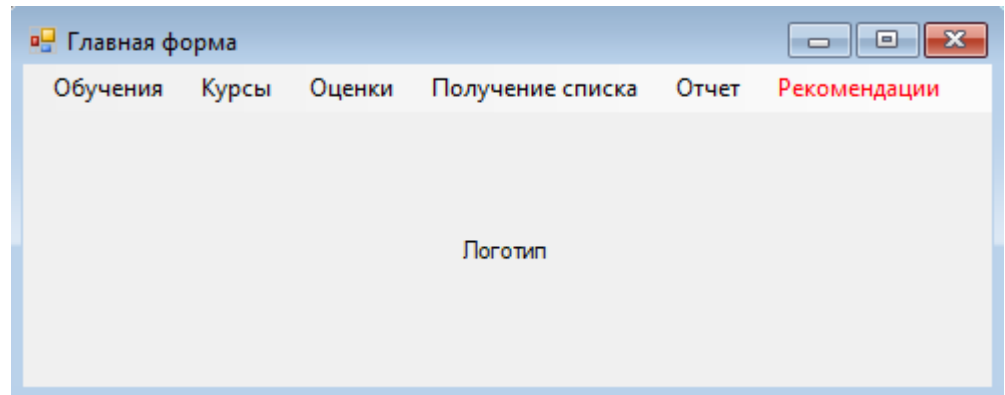


- Авторизация. Пользователь вводит данные. Если такой пользователь есть в системе, то происходит переход к основной форме. Если такого пользователя нет в системе, выводится сообщение об этом с просьбой ввести заново данные.



- Основная форма. Состоит из пунктов меню и логотипа. Через пункты меню можно перейти на формы работы с **обучениями**,

курсами (в том числе и **привязка курсов**), **оценками**, на форму получения списка **литературы** и на форму получения отчета. Дополнительно, если реализовано, то вызов формы рекомендательной подсистемы.

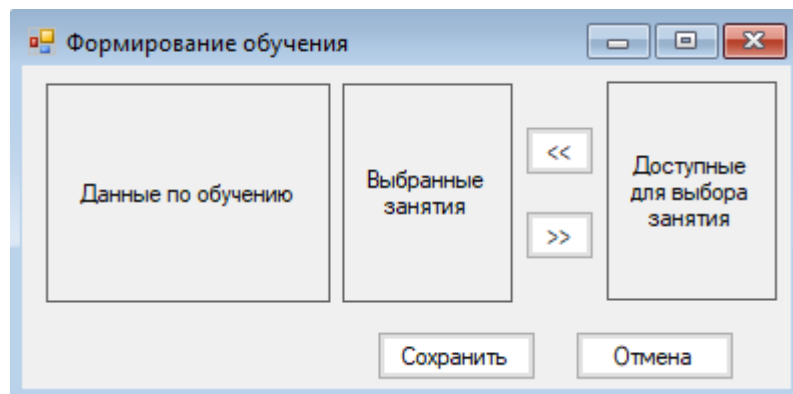


- Формирование **обучений** (CRUD). Имеется форма со списком всех **обучений** пользователя и кнопки для создания, изменения и удаления **обучения**.

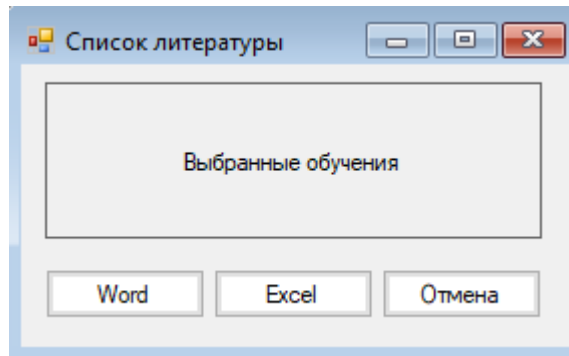
Пользователь может создать **запись обучения**, включающую следующие поля:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступные **занятия** из списка (несколько) и сохранить изменения.



- Получение списка. Пользователь может получить список **литературы** по выбранным **обучениям**. Пользователь отмечает интересующие его **обучения**, указывает формат файла (doc или xls), указывает имя и место сохранения файла и получает интересующий его список.



При выводе в doc-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

При выводе в xls-файл формат документа будет следующим:

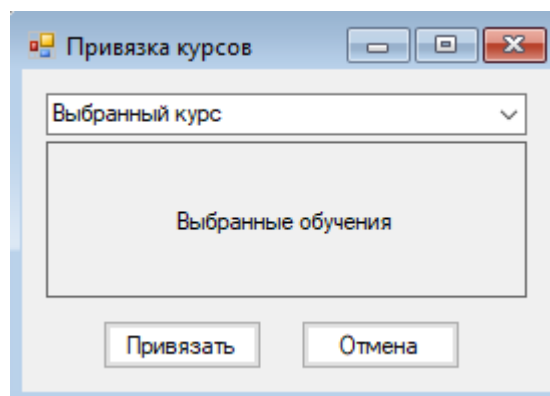
указать как будет выглядеть документ, можно приложить скрин.

- Формирование **курсов** (CRUD). Имеется форма со списком всех **курсов** пользователя и кнопки для создания, изменения и удаления **курса**.

Запись **курса** включает в себя следующие поля:

Данные для записи (2-3 поля).

- Привязка **курсов** к **обучениям**. Пользователь может выбрать **курс**, **обучения** и связать их.

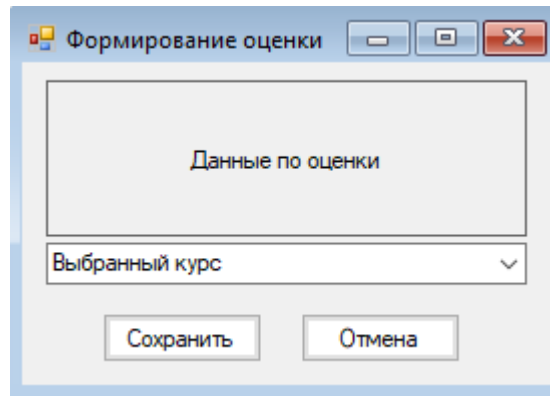


- Формирование **оценок от преподавателя** (CRUD). Имеется форма со списком всех **оценок** пользователя и кнопки для создания, изменения и удаления **оценки**.

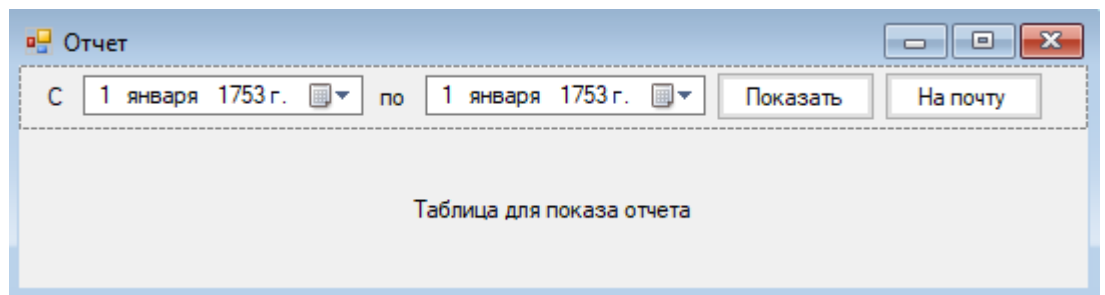
Запись **оценки** включает в себя следующие поля:

Данные для записи (2-3 поля).

Имеется возможность выбрать запись **курса** из списка к записи **оценки**.



- Пользователь может получить сведения по полученной им литературе и полученным оценкам за период по различным обучением. Для этого он должен указать интересующий его период, выбрать вариант получения сведений (отправка на почту или вывод на форму) и получить отчет.



Указать, как будет выглядеть отчет, можно приложить скрин.

Если проект разрабатывается индивидуально, то указать следующие пункты:

- Метод создания литературы. Определить, когда должен вызываться метод. Описать как будет выполняться работа метода (либо случайно, либо по какому-то алгоритму).
- Метод создания занятий. Определить, когда должен вызываться метод. Описать как будет выполняться работа метода (либо случайно, либо по какому-то алгоритму).
- Метод привязки литературы к занятиям. Метод должен привязывать существующую литературу к созданным занятиям. Определить, когда должен вызываться метод. Описать как будет выполняться работа метода (либо случайно, либо по какому-то алгоритму).

В качестве интерфейса пользователя будет выступать Web-клиент, разработанный на ASP.NET.

Шаблон технического задания для роли «Кладовщик (вторая роль)»

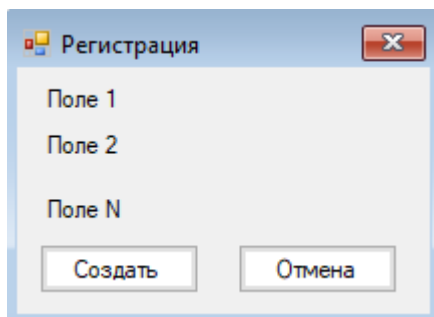
Техническое задание на разработку программного продукта «Учет литературы. Кладовщик»

Программа «Учет литературы. Кладовщик» предназначена для ведения учета выдаваемой студентам литературы для проведения занятий. Преподаватели могут формировать список проводимых занятий и списки литературы к ним, а также акты выставления оценок на занятия. Программа предоставляет возможность получения отчетов по выдаваемой литературе.

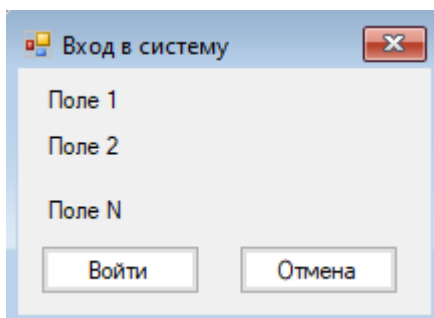
Основной функционал программы:

- Регистрация. Для регистрации пользователь должен заполнить следующие данные:

Данные от пользователя. Указать требования к каждому полю. Пример: логин. Логин должен быть уникальным, иметь длину не более 50 символов. Обязателен к заполнению.

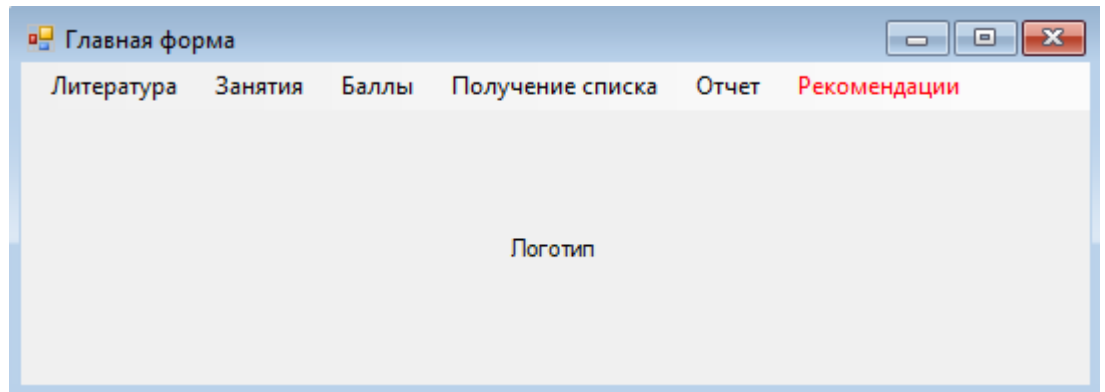


- Авторизация. Пользователь вводит данные. Если такой пользователь есть в системе, то происходит переход к основной форме. Если такого пользователя нет в системе, выводится сообщение об этом с просьбой ввести заново данные.



- Основная форма. Состоит из пунктов меню и логотипа. Через пункты меню можно перейти на формы работы с литературой (в

том числе и **привязка курсов**), **занятиями**, **баллами**, на форму получения списка **курсов** и на форму получения отчета по **литературе**. Дополнительно, если реализовано, то вызов формы рекомендательной подсистемы.



- Формирование **литературы** (CRUD). Имеется форма со списком всей **литературы** пользователя и кнопки для создания, изменения и удаления **литературы**.

По **литературе** хранятся следующие сведения:

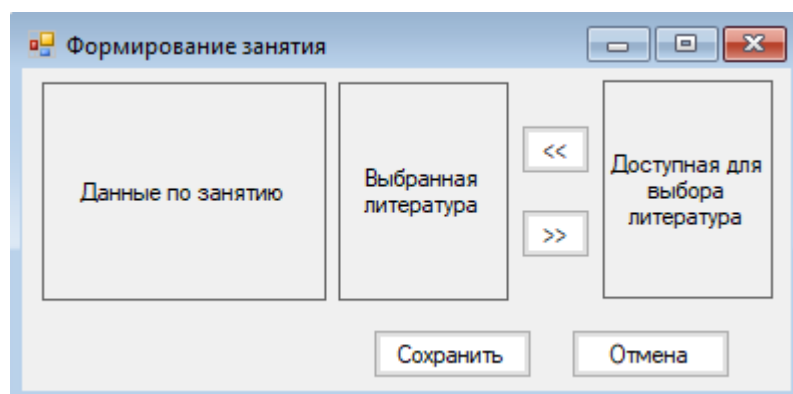
Данные для записи (2-3 поля).

- Формирование **занятиям** (CRUD). Имеется форма со списком всех **занятий** пользователя и кнопки для создания, изменения и удаления **занятия**.

По **занятиям** хранятся следующие сведения:

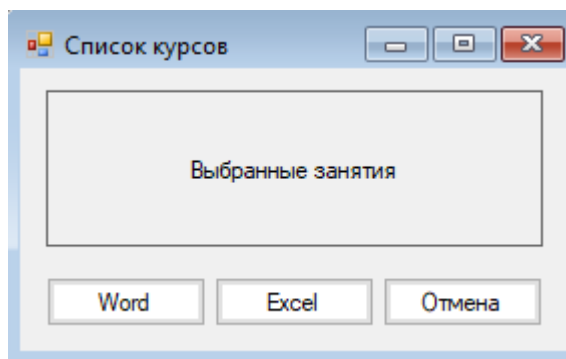
Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступную **литературу** из списка (несколько) и сохранить изменения.



- Получение списка. Пользователь может получить список **курсов** по выбранным **занятиям**. Пользователь отмечает интересующие

его **занятия**, указывает формат файла (doc или xls), указывает имя и место сохранения файла и получает интересующий его список.



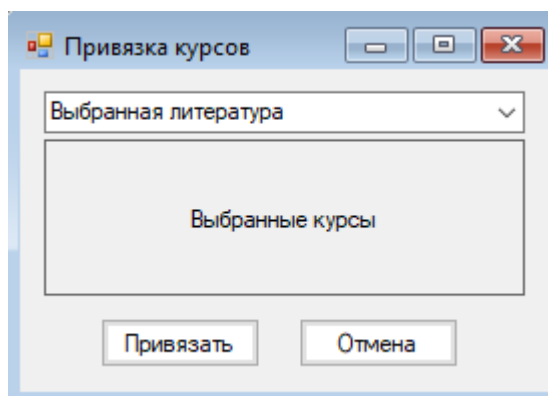
При выводе в doc-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

При выводе в xls-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

- Привязка **литературы** к **курсам**. Пользователь может выбрать **литературу**, **курс** и связать их.

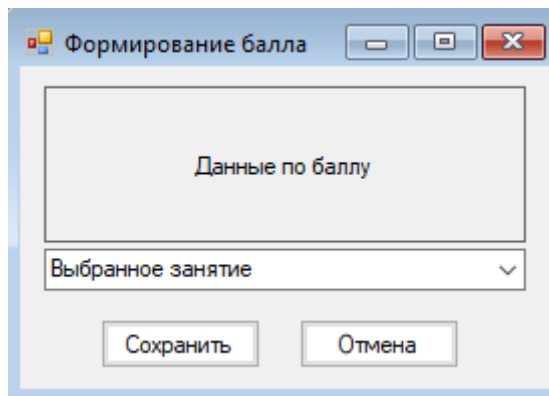


- Формирование **отметок выставления баллов** (CRUD). Имеется форма со списком всех **отметок** пользователя и кнопки для создания, изменения и удаления **отметки**.

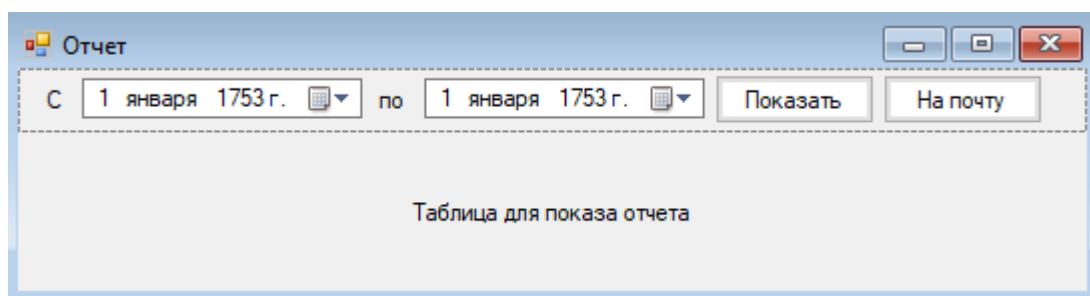
Запись **отметки** включает в себя следующие поля:

Данные для записи (2-3 поля).

Имеется возможность выбрать запись **занятия** из списка к записи **отметки**.



- Пользователь может получить сведения по полученной им литературе за период, с указанием на каких занятиях и курсах она использовалась. Для этого он должен указать интересующий его период, выбрать вариант получения сведений (отправка на почту или вывод на форму) и получить отчет.



Указать, как будет выглядеть отчет, можно приложить скрин.

Если проект разрабатывается индивидуально, то указать следующие пункты:

- Метод создания курсов. Определить, когда должен вызываться метод. Описать как будет проходить формирования (либо случайно, либо по какому-то алгоритму).
- Метод привязки курсов к обучением. Определить, когда должен вызываться метод. Описать как будет проходить формирования (либо случайно, либо по какому-то алгоритму).
- Метод формирования обучений. Метод должен формировать обучения и выбирать туда занятия. Определить, когда должен вызываться метод. Описать как будет проходить формирования (либо случайно, либо по какому-то алгоритму).

В качестве интерфейса пользователя будет выступать desktop-приложение, разработанное на WPF.

ПРИЛОЖЕНИЕ 6

Шаблон технического задания для роли «Исполнитель (третья роль)»

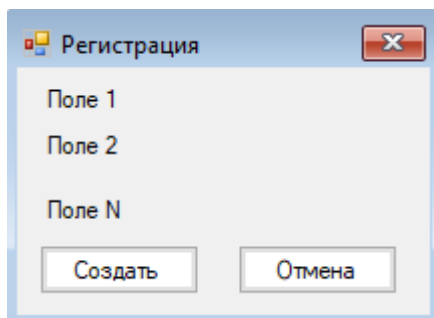
Техническое задание на разработку программного продукта «Учет литературы. Исполнитель»

Программа «Учет литературы. Исполнитель» предназначена для ведения учета выдаваемой студентам в библиотеке литературы для чтения. Студенты указывают занятия, под которые им нужна литература, составляют заявки на получение литературы в библиотеке и отметке о посещении читального зала и какой литературой они там пользовались. Программа предоставляет возможность получения отчетов по заявкам на литературу и посещениям читального зала.

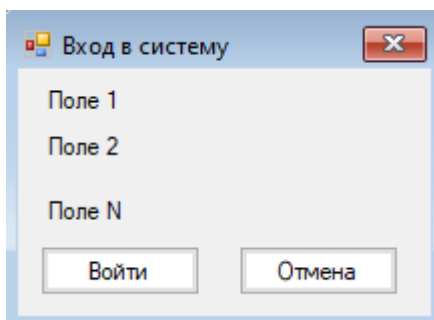
Основной функционал программы:

- Регистрация. Для регистрации пользователь должен заполнить следующие данные:

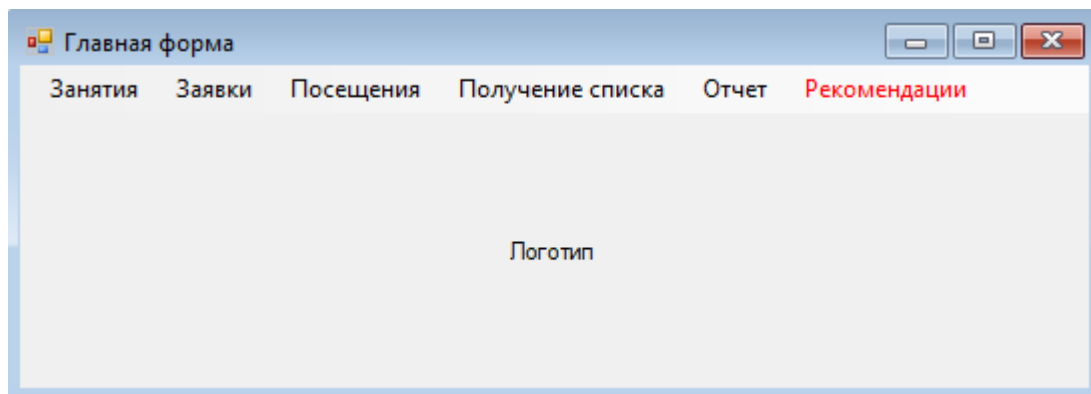
Данные от пользователя. Указать требования к каждому полю. Пример: логин. Логин должен быть уникальным, иметь длину не более 50 символов. Обязателен к заполнению.



- Авторизация. Пользователь вводит данные. Если такой пользователь есть в системе, то происходит переход к основной форме. Если такого пользователя нет в системе, выводится сообщение об этом с просьбой ввести заново данные.



- Основная форма. Состоит из пунктов меню и логотипа. Через пункты меню можно перейти на формы работы с занятиями, заявками, посещениями (в том числе и привязка передач), на форму получения списка занятий и на форму получения отчета по обучением. Дополнительно, если реализовано, то вызов формы рекомендательной подсистемы.



- Формирование занятия (CRUD). Имеется форма со списком всей занятий пользователя и кнопки для создания, изменения и удаления занятия.

По занятию хранятся следующие сведения:

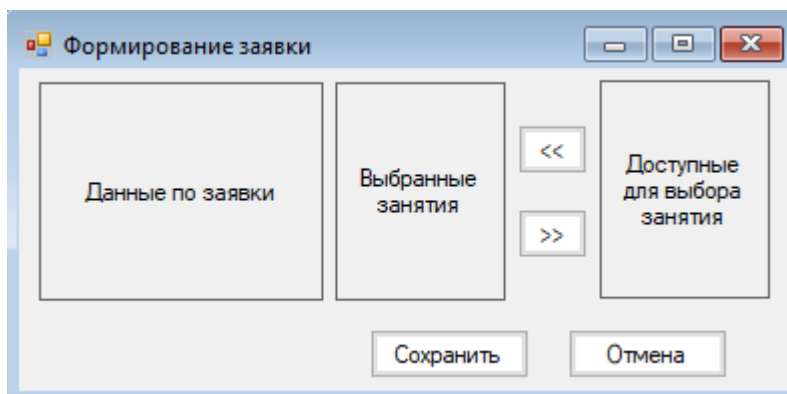
Данные для записи (2-3 поля).

- Формирование заявок на литературу (CRUD). Имеется форма со списком всех заявок пользователя и кнопки для создания, изменения и удаления заявки.

По заявке хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступные занятия из списка (несколько) и сохранить изменения.

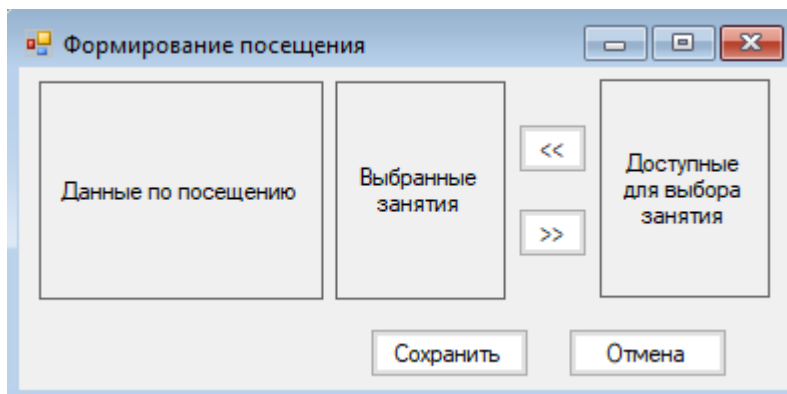


- Формирование **записи посещения читального зала** (CRUD).
Имеется форма со списком всех **посещений** пользователя и кнопки для создания, изменения и удаления **посещения**.

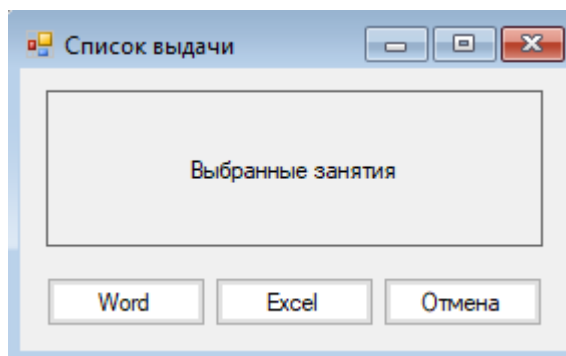
По **посещению** хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступные **занятия** из списка (несколько) и сохранить изменения.



- Получение списка. Пользователь может получить список **записей выдачи литературы** по выбранным **занятиям**. Пользователь отмечает интересующие его **занятия**, указывает формат файла (doc или xls), указывает имя и место сохранения файла и получает интересующий его список.



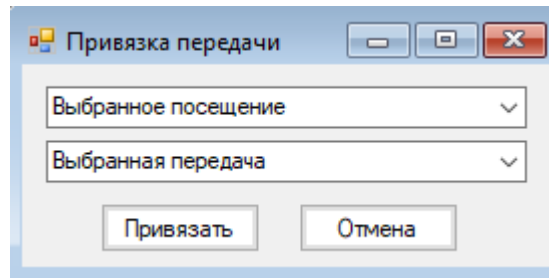
При выводе в дос-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

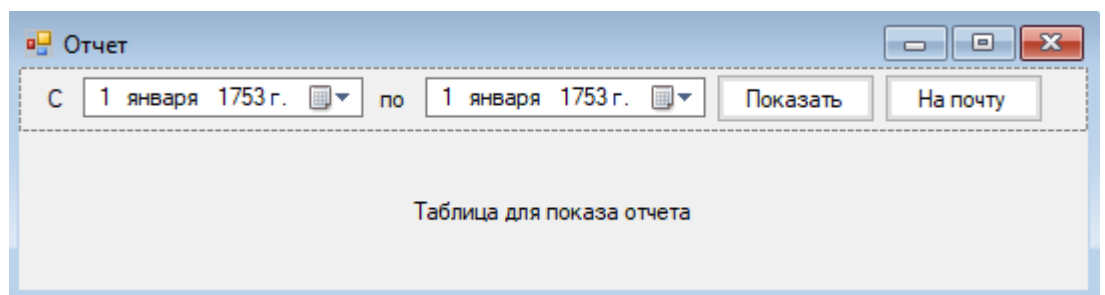
При выводе в xls-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

- Привязка **записи посещения** к **записи передачи литературы**.
Пользователь может выбрать **запись посещения**, **запись передачи** и связать их.



- Пользователь может получить сведения за период по занятиям, для которых требовалось посещения библиотеки и читального зала. Для этого он должен указать интересующий его период, выбрать вариант получения сведений (отправка на почту или вывод на форму) и получить отчет.



Указать, как будет выглядеть отчет, можно приложить скрин.

Если проект разрабатывается индивидуально, то указать следующие пункты:

- Метод создания литературы. Метод может вызываться при старте программы, проверять, есть ли литература в базе данных и, если их нет, создавать ее там (создаваться может случайно, либо загружать из файла (путь до файла можно прописать жестко в коде)).
- Метод создания записей выдачи литературы. Определить, когда должен вызываться метод. Описать как будет проходить создание (либо случайно, либо по определенному алгоритму).
- Метод привязки литературы к записям выдачи литературы. Определить, когда должен вызываться метод. Описать как будет проходить формирования (либо случайно, либо по какому-то алгоритму).

В качестве интерфейса пользователя будет выступать Web-клиент, разработанный на Web API.

Шаблон технического задания для роли «Поручитель (четвертая роль)»

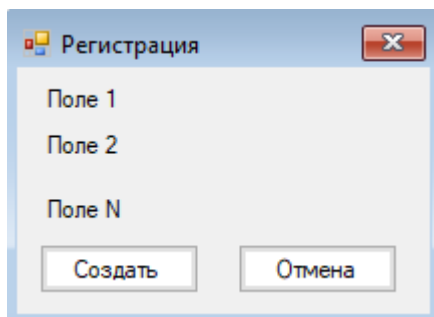
Техническое задание на разработку программного продукта «Учет литературы. Поручитель»

Программа «Учет литературы. Поручитель» предназначена для ведения учета выдаваемой библиотекарем литературы студентам для занятий. Библиотекари могут формировать литературу, записи ее выдачи студентам на занятия и в читальном зале. Программа предоставляет возможность получения отчетов по выдаваемой студентам литературе.

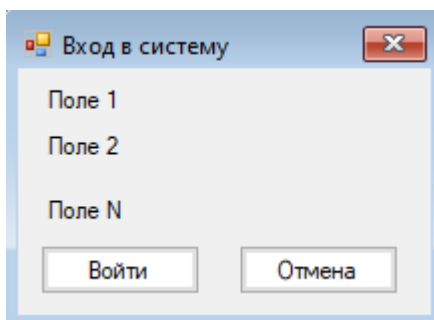
Основной функционал программы:

- Регистрация. Для регистрации пользователь должен заполнить следующие данные:

Данные от пользователя. Указать требования к каждому полю. Пример: логин. Логин должен быть уникальным, иметь длину не более 50 символов. Обязателен к заполнению.

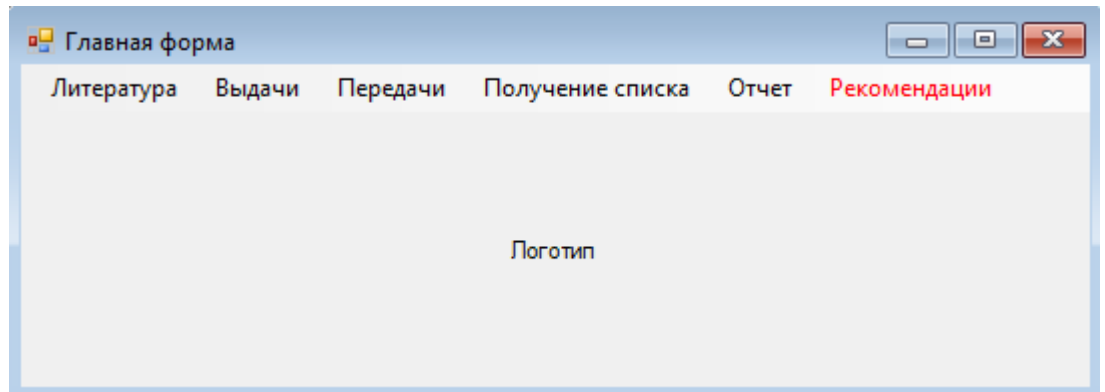


- Авторизация. Пользователь вводит данные. Если такой пользователь есть в системе, то происходит переход к основной форме. Если такого пользователя нет в системе, выводится сообщение об этом с просьбой ввести заново данные.



- Основная форма. Состоит из пунктов меню и логотипа. Через пункты меню можно перейти на формы работы с литературой,

выдачами (в том числе и привязка заявки), передачами, на форму получения списка заявок и на форму получения отчета по литературе. Дополнительно, если реализовано, то вызов формы рекомендательной подсистемы.



- Формирование литературы (CRUD). Имеется форма со списком всей литературы пользователя и кнопки для создания, изменения и удаления литературы.

По литературе хранятся следующие сведения:

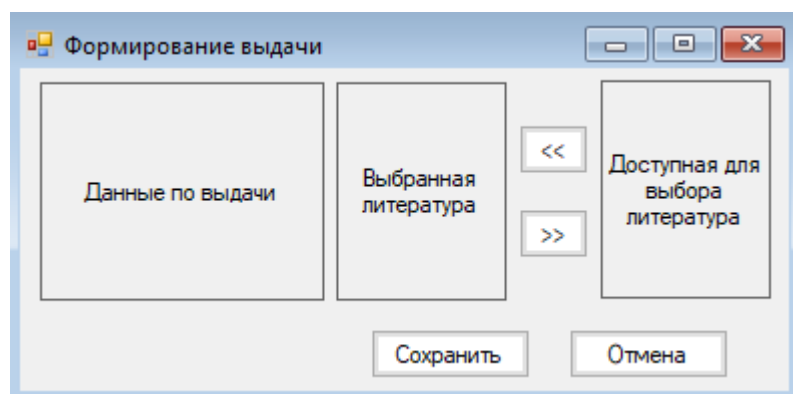
Данные для записи (2-3 поля).

- Формирование записи выдачи литературы (CRUD). Имеется форма со списком всех записей выдач пользователя и кнопки для создания, изменения и удаления записи выдачи.

По записи хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступную литературу из списка (несколько) и сохранить изменения.

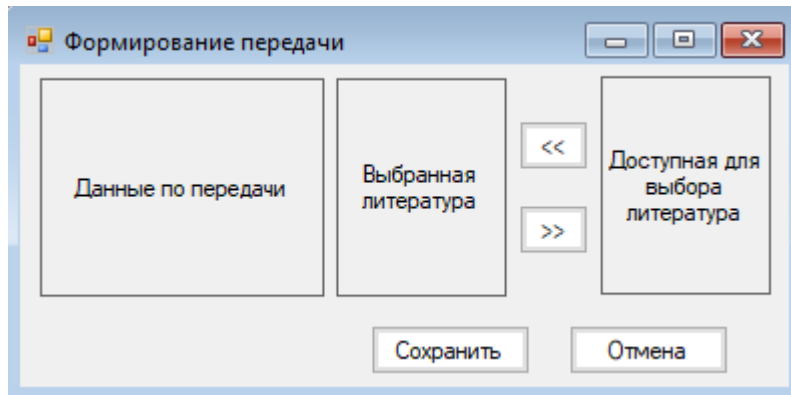


- Формирование записи передачи в руки литературы (CRUD). Имеется форма со списком всех записей передач пользователя и кнопки для создания, изменения и удаления записи передачи.

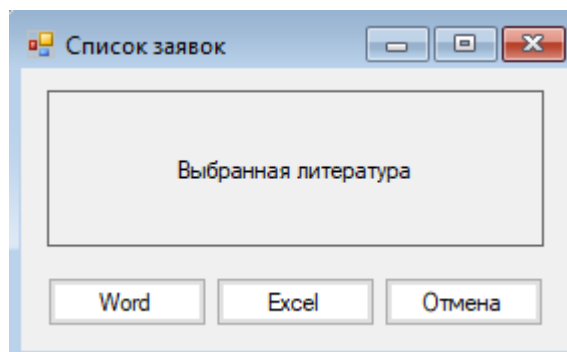
По **записи** хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступную **литературу** из списка (несколько) и сохранить изменения.



- Получение списка. Пользователь может получить список **заявок студентов** по выбранной **литературе**. Пользователь отмечает интересующую его **литературу**, указывает формат файла (doc или xls), указывает имя и место сохранения файла и получает интересующий его список.



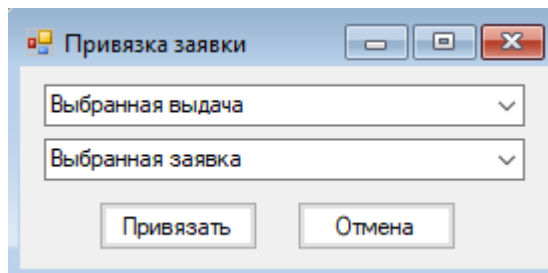
При выводе в doc-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

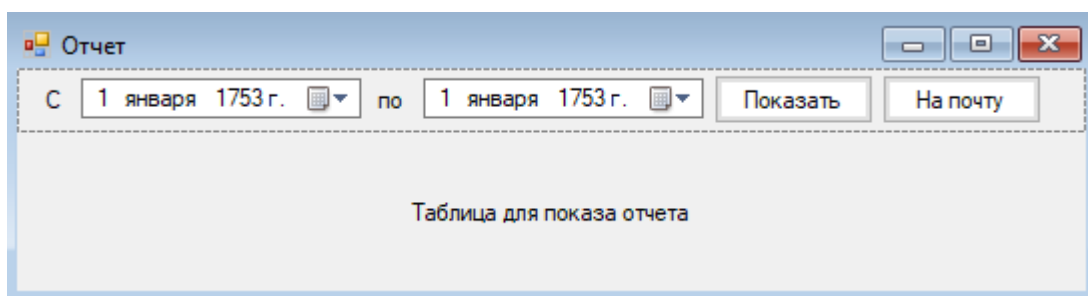
При выводе в xls-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

- Привязка **записи выдачи литературы** к **записи заявки студента**. Пользователь может выбрать **запись выдачи**, **запись заявки** и связать их.



- Пользователь может получить сведения за период по литературе, которая указывалась в записях выдачи и передачах на руки. Для этого он должен указать интересующий его период, выбрать вариант получения сведений (отправка на почту или вывод на форму) и получить отчет.



Указать, как будет выглядеть отчет, можно приложить скрин.

Если проект разрабатывается индивидуально, то указать следующие пункты:

- Метод создания занятий. Метод может вызываться при старте программы, проверять, есть ли занятия в базе данных и, если их нет, создавать ее там (создаваться может случайно, либо загружать из файла (путь до файла можно прописать жестко в коде)).
- Метод создания записей заявок студентов. Определить, когда должен вызываться метод. Описать как будет проходить создание (либо случайно, либо по определенному алгоритму).
- Метод привязки занятий к записям заявок. Определить, когда должен вызываться метод. Описать как будет проходить формирования (либо случайно, либо по какому-то алгоритму).

В качестве интерфейса пользователя будет выступать desktop-приложение, разработанное на WPF.

ПРИЛОЖЕНИЕ 8

Шаблон технического задания для роли «Поставщик (пятая роль)»

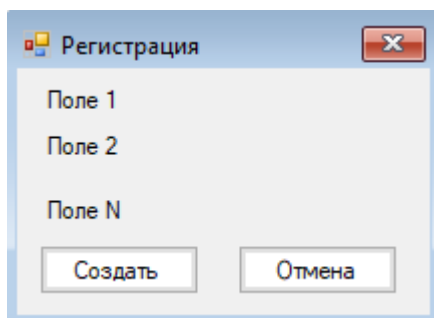
Техническое задание на разработку программного продукта «Учет литературы. Поставщик»

Программа «Учет литературы. Поставщик» предназначена для ведения учета поставляемой университету литературы. Поставщик может фиксировать производство и поставку литературы университетам. Программа предоставляет возможность получения отчетов по изготовленной литературе.

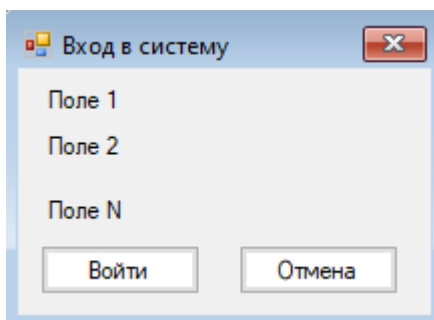
Основной функционал программы:

- Регистрация. Для регистрации пользователь должен заполнить следующие данные:

Данные от пользователя. Указать требования к каждому полю.
Пример: логин. Логин должен быть уникальным, иметь длину не более 50 символов. Обязателен к заполнению.

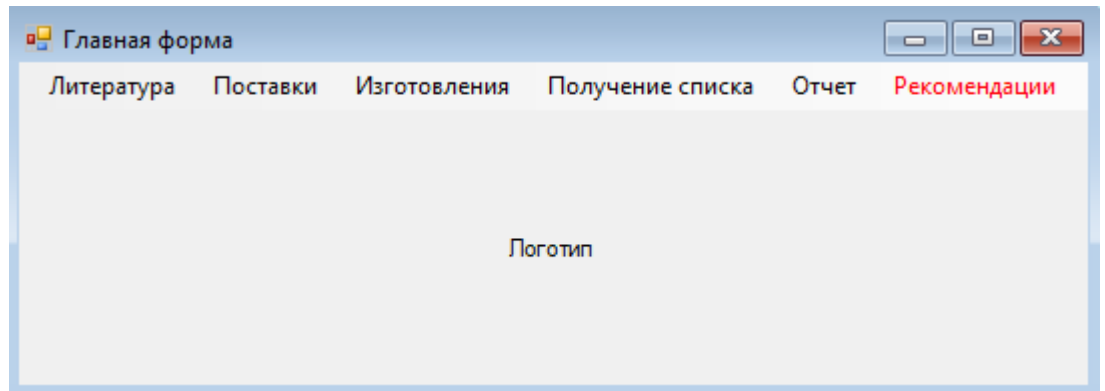


- Авторизация. Пользователь вводит данные. Если такой пользователь есть в системе, то происходит переход к основной форме. Если такого пользователя нет в системе, выводится сообщение об этом с просьбой ввести заново данные.



- Основная форма. Состоит из пунктов меню и логотипа. Через пункты меню можно перейти на формы работы с литературой, поставками (в том числе и привязка заявок), изготовлением, на

форму получения списка **актов приемки** и на форму получения отчета по **литературе**. Дополнительно, если реализовано, то вызов формы рекомендательной подсистемы.



- Формирование **литературы** (CRUD). Имеется форма со списком всей **литературы** пользователя и кнопки для создания, изменения и удаления **литературы**.

По **литературе** хранятся следующие сведения:

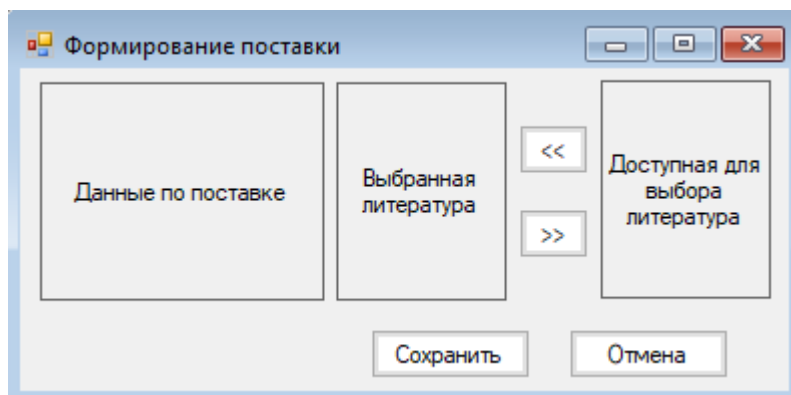
Данные для записи (2-3 поля).

- Формирование **записи поставки литературы** (CRUD). Имеется форма со списком всех **поставок** пользователя и кнопки для создания, изменения и удаления **поставки**.

По **поставке** хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступную **литературу** из списка (несколько) и сохранить изменения.

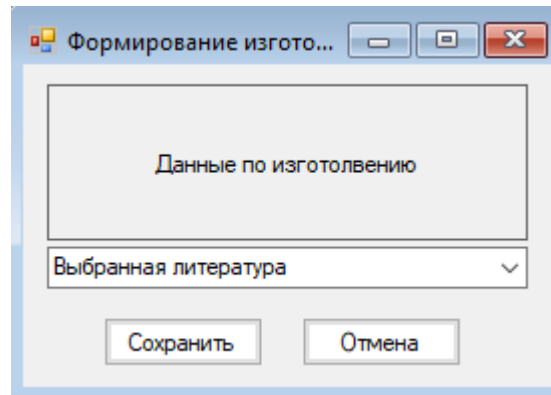


- Формирование **записи изготовления литературы** (CRUD). Имеется форма со списком всех **записей изготовления** пользователя и кнопки для создания, изменения и удаления **записи изготовления**.

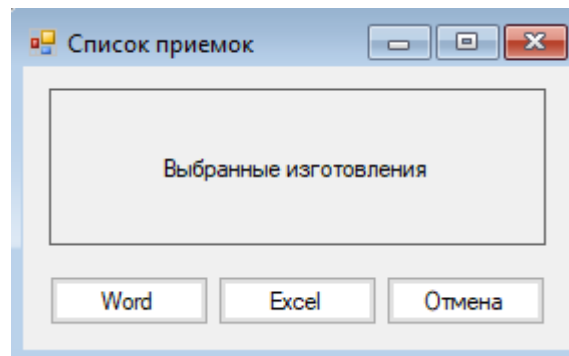
По **записи изготовления** хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись из списка **литературы** одну запись и сохранить изменения.



- Получение списка. Пользователь может получить список **актов приемки литературы** по выбранным **записям изготовления литературы**. Пользователь отмечает интересующую его **записи изготовления**, указывает формат файла (doc или xls), указывает имя и место сохранения файла и получает интересующий его список.



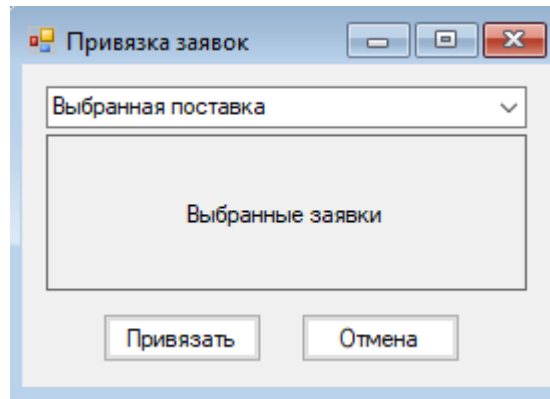
При выводе в doc-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

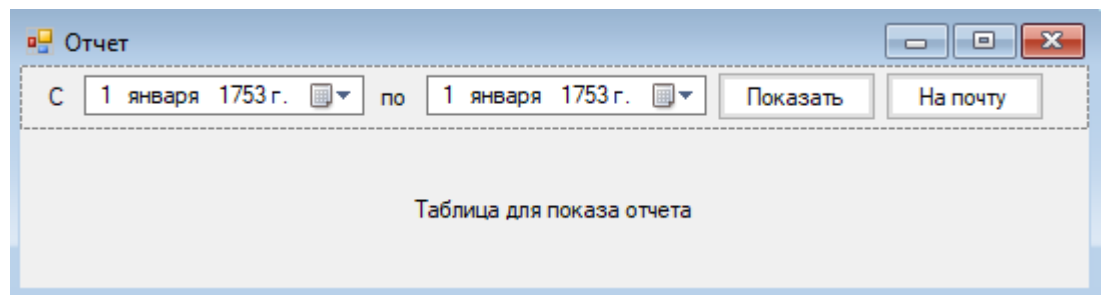
При выводе в xls-файл формат документа будет следующим:

указать как будет выглядеть документ, можно приложить скрин.

- Привязка **записи поставки литературы** к **записям заявок на поставки**. Пользователь может выбрать **запись поставки, записи заявок** и связать их.



- Пользователь может получить сведения за период по **литературе**, которая была **изготовлена** и **поставлена**. Для этого он должен указать интересующий его период, выбрать вариант получения сведений (отправка на почту или вывод на форму) и получить отчет.



Указать, как будет выглядеть отчет, можно приложить скрин.

Если проект разрабатывается индивидуально, то указать следующие пункты:

- Метод создания **заявок на поставки**. Определить, когда должен вызываться метод. Описать как будет проходить формирования (либо случайно, либо по какому-то алгоритму).
- Метод создания **записей поступления**. Метод должен формировать **записи** и выбирать туда **литературу**. Определить, когда должен вызываться метод. Описать как будет проходить создание (либо случайно, либо по определенному алгоритму).
- Метод привязки **заявок к записям поступлений**. Определить, когда должен вызываться метод. Описать как будет проходить формирования (либо случайно, либо по какому-то алгоритму).

В качестве интерфейса пользователя будет выступать **Web-клиент**, разработанный на **ASP.NET**.

Шаблон технического задания для роли «Заказчик (шестая роль)»

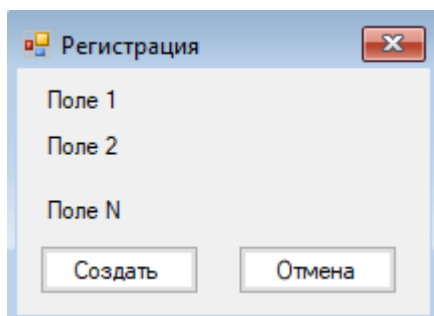
Техническое задание на разработку программного продукта «Учет литературы. Заказчик»

Программа «Учет литературы. Заказчик» предназначена для ведения учета получаемой университетом литературы. Заказчики могут формировать заявки на литературу, записи из поступления и акты получения. Программа предоставляет возможность получения отчетов по полученной литературе.

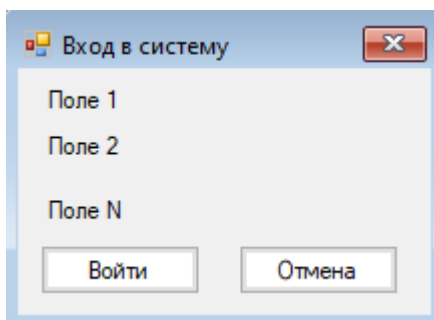
Основной функционал программы:

- Регистрация. Для регистрации пользователь должен заполнить следующие данные:

Данные от пользователя. Указать требования к каждому полю.
Пример: логин. Логин должен быть уникальным, иметь длину не более 50 символов. Обязателен к заполнению.

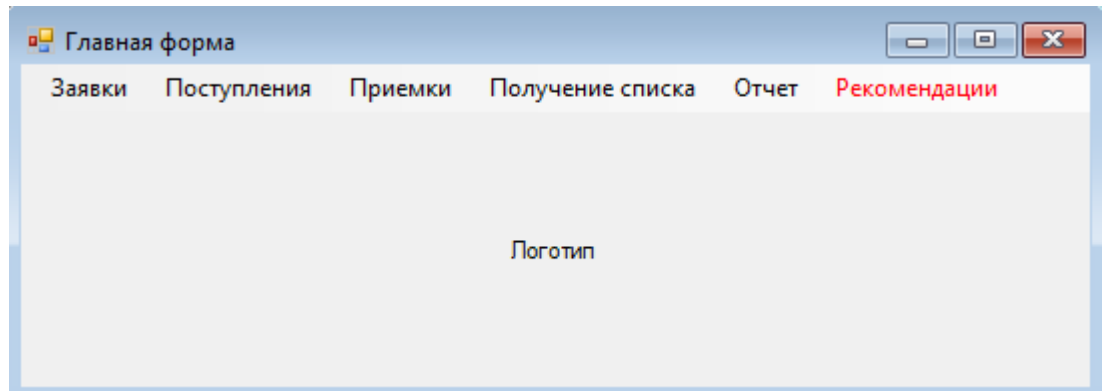


- Авторизация. Пользователь вводит данные. Если такой пользователь есть в системе, то происходит переход к основной форме. Если такого пользователя нет в системе, выводится сообщение об этом с просьбой ввести заново данные.



- Основная форма. Состоит из пунктов меню и логотипа. Через пункты меню можно перейти на формы работы с заявками, поступлениями (в том числе и привязка литературы), приемками

на форму получения списка **изготовлений** и на форму получения отчета по **литературе**. Дополнительно, если реализовано, то вызов формы рекомендательной подсистемы.



- Формирование **заявок на поставки** (CRUD). Имеется форма со списком всей **заявок** пользователя и кнопки для создания, изменения и удаления **заявки**.

По **заявке** хранятся следующие сведения:

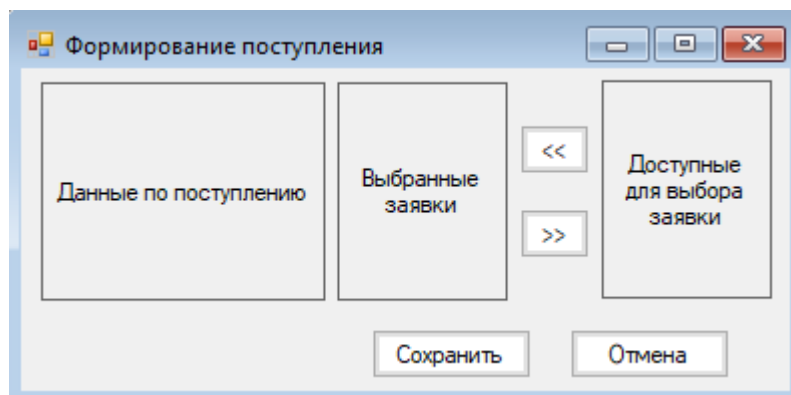
Данные для записи (2-3 поля).

- Формирование **записи поступления литературы** (CRUD). Имеется форма со списком всех **поступлений** пользователя и кнопки для создания, изменения и удаления **поступления**.

По **поступлению** хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись доступные **заявки на поставки** из списка (несколько) и сохранить изменения.

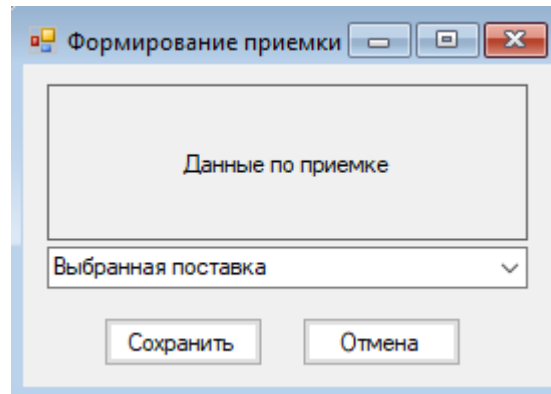


- Формирование **актов получения литературы** (CRUD). Имеется форма со списком всех **актов** пользователя и кнопки для создания, изменения и удаления **акта**.

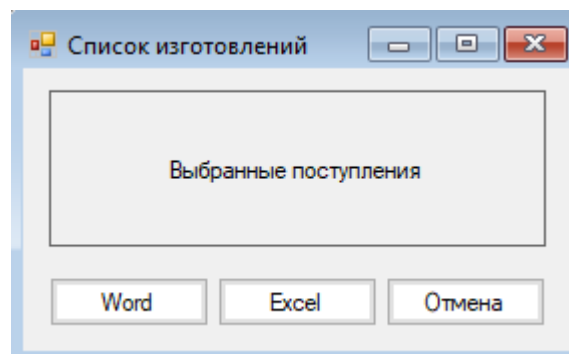
По **акту** хранятся следующие сведения:

Данные для записи (2-3 поля).

Имеется возможность выбрать в создаваемую запись из списка **поставок** одну запись и сохранить изменения.



- Получение списка. Пользователь может получить список **записей изготовления литературы** по выбранным **записям поступления литературы**. Пользователь отмечает интересующую его **записи поступления**, указывает формат файла (doc или xls), указывает имя и место сохранения файла и получает интересующий его список.



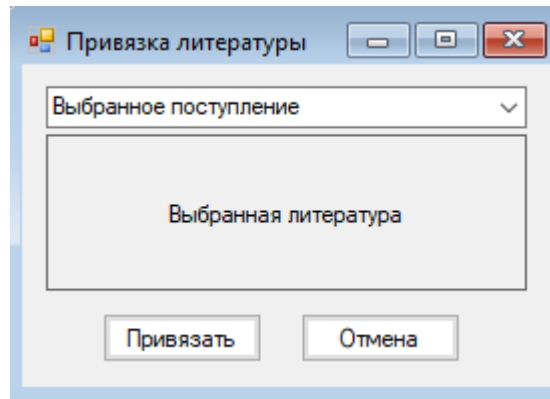
При выводе в doc-файл формат документа будет следующим:

[указать как будет выглядеть документ, можно приложить скрин.](#)

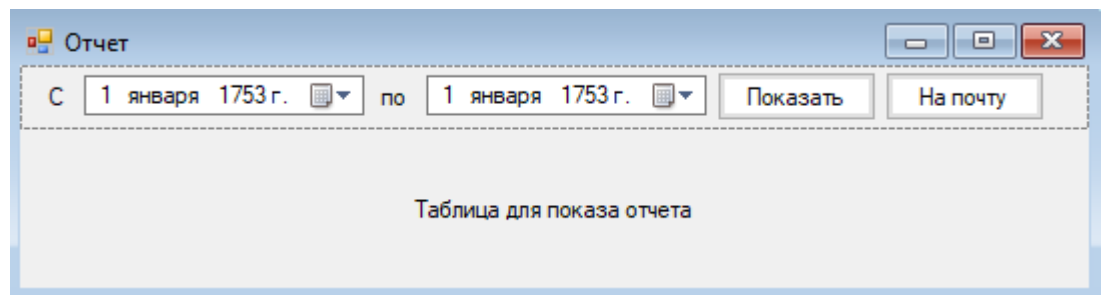
При выводе в xls-файл формат документа будет следующим:

[указать как будет выглядеть документ, можно приложить скрин.](#)

- Привязка **записи поступления литературы** к **литературе**. Пользователь может выбрать **запись поступления, литературу** и связать их.



- Пользователь может получить сведения за период по **литературе**, которая была **изготовлена** и **получена (через акты)**. Для этого он должен указать интересующий его период, выбрать вариант получения сведений (отправка на почту или вывод на форму) и получить отчет.



Указать, как будет выглядеть отчет, можно приложить скрин.

Если проект разрабатывается индивидуально, то указать следующие пункты:

- Метод создания **литературы**. Определить, когда должен вызываться метод. Описать как будет проходить формирование (либо случайно, либо по какому-то алгоритму).
- Метод создания **записей поставки**. Метод должен формировать **записи** и выбирать туда **литературу**. Определить, когда должен вызываться метод. Описать как будет проходить создание (либо случайно, либо по определенному алгоритму).
- Метод создания **записей изготовления**. Метод должен формировать **записи** и выбирать туда **литературу**. Определить, когда должен вызываться метод. Описать как будет проходить создание (либо случайно, либо по определенному алгоритму).

В качестве интерфейса пользователя будет выступать **desktop-приложение**, разработанное на **WPF**.

РАЗРАБОТКА ПРОГРАММЫ УЧЕТА ЛИТЕРАТУРЫ В
АБСТРАКТНОМ УНИВЕРСИТЕТЕ:

практикум к выполнению
курсовых работ по дисциплине
«Технологии программирования»

Автор ЭГОВ Евгений Николаевич

Усл. печ. л. 3

УлГТУ, 432027, г. Ульяновск, ул. Сев. Венец, д. 32.