

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шебзухова Татьяна Александровна
Должность: Директор Пятигорского института (филиал) Северо-Кавказского
федерального университета
Дата подписания: 25.07.2023 14:41:06
Уникальный программный ключ:
d74ce93cd40e39275c3ba2f58486412a1c8ef96f

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ

Методические указания к выполнению курсовой работы
по дисциплине

ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Направление подготовки 09.04.02 Информационные системы и технологии
Профиль подготовки «Технологии работы с данными и знаниями, анализ информации»

Квалификация выпускника Магистр

Пятигорск 2023 г.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C00000043E9AB8B952205E7BA5000600000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Печатается по решению
Учебно-методического совета
Северо-Кавказского федерального
университета

Технологии разработки программного обеспечения: Методические указания к курсовой работе по дисциплине / сост. Флоринский О.С. – Пятигорск: Изд-во СКФУ, 2023. – 28 с.

Составитель

Флоринский О.С. – к.т.н., доцент кафедры «Систем управления и информационных технологий» ФГАОУ ВО «Северо-Кавказский федеральный университет», Пятигорский институт (филиал) СКФУ в г. Пятигорске

Методические указания содержат основные требования к написанию курсовой работы, порядок сбора и анализа материалов исследования, рекомендуемую литературу. Предназначается для студентов направления **09.04.02 «Информационные системы и технологии»**

© ФГАОУ ВО «Северо-Кавказский
федеральный университет», 2023

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

СОДЕРЖАНИЕ

ОБЩИЕ ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ КУРСОВЫХ РАБОТ.....	4
РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ КУРСОВОЙ РАБОТЫ.....	4
ВАРИАНТЫ ЗАДАНИЙ НА КУРСОВУЮ РАБОТУ	6
ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ.....	27
ПОРЯДОК ПРЕДСТАВЛЕНИЯ КУРСОВОЙ РАБОТЫ К ЗАЩИТЕ.....	27
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	27

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

ОБЩИЕ ТРЕБОВАНИЯ К ВЫПОЛНЕНИЮ КУРСОВЫХ РАБОТ

Одной из важных форм обучения студентов является курсовое проектирование. Разработка курсовой работы по дисциплине «Технологии разработки программного обеспечения» завершает изучение этого предмета. Написание курсовой работы вырабатывает у студента навыки самостоятельной работы, навыки решения задач с использованием компьютерных систем, а также навыки системного подхода к решению задач, что является немаловажным в процессе обучения. Разработка курсовой работы является частью целенаправленного и систематизированного подхода к обучению студентов. Помогая им утвердиться в будущей специальности.

Цель курсовой работы по дисциплине «Технологии разработки программного обеспечения» – «Формирование способности проводить предпроектное обследование объекта проектирования, анализ научно-технической информации, отечественного и зарубежного опыта, анализ разработанных программных приложений, техническое проектирование и сопровождение программных приложений, разрабатывать отдельные компоненты информационной системы».

При выполнении курсовой работы студент самостоятельно осваивает все этапы создания программных приложений от постановки задачи до практической реализации.

В результате освоения данного курса студенты должны приобрести навыки по применению вычислительной техники для решения профессиональных задач.

Выполненная курсовая работа представляется на кафедру согласно графику ее выполнения. Курсовая работа регистрируется на кафедре и передается научному руководителю. В течение 10-ти дней руководитель проверяет и рецензирует работу. Если у руководителя есть замечания, то она возвращается на доработку. Студент знакомится с рецензией, вносит в случае необходимости исправления и защищает курсовую работу у руководителя.

РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ КУРСОВОЙ РАБОТЫ

Курсовая работа выполняется на персональном компьютере и должна содержать распечатку с выполненной курсовой работой и диск, содержащий все части работы, записанные в электронном виде (каждое задание должно находиться в отдельной папке, кроме того, должен быть файл, содержащий полную электронную версию всей распечатки курсовой работы).

В распечатку курсовой работы входит: Содержание, Введение, Основная часть, состоящая из двух основных подразделов: Теоретическая часть и Практическая часть, Заключение с анализом и основными выводами, Список использованных источников, Приложения (при необходимости). Ниже представлена примерная структура содержания работы:

СОДЕРЖАНИЕ

ВВЕДЕНИЕ (СТРАНИЦА 3)

3

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Раздел 1

1.1.1 Xxx

1.1.2 Yyy

1.1.n Zzz

1.2 Раздел 2

1.2.1 Xxx

1.2.2 Yyy

1.2.n Zzz

1.n Раздел n

1.n.1 Xxx

1.n.2 Yyy

1.n.n Zzz

2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 Практическое задание № 1

2.2 Практическое задание № 2

2.3 Практическое задание № 3

2.4 Практическое задание № 4

2.5 Практическое задание № 5

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.06.2022 по 19.06.2023

ЗАКЛЮЧЕНИЕ
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ
ПРИЛОЖЕНИЯ (если требуются)

ВВЕДЕНИЕ (страница 3)

Во введении обосновывается актуальность выбранной темы исследования, отражаются цель и задачи, дается краткий обзор материалов теме исследования.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Раскрытие теоретической темы курсовой работы, объем ориентировочно 20-30 страниц. Тема курсовой работы выбирается студентом индивидуально из примерной тематики теоретических заданий и согласуется с преподавателем, также студентом может быть предложена своя индивидуальная тема также согласуемая с преподавателем.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Результат решения практических заданий – в выбранной студентом (и согласованной с преподавателем) среде программирования.

Вариант практического задания выбирается студентом в соответствии с последней цифрой номера зачетной книжки (студенческого билета). Студентом может быть предложена своя индивидуальная тема практического задания, которая должна быть согласована с преподавателем.

Результат решения практических заданий сдается в бумажном и электронном виде. Бумажная версия должна содержать, по каждому заданию, листинг программного кода, пояснения и распечатки всех этапов решения задач в среде программирования. Электронная версия должна содержать следующие материалы:

Задание 1 – все файлы программы и откомпилированный exe-модуль, которые должны находиться в отдельной папке «Задание 1»;

в текстовом файле (word) – листинг программного кода, пояснения и распечатки всех этапов решения задач в среде программирования;

Задание 2 – все файлы программы и откомпилированный exe-модуль, которые должны находиться в отдельной папке «Задание 2»;

в текстовом файле (word) – листинг программного кода, пояснения и распечатки всех этапов решения задач в среде программирования;

Задание 3 – все файлы программы и откомпилированный exe-модуль, которые должны находиться в отдельной папке «Задание 3»;

в текстовом файле (word) – листинг программного кода, пояснения и распечатки всех этапов решения задач в среде программирования;

Задание 4 – файл созданной базы данных, все файлы программы и откомпилированный exe-модуль, которые должны находиться в отдельной папке «Задание 4»;

в текстовом файле (word) – пояснения и распечатки всех этапов решения задач в СУБД Access. Листинг программного кода, пояснения и распечатки всех этапов решения задач в среде программирования.

Задание 5 – все файлы программы и откомпилированный exe-модуль, которые должны находиться в отдельной папке «Задание 5»;

в текстовом файле (word) - листинг программного кода, пояснения и распечатки всех этапов решения задач в среде программирования.

ЗАКЛЮЧЕНИЕ

Заключение может содержать основные выводы и анализ достигнутых результатов по теоретической и практической частям работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Приводится список использованных источников (рекомендуется использовать источники

не старше 5 лет)

ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебукова Татьяна Александровна

ПРИЛОЖЕНИЯ

Действителен: с 19.08.2022 по 19.08.2023

Приложения в работе формируются в случае их необходимости. Приложения могут включать различный дополнительный материал, необходимый для подтверждения рассматриваемых положений.

ВАРИАНТЫ ЗАДАНИЙ НА КУРСОВУЮ РАБОТУ

Теоретическая часть:

Тема курсовой работы выбирается студентом индивидуально из примерной тематики теоретических заданий и согласуется с преподавателем, также студентом может быть предложена своя индивидуальная тема также согласуемая с преподавателем.

Примерный перечень тем на курсовую работу

1. Жизненный цикл программного обеспечения.
2. Этапы проектирования программного обеспечения
3. Этап постановки задачи при разработке программного обеспечения.
4. Обеспечение диалога с пользователем при разработке программного обеспечения.
5. Отладка и тестирование разрабатываемого программного обеспечения.
6. Разработка программного обеспечения с использованием CASE – технологий.
7. Методы борьбы с ошибками в программном обеспечении
8. Проблемы обеспечения контроля качества разрабатываемого программного обеспечения.
9. Стандартизация в области разработки программного обеспечения.
10. Сертификация в области разработки программного обеспечения.

Практическая часть:

Вариант практического задания выбирается студентом в соответствии с последней цифрой номера зачетной книжки. Студентом может быть предложена своя индивидуальная тема практического задания, которая должна быть согласована с преподавателем.

Варианты практических заданий

Вариант 1.

Задание 1.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

В первой форме должны заноситься ФИО и номер группы студента, при нажатии на кнопку «Старт» программа должна переходить ко второй форме.

Во второй форме:

- ФИО студента и номер группы, указанные в первой форме, должны быть отображены в соответствующих местах;
- вместо слов «Дата» и «Время» должны выводиться текущие дата и время;
- при нажатии на кнопку «Выход» программа должна закрываться;
- при нажатии на кнопку «Сообщение о создателе программы» должна появляться надпись, информирующая о создателе программы;
- при нажатии на кнопку «Описание картинки 1» должна появляться поясняющая надпись об изображении на картинке 1;
- при нажатии на кнопку «Описание картинки 2» должна появляться поясняющая надпись об изображении на картинке 2.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Создайте программу нахождения максимального значения Y для множества X, если
известно:
$$Y=32X^2 - 123X + 54$$

Действителен: с 19.08.2022 по 19.08.2023

Значения X начальное, X конечное и шаг должны вводиться пользователем при работе с программой. Как результат программа должна выводить максимальное значение Y и соответствующее ему значение X .

Задание 3.

Создайте программу, позволяющую тестировать студентов по дисциплине «Информатика», тема: «Текстовый процессор Word», состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

Форма 1

Данные студента

ФИО
Text1

Номер группы
Text2

Старт Выход

Форма 2

Ответьте на следующий вопрос

Label1

Варианты ответов

Command1
Command2
Command3

После внесения данных в форму 1 и нажатия кнопки «Старт», должен происходить переход ко второй форме, в которой поочередно должны задаваться вопросы с тремя вариантами ответов (два неверных, один верный). После прохождения всех вопросов программа должна выдавать ФИО и номер группы тестируемого, а также результаты тестирования.

Задание 4.

Создайте с помощью MS Access базу данных по реализуемым товарам фирмы «Дежавю»; фирма реализует 20 разновидностей товара, база данных должна содержать данные, необходимые для реализации товара: наименование товара, его приходная цена, наценка, налоги, кроме того, база данных должна содержать сведения о каждом из товаров: сорт, фирма производитель, дата производства, срок хранения.

Создать с помощью средства разработки (среды программирования) программу, предоставляющую доступ к базе данных. Программа должна позволять получать доступ к данным по каждому из 20 разновидностей товара, т.е. при выборе кого-либо товара должны отображаться все данные по нему. При нажатии на кнопку «Расчет отпускной цены» программа должна выдавать следующие сведения по выбранному товару: наименование товара, приходная цена, наценка, налоги и итоговая отпускная цена.

Задание 5.

Разработайте программное обеспечение, для решения какой либо прикладной задачи. При разработке данного программного обеспечения студент должен продемонстрировать свои навыки и умения, полученные в ходе изучения литературы по технологиям программирования при освоении данной дисциплины. Выбранная тематика разрабатываемого программного обеспечения должна быть согласована с преподавателем.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 250400043E94B68352269E7BA56007400043E
Владелец: 250400043E94B68352269E7BA56007400043E
Действителен: с 19.08.2022 по 19.08.2023

Вариант 2.

Задание 1.

Создайте программу, позволяющую тестировать студентов по дисциплине «Информатика», тема: «Операционная система Windows». Программа должна состоять из одной формы, в которой поочередно должны задаваться вопросы (не менее 15 вопросов) с четырьмя вариантами ответов (три неверных, один верный). После прохождения всех вопросов программа должна выдавать результаты тестирования и время начала и окончания прохождения теста.

Задание 2.

Составьте программу сортировки пяти заданных чисел (2, 5, 7, 12, 8) по возрастанию.

Задание 3.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

Форма 1

Настройки

☒ Непрозрачный

Цвет R G B

☒ Прямоугольник
☐ Круг
☐ Овал

Выход

Сведения о разработчике

Работу выполнил

Фамилия Имя Отчество Группа

Форма 2

Работу выполнил

Text1

Группы

Text2

Время входа в программу

Время окончания работы с программой

Выход

В первой форме:

- должны заноситься ФИО и номер группы студента;
- В данной форме фигура, занимающая основное место должна:
 - при переключении менять форму: прямоугольник, круг и овал;
 - при установке флажка должна включаться и отключаться ее прозрачность;
 - при изменении положения полос прокрутки - менять цвет;
- При нажатии на кнопку «Сведения о разработчике» должна появляться надпись, информирующая о разработчике программы;
- при нажатии на кнопку «Выход» программа должна переходить ко второй форме.

Сертификат:
Владелец:

Документ подписан
Электронной подписью
Щебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Во второй форме:

- ФИО студента и номер группы, указанные в первой форме, должны быть отображены в соответствующих местах;
- вместо слов «Время входа в программу» должно быть отображено время открытия первой формы;
- вместо слов «Время окончания работы с программой» должно быть отображено время закрытия первой формы;
- при нажатии на кнопку «Выход» программа должна закрываться.

Задание 4.

Создайте с помощью MS Access базу данных по сотрудникам фирмы «Самоцвет»; в фирме работает 15 человек, база данных должна содержать данные, необходимые для расчета заработной платы: ФИО сотрудника, его должность, квалификационный разряд, количество отработанных дней в месяце и сумму оплаты за один день, кроме того, база данных должна содержать личные данные сотрудников: год рождения, номер паспорта, адрес проживания, телефон.

Создать с помощью средства разработки (среды программирования) программу, предоставляющую доступ к базе данных. Программа должна позволять получать доступ к данным по каждому из сотрудников фирмы, т.е. при выборе кого-либо сотрудника должны отображаться все данные по нему. При нажатии на кнопку «Расчет заработной платы» программа должна выдавать следующие сведения по выбранному сотруднику: ФИО сотрудника, общую начисленную сумму и сумму к выдаче, после вычитания налогов.

Задание 5.

Разработайте программное обеспечение, для решения какой либо прикладной задачи. При разработке данного программного обеспечения студент должен продемонстрировать свои навыки и умения, полученные в ходе изучения литературы по технологиям программирования при освоении данной дисциплины. Выбранная тематика разрабатываемого программного обеспечения должна быть согласована с преподавателем.

Вариант 3.

Задание 1.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

В первой форме должны заноситься ФИО и номер группы студента, при нажатии на кнопку «Старт» программа должна переходить ко второй форме.

Во второй форме:

- ФИО студента, указанные в первой форме, должны быть отображены в соответствующих местах;
- при переключении менять форму: прямоугольник, круг и овал;
- при установке флажка должна включаться и отключаться ее прозрачность;
- при изменении положения полос прокрутки менять цвет;
- при нажатии на кнопку «Выход» программа перед закрытием должна выдавать время начала и окончания работы с программой.

Задание 2.

Напишите программу определения наличия задаваемого слова в произвольном тексте (текст и искомое слово должны вводиться пользователем при работе с программой).

Задание 3.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

1

3

4

2

В первой форме:

- должны заноситься ФИО и номер группы студента;
- вместо слов «Дата» и «Время» должны выводиться текущие дата и время;
- при нажатии на кнопку «Старт» программа должна переходить ко второй форме.

Во второй форме:

- в поля «1» и «2» должны вводиться числовые значения, причем, при вводе числового значения вместо надписи «Комментарий» должна появляться надпись «ОК», а при вводе текстового значения должна появляться надпись «Введено неверное значение» и кнопки «+ - * / », должны блокироваться.
- при нажатии кнопок «+» (плюс), «-» (минус), «*» (умножить), «/» (разделить) должны производиться соответствующие действия со значениями, введенными в поля «1» и «2». Результат вычислений должен отображаться в поле «3». В списке «4» должны отображаться все получаемые значения поля «3».
- при нажатии на кнопку «C» должно происходить обнуление полей «1», «2» и «3».
- при нажатии на кнопку «Функция» должна вычисляться функция по значениям полей «1» и «2», причем, в случае деления на ноль и вычисления квадратного корня из отрицательного числа программа должна выдавать сообщение об ошибке и предлагать произвести ввод данных заново:

$$A^{\frac{B}{2}} \cdot (\sqrt{A} - \sqrt{B}) \cdot \frac{\sqrt[3]{AB}}{A \cdot B}$$

- при нажатии на кнопку «Сведения о разработчике» должна появляться надпись, информирующая о разработчике программы;
- при нажатии на кнопку «Выход» программа должна закрываться.

ДОКУМЕНТ ПОДПИСАН
нажатии на кнопку
Сертификат: 20200818584202200575450060000143E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Задание 4.

Создайте с помощью MS Access базу данных по реализуемым товарам фирмы «Апокалипсис»; фирма реализует 20 разновидностей товара, база данных должна содержать данные, необходимые для реализации товара: наименование товара, его приходная цена, наценка, налоги, кроме того, база данных должна содержать сведения о каждом из товаров: сорт, фирма производитель, дата производства, срок хранения.

Создать с помощью средства разработки (среды программирования) программу, предоставляющую доступ к базе данных. Программа должна позволять получать доступ к данным по каждому из 20 разновидностей товара, т.е. при выборе кого-либо товара должны отображаться все данные по нему. При нажатии на кнопку «Дата реализации» программа должна выдавать следующие сведения по выбранному товару: наименование товара, дата производства, срок хранения и дату, до которой должен быть реализован данный товар.

Задание 5.

Разработайте программное обеспечение, для решения какой либо прикладной задачи. При разработке данного программного обеспечения студент должен продемонстрировать свои навыки и умения, полученные в ходе изучения литературы по технологиям программирования при освоении данной дисциплины. Выбранная тематика разрабатываемого программного обеспечения должна быть согласована с преподавателем.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Вариант 4.

Задание 1.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

В первой форме:

- должны заноситься ФИО и номер группы студента;
- при нажатии на кнопку «Старт» программа должна переходить ко второй форме.

Во второй форме:

- в поля «1» и «2» должны вводиться числовые значения. При нажатии кнопок «+» (плюс), «-» (минус), «*» (умножить), «/» (разделить), «^» (возвести в степень), «SQR» (извлечь квадратный корень) должны производиться соответствующие действия со значениями, введенными в поля «1» и «2». Результат вычислений должен отображаться в поле «3». В списке «4» должны отображаться все получаемые значения поля «3»;
- при нажатии на кнопку «C» должно происходить обнуление полей «1, 2, 3»;
- при нажатии на кнопку «Функция» должна вычисляться нижеследующая функция по значениям полей «1» и «2», причем, в случае деления на ноль и вычислении квадратного корня из отрицательного числа программа должна выдавать сообщение об ошибке и предлагать произвести ввод данных заново:

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

$$\frac{A^2 \cdot \sqrt[3]{AB}}{A - B}$$

- при нажатии на кнопку «Сведения о разработчике» должна появляться надпись, информирующая о разработчике программы;
- при нажатии на кнопку «Выход» программа перед закрытием должна выдавать сообщение типа: «Студент: Иванов И. И. Группы: ИСТ – 001 завершил работу с программой», соответственно подставляя номер группы и ФИО, из указанных в первой форме.

Задание 2.

Составьте программу сортировки любого количества чисел, вводимых пользователем при работе с программой.

Задание 3.

Создайте программу, позволяющую тестировать студентов по дисциплине «Информатика», тема: «Табличный процессор Excel», состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

После запуска программы должна открываться первая форма «Тестирование», в которой поочередно должны задаваться вопросы с тремя вариантами ответов (два неверных, один верный). После прохождения всех вопросов программа должна переходить ко второй форме: «Результаты опроса», в которой вместо слов «Начало тестирования» и «Окончание тестирования» должны указываться фактическое время соответственно открытия и закрытия первой формы. Также, должны быть указаны результаты теста – количество правильных и неправильных ответов и оценка работы тестируемого. При нажатии кнопки «Выход» программа должна завершать свою работу.

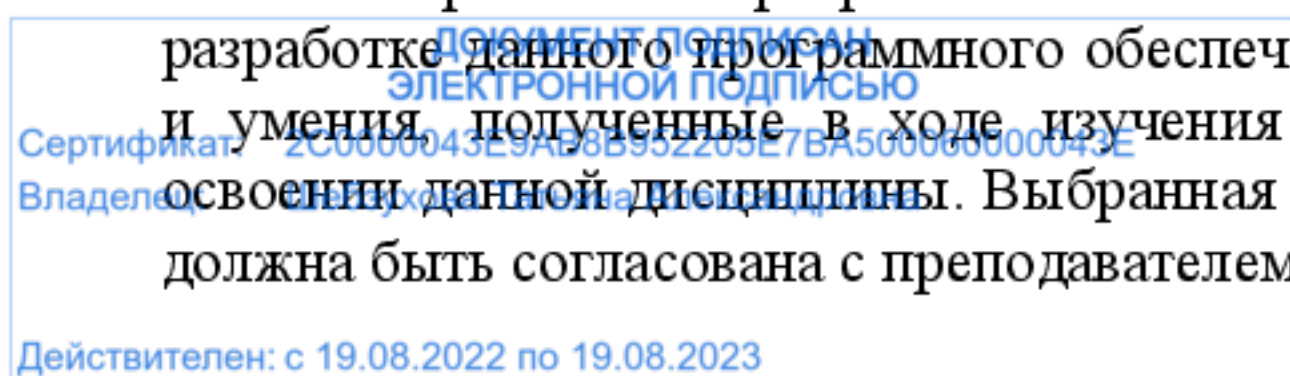
Задание 4.

Создайте с помощью MS Access базу данных по студентам группы ИСТ-001; база данных должна содержать данные по успеваемости студентов: ФИО студента и его оценки по всем предметам, кроме того, база данных должна содержать личные данные студентов: год рождения, номер паспорта, адрес проживания, телефон.

Создать с помощью средства разработки (среды программирования) программу, предоставляющую доступ к базе данных. Программа должна позволять получать доступ к данным по каждому из студентов группы ИСТ-001, т.е. при выборе кого-либо студента должны отображаться все данные по нему. При нажатии на кнопку «Успеваемость» программа должна выдавать следующие сведения по выбранному студенту: ФИО студента, оценки по всем предметам и средний балл.

Задание 5.

Разработайте программное обеспечение, для решения какой либо прикладной задачи. При разработке данного программного обеспечения студент должен продемонстрировать свои навыки и умения, полученные в ходе изучения литературы по технологиям программирования при освоении данной дисциплины. Выбранная тематика разрабатываемого программного обеспечения должна быть согласована с преподавателем.



Вариант 5.

Задание 1.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

Форма 1

Группа: 001

ФИО
Иван
Андреевич
Клепалко

Сейчас
Дата
Время

Писания картинка 1

Писания картинка 2

Задание выполнил...

В
Ы
Х
О
Д

Форма 2

Работу выполнил
Text1

Группы
Text2

Время входа в программу

Время окончания работы с программой

Выход

В первой форме:

- должны заноситься ФИО и номер группы студента;
- вместо слов «Дата» и «Время» должны выводиться текущие дата и время;
- при нажатии на кнопку «Задание выполнил» должна появляться надпись, о создателе программы;
- при нажатии на кнопку «Описание картинки 1» должна появляться соответствующая надпись;
- при нажатии на кнопку «Описание картинки 2» должна появляться соответствующая надпись;
- при нажатии на кнопку «Выход» программа должна переходить ко второй форме.

Во второй форме:

- ФИО студента и номер группы, указанные в первой форме, должны быть отображены в соответствующих местах;
- вместо слов «Время входа в программу» должно быть отображено время открытия первой формы;
- вместо слов «Время окончания работы с программой» должно быть отображено время закрытия первой формы;
- при нажатии на кнопку «Выход» программа должна закрываться.

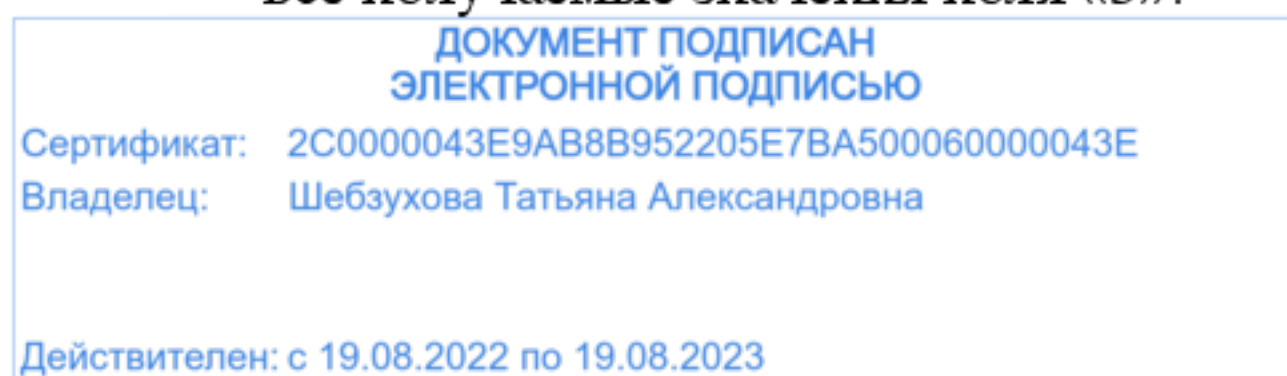
Задание 2.

Напишите программу определения наличия слова «монитор» в произвольном тексте (текст должен вводиться пользователем при работе с программой).

Задание 3.

Создайте программу, состоящую из одной формы, и внесите все необходимые элементы, как представлено на нижеследующем рисунке.

- В поля «1» и «2» должны вводиться числовые значения, причем, при вводе числового значения вместо надписи «Комментарий» должна появляться надпись «ОК», а при вводе текстового значения должна появляться надпись «Введено неверное значение» и кнопки «+ - * /», должны блокироваться.
- при нажатии кнопок «+» (плюс), «-» (минус), «*» (умножить), «/» (разделить) должны производиться соответствующие действия со значениями, введенными в поля «1» и «2». Результат вычислений должен отображаться в поле «3». В списке «4» должны отображаться все получаемые значения поля «3».



1

2

3

4

Задание к билету № n

Окно ввода

0

0

Комментарий

Окно результата

0

List1

Комманда

+

-

Функция

*

/

С

Сведения о разработчике

Выход

- при нажатии на кнопку «Функция» должна вычисляться функция по значениям поля «1», представленная следующим выражением:

$$\text{Функция (x)} = \begin{cases} 1, & \text{при } x = 0 \\ 10, & \text{при } x = 1 \\ 50, & \text{при } x = 2 \\ \text{сообщение об ошибке, при других значениях } x. \end{cases}$$

при нажатии на кнопку «С» должно происходить обнуление полей «1, 2, 3»;

при нажатии на кнопку «Сведения о разработчике» должна появляться надпись, информирующая о разработчике программы;

дополните форму полями для ввода «Фамилии», «Имени» и «Отчества» работающего в данный момент с программой пользователя;

при нажатии на кнопку «Выход» программа перед закрытием должна выдавать сообщение типа: «Студент Иванов И. И. завершил работу с программой, время начала работы 15:25:28, время окончания работы 15:30:12», соответственно подставляя ФИО, из указанных самим пользователем в форме, а время – фактическое время начала и окончания работы с данной программой.

Задание 4.

Создайте с помощью MS Access базу данных по реализуемым товарам фирмы «Forte»; фирма реализует 20 разновидностей товара, база данных должна содержать данные, необходимые для реализации товара: наименование товара, его приходная цена, наценка, налоги, кроме того, база данных должна содержать сведения о каждом из товаров: количество товаров на складе, сорт, фирма производитель, дата производства, срок хранения.

Создать с помощью средства разработки (среды программирования) программу, предоставляющую доступ к базе данных. Программа должна позволять получать доступ к данным по каждому из 20 разновидностей товара, т.е. при выборе кого-либо товара должны отображаться все данные по нему. При нажатии на кнопку «Расчет» программа должна выдавать следующие сведения по выбранному товару: наименование товара, приходная цена всего товара данного вида, находящегося на складе, и суммарная отпускная цена этого товара на складе.

Задание 5.

Разработайте программное обеспечение, для решения какой либо прикладной задачи. При разработке данного программного обеспечения студент должен продемонстрировать свои навыки и умения, полученные в ходе изучения литературы по технологиям программирования при освоении данной дисциплины. Выбранная тематика разрабатываемого программного обеспечения должна быть согласована с преподавателем.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Задание 1.

Создайте программу, состоящую из одной формы, и внесите все необходимые элементы, как представлено на нижеследующем рисунке:

Вариант 6.

1

2

Задание к билету № n

Комментарий

+-Функция

*/C

List1

Сведения о разработчикеВыход

3

4

- В поля «1» и «2» должны вводиться числовые значения, причем, при вводе числового значения вместо надписи «Комментарий» должна появляться надпись «ОК», а при вводе текстового значения должна появляться надпись «Введено неверное значение» и кнопки «+ - * / », должны блокироваться.
- при нажатии кнопок «+» (плюс), «-» (минус), «*» (умножить), «/» (разделить) должны производиться соответствующие действия со значениями, введенными в поля «1» и «2». Результат вычислений должен отображаться в поле «3». В списке «4» должны отображаться все получаемые значения поля «3».
- при нажатии на кнопку «Функция» должна вычисляться функция по значениям поля «1», представленная следующим выражением:
$$\text{Функция}(x) = \begin{cases} 1, & \text{при } x = 0 \\ 10, & \text{при } x = 1 \\ 50, & \text{при } x = 2 \\ \text{сообщение об ошибке, при других значениях } x. \end{cases}$$
- при нажатии на кнопку «C» должно происходить обнуление полей «1, 2, 3»;
- при нажатии на кнопку «Сведения о разработчике» должна появляться надпись, информирующая о разработке программы;
- дополните форму полями для ввода «Фамилии», «Имени» и «Отчества» работающего в данный момент с программой пользователя;
- при нажатии на кнопку «Выход» программа перед закрытием должна выдавать сообщение типа: «Студент Иванов И. И. завершил работу с программой, время начала работы 15:25:28, время окончания работы 15:30:12», соответственно подставляя ФИО, из указанных самим пользователем в форме, а время – фактическое время начала и окончания работы с данной программой.

Задание 2.

Составьте программу нахождения максимального и минимального числа из любого количества чисел, вводимых пользователем при работе с программой.

Задание 3.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
 Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

В первой форме должны заноситься ФИО и номер группы студента, при нажатии на кнопку «Старт» программа должна переходить ко второй форме.

Во второй форме:

- ФИО студента, указанные в первой форме, должны быть отображены в соответствующих местах;
- при переключении менять форму: прямоугольник, круг и овал;
- при установке флажка должна включаться и отключаться ее прозрачность;
- при изменении положения полос прокрутки менять цвет;
- при нажатии на кнопку «Выход» программа перед закрытием должна выдавать время начала и окончания работы с программой.

Задание 4.

Создайте с помощью MS Access базу данных по реализуемым товарам фирмы «Био-Ком»; фирма реализует 20 разновидностей товара, база данных должна содержать данные, необходимые для реализации товара: наименование товара, его приходная цена, наценка, налоги, кроме того, база данных должна содержать сведения о каждом из товаров: сорт, фирма производитель, дата производства, срок хранения.

Создать с помощью средства разработки (среды программирования) программу, предоставляющую доступ к базе данных. Программа должна позволять получать доступ к данным по каждому из 20 разновидностей товара, т.е. при выборе кого-либо товара должны отображаться все данные по нему. При нажатии на кнопку «Расчет отпускной цены» программа должна выдавать следующие сведения по выбранному товару: наименование товара, приходная цена, наценка, налоги и итоговая отпускная цена.

Задание 5.

Разработайте программное обеспечение, для решения какой либо прикладной задачи. При разработке данного программного обеспечения студент должен продемонстрировать свои навыки и умения, полученные в ходе изучения литературы по технологиям программирования при освоении данной дисциплины. Выбранная тематика разрабатываемого программного обеспечения должна быть согласована с преподавателем.

Вариант 7.

Задание 1.

Создайте программу, позволяющую тестировать студентов по дисциплине «Информатика», тема: «Текстовый процессор Word», состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

Форма 1

Данные студента

ФИО
Text1

Номер группы
Text2

Старт Выход

Форма 2

Ответьте на следующий вопрос

Label1

Варианты ответов

Command1
Command2
Command3

После внесения данных в форму 1 и нажатия кнопки «Старт», должен происходить переход ко второй форме, в которой поочередно должны задаваться вопросы с тремя вариантами ответов (два неверных, один верный). После прохождения всех вопросов программа должна выдавать ФИО и номер группы тестируемого, а также результаты тестирования.

Задание 2.

Составьте программу определения функции:

$$Y = \begin{cases} \text{Отрицательное значение, если } (X < 0); \\ \text{Критическая точка, если } (X = 0); \\ \text{Положительное значение, если } (X > 0); \end{cases}$$

Для X , изменяющегося от $X_{\text{начальное}}$ (X_n) до $X_{\text{конечное}}$ (X_k) с шагом C . Значения $X_{\text{начальное}}$, $X_{\text{конечное}}$ и шаг должны вводиться пользователем при работе с программой.

Задание 3.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

В первой форме:

- должны заноситься ФИО и номер группы студента;
- при нажатии на кнопку «Старт» программа должна переходить ко второй форме.

Во второй форме:

- в поля «1» и «2» должны вводиться числовые значения. При нажатии кнопок «+» (плюс), «-» (минус), «*» (умножить), «/» (разделить), «^» (возвести в степень), «SQR» (извлечь квадратный корень) должны производиться соответствующие действия со значениями, введенными в поля «1» и «2». Результат вычислений должен отображаться в поле «3». В списке «4» должны отображаться все получаемые значения поля «3»;
- при нажатии на кнопку «C» должно происходить обнуление полей «1, 2, 3»;
- при нажатии на кнопку «Функция» должна вычисляться функция по значениям полей «1» и «2», причем, в случае деления на ноль и вычислении квадратного корня из отрицательного числа программа должна выдавать сообщение об ошибке и предлагать произвести ввод данных заново:

$$\frac{A^2 \cdot \sqrt[3]{AB}}{A - B}$$

- при нажатии на кнопку «Сведения о разработчике» должна появляться надпись, информирующая о разработчике программы;
- при нажатии на кнопку «Выход» программа перед закрытием должна выдавать сообщение типа: «Студент: Иванов И. И. Группы: ПИЭ – 001 завершил работу с программой», соответствующий подставив номер группы и ФИО, из указанных в первой форме.

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Задание 4.

Действителен: с 19.08.2022 по 19.08.2023

Создайте с помощью MS Access базу данных по сотрудникам фирмы «Кристалл»; в фирме работает 15 человек, база данных должна содержать данные, необходимые для расчета заработной платы: ФИО сотрудника, его должность, квалификационный разряд, количество отработанных дней в месяце и сумму оплаты за один день, кроме того, база данных должна содержать личные данные сотрудников: год рождения, номер паспорта, адрес проживания, телефон.

Создать с помощью средства разработки (среды программирования) программу, предоставляющую доступ к базе данных. Программа должна позволять получать доступ к данным по каждому из сотрудников фирмы, т.е. при выборе кого-либо сотрудника должны отображаться все данные по нему. При нажатии на кнопку «Расчет заработной платы» программа должна выдавать следующие сведения по выбранному сотруднику: ФИО сотрудника, общую начисленную сумму и сумму к выдаче, после вычитания налогов.

Задание 5.

Разработайте программное обеспечение, для решения какой либо прикладной задачи. При разработке данного программного обеспечения студент должен продемонстрировать свои навыки и умения, полученные в ходе изучения литературы по технологиям программирования при освоении данной дисциплины. Выбранная тематика разрабатываемого программного обеспечения должна быть согласована с преподавателем.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Вариант 8.

Задание 1.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

Форма 1

Настройки

☒ Непрозрачный

Цвет R G B

☒ Прямоугольник
☐ Круг
☐ Овал

Выход

Сведения о разработчике

Работу выполнил

Фамилия Имя Отчество Группа

Форма 2

Работу выполнил

Text1

Группы

Text2

Время входа в программу

Время окончания работы с программой

Выход

В первой форме:

- должны заноситься ФИО и номер группы студента;
- В данной форме фигура, занимающая основное место должна:
 - при переключении менять форму: прямоугольник, круг и овал;
 - при установке флажка должна включаться и отключаться ее прозрачность;
 - при изменении положения полос прокрутки - менять цвет;
- При нажатии на кнопку «Сведения о разработчике» должна появляться надпись, информирующая о разработчике программы;
- при нажатии на кнопку «Выход» программа должна переходить ко второй форме.

Во второй форме:

- ФИО студента и номер группы, указанные в первой форме, должны быть отображены в соответствующих местах;
- вместо слов «Время входа в программу» должно быть отображено время открытия первой формы;
- вместо слов «Время окончания работы с программой» должно быть отображено время закрытия первой формы;
- при нажатии на кнопку «Выход» программа должна закрываться.

Документ подписан
Электронный документ
Сертификат: 2C0000043E9AB8B952285E7BA500060000043E
Владелец: Шебухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Задание 2.

Напишите программу определения наличия слова «компьютер» в произвольном тексте (текст должен вводиться пользователем при работе с программой).

Задание 3.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

Форма 1

Группа: 001

ФИО: Иван Андреевич Клепалко

Дата

Время

Описание картинки 1

Описание картинки 2

Задание выполнил...

Выход

Форма 2

Работу выполнил: Text1

Группы: Text2

Время входа в программу

Время окончания работы с программой

Выход

В первой форме:

- должны заноситься ФИО и номер группы студента;
- вместо слов «Дата» и «Время» должны выводиться текущие дата и время;
- при нажатии на кнопку «Задание выполнил» должна появляться надпись о создателе программы;
- при нажатии на кнопку «Описание картинки 1» должна появляться соответствующая надпись;
- при нажатии на кнопку «Описание картинки 2» должна появляться соответствующая надпись;
- при нажатии на кнопку «Выход» программа должна переходить ко второй форме.

Во второй форме:

- ФИО студента и номер группы, указанные в первой форме, должны быть отображены в соответствующих местах;
- вместо слов «Время входа в программу» должно быть отображено время открытия первой формы;
- вместо слов «Время окончания работы с программой» должно быть отображено время закрытия первой формы;
- при нажатии на кнопку «Выход» программа должна закрываться.

Задание 4.

Создайте с помощью MS Access базу данных по реализуемым товарам фирмы «Руно»; фирма реализует 20 разновидностей товара, база данных должна содержать данные, необходимые для реализации товара: наименование товара, его приходная цена, наценка, налоги, кроме того, база данных должна содержать сведения о каждом из товаров: сорт, фирма производитель, дата производства, срок хранения.

Создать с помощью средства разработки (среды программирования) программу, предоставляющую доступ к базе данных. Программа должна позволять получать доступ к данным по каждому из 20 разновидностей товара, т.е. при выборе кого-либо товара должны отображаться все данные по нему. При нажатии на кнопку «Дата реализации» программа должна выдавать следующие сведения по выбранному товару: наименование товара, дату производства, срок хранения и дату, до которой должен быть реализован данный товар.

Задание 5.

Разработайте программное обеспечение, для решения какой либо прикладной задачи. При разработке данного программного обеспечения студент должен продемонстрировать свои навыки и умения, полученные в ходе изучения литературы по технологиям программирования при

освоении данной дисциплины. Выбранная тематика разрабатываемого программного обеспечения должна быть согласована с преподавателем.

Вариант 9.

Задание 1.

Создайте программу, позволяющую тестировать студентов по дисциплине «Информатика», тема: «Табличный процессор Excel», состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

После запуска программы должна открываться первая форма «Тестирование», в которой поочередно должны задаваться вопросы с тремя вариантами ответов (два неверных, один верный). После прохождения всех вопросов программа должна переходить ко второй форме: «Результаты опроса», в которой вместо слов «Начало тестирования» и «Окончание тестирования» должны указываться фактическое время соответственно открытия и закрытия первой формы. Также, должны быть указаны результаты теста – количество правильных и неправильных ответов и оценка работы тестируемого. При нажатии кнопки «Выход» программа должна завершать свою работу.

Задание 2.

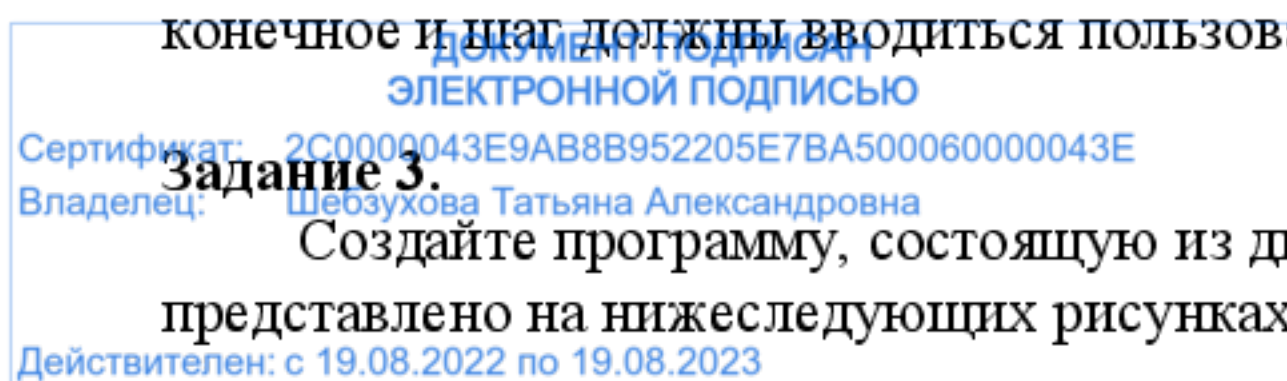
Составьте программу нахождения значения Y , если известно:

$$15 Y = \sum_{X=X_n}^{X=X_k} X^{\frac{1}{2X}}$$

Для X , изменяющегося от $X_{\text{начальное}}$ (X_n) до $X_{\text{конечное}}$ (X_k) с шагом C . Значения $X_{\text{начальное}}$, $X_{\text{конечное}}$ и шаг должны вводиться пользователем при работе с программой.

Задание 3.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках.



В первой форме должны заноситься ФИО и номер группы студента, при нажатии на кнопку «Старт» программа должна переходить ко второй форме.

Во второй форме:

- ФИО студента и номер группы, указанные в первой форме, должны быть отображены в соответствующих местах;
- вместо слов «Дата» и «Время» должны выводиться текущие дата и время;
- при нажатии на кнопку «Выход» программа должна закрываться;
- при нажатии на кнопку «Сообщение о создателе программы» должна появляться надпись, информирующая о создателе программы;
- при нажатии на кнопку «Описание картинки 1» должна появляться поясняющая надпись об изображении на картинке 1;
- при нажатии на кнопку «Описание картинки 2» должна появляться поясняющая надпись об изображении на картинке 2.

Задание 4.

Создайте с помощью MS Access базу данных по студентам группы 357 база данных должна содержать данные по успеваемости студентов: ФИО студента и его оценки по всем предметам, кроме того, база данных должна содержать личные данные студентов: год рождения, номер паспорта, адрес проживания, телефон.

Создать с помощью средства разработки (среды программирования) программу, предоставляющую доступ к базе данных. Программа должна позволять получать доступ к данным

Сертификат:
Владелец:
Действителен: с 19.08.2022 по 19.08.2023

по каждому из студентов группы 357, т.е. при выборе кого-либо студента должны отображаться все данные по нему. При нажатии на кнопку «Успеваемость» программа должна выдавать следующие сведения по выбранному студенту: ФИО студента, оценки по всем предметам и средний балл.

Задание 5.

Разработайте программное обеспечение, для решения какой либо прикладной задачи. При разработке данного программного обеспечения студент должен продемонстрировать свои навыки и умения, полученные в ходе изучения литературы по технологиям программирования при освоении данной дисциплины. Выбранная тематика разрабатываемого программного обеспечения должна быть согласована с преподавателем.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Вариант 10.

Задание 1.

Создайте программу, состоящую из двух форм, и внесите все необходимые элементы, как представлено на нижеследующих рисунках:

1

3

4

2

В первой форме:

- должны заноситься ФИО и номер группы студента;
- вместо слов «Дата» и «Время» должны выводиться текущие дата и время;
- при нажатии на кнопку «Старт» программа должна переходить ко второй форме.

Во второй форме:

- в поля «1» и «2» должны вводиться числовые значения, причем, при вводе числового значения вместо надписи «Комментарий» должна появляться надпись «ОК», а при вводе текстового значения должна появляться надпись «Введено неверное значение» и кнопки «+ - * / », должны блокироваться.
- при нажатии кнопок «+» (плюс), «-» (минус), «*» (умножить), «/» (разделить) должны производиться соответствующие действия со значениями, введенными в поля «1» и «2». Результат вычислений должен отображаться в поле «3». В списке «4» должны отображаться все получаемые значения поля «3».
- при нажатии на кнопку «C» должно происходить обнуление полей «1», «2» и «3».
- при нажатии на кнопку «Функция» должна вычисляться следующая функция по значениям полей «1» и «2», причем, в случае деления на ноль и вычислении квадратного корня из отрицательного числа программа должна выдавать сообщение об ошибке и предлагать произвести ввод данных заново:

Документ подписан
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

$$A^{\frac{B}{2}} \cdot (\sqrt{A} - \sqrt{B}) \cdot \frac{\sqrt[3]{AB}}{A \cdot B}$$

- при нажатии на кнопку «Сведения о разработчике» должна появляться надпись, информирующая о разработчике программы;
- при нажатии на кнопку «Выход» программа должна закрываться.

Задание 2.

Составьте программу определения функции:

$$Y = \begin{cases} 15-X, & \text{если } (X < -1); \\ 15-X^2, & \text{если } (-1 \leq X \leq 1); \\ X^3 - 15, & \text{если } (X > 1); \end{cases}$$

Для X , изменяющегося от $X_{\text{начальное}}$ (X_n) до $X_{\text{конечное}}$ (X_k) с шагом C . Значения X начальное, X конечное и шаг должны вводиться пользователем при работе с программой.

Задание 3.

Создайте программу, позволяющую тестировать студентов по дисциплине «Информатика», тема: «СУБД MS Access». Программа должна состоять из одной формы, в которой поочередно должны задаваться вопросы (не менее 15 вопросов) с четырьмя вариантами ответов (три неверных, один верный). После прохождения всех вопросов программа должна выдавать результаты тестирования и время начала и окончания прохождения теста.

Задание 4.

Создайте с помощью MS Access базу данных по реализуемым товарам фирмы «Милена»; фирма реализует 20 разновидностей товара, база данных должна содержать данные, необходимые для реализации товара: наименование товара, его приходная цена, наценка, налоги, кроме того, база данных должна содержать сведения о каждом из товаров: количество товаров на складе, сорт, фирма производитель, дата производства, срок хранения.

Создать с помощью средства разработки (среды программирования) программу, предоставляющую доступ к базе данных. Программа должна позволять получать доступ к данным по каждому из 20 разновидностей товара, т.е. при выборе кого-либо товара должны отображаться все данные по нему. При нажатии на кнопку «Расчет» программа должна выдавать следующие сведения по выбранному товару: наименование товара, приходная цена всего товара данного вида, находящегося на складе, и суммарная отпускная цена этого товара на складе.

Задание 5.

Разработайте программное обеспечение, для решения какой либо прикладной задачи. При разработке данного программного обеспечения студент должен продемонстрировать свои навыки и умения, полученные в ходе изучения литературы по технологиям программирования при освоении данной дисциплины. Выбранная тематика разрабатываемого программного обеспечения должна быть согласована с преподавателем.

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

При написании курсовой работы необходимо изучить методические рекомендации к курсовой работе, учебную литературу, сформулировав главы по своей тематике.

Предлагается придерживаться следующего примерного плана представления отдельных разделов курсовой работы (таблица 1).

Таблица 1 – Примерный план представления отдельных разделов курсовой работы

25 %	Сбор и анализ материалов необходимых для выполнения курсовой работы
50 %	Разработка теоретической части работы
75 %	Разработка практической части работы
100 %	Оформление курсовой работы

ПОРЯДОК ПРЕДСТАВЛЕНИЯ КУРСОВОЙ РАБОТЫ К ЗАЩИТЕ

Курсовая работа сдается на кафедру в установленные сроки и регистрируется в присутствии студента. Кафедра не несет ответственности за работы, сданные без регистрации. В случае пропажи таких работ они считаются несданными.

Курсовая работа проверяется преподавателем, и впоследствии допускается или не допускается к защите. Защита работы является обязательной.

В ходе защиты работы студент должен устно изложить содержание работы и ответить на дополнительные вопросы преподавателя по дисциплине.

Преподаватель дает оценку качественному уровню проделанной работы, усвоению теоретического материала.

Курсовая работа оценивается дифференцированно. Преподаватель может вернуть курсовую работу на доработку в случае неправильного её оформления или не раскрытия должным образом тематики курсовой работы. Незачтенная работа перерабатывается и сдается на повторную проверку.

Если работа выполнена не по теме закрепленной приказом кафедры, то она возвращается студенту как не выполненная работу.

Получив проверенную курсовую работу, необходимо внимательно ознакомиться со всеми замечаниями, внести исправления. Если курсовая работа не защищена, то студент получает оценку не удовлетворительно.

Учебно-методическое и информационное обеспечение дисциплины

Перечень основной и дополнительной литературы, необходимой для освоения дисциплины (модуля)

Перечень основной литературы:

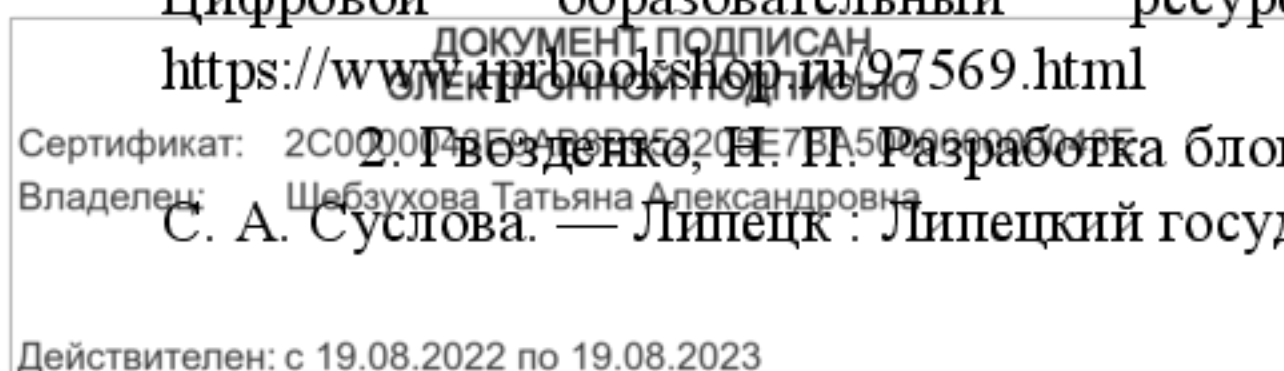
1. Битюцкая, Н. И. Разработка программных приложений : лабораторный практикум / Н. И. Битюцкая. — Ставрополь : Северо-Кавказский федеральный университет, 2015. — 140 с.

2. Абрамов, Г. В. Проектирование и разработка информационных систем : учебное пособие для СПО / Г. В. Абрамов, И. Е. Медведкова, Л. А. Коробова. — Саратов : Профобразование, 2020. — 169 с. — ISBN 978-5-4488-0730-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/88888.html>

Перечень дополнительной литературы:

1. Баженова, И. Ю. Основы проектирования приложений баз данных : учебное пособие / И. Ю. Баженова. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 324 с. — ISBN 978-5-4497-0682-9. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/97569.html>

2. Гвозденко, Н. П. Разработка блок-схем алгоритмов : учебное пособие / Н. П. Гвозденко, С. А. Суслова. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2021.



— 59 с. — ISBN 978-5-00175-055-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/116169.html>

3. Савельев, А. О. Проектирование и разработка веб-приложений на основе технологий Microsoft : учебное пособие / А. О. Савельев, А. А. Алексеев. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2022. — 418 с. — ISBN 978-5-4497-1650-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/120486.html>

Перечень учебно-методического обеспечения самостоятельной работы обучающихся по дисциплине (модулю)

1. Методические рекомендации по выполнению лабораторных работ по дисциплине "Технология разработки программного обеспечения "

2. Методические рекомендации по организации самостоятельной работы студентов по дисциплине " Технология разработки программного обеспечения "

1. Методические указания по выполнению курсовой работы по дисциплине «Технология разработки программного обеспечения».

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)

1. <http://el.ncfu.ru/> – система управления обучением ФГАОУ ВО СКФУ. Дистанционная поддержка дисциплины «Цифровая грамотность и обработка данных»

2. <http://www.un.org> - Сайт ООН Информационно-коммуникационные технологии

3. <http://www.intuit.ru> – Интернет-Университет Компьютерных технологий.

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем

При чтении лекций используется компьютерная техника, демонстрации презентационных мультимедийных материалов. На семинарских и практических занятиях студенты представляют презентации, подготовленные ими в часы самостоятельной работы.

Информационные справочные системы:

Информационно-справочные и информационно-правовые системы, используемые при изучении дисциплины:

1	КонсультантПлюс - http://www.consultant.ru/
---	---

Программное обеспечение:

1	Операционная система: Microsoft Windows 8: 2013-02(3000). Бессрочная лицензия. Договор № 01-за/13 от 25.02.2013. Окончание бесплатной поддержки – 2023-01 ИЛИ Операционная система: Microsoft Windows 10: 2016-08(20), 2017-10(67), 2018-01(18), 2018-04(6), 2018-05(6), 2019-02(7). Бессрочная лицензия. Договоры № 27-за/16 от 02.08.2016. и № 0321100021117000009_229123 от 10.10.2017. На текущий момент окончания поддержки не анонсировано.
2	Базовый пакет программ Microsoft Office (Word, Excel, PowerPoint). MicrosoftOfficeStandard 2013: договор № 01-за/13 от 25.02.2013г., Лицензирование Microsoft Office https://support.microsoft.com/ru-ru/lifecycle/search/16674 Дата начала жизненного цикла 09.01.2013г.; набор обновлений Office 2013 Service Pack1 Дата начала жизненного цикла 25.02.2014г., Дата окончания основной фазы поддержки 10.04.2018; Дополнительная дата окончания поддержки 11.04.2023г.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шесухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Лабораторные занятия	Учебная аудитория для проведения занятий семинарского типа (практических работ) — компьютерная аудитория	Количество рабочих мест – 12 Оборудование: Персональные компьютеры, объединенные в локальную сеть и имеющие выход в интернет
Самостоятельная работа	Помещения для самостоятельной работы	Компьютеры с выходом в Интернет и обеспечением доступа в электронную информационно-образовательную среду ИСУ СКФУ.

Учебные аудитории для проведения учебных занятий, оснащены оборудованием и техническими средствами обучения. Помещения для самостоятельной работы обучающихся, оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа к электронной информационно-образовательной среде. Специализированная мебель и технические средства обучения, служащие для представления учебной информации.

Материально-техническая база обеспечивает проведение всех видов дисциплинарной и междисциплинарной подготовки, лабораторной, научно-исследовательской работы обучающихся (переносной ноутбук, переносной проектор, компьютеры с необходимым программным обеспечением и выходом в интернет).

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду образовательной организации.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ

Методические указания
по выполнению лабораторных работ
по дисциплине «ТЕХНОЛОГИИ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

Направление подготовки/специальность 09.04.02 Информационные системы и технологии
Направленность (профиль)/специализация Технологии работы с данными и знаниями, анализ информации
Форма обучения очная, заочная
Год начала обучения 2023
Изучается в 3 семестре

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

СОДЕРЖАНИЕ

1. Цель и задачи освоения дисциплины.....	3
2. Компетенции обучающегося, формируемые в результате изучения дисциплины.....	3
3. Наименование лабораторных занятий.....	3
4. Содержание лабораторных занятий.....	4
5. Критерии оценивания компетенций.....	1 13
6. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.....	1 13
7. Учебно-методическое и информационное обеспечение дисциплины.....	1 14

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

1. ЦЕЛЬ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью изучения дисциплины «Технологии разработки программного обеспечения» учебного плана подготовки магистров по направлению подготовки 09.04.02 «Информационные системы и технологии», является получение компетенций, достаточных для анализа требований к программным системам, их документирования, проектирования, разработки, тестирования, внедрения, управления программными проектами и управления качеством разработки программных систем.

Задачами курса является приобретение и развитие знаний, умений и навыков для производственно-технологической, организационно-управленческой, проектной и научно-исследовательской деятельности.

2. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесённых с планируемыми результатами освоения образовательной программы

Код, формулировка компетенции	Код, формулировка индикатора	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций, индикаторов
ПК-2 способен создания технической документации информационно-методического и маркетингового назначения в сфере информационных технологий и систем	ИД-1 ПК-2 Создает техническую документации информационно-методического назначения в сфере информационных технологий и систем ИД-2 ПК-2 Применяет техническую документацию для создания информационных технологий и систем ИД-3 ПК-2 Использует техническую документацию для решения маркетинговых задач в сфере информационных технологий и систем;	Проводит предпроектное обследование объекта проектирования, анализ научно-технической информации, отечественного и зарубежного опыта
ПК-4 способен выполнять разработку систем управления базами данных, операционных систем, организацию разработки системного программного обеспечения, интеграция разработанного системного программного обеспечения	ИД-1 ПК-4 Выполняет разработку систем управления базами данных. ИД-2 ПК-4 Проводить непосредственное руководство процессами разработки программного обеспечения; ИД-3 ПК-4 Проводить организацию разработки системного программного обеспечения, интеграцию разработанного системного программного обеспечения.	Проводит анализ разработанных программных приложений
ПК-6 способен проводить организационное сопровождение разработки, отладки, модификации и поддержки информационных технологий и систем	ИД-1 ПК-6 Проводить организационное сопровождение процессом разработки ПО ИД-2 ПК-6 Выполняет управление проектами в области ИТ любого масштаба ИД-3 ПК-6 Проводить отладку, модификацию и	Проводит техническое проектирование и сопровождение программных приложений

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043
Владелец: Шебзухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

	поддержку информационных технологий и систем	
ПК-10 способен выполнять управление аналитическими работами и подразделением	ИД-1 ПК-10 Выполняет разработку новых инструментов и методов управления проектами в области ИТ. ИД-2 ПК-10 Проводить разработку новых инструментов; ИД-3 ПК-10 Использует методы управления проектами в области ИТ	Разрабатывает отдельные компоненты информационной системы

3. НАИМЕНОВАНИЕ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

№ Темы дисциплины	Наименование тем дисциплины, их краткое содержание	Очная	Заочная
2	Лабораторная работа 1. Моделирование объекта автоматизации.	4,5	1,5
3	Лабораторная работа 2. <i>Разработка модели вариантов использования и их спецификаций.</i>	3	1,5
4	Лабораторная работа 3. <i>Реализация архитектуры на базе объектно-реляционного отображения с типизированными объектами.</i>	3	1,5
	Итого за семестр	3	4,5
	Итого	13,5	4,5

4. СОДЕРЖАНИЕ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

Тема 2. Анализ проблемы и постановка задачи.

Лабораторная работа № 1. Анализ проблемы, постановка задачи

Форма проведения: Практическое выполнение задания

Цель работы – сформировать навыки:

- работы с заказчиками программных систем;
- идентификации заинтересованных лиц и интервью с ними;
- анализа полученного материала;
- формулирования проблемы, ее актуальности и потребностей заинтересованных лиц.

Краткие теоретические сведения

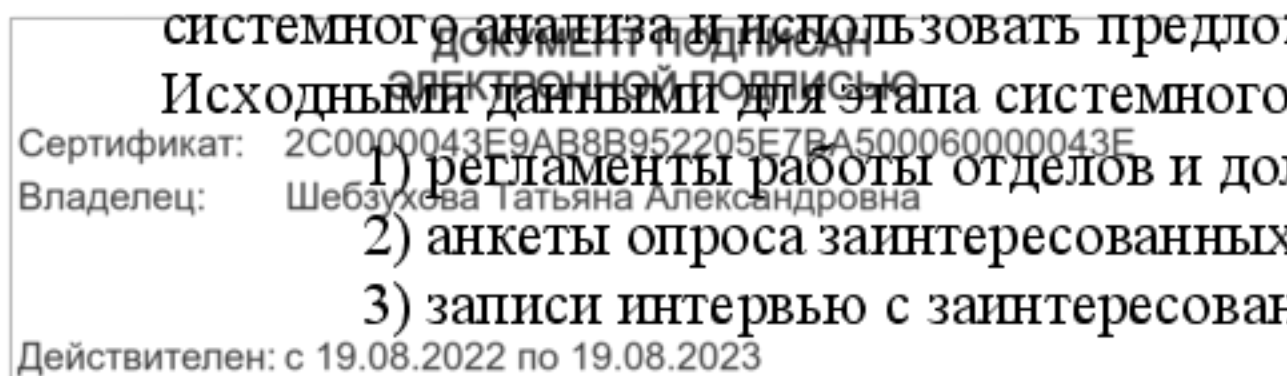
На этапе анализа проблемы проводится анализ предметной области, для которой разрабатывается ПО. Цели этапа:

- 1) определение границ, или контура, системы;
- 2) описание объектов автоматизации и/или формализации знаний об этих объектах;
- 3) выявление или определение потребностей заказчика ПО.

Анализ предметной области можно проводить, например, основываясь на теории системного анализа и использовать предложенные в ней методы.

Исходными данными для этапа системного анализа являются:

- 1) регламенты работы отделов и должностные инструкции сотрудников этих отделов;
- 2) анкеты опроса заинтересованных лиц;
- 3) записи интервью с заинтересованными лицами;



4) другие документы, имеющие отношение к исследуемому объекту.
Выходными данными, или результатом, этапа системного анализа являются:

- 1) перечень заинтересованных лиц;
- 2) список потребностей заинтересованных лиц в разрабатываемом ПО;
- 3) описание объектов автоматизации;
- 4) модель объектов автоматизации или предметной области.

Описание примера

Здесь формулируется задача, решением которой является разработка программного обеспечения.

Итак, в результате вступления России в Болонский процесс в РФ была инициирована реформа высшего профессионального образования, в соответствии с которой Министерством образования и науки РФ была разработана программа перевода традиционной системы оценки успеваемости студентов в систему зачетных единиц (кредитов). Это объясняется необходимостью унификации систем высшего образования с целью создания единого образовательного пространства в тех странах, которые уже вступили в Болонский процесс.

В рамках этой программы все вузы страны должны к установленной дате перейти на новую систему.

Некоторый вуз, будем называть его просто «Университет» начал решать поставленную перед ним задачу поэтапно. Одной из задач перехода на новую систему в «Университете» являлась автоматизация учета текущей успеваемости и промежуточных аттестаций студентов в целях унификации этого процесса на всех кафедрах и факультетах вуза, реализации возможности автоматизированного формирования отчетов, публикации на сайте вуза рейтингов успеваемости студентов.

Составление списка заинтересованных лиц

Заинтересованные лица – это все те, кто имеет прямое или косвенное отношение к процессу, автоматизация которого производится.

Для выявления заинтересованных лиц необходимо ответить на следующие вопросы:

кто является пользователем системы?

кто является заказчиком (покупателем) системы?

на кого еще окажут влияние результаты работы системы?

кто будет оценивать и принимать систему, когда она будет представлена и развернута?

существуют ли другие внутренние или внешние пользователи системы, чьи потребности необходимо учесть?

кто будет заниматься сопровождением новой системы?

не забыли ли мы кого-нибудь?

В нашем примере определим будущих пользователей системы – это преподаватели, секретари кафедр и деканатов, заведующие кафедрами, системный администратор и сотрудники Учебного управления. Заказчиком нашей системы является вуз в лице первого проректора.

Теперь попытаемся выяснить, на кого еще будут оказывать влияние результаты работы нашей системы. Во-первых, на студентов, ведь рейтинги успеваемости всех студентов будут доступны на сайте вуза. Во-вторых, по этой же причине, на родителей. В-третьих, на деканов факультетов и заместителей деканов по учебной работе, поскольку любые изменения в учебном процессе касаются их профессиональной деятельности.

Итак, рассмотрев первые три вопроса, мы практически охватили всех заинтересованных лиц. Поскольку сопровождать систему будет разработчик этой системы, то в список заинтересованных лиц мы его не включаем. Таким образом, получаем следующий список заинтересованных лиц для нашей системы:

преподаватели;

секретари кафедр;

секретари деканатов;

заведующие кафедрами;

Учебное управление (в лице начальника);

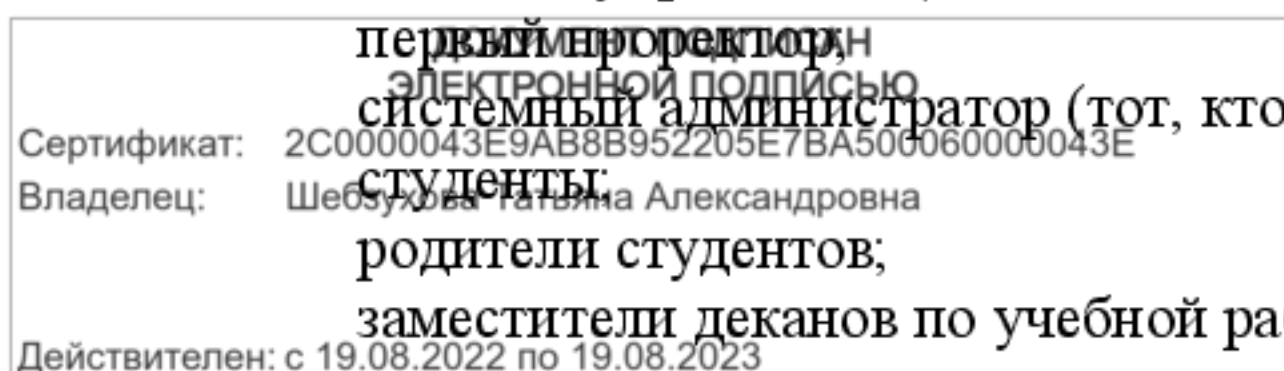
первый проректор;

системный администратор (тот, кто будет администрировать нашу систему);

студенты;

родители студентов;

заместители деканов по учебной работе;



деканаты факультетов.

Анкетирование и проведение интервью

Для выявления потребностей заказчика и описания объектов автоматизации можно проводить как анкетирование, так и интервью. Но наибольший эффект возможен только при проведении и того и другого.

Примеры анкеты и перечня вопросов для интервью приведены ниже.

Анкета для опроса заинтересованных лиц

1. Имя.
2. Наименование организации.
3. Наименование структурного подразделения.
4. Должность.
5. Кому Вы непосредственно подчиняетесь?
6. Каковы Ваши основные обязанности?
7. Что Вы в основном производите?
8. Для кого?
9. Какие документы или какую информацию можно считать входящими, или необходимыми, для Вашей деятельности?
10. Какие документы или какую информацию можно считать исходящими, или результатом Вашей деятельности?
11. Как измеряется успех Вашей деятельности?
12. Какие проблемы влияют на успешность Вашей деятельности?
13. Какие тенденции, если такие существуют, делают Вашу работу проще или сложнее?
14. Какой интерес или какие потребности у Вас есть относительно будущего решения (разрабатываемого ПО)?

Перечень вопросов для интервью

Оценка проблемы

Для каких проблем (прикладного типа) Вы ощущаете нехватку хороших решений? Назовите их. (Не забывайте спрашивать: «А еще?»)

По каждой проблеме выясните следующее:

почему существует эта проблема?

как она решается в настоящее время?

как заказчик (пользователь) хотел бы ее решать?

Понимание пользовательской среды

Каковы Ваши навыки в компьютерной области?

С какими типами приложений Вы имеете опыт работы?

Какая платформа используется?

Каковы Ваши планы относительно будущих платформ?

Используется ли ПО, которое имеет отношение к данной проблеме?

(Если да, то пусть о нем немного расскажут).

Каковы Ваши ожидания относительно практичности продукта?

В каком виде должна быть представлена справочная информация для пользователя (в интерактивном или печатном)?

Резюме (перечисляются основные пункты, чтобы проверить, все ли правильно вы поняли):

Итак, Вы сказали мне... (перечислите описанные заказчиком проблемы своими словами).

Адекватно ли этот список представляет проблемы, которые имеются при существующем решении?

Какие еще проблемы Вы испытываете?

Заключение аналитика. После интервью, пока его данные еще свежи в вашей памяти, зафиксируйте не менее трех потребностей или проблем с наивысшими приоритетами, выявленных вами в беседе с данным заказчиком (пользователем).

После проведения анкетирования и интервьюирования необходимо обработать собранную информацию. На основе этих данных нужно сформулировать перечень потребностей заказчиков, построить ~~документ предметной области~~ и описать объект/объекты автоматизации. Все эти результаты в дальнейшем будут использованы при написании технического задания (ТЗ) на разрабатываемую систему.

Допустим, что в качестве примера было проведено анкетирование и интервью с начальником Учебного управления. Для этого, мы воспользовались одним перечнем вопросов,

совместив две эти процедуры воедино. По сути, мы провели только интервью. Предполагаемый пример результата интервью (вопросы и ответы) приведен в таблице 1.1.

Таблица 1.1 - Интервью с начальником учебного отдела

Вопрос	Ответ
1. Имя	Иванов Иван Иванович
2. Наименование структурного подразделения	Учебное управление
3. Должность	Начальник Учебного управления
4. Кому Вы непосредственно подчиняетесь?	Первому проректору
5. Каковы Ваши основные обязанности?	Обеспечение общего управления учебной деятельностью. Обеспечение качества и своевременного выполнения возложенных на Учебное управление задач и функций. Обеспечение выполнения мероприятий по защите конфиденциальной и информационной безопасности в подразделениях управления.
6. Что Вы в основном производите?	Функции, выполняемые Учебным управлением: 1. Создание единого нормативно-справочного поля учебного процесса. 2. Организационное сопровождение учебного процесса. 3. Планирование ресурсов учебного процесса (объемов работ кафедр, площадей, численности контингента, нагрузки на условиях почасовой оплаты, бланочной документации и др.). 4. Контроль за ходом организации и анализ показателей учебного процесса (как внешних, так и внутренних). 5. Ответственность на различных уровнях управления. 6. Анализ выполнения учебной нагрузки, трудоемкости отдельных параметров учебного процесса, результатов сессии, результатов государственных аттестационных комиссий. Разработка графика учебного процесса. 7. Составление расписания учебных занятий, экзаменов, зачетов и графика защит. 8. Контроль занятости аудиторного фонда, составления семестровых планов, проведения занятий, заседаний государственных аттестационных комиссий, хода сессий, делопроизводства факультетов, оформления приложений к дипломам. 9. Учет движения студентов по всем формам обучения (очное, очно-заочное и заочное, второе высшее образование, обучение по сокращенным программам).
7. Какие документы или какую информацию можно считать входящими, или необходимыми, для Вашей деятельности?	1. Приказы и инструктивные письма Минобразования России по учебно-методическим вопросам. 2. Положение об Учебном управлении. 3. Должностные инструкции сотрудников. 4. Приказы ректора по контингенту студентов (первые экземпляры). 5. Годовые планы приема абитуриентов. 6. Планы работы факультетов. 7. Отчеты председателей государственных аттестационных комиссий по всем специальностям и направлениям.
8. Какие документы или какую информацию можно считать исходящими, или результатом Вашей деятельности?	1. Семестровые планы занятий. 2. Расписание учебных занятий. 3. Расписание экзаменов. 4. Отчеты вуза по учебно-методической работе за учебный год. 5. Сводные статистические отчеты вуза о движении контингента студентов на начало и конец учебного года.

Сертификат
Владелец: Алло Бризков, Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

9. Как измеряется успех Вашей деятельности?	В настоящее время отсутствуют количественные показатели оценки деятельности управления
10. Какие проблемы влияют на успешность Вашей деятельности?	Проблемы, связанные с организацией учебного процесса. В настоящее время основной проблемой является перевод учебного процесса в систему зачетных единиц.
11. Какой интерес или какие потребности у Вас есть относительно будущего решения (разрабатываемого ПО)?	Разрабатываемая система должна быть максимально эргономичной, работать стабильно (без сбоев); отклик системы не должен вызывать у пользователей раздражения; реализуемая функциональность должна полностью удовлетворить потребности пользователя.

Список потребностей заинтересованных лиц

В результате анкетирования и интервьюирования всех заинтересованных лиц были сформулированы потребности заказчика относительно разрабатываемого ПО. Далее необходимо провести аналогию между выявленными потребностями и структурой и требованиями ТЗ в соответствии с ГОСТ 34.602–89. Таким образом, потребности заказчика в ТЗ могут быть описаны в разделе «Назначение и цели создания системы».

Допустим, что в нашем примере были выявлены следующие потребности:

- 1) унифицировать процесс оценивания знаний в системе кредитов на всех кафедрах и факультетах вуза;
- 2) минимизировать субъективность при оценивании студентов в промежуточных аттестациях;
- 3) реализовать возможность автоматического формирования рейтингов студентов по разным параметрам;
- 4) реализовать возможность формирования единой отчетности на кафедрах и факультетах.

Задание и порядок проведения работы

1. Составить перечень заинтересованных лиц.
2. Провести интервью и/или анкетирование с каждым заинтересованным лицом.
3. Проанализировать полученную информацию и сформулировать актуальность проблемы и потребности заинтересованных лиц.

Контрольные вопросы

1. Что является исходными данными для анализа проблемы (предметной области)?
2. Что является результатом этапа системного анализа предметной области?
3. Как определить заинтересованных лиц?
4. Какой метод сбора информации наиболее эффективен?
5. Для чего проводятся интервьюирование и анкетирование?

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-3	1-3	1-3	1-2

Оценочные средства: контрольные вопросы

Тема 2. Анализ проблемы и постановка задачи.

Лабораторная работа № 2. Моделирование объекта автоматизации

Форма проведения: Практическое выполнение задания

Цели работы:

- познакомиться с методологией ARIS;
- сформировать навыки работы в среде ARISToolset;
- сформировать навыки разработки организационных диаграмм и диаграмм бизнес-процессов (VAD и EPC).

Краткие теоретические сведения

На основе собранных данных разрабатывается модель предметной области. Модель – это мысленно представленный, или изображенный (например, нарисованный на бумаге), или

изготовленный (например, бумажный макет) образ, аналог оригинала (объекта, процесса или явления), который «замещает» оригинал и отражает его наиболее важные черты и свойства.

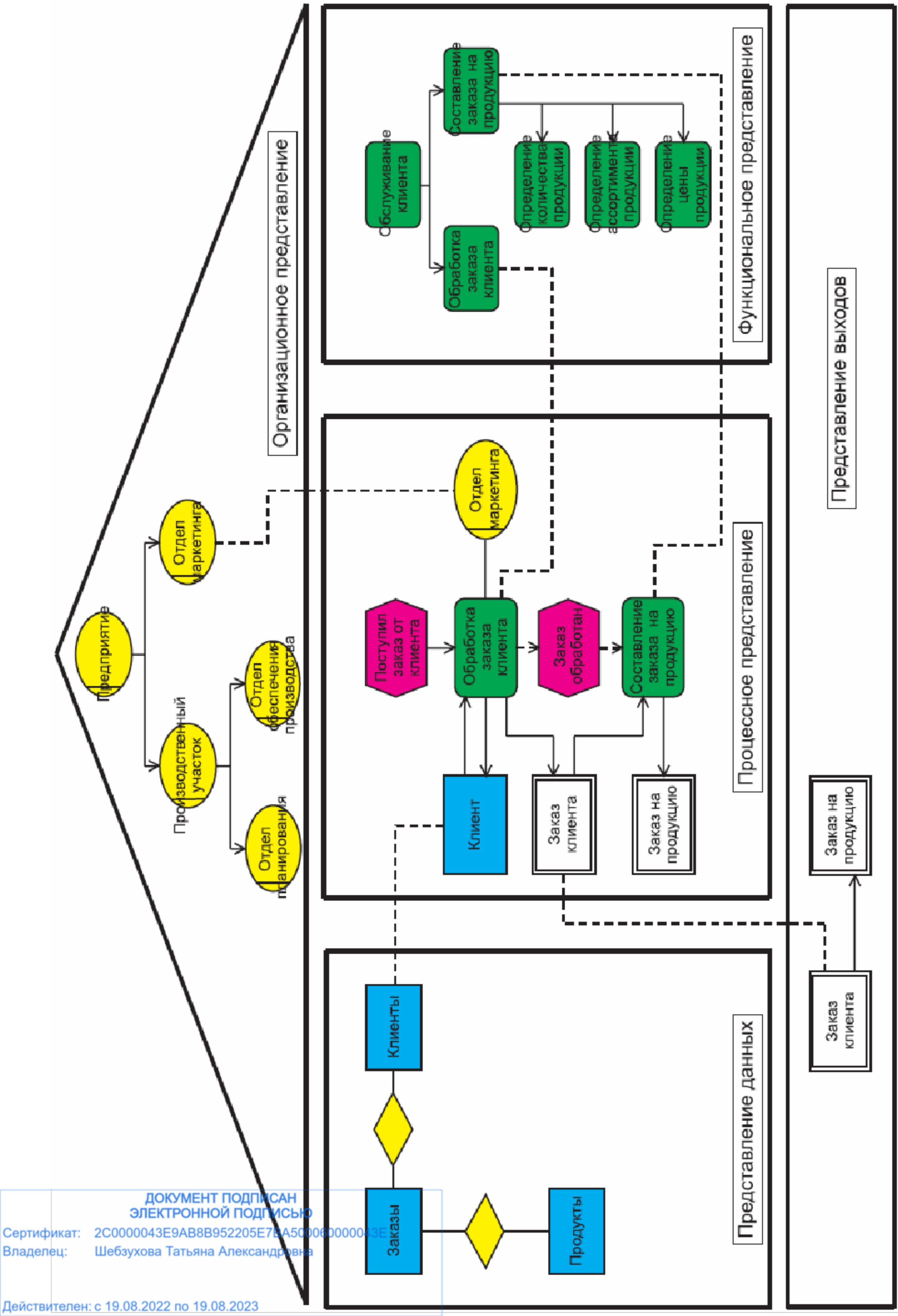


Рисунок 2.1 - «Здание» ARIS. Представления

ARIS-модели – это диаграммы экономических систем и процессов, на основе которых можно анализировать, оптимизировать бизнес-процессы (БП) и управлять ими.

В рамках данной лабораторной работы предлагается строить модель с использованием инструмента ARIS. Выбор этого инструмента и методологии обусловлен заложенными в них принципами, которые основаны на теории систем.

Введение в методологию ARIS

В теории систем различают структуру системы и ее поведение. Структура описывает статику системы, а поведение – ее динамику. Четыре первых представления в ARIS (организационное, функциональное, представление выходов и представление данных) описывают структуру системы. В процессном представлении устанавливаются связи перечисленных представлений и описывается динамическое поведение системы. На рисунке 2.1 перечисленные представления сгруппированы в пять частей, образующих «здание» ARIS.

Функциональное представление содержит описание целей, выполняемых функций, перечень отдельных подфункций, а также общие взаимосвязи и связи подчиненности, которые существуют между функциями, целями и факторами успеха.

Организационное представление описывает взаимодействие пользователей и организационных единиц, а также их связи и имеющие к ним отношение структуры. Организационные модели служат для описания иерархической структуры организации (где группируются субъекты ответственности), организационных единиц, должностей и средств, выполняющих работу над объектами, и др.

В представлении данных описывают информационную среду предприятия (среду обработки данных), события, сообщения и др., где неявно фиксируются также объекты в виде информационных услуг.

Процессное представление используется для описания связей между тремя предшествующими представлениями. Интеграция этих связей в пределах отдельного представления делает возможным учесть все связи без избыточности. Модели представления процессов описывают отношения между отдельными моделями в рамках всего бизнес-процесса. Это позволяет отслеживать все двусторонние и многосторонние отношения между моделями различных видов, а также полностью описать бизнес-процесс.

Представление выходов предназначено для описания результатов выполнения процессов.

Описание инструмента ARIS. Начало работы

Аналогично большинству приложений Windows, ARIS имеет многооконный интерфейс. Главное окно ARIS (рисунок 2.2) состоит из главного меню, панели инструментов, панели проводника (рисунок 2.3) и обзорного окна.

Прежде всего создайте новую базу данных на локальном сервере. Для этого щелкните правой кнопкой мыши по панели проводника на сервере LOCAL – появится контекстное меню. Выберите в нем New→Database.

В появившемся окне введите имя новой базы данных латинскими буквами, например Sample.

Для того чтобы элементы моделей имели русскоязычные имена, добавьте в список языков русский язык. Для этого откройте созданную вами базу данных, выберите папку Languages и через контекстное меню добавьте русский язык. Затем в главном меню выберите View→Options – появится окно настроек ARIS, общих для всех баз данных (рисунок 2.4). На закладке General измените язык English на «Русский» (рисунок 2.4) в поле Database Language.

Перезапустите ARIS.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA5000600001005
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

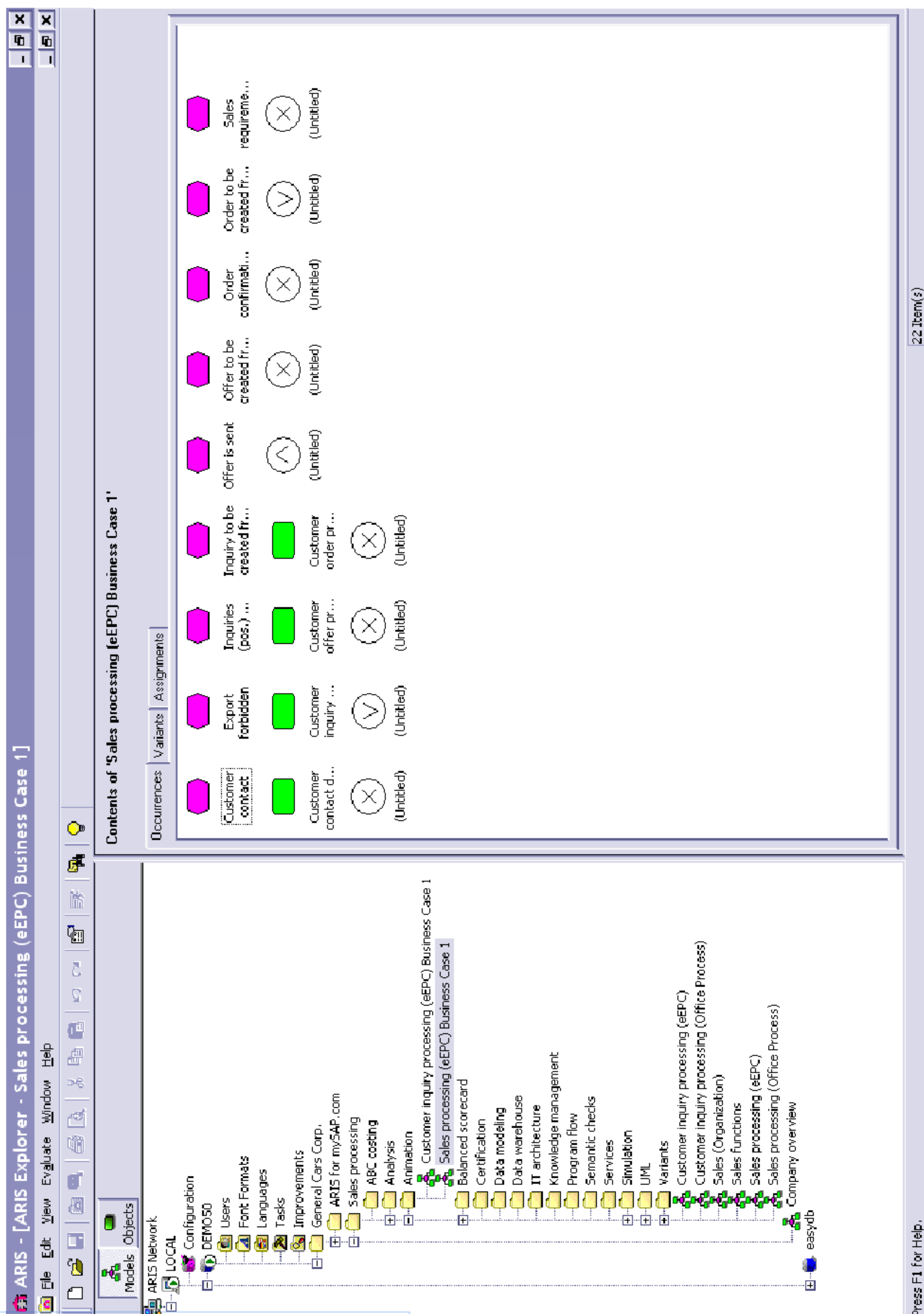


Рисунок 2.2 - Главное окно ARIS

промежуток (как в Windows-проводнике). На этом настройку инструмента и базы данных закончим и перейдем к построению моделей или моделированию предметной области.

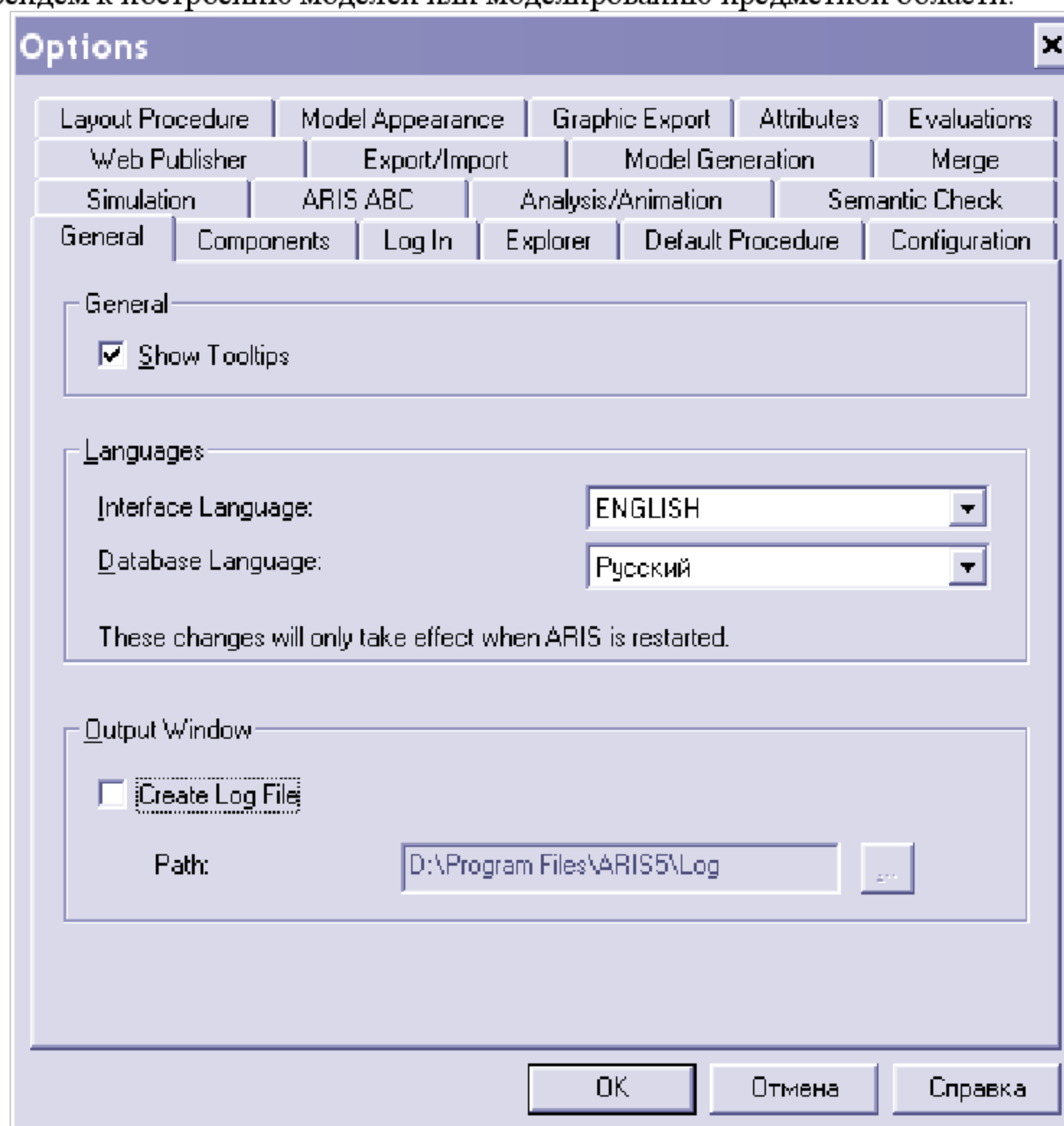


Рисунок 2.4 - Окно настроек

Построение организационной модели

Щелкните правой кнопкой мыши по главной папке и в появившемся контекстном меню выберите New→Model – появится окно выбора модели (рисунок 2.5). Поставьте галочку в поле Organization (рисунок 2.5), что соответствует организационному представлению. Ниже появится список всех моделей, имеющих отношение к выбранному представлению или к выбранным представлениям, если галочками будет помечено несколько полей.

Выберите диаграмму Organization Chart и нажмите кнопку Далее. Введите имя новой модели, например «Организационная структура».

Организационная диаграмма позволяет всесторонне описать организационно-штатную структуру предприятия. Организационные диаграммы показывают имеющиеся организационные подразделения (как исполнители функций) и их взаимозависимости в соответствии с выбранными критериями структурирования. Отдельные организационные единицы соединяются связями для указания иерархии.

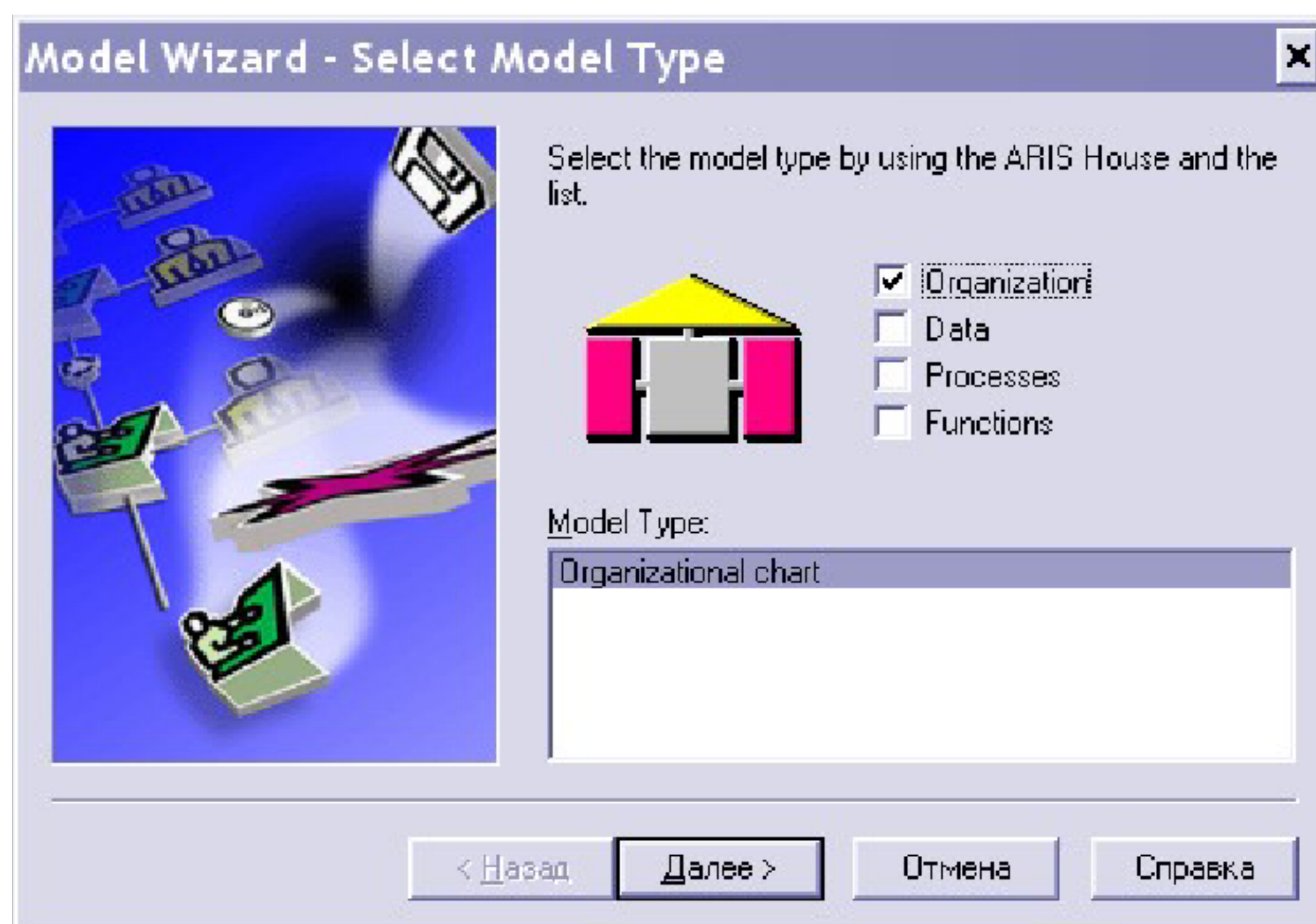


Рисунок 2.5 - Окно создания новой модели

Таблица 2.1 - Элементы организационной диаграммы

Графическая нотация	Наименование	Описание
	Организационная единица (Organizational unit)	Элемент организационной структуры (структурное подразделение), который отвечает за выполнение определенных задач и преследует определенные цели
	Должность или позиция (Position)	Наиболее мелкий организационный элемент на предприятии. Обязанности и административные полномочия такого элемента задаются должностными инструкциями
	Группа (Group)	Несколько сотрудников, которые вместе работают над конкретной задачей в определенный период времени (например, группа проектирования)
	Внешний сотрудник (External person)	Лицо, выполняющее определенные функции в компании, но не являющееся служащим этой компании. Может назначаться к организационным единицам или функциям, к выполнению которых он привлекается или за которые он отвечает
	Сотрудник (Internal person)	Лицо, являющееся служащим компании. Может назначаться к организационным единицам или функциям, которые он выполняет или за которые несет ответственность
	Тип сотрудника (Person type)	Обобщает индивидуальных лиц, имеющих одинаковые характеристики. Эти характеристики могут иметь отношение, например, к одним и тем же полномочиям. Начальники отделов или бригадиры, например, должны следовать определенным правилам и выполнять определенные обязанности, которые достаточно описать только один раз
	Местонахождение (Location)	Определяет физическое положение (область, город, предприятие, здание и т. д.) организационных единиц, рабочих мест, образцов аппаратных компонентов и технических ресурсов компании

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7B8A50C810143E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Модель организационной структуры предприятия может содержать следующие структурные элементы: Организационная единица, Должность или позиция, Группа и др. (таблица 2.1).

При построении модели организационной структуры между структурными элементами могут устанавливаться следующие типы связей: Имеет в подчинении (Is superior), Связан с (Is assigned to), Принадлежит (Belongs to), Является организационным управляющим (Is Organization Manager for), Имеет в своем составе (Has member), Состоит из (Is composed of), Располагает (Is located at), Взаимодействует с (Cooperates with), Содержит (Subsumes), Занимает (Occupies).

Тип связи устанавливается автоматически после соединения двух элементов модели и зависит от последовательности соединения двух элементов. Для того чтобы посмотреть, какой тип связи установился между элементами, щелкните двойным кликом по связи на диаграмме – отобразится окно свойств этой связи (рисунок 2.6), где в поле Type можно увидеть тип связи.

Пример организационной диаграммы приведен на рис. 2.7.

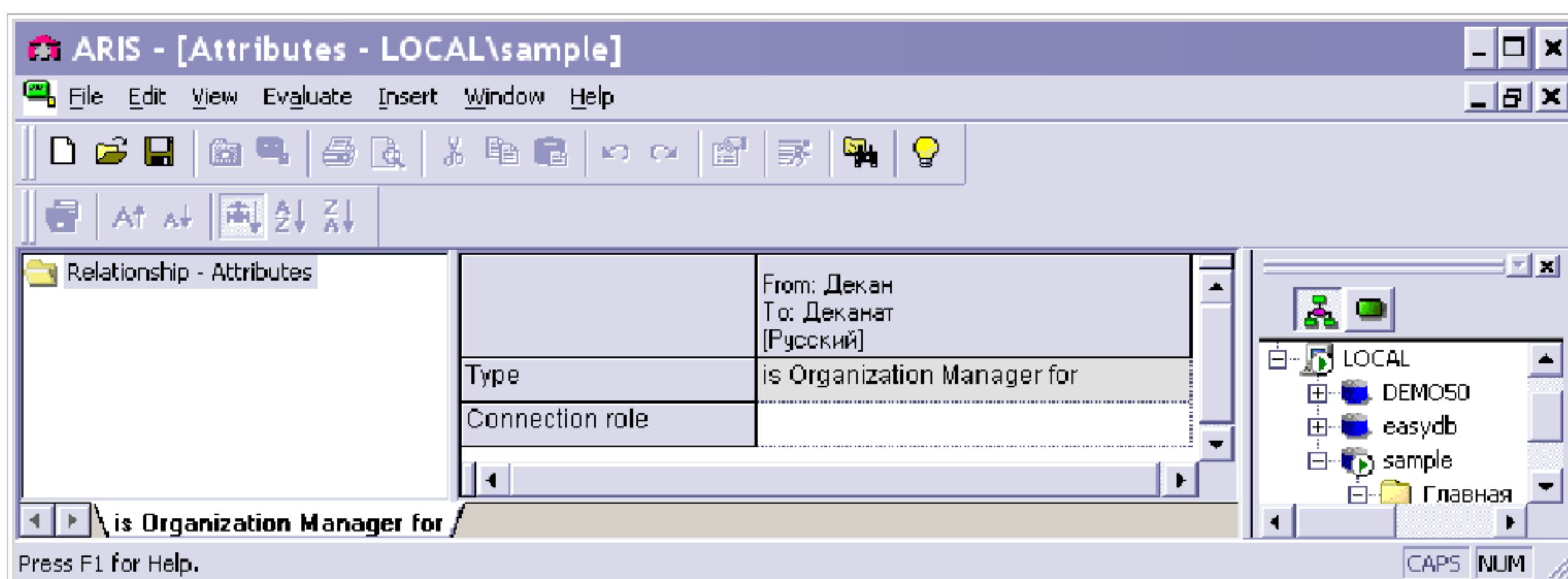


Рисунок 2.6 - Свойства связи между элементами

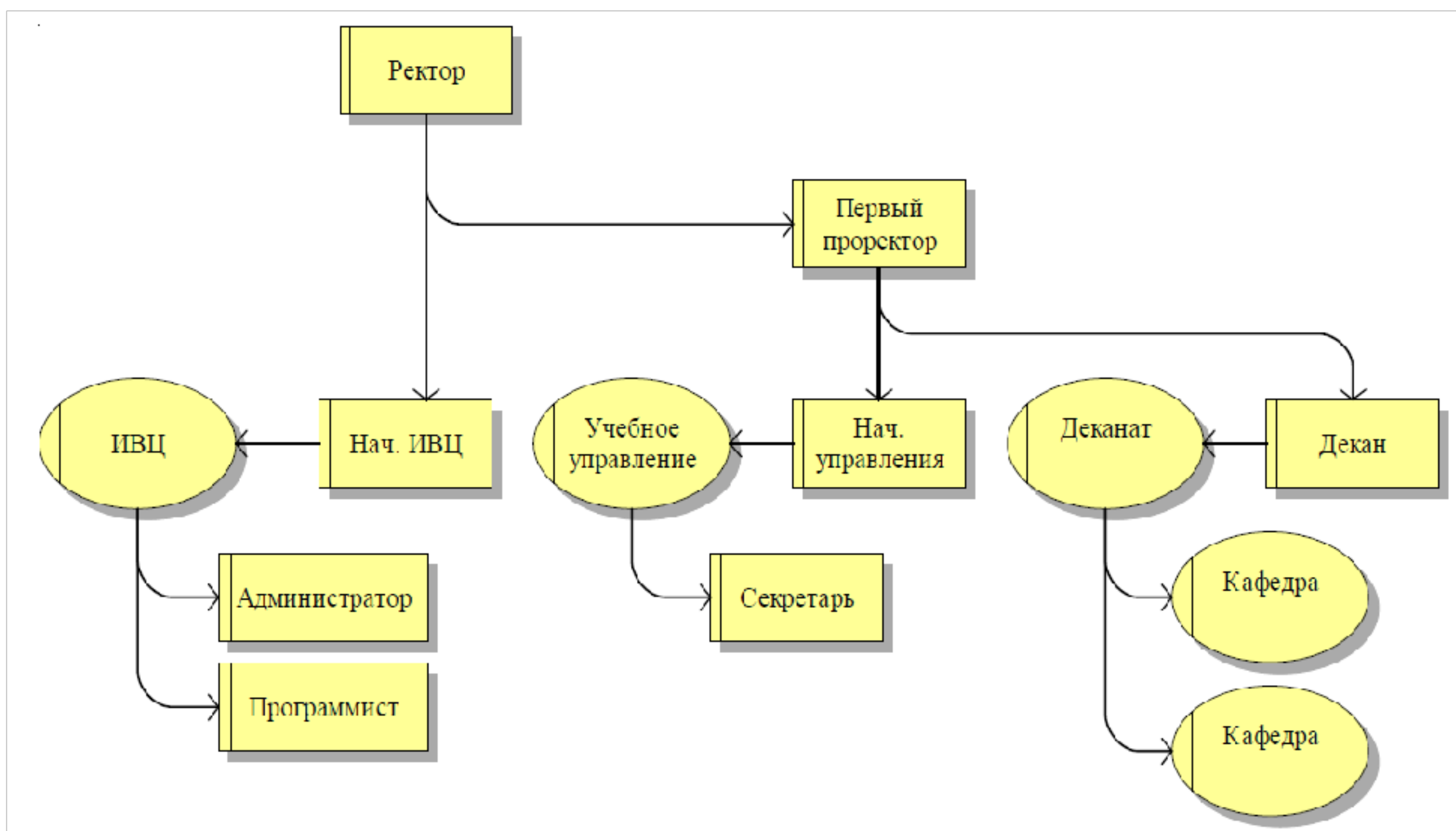


Рисунок 2.7 - Часть организационной диаграммы «Университет»

Построение диаграммы цепочек добавленного качества

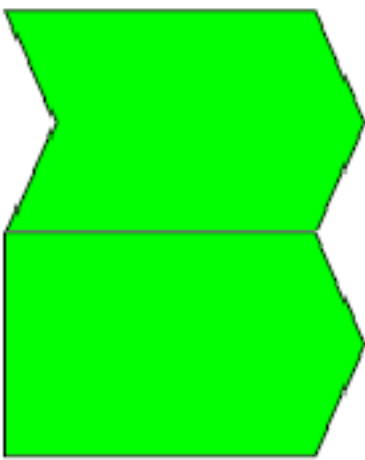
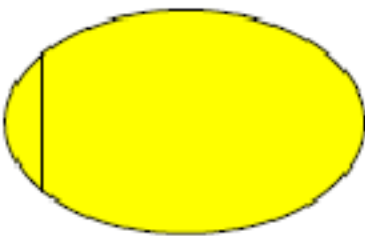
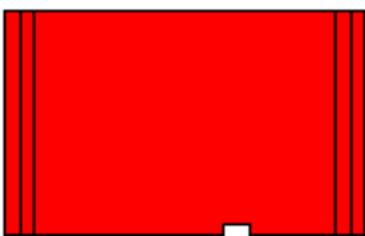
Для упрощения описания деятельности организации необходимо разработать иерархию моделей бизнес-процессов (БП) организации, начиная с самого верхнего уровня и заканчивая моделями отдельных БП на нижнем уровне. Для описания модели верхнего уровня используется диаграмма типа Value-added chain diagram (VAD), название которой можно перевести как «модель цепочки добавленного качества (стоимости)».

Диаграмма цепочек добавленного качества описывает функции организации, которые непосредственно влияют на реальный выход ее продукции. Эти функции создают последовательность действий, формируя добавленные значения: стоимость, количество, качество и т. д.

Для создания диаграммы VAD щелкните правой кнопкой мыши по главной папке и в появившемся контекстном меню выберите New→Model. Появится окно выбора модели (рисунок 2.5). Поставьте галочку в поле Processes, что соответствует процессному представлению.

Выберите диаграмму Value-added chain diagram и нажмите кнопку Далее. Введите имя новой модели, например «Основные процессы». Диаграмма цепочек добавленной стоимости может содержать следующие структурные элементы: Организационная единица, Должность или позиция, Функция и др. (таблица 2.2).

Таблица 2.2 - Элементы диаграммы цепочки добавленного качества

Графическая нотация	Наименование	Описание
	Функция	Действие или набор действий, выполняемых над исходным объектом (документом, материалом и т. п.) с целью получения заданного результата
	Технический термин	—
	Организационная единица (Organizational unit)	Элемент организационной структуры (структурное подразделение), который отвечает за выполнение определенных задач и преследует определенные цели
	Кластер (экземпляр)	Экземпляр объекта данных. Он представляет набор объектов как структуру данных

Пример диаграммы цепочек добавленной стоимости представлен на рисунке 2.8.

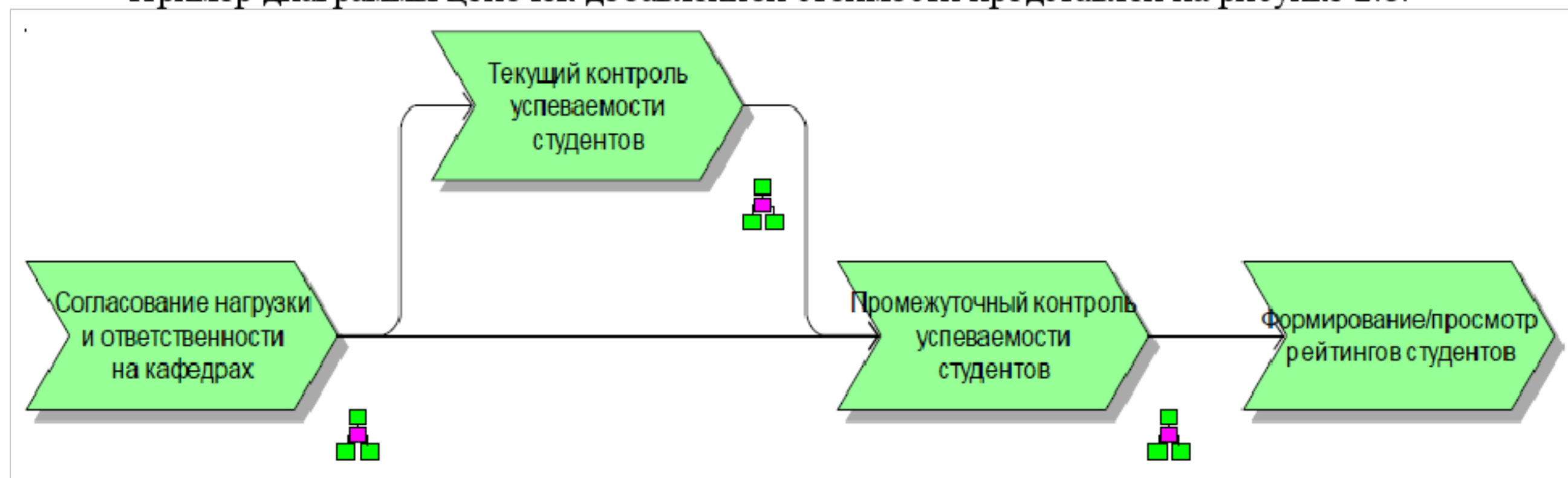


Рисунок 2.8 - Основные процессы учета успеваемости студентов в системе рейтингов

Пиктограммой  помечаются те функции, для которых существуют модели, раскрывающие эту функцию. Таким механизмом реализуется принцип декомпозиции.

Построение eEPC модели

Extended event driven process chain (eEPC) – цепочка процесса, управляемая событиями.

С помощью диаграмм eEPC процедуры БП представляются как логические последовательности событий. События определяют, какое состояние или отношение будет переключать функцию и какое состояние наступит после ее выполнения. Поэтому модель eEPC должна всегда иметь запускающие и завершающие события.

Документ подписан
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060008042E
Владелец: Шебухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Одно событие может инициировать выполнение одновременно нескольких функций, и, наоборот, функция может быть результатом наступления нескольких событий. Объединения нескольких событий или функций отображаются на диаграмме eEPC с помощью соединителей в виде небольшого кружка. Эти соединители не только отображают графические связи между объектами модели, но и определяют логические связи между ними. Различают два типа связи логических операторов – связи событий и связи функций.

Для создания диаграммы eEPC щелкните правой кнопкой мыши по главной папке и в появившемся контекстном меню выберите New→Model. Появится окно выбора модели (рисунок 2.5). Поставьте галочку в поле Processes, что соответствует процессному представлению.

Выберите диаграмму eEPC и нажмите кнопку Далее. Введите имя новой модели.

Таблица 2.3 - Элементы диаграммы цепочек, управляемых событиями

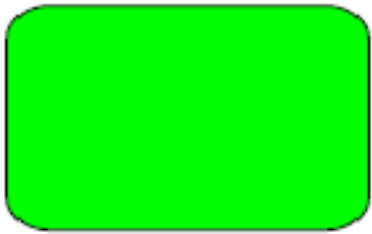
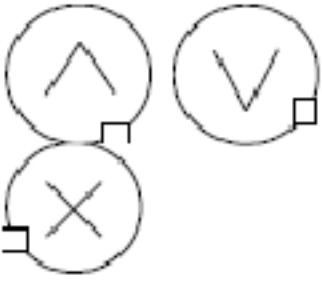
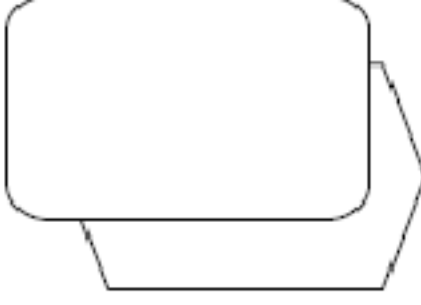
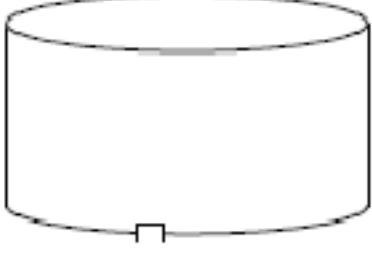
Графическая нотация	Наименование	Описание
	Функция	Действие или набор действий, выполняемых над исходным объектом (документом, материалом и т. п.) с целью получения заданного результата
	Событие	Состояние, которое является существенным для управления БП и оказывает влияние или контролирует дальнейшее развитие одного или более БП. Изменения состояния отражаются с помощью информационных объектов
	Правило	Представляет собой правило разветвления и слияния веток БП. Если перейти к рассмотрению каждой функции БП, то можно сказать, что правило отражает логическое соотношение между несколькими исходными для функции событиями и несколькими результирующими
	Интерфейс функции	Используется для обозначения функции внешнего БП, который либо не описывается в данной модели, либо является БП другой предметной области
	Носитель информации	Представляет собой средство хранения информации. Оно может быть реализовано, к примеру, в виде картотеки или компьютерных файлов или БД
	Документ	Обозначает любой вид документов

Диаграмма цепочек, управляемых событиями, кроме элементов организационной диаграммы и диаграммы данных может содержать следующие структурные элементы: Функция, Событие и др. (таблица 2.3).

При построении диаграмм eEPC нужно соблюдать следующие правила:

- 1) каждая функция инициируется событием (или несколькими событиями) и завершается событием (или несколькими событиями);
- 2) каждая функция должна иметь входную и выходную информацию;
- 3) у каждой функции должно быть ответственное лицо за ее выполнение.

На рисунке 2.9 приведен пример блока, на базе которого строится вся модель eEPC.

Пример диаграмм цепочек, управляемых событиями, представлен на рисунках 2.10 - 2.12.

Эти диаграммы раскрывают (детально описывают) процессы, изображенные на рисунке 2.8.

Сертификат: 2C0000043E94B8B952205E4BA50006000043E
Владелец: Шебунин Сергей Александрович

Действителен: с 19.08.2022 по 19.08.2023

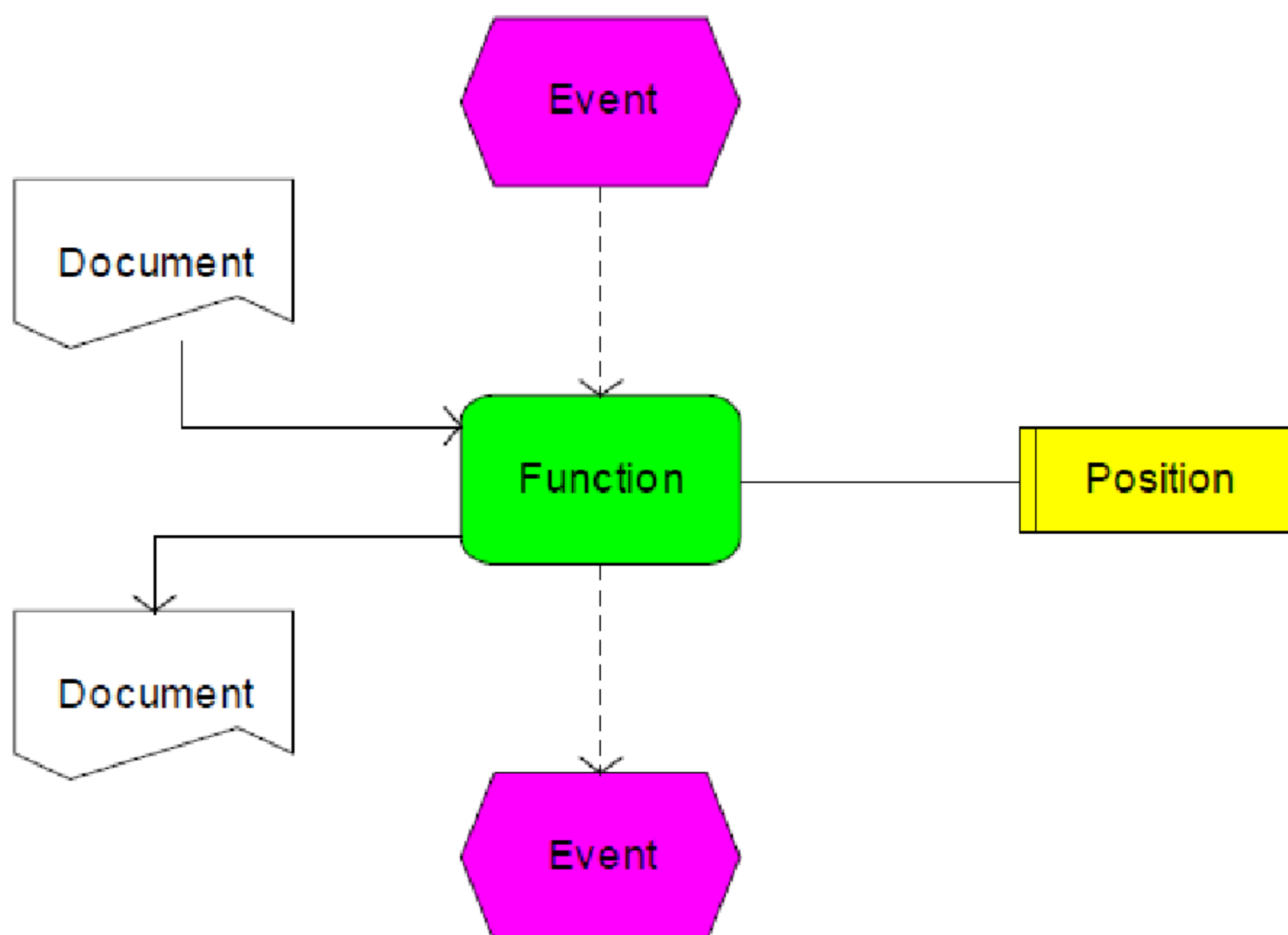


Рисунок 2.9. Блок, на базе которого строится модель eEPC

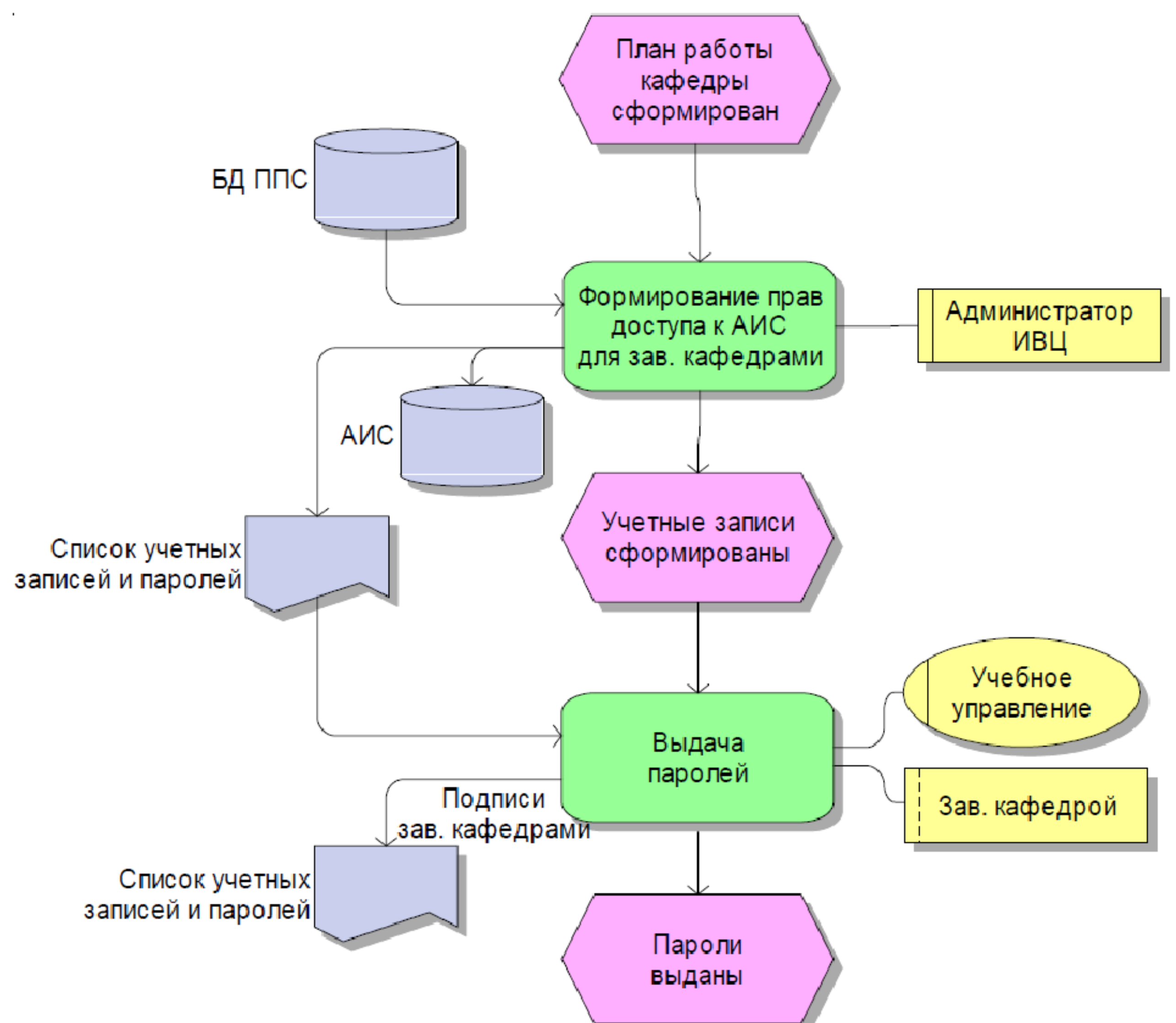
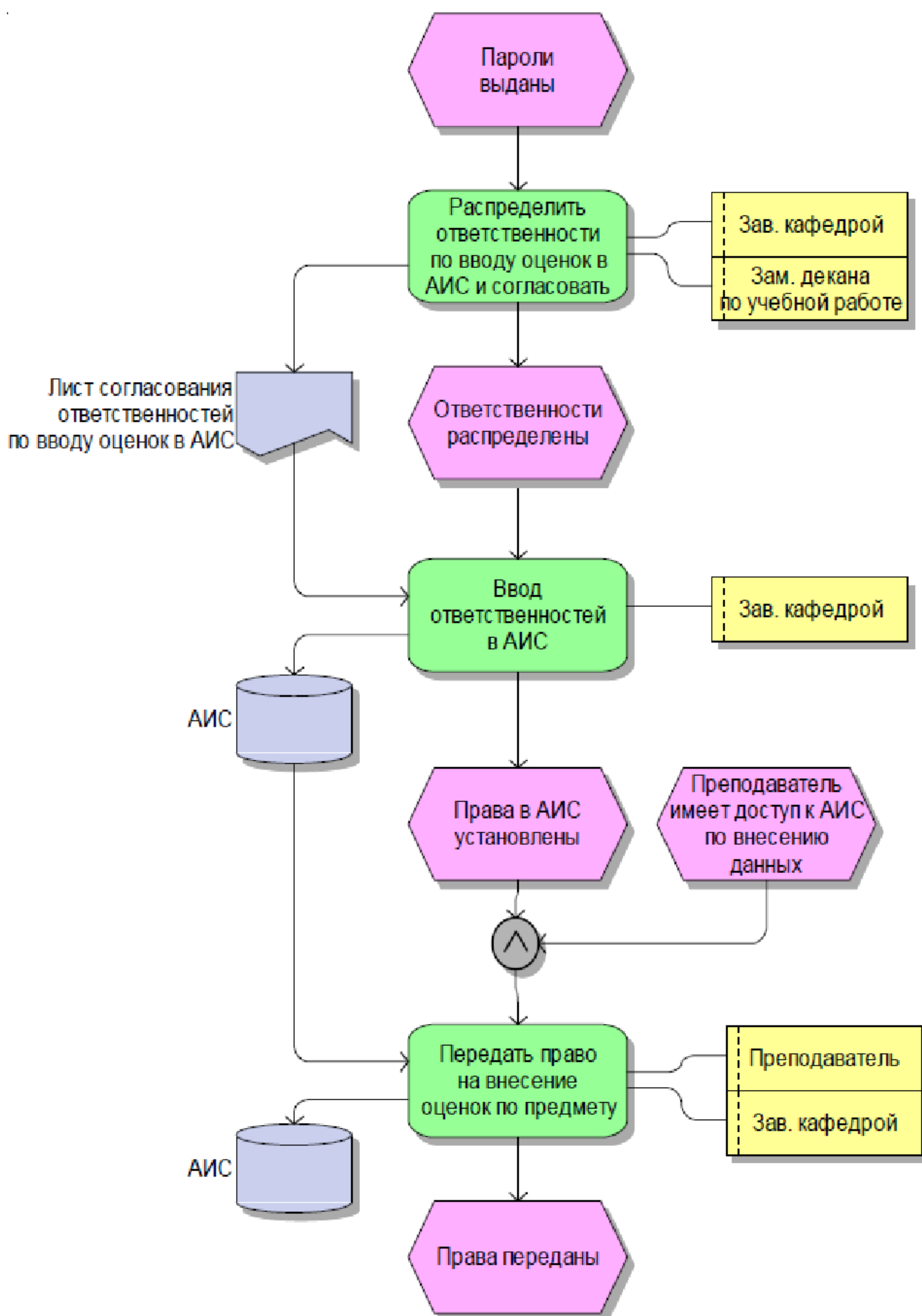


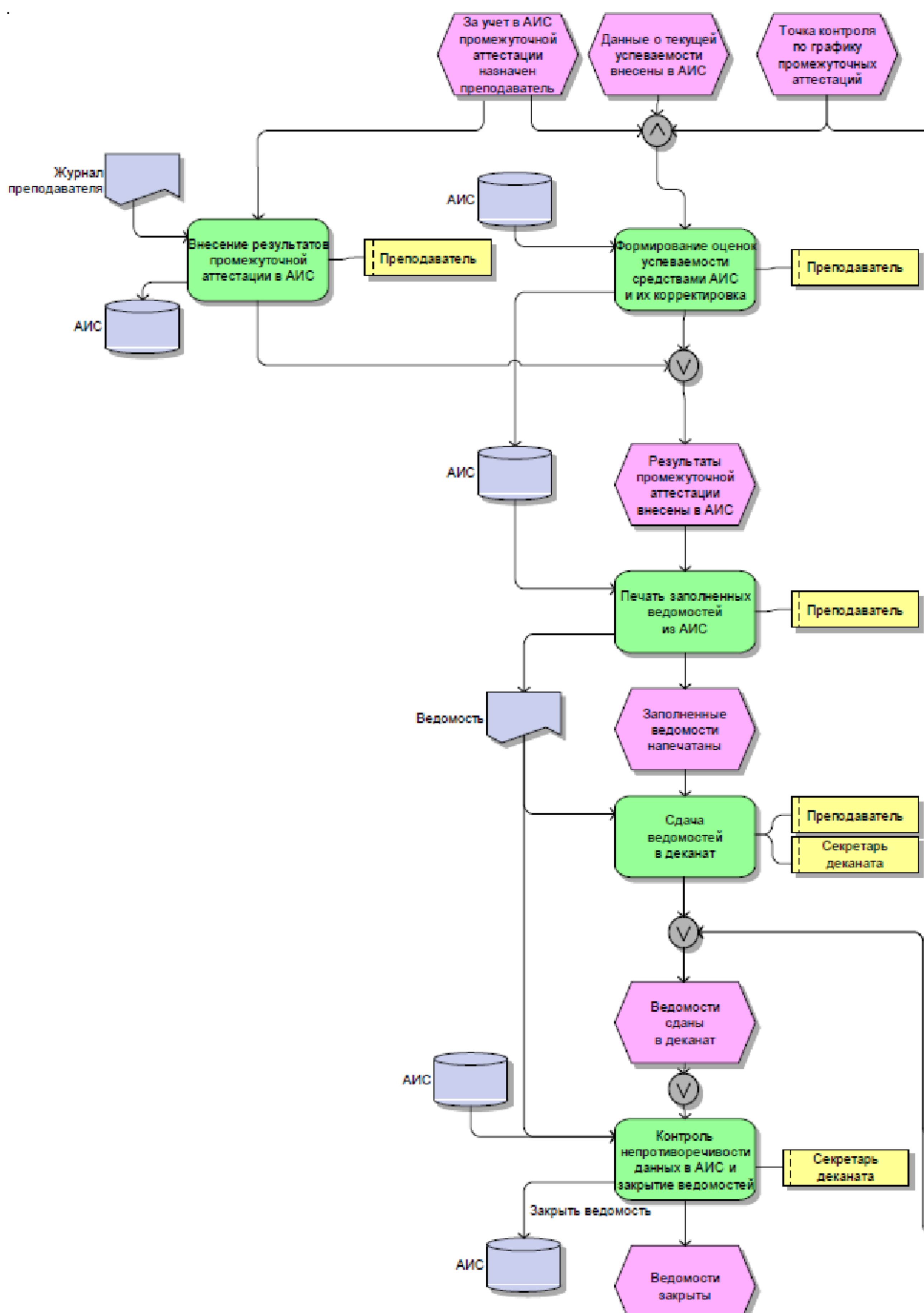
Рисунок 2.10 - Согласование нагрузки и ответственности на кафедрах



Продолжение рисунка 2.10 – Согласование нагрузки и ответственности на кафедрах

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

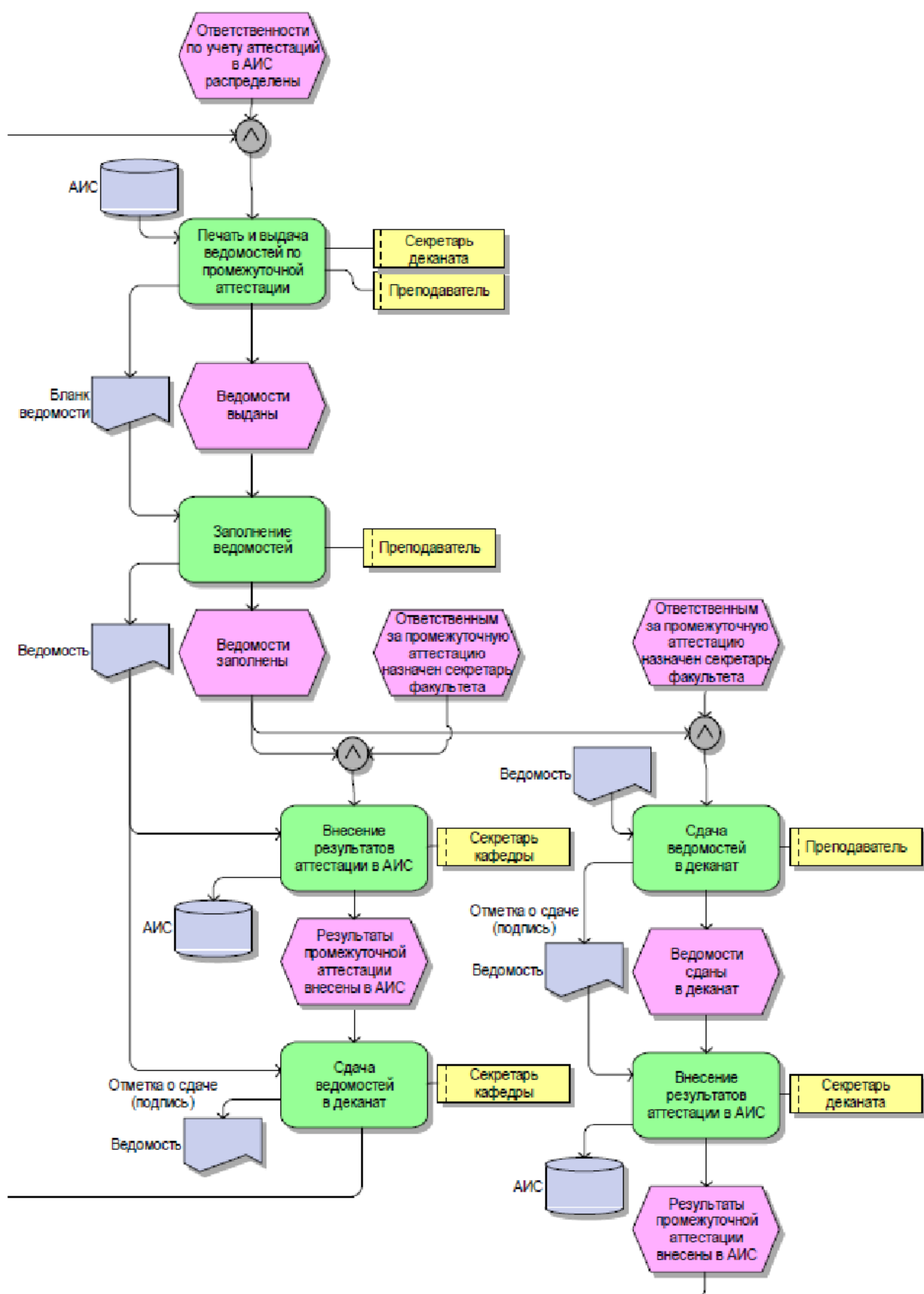


ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Рисунок 2.11. Промежуточный контроль успеваемости студентов

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023



Продолжение рисунка 2.11 - Промежуточный контроль успеваемости студентов

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

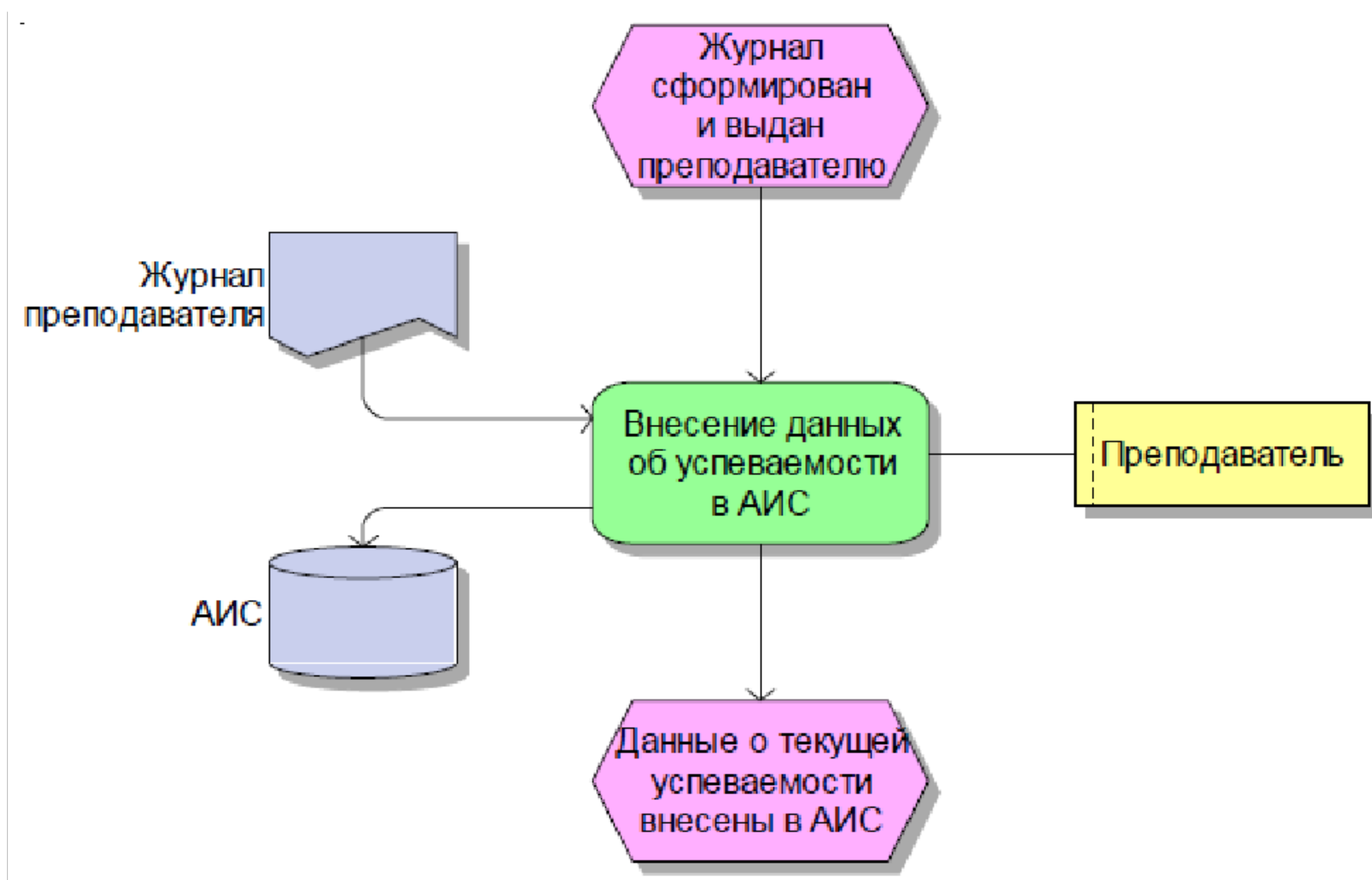


Рисунок 2.12. Текущий контроль успеваемости студентов

Описание объектов автоматизации

Объект автоматизации необходимо описывать с целью структурирования полученных данных об этом объекте и представления их в письменном и/или модельном виде. Представленная таким образом информация об объекте является незаменимым помощником при выполнении последующих работ в процессе разработки ПО, таких как проектирование подсистем и модулей, проектирование базы данных, разработка пользовательского интерфейса, формулирование функций и требований к ПО. Кроме того, описание объекта автоматизации требует ГОСТ при разработке ТЗ.

Как правило, для объекта строится его модель (например, по методологии ARIS), делаются пояснения к модели и описываются нюансы, которые сложно или невозможно отобразить на модели.

Задания и порядок выполнения работы

1. Установить и настроить среду моделирования ARISToolset.
2. Разработать организационную диаграмму.
3. Разработать диаграмму добавленного качества (VAD).
4. Разработать для каждого процесса VAD диаграмму eEPC.

Контрольные вопросы

1. Что такое модель?
2. Методология ARIS.
3. Организация хранения данных о модели в ARISToolset.
4. Как взаимосвязаны разные диаграммы в рамках одной модели?
5. Какие вы знаете связи между элементами в организационной диаграмме?
6. Какие процессы отражены на диаграмме VAD?
7. Как реализован механизм декомпозиции в ARIS?
8. Из каких блоков строится диаграмма eEPC?
9. Что такое событие в eEPC?
10. Каким образом может использоваться модель ARIS для описания объекта автоматизации?

Работа с литературой.

Рекомендуемые источники информации
(№ источника)

Основная	Дополнительная	Методическая	Интернет-ресурсы
1-3	1-3	1-3	1-2

Оценочные средства: контрольные вопросы

Тема 3. Анализ требований и их формализация.

Лабораторная работа № 3. Разработка модели вариантов использования и их спецификаций

Форма проведения: Практическое выполнение задания

Цель работы – сформировать навыки:

- разработки модели вариантов использования;
- организации модели вариантов использования;
- разработки спецификации вариантов использования.

Краткие теоретические сведения

Цель данного этапа заключается в разработке требований к ПО. В России стандартом оформления требований является ГОСТ 34.602–89, который регламентирует содержание ТЗ на разработку автоматизированных информационных систем.

Существуют и другие широко применяемые в мире стандарты, например IEEE (Institute of Electrical and Electronics Engineers) 830: Standard for Software Requirements Specification (1994).

Процесс разработки требований к ПО (или спецификаций требований) условно можно разделить на три этапа, которые позволяют описать требования на трех уровнях (рисунок 3.1), причем настолько детально, насколько этого требует уровень абстракции.



Рисунок 3.1 - Характеристика уровней описания требований

Выявление и формулирование требований первого уровня «Потребности» было осуществлено на предыдущем этапе – этапе системного анализа. На настоящем этапе нужно описать требования для второго и третьего уровней.

Модель вариантов использования

Для определения (выявления) функций разрабатываемой системы необходимо провести анализ требований. На данном этапе для такого анализа имеется уже достаточно много материала, который был собран на предыдущем этапе. Таким образом, исходными данными для анализа требований являются потребности заинтересованных лиц, данные интервьюирования, описание объектов автоматизации, цели и задачи разработки системы.

Процесс формулирования функций системы можно частично отнести к процессу проектирования, поскольку он определяет функциональность будущей системы, что, в свою

очередь, уже на этапе анализа требований заставляет задумываться об архитектуре системы и об элементах пользовательского интерфейса. И этот подход верен: он позволяет безболезненно перейти от этапа к этапу и поддерживает итеративную модель разработки программных систем.

При определении функций можно использовать элементы системного анализа. Но существуют методы более адаптированные к процессам разработки ПО. В данной лабораторной работе рекомендуется использовать метод вариантов использования для идентификации и представления функций системы.

Метод вариантов использования (прецедентов) является частью методологии объектно-ориентированного проектирования. Это метод анализа и проектирования сложных систем, представляющий собой способ описания поведения системы с точки зрения того, как различные пользователи взаимодействуют с ней для достижения своих целей. Такой ориентированный на пользователя подход предоставляет возможность исследовать различные варианты поведения системы при раннем привлечении пользователя.

Варианты использования можно успешно применять на протяжении всего жизненного цикла ПО: при анализе, проектировании и в процессе тестирования. В этом случае процесс разработки ПО называют «основанный на вариантах использования».

Вариант использования – это функциональный связный блок, выраженный в виде транзакции между актантом и системой (рисунок 3.2). Вариант использования описывает последовательность действий, выполняемых системой с целью получения полезного результата пользователем системы.

Актантом называют пользователя системы (человек, другая система). Между вариантами использования и актантами существует связь, которая показывает, какие функции системы доступны каждому пользователю.

Кроме того, в модели вариантов использования есть связи между самими вариантами использования. В соответствии со стандартом языка UML таких связей достаточно много.

В данной работе будут рассмотрены наиболее часто употребляемые связи при построении модели – это расширение (обобщение) и использование (рисунок 3.2).

Расширение применяется в случае, если у ряда вариантов использования есть общая часть, которую можно выделить в отдельный вариант использования. Этот тип связи аналогичен наследованию в классах. Пример такого типа связи приведен на рисунке 3.3.

Использование применяется в случае, если несколько вариантов использования в основном или альтернативных потоках имеют одинаковое поведение. В отличие от обобщения использование не подразумевает обязательное наличие общего поведения.

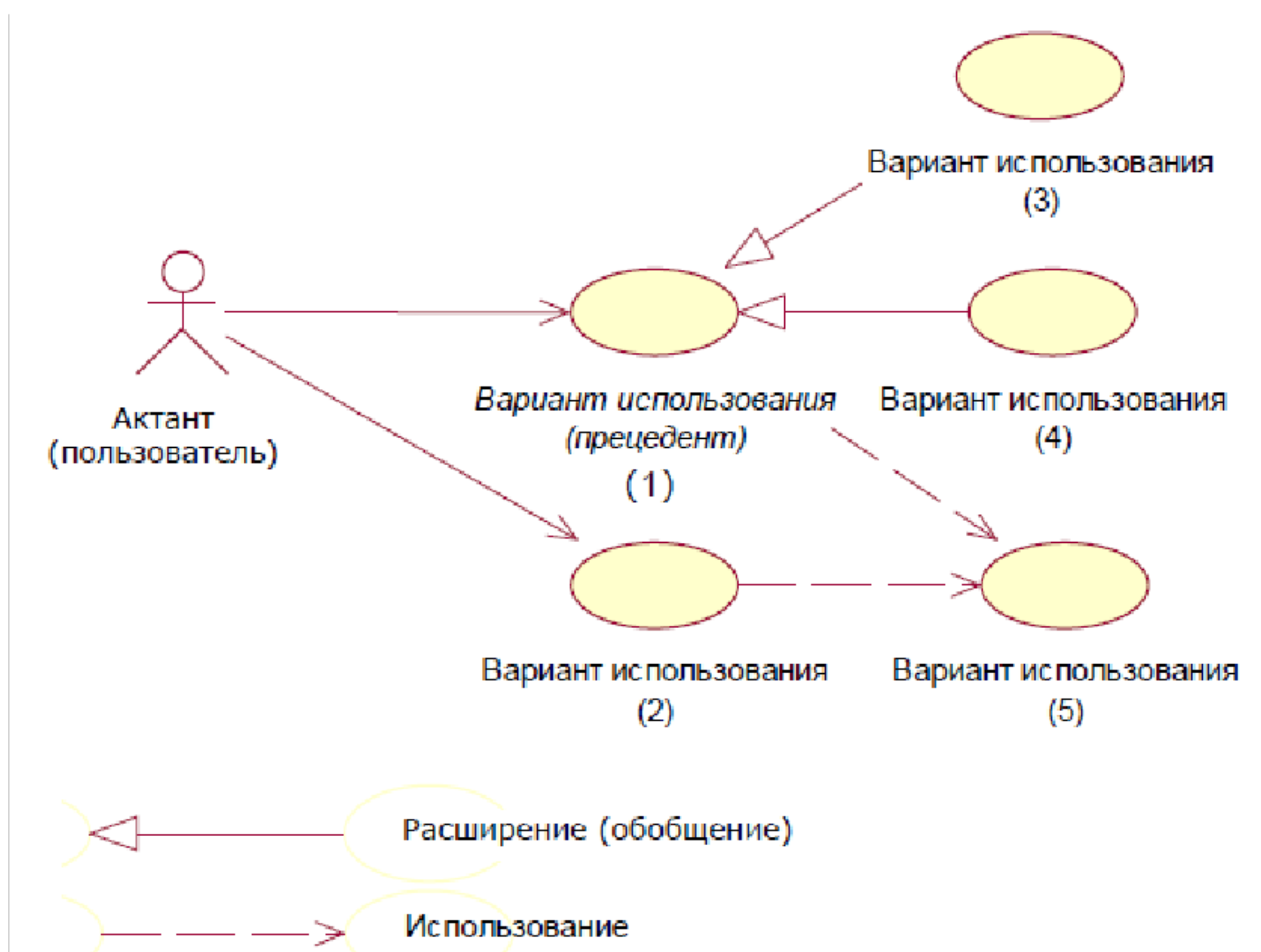


Рисунок 3.2 - Обозначения в модели вариантов использования

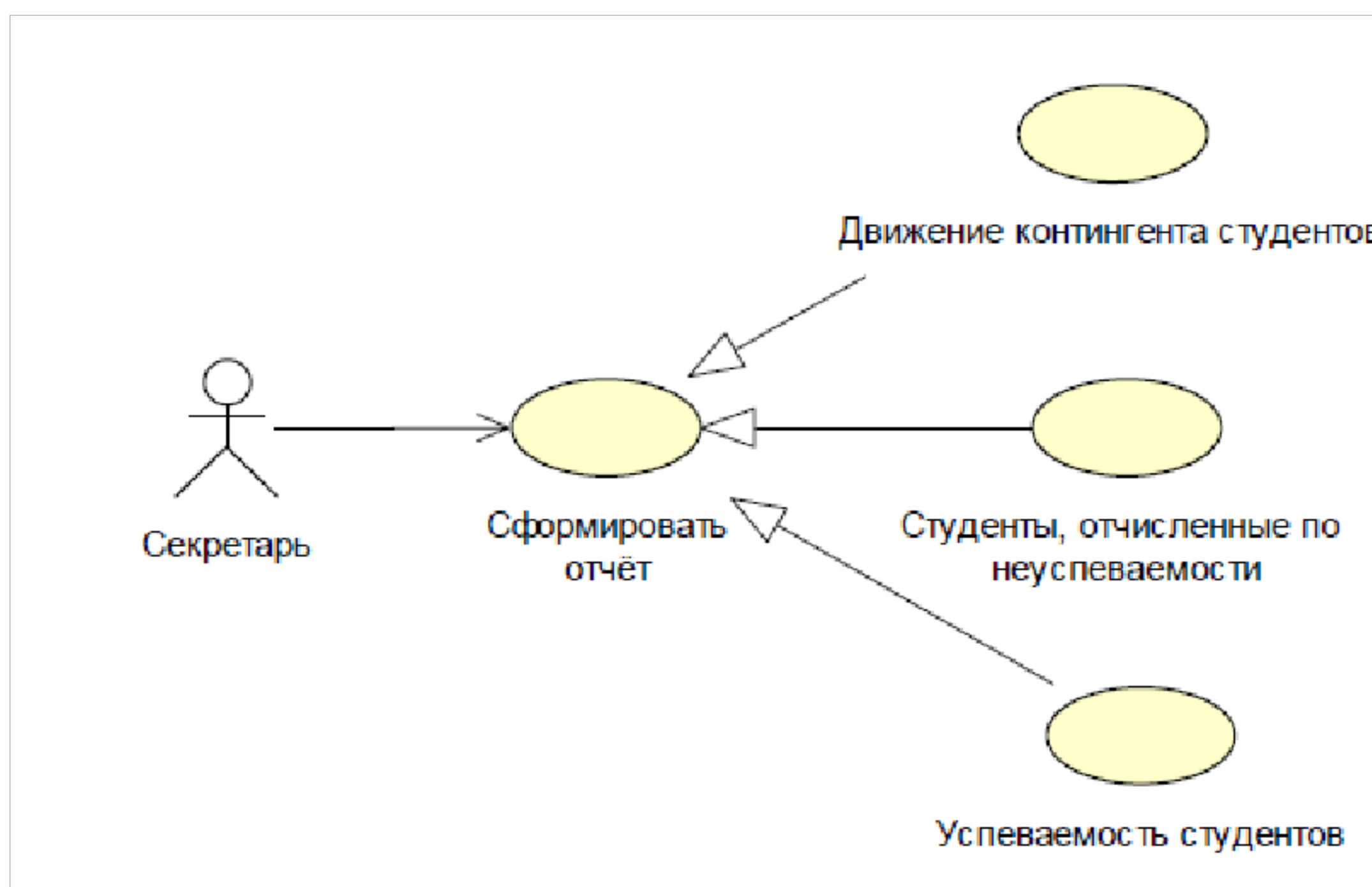


Рисунок 3.3 - Пример связи обобщения

Построение модели вариантов использования.

Модель вариантов использования системы состоит из всех актантов системы и различных вариантов использования, посредством которых актанты взаимодействуют с системой, тем самым описывая многообразие ее функционального поведения. Она также показывает связи между вариантами использования, что углубляет наше понимание системы.

Первый шаг моделирования вариантов использования состоит в создании списка актантов системы. Второй шаг описывает системное поведение, т. е. то, как пользователи взаимодействуют с системой, осуществляя некоторые последовательности действий, для достижения определенных

целей. Если модель вариантов использования получается слишком громоздкой и нечитабельной, то ее необходимо декомпозировать, т. е. разбить на подсистемы.

Пример модели вариантов использования приведен на рисунках 3.4 - 3.8.

Следующий шаг состоит в уточнении деталей функционального поведения каждого варианта использования, т. е. в разработке спецификаций. Спецификации вариантов использования состоят из текстовых и/или графических описаний каждого варианта использования, написанных с позиции пользователя.

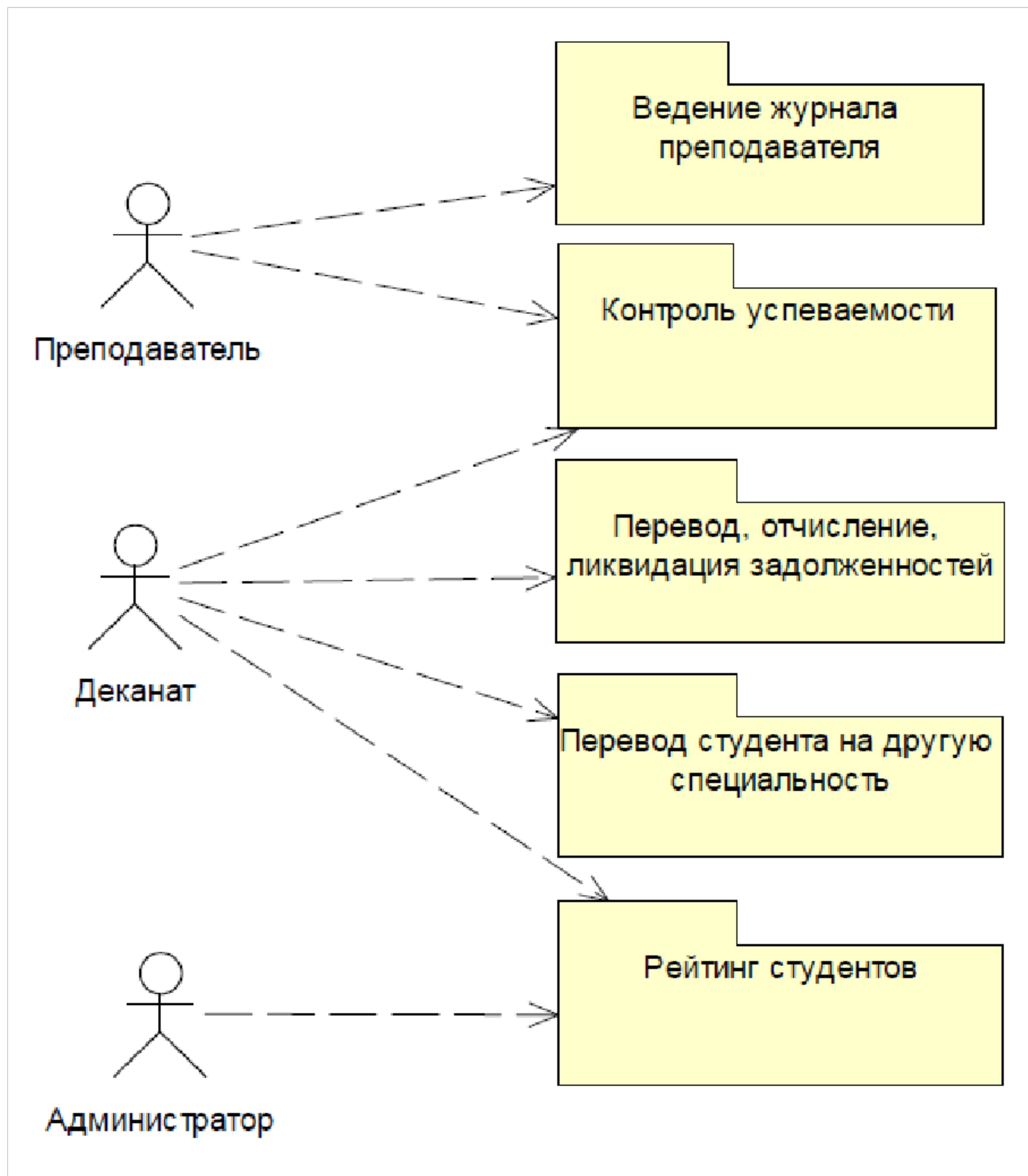


Рисунок 3.4 - Подсистемы в модели вариантов использования

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

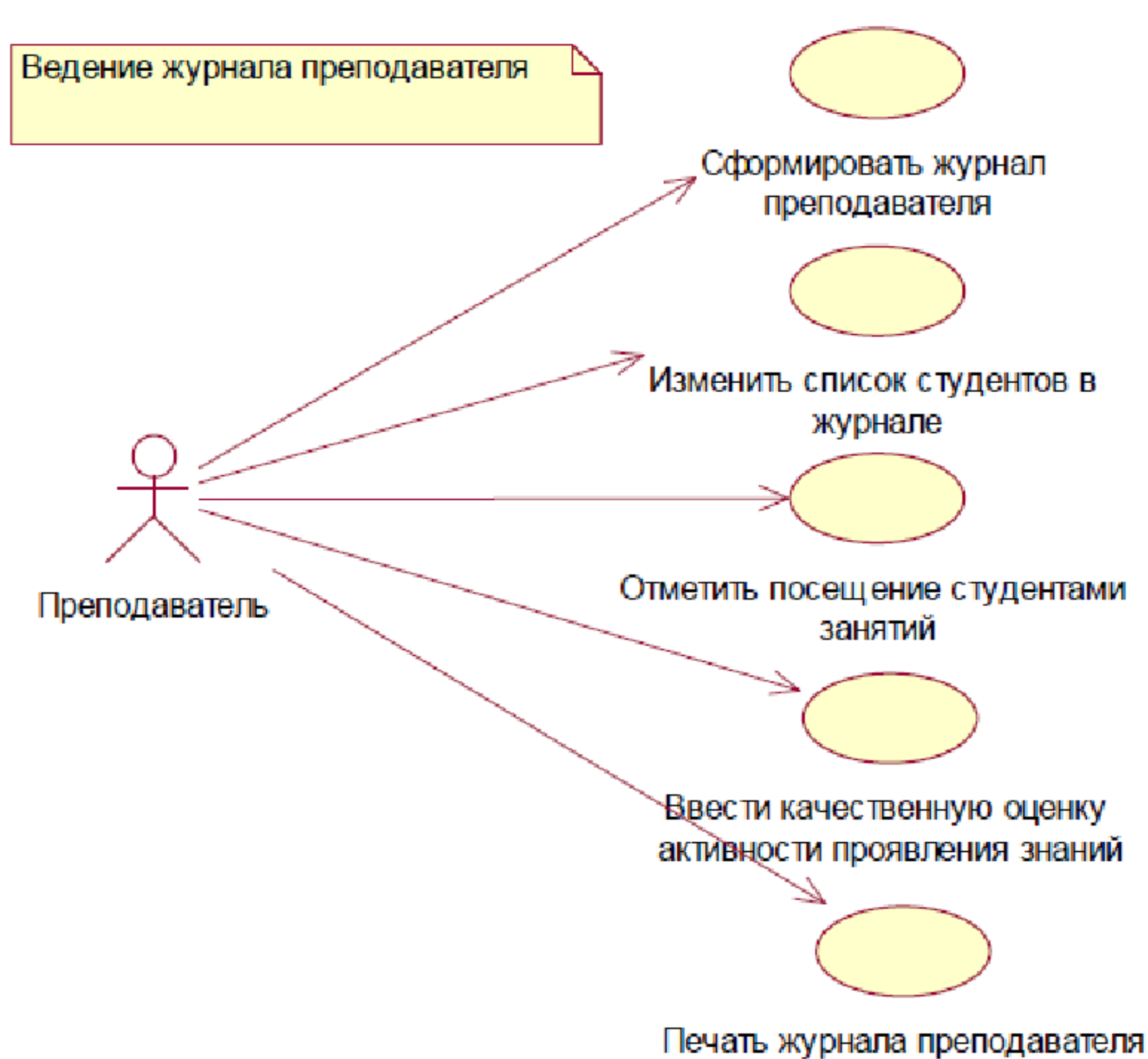


Рисунок 3.5 - Диаграмма вариантов использования «Ведение журнала преподавателя»

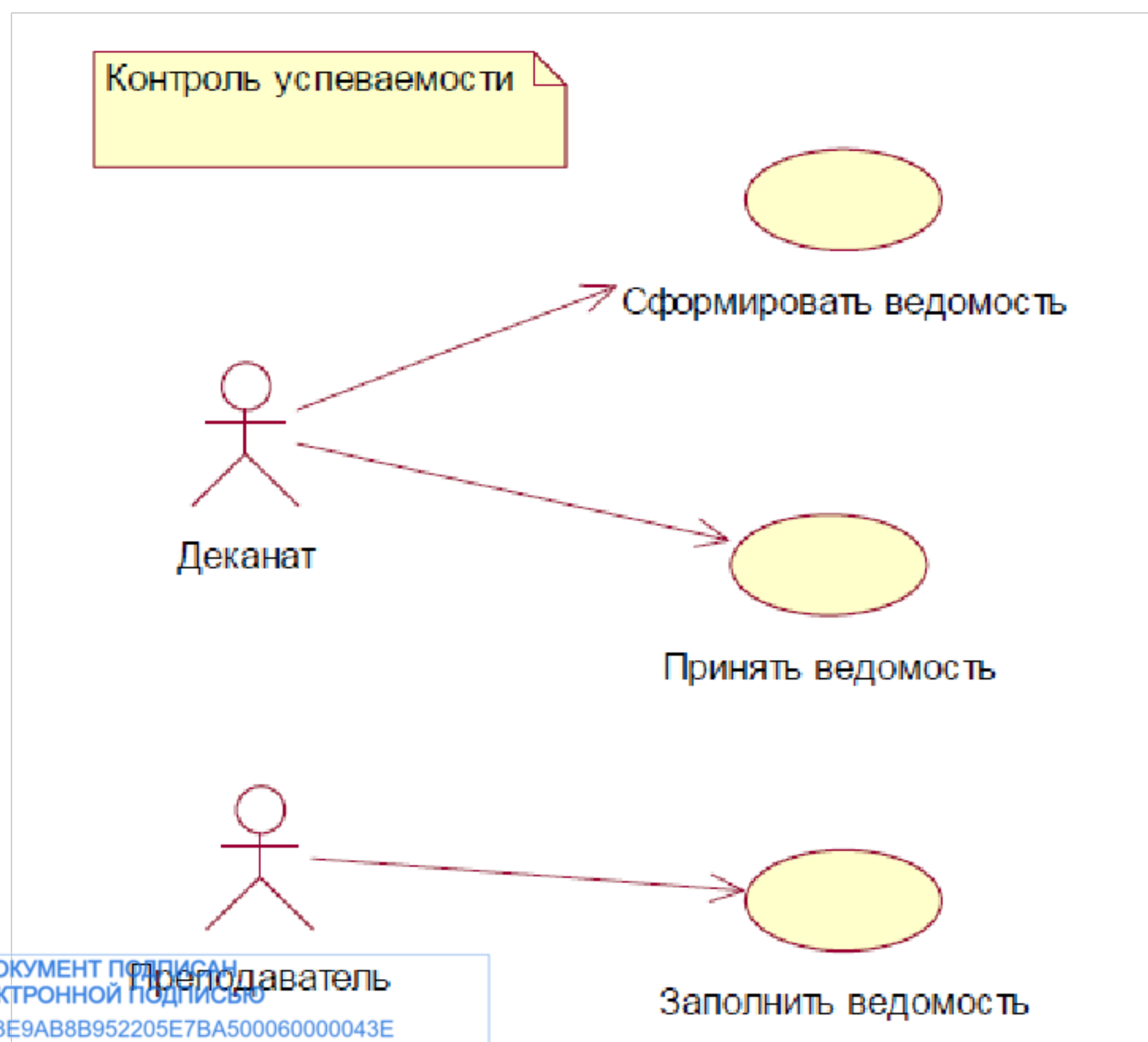


Рисунок 3.6 - Диаграмма вариантов использования «Контроль успеваемости»

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Перевод, отчисление, ликвидация задолженностей

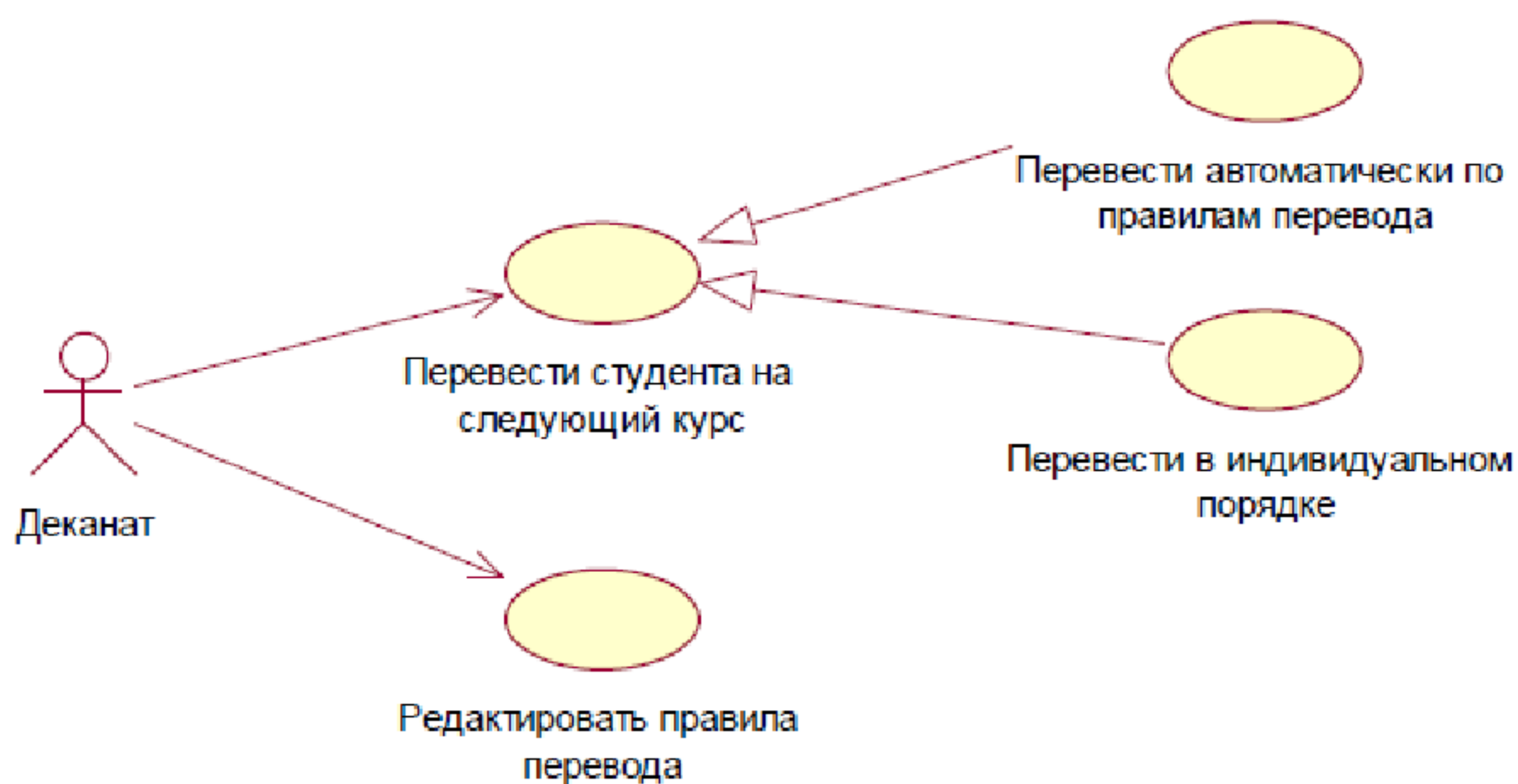


Рисунок 3.7 - Диаграмма вариантов использования «Перевод, отчисление, ликвидация задолженностей»

Рейтинги студентов

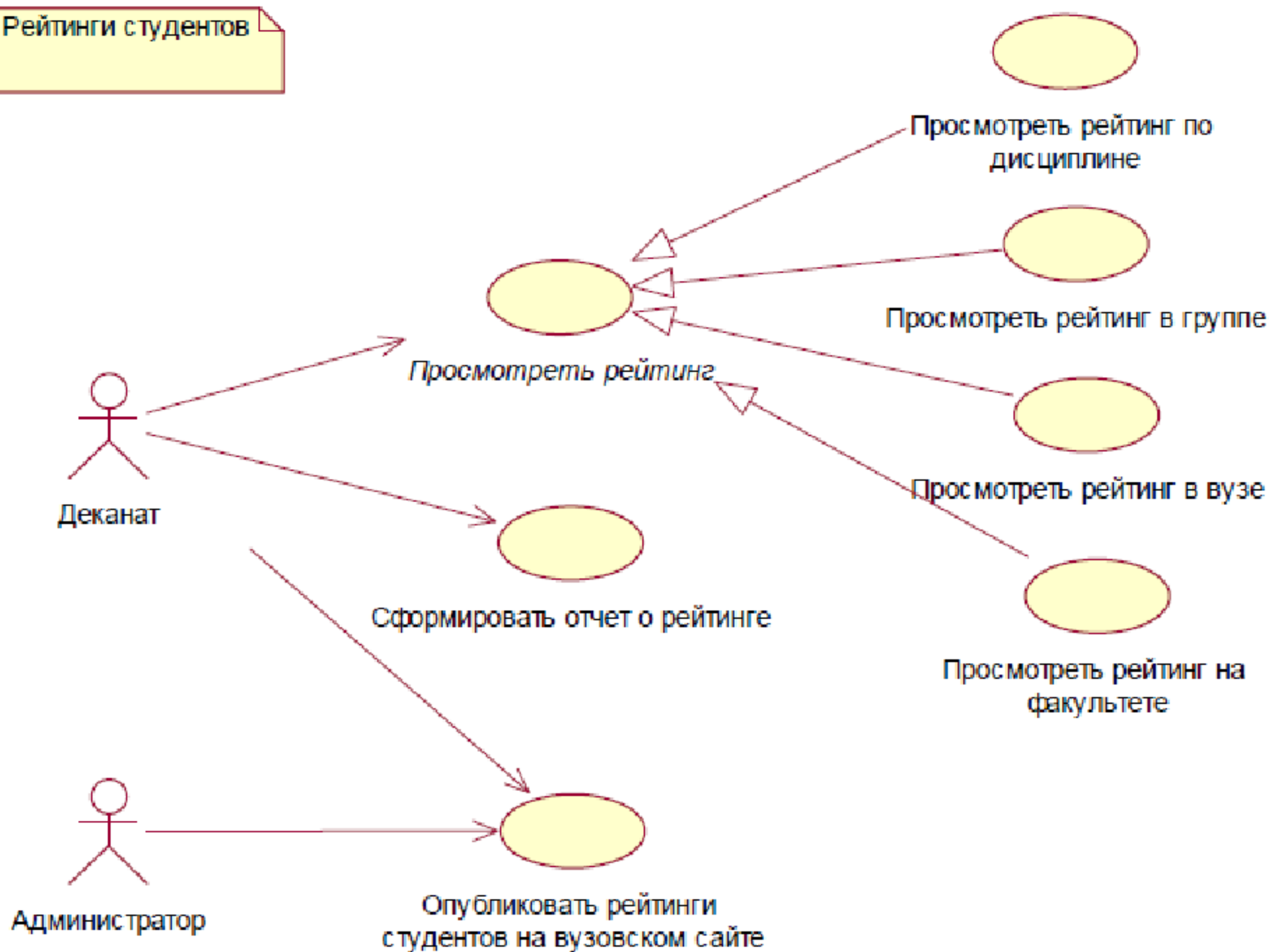


Рисунок 3.8 - Диаграмма вариантов использования «Рейтинги студентов»

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Спецификация вариантов использования

Действителен: с 19.08.2022 по 19.08.2023

Как было отмечено выше, спецификация служит для описания последовательности действий пользователя и системы в рамках одного варианта использования. Цель разработки спецификаций заключается в описании требований к ПО, которые позволят заказчику и разработчику системы однозначно (недвусмысленно) понимать требования. Спецификации вариантов использования позволяют проектировщикам и программистам без дополнительных уточнений и домысливания разрабатывать систему.

Спецификации вариантов использования можно оформлять, придерживаясь разных стандартов, начиная с международных и заканчивая корпоративными. Спецификация должна включать следующие разделы:

- название варианта использования;
- краткое описание варианта использования;
- его роль и цель (для описания достаточно одного абзаца);
- основной поток;
- альтернативные потоки;
- специальные требования.

Основной поток. Вариант использования всегда инициируется неким актантом, когда последний производит некое действие. Вариант использования должен описывать, что делает актант и что система делает в ответ; он должен выглядеть как диалог между актантом и системой.

Вариант использования должен описывать, что происходит внутри системы, а не как или почему это происходит. Если происходит обмен информацией, то нужно указать, какая информация поступает, а какая – отправляется.

Кроме того, не очень понятно, что имеется в виду, если сказать, что актант вводит информацию о клиенте; лучше сказать, что актант вводит имя и адрес клиента. Для того чтобы не делать вариант использования очень сложным и не запутаться в деталях, полезно использовать глоссарий, где можно определить, что понимается под информацией о клиенте.

Простые альтернативы можно описать в тексте варианта использования. Если для описания того, что происходит в альтернативном случае, достаточно нескольких предложений, следует сделать это непосредственно в разделе, посвященном основному потоку событий. Если альтернативные потоки более сложные, нужно использовать отдельный раздел «Альтернативные потоки».

Иногда рисунок информативнее многих слов. Если это поможет добиться большей ясности, можно включать в вариант использования графические описания интерфейсов пользователя, потоков процессов или другие рисунки. Если для представления сложного процесса принятия решения необходимо использовать формальные средства спецификации, такие как диаграммы деятельности, несомненно, это следует делать. Аналогично, если поведение системы зависит от состояния, то диаграмма перехода состояний лучше прояснит его, чем это сделают страницы текста. Используйте представление, которое лучше всего подходит для отображения вашей проблемы, но будьте осторожны с терминологией, обозначениями или рисунками, которые могут быть непонятны вашей аудитории. Помните, что ваша задача прояснить, а не запутать.

Альтернативные потоки

Первый альтернативный поток. Более сложные альтернативы нужно описывать в отдельном разделе. Следует воспринимать альтернативные потоки как варианты альтернативного поведения: каждый альтернативный поток представляет некое альтернативное поведение (вариантов много из-за исключительных ситуаций, возникающих в основном потоке). Они могут быть произвольной длины, которая требуется для описания связанных с альтернативным поведением событий. Когда альтернативный поток заканчивается, события основного потока продолжают, если не оговорено противное.

Альтернативные потоки могут, в свою очередь, также состоять из подразделов.

Второй альтернативный поток. Достаточно часто в варианте использования встречается несколько альтернативных потоков. Для большей ясности следует описывать каждую альтернативу отдельно. Использование альтернативных потоков упрощает понимание варианта использования, а также предотвращает декомпозицию вариантов использования на их иерархии.

Следует помнить, что варианты использования – это только текстовые описания, и их главная задача в том, чтобы документировать поведение системы ясно, кратко и понятно.

Специальные требования. Здесь обычно приводятся имеющие отношение к данному варианту использования нефункциональные требования, которые сложно описать в тексте потока событий варианта использования.

Электронной подписью
Сертификат: 2C0000043E9AB8B952205E7BA5000660000043E
Владелец: Шебухова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Примерами таких требований могут служить положения законодательства и инструкций, применяемые стандарты, атрибуты качества создаваемой системы, в том числе требования практичности, надежности, производительности или возможности сопровождения. В данном разделе следует также фиксировать другие требования, такие как описание операционных систем и сред, требования совместимости и ограничения проектирования.

Специальные требования включают *предварительные условия варианта использования* (состояние системы, в котором она должна находиться перед началом выполнения варианта использования) и *постусловия варианта использования* (перечень возможных состояний системы непосредственно после завершения варианта использования).

Пример спецификации варианта использования.

Пример приводится для варианта использования «Сформировать отчет о рейтинге».

Краткое описание.

Отчет показывает рейтинги студентов, дисциплин, специальностей, кафедр и факультетов по семестрам, годам и за все время обучения. Перед формированием отчета может быть установлен фильтр для уменьшения полезной (необходимой в данный момент) информации.

Предварительные условия.

Все данные о студентах и результатах аттестаций должны быть внесены в базу данных.

Поток событий. Основной поток.

Пользователь выбирает пункт меню – кнопку, ссылку на усмотрение проектировщика интерфейса пользователя. Система запускает форму выбора параметров отчета «Рейтинги студентов». На этой форме должны быть доступны для выбора следующие параметры:

1. *Дата формирования отчета.* Отчет может быть сформирован за любую дату, но не позднее текущей. Соответственно данные, используемые для формирования отчета, должны соответствовать указанной дате. Это означает, что при выполнении запроса к БД необходимо учитывать даты создания обрабатываемых документов, которые не должны превышать значения, указанного в данном параметре.

2. *Факультет.* Пользователь должен иметь возможность выбрать любой факультет или все факультеты.

3. *Кафедра.* Пользователь должен иметь возможность выбрать любую кафедру или все кафедры. Если в предыдущем пункте был выбран конкретный факультет, то в список кафедр должны попасть только те, которые прикреплены к выбранному факультету.

4. *Направление, специальность, специализация.* Правило выбора аналогично п. 3.

5. *Учебная группа.* Правило выбора аналогично п. 3.

6. *Студент.* Правило выбора аналогично п. 3.

7. *Результаты всех сессий* на дату, указанную в п. 1.

8. *Учебный год.* Список параметров должен формироваться на основе выбранных в п. 1–6 параметров.

9. *Результаты сессии.* Выбирается номер семестра. Если выбран параметр п. 8, то в списке семестров должны находиться только те, которые попадают в выбранный учебный год.

10. *Результаты аттестаций.* В список аттестаций должны попасть только те, которые соответствуют параметрам, выбранным в п. 7, 8.

После выбора на форме соответствующих параметров пользователь нажимает кнопку Сформировать отчет. Система формирует отчет в формате MS Excel в соответствии с формой, приведенной в таблице 3.1.

Таблица 3.1 - Отчет по рейтингам

Объект	Общий рейтинг	1-й курс						2-й курс						3-й курс					
		1-й семестр			2-й семестр			3-й семестр			4-й семестр			5-й семестр			6-й семестр		
		1 ат.	2 ат.	3 ат.	1 ат.	2 ат.	3 ат.	1 ат.	2 ат.	3 ат.	1 ат.	2 ат.	3 ат.	1 ат.	2 ат.	3 ат.	1 ат.	2 ат.	3 ат.
Факультет1																			
Кафедра1																			
Специальность																			
Группа1																			
ФИО студента1																			
ФИО студента2																			
...																			
Группа2																			
...																			
Специальность																			
...																			
Кафедра2																			
...																			
Факультет2																			
...																			

В этой таблице представлен отчет за первые три курса, 4-й и 5-й курсы добавляются с правой стороны отчета по аналогии с 1–3-м курсами. В зависимости от выбора параметров в п. 1–9 форма отчета может изменять свое содержание, как по вертикали, так и по горизонтали. Рейтинг для каждой ячейки отчета рассчитывается по алгоритму, описываемому в следующем разделе.

Алгоритм расчета рейтингов

В таблице 3.2 приведены обозначения, используемые в формулах по расчету рейтингов.

Трудоемкость поддисциплины по видам занятий (без учета форм отчетности) рассчитывается по формуле

$$T_{i_{jk},pd_d}^p = \sum_{t,m_{pd_d}} T_{i_{jk},m_{pd_d}}^{p,t}.$$

Трудоемкость поддисциплины в целом рассчитывается с учетом формы отчетности по формуле

$$T_{i_{jk},pd_d} = \sum_p T_{i_{jk},pd_d}^p + T_{i_{jk},pd_d}^{\Phi}.$$

Средневзвешенная оценка для каждого студента по модулю за каждую аттестацию и по поддисциплине рассчитывается по нижеприведенным формулам:

$$O_{l,f,s_{i_{jk}}}^{t,m_{pd_d}} = \frac{\sum_p O_{l,f,s_{i_{jk}}}^{p,t,m_{pd_d}} T_{i_{jk},m_{pd_d}}^{p,t}}{\sum_p T_{i_{jk},m_{pd_d}}^{p,t}};$$

$$O_{l,f,s_{i_{jk}}}^{pd_d} = \frac{\sum_{t,m_{pd_d}} \sum_p O_{l,f,s_{i_{jk}}}^{p,t,m_{pd_d}} T_{i_{jk},m_{pd_d}}^{p,t} + O_{l,f,s_{i_{jk}}}^{pd_d} T_{i_{jk},pd_d}^{\Phi}}{\sum_{t,m_{pd_d}} \sum_p T_{i_{jk},m_{pd_d}}^{p,t} + T_{i_{jk},pd_d}^{\Phi}}.$$

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Соответственно рейтинги студентов по модулям за текущие аттестации и рейтинги студентов по поддисциплине рассчитываются по формулам

Действителен: с 19.08.2022 по 19.08.2023

$$R_{l,f,s_{g_{ijk}}}^{t,m_{pd_d}} = O_{l,f,s_{g_{ijk}}}^{t,m_{pd_d}} T_{ijk,m_{pd_d}}^t,$$

$$R_{l,f,s_{g_{ijk}}}^{pd_d} = O_{l,f,s_{g_{ijk}}}^{pd_d} \left(\sum_t T_{ijk,m_{pd_d}}^t + T_{ijk,pd_d}^{\Phi} \right).$$

Таблица 3.2 – Обозначения формул расчета рейтинга

Обозначения	Пояснения
$O_{l,f,s_{g_{ijk}}}^{p,t,m_{pd_d}}$	Оценка за текущую работу в 100-балльной шкале, где: p – вид занятий (лекции, лабораторные, экзамен и т. д.); t – номер аттестации в семестре; m_{pd_d} – модуль поддисциплины pd дисциплины d ; l – форма оплаты (бюджетники, контрактники); f – категории обучаемых (типы: студент, аспирант и т. д.) и формы обучения (очная, заочная и очно-заочная)); $s_{g_{ijk}}$ – студент g_{ijk} -й группы i_{jk} -й специальности, направления, специализации j -й кафедры k -го факультета (института)
$T_{ijk,m_{pd_d}}^{p,t}$	Трудоемкость модуля m_{pd_d} за аттестацию t в зачетных единицах
$T_{ijk,m_{pd_d}}^p$	Трудоемкость дисциплины (поддисциплины) pd_d по видам занятий за семестр в зачетных единицах
$T_{ijk,m_{pd_d}}$	Трудоемкость дисциплины (поддисциплины) pd_d за семестр в зачетных единицах
T_{ijk,pd_d}^{Φ}	Трудоемкость по форме отчетности (зачет или экзамен) в зачетных единицах по поддисциплине (дисциплине) за семестр
$O_{l,f,s_{g_{ijk}}}^{t,m_{pd_d}}$	Средневзвешенная оценка по модулю
$O_{l,f,s_{g_{ijk}}}^{pd_d}$	Средневзвешенная оценка за семестр
$R_{l,f,s_{g_{ijk}}}^{t,m_{pd_d}}$	Рейтинг $s_{g_{ijk}}$ -го студента за аттестацию t по модулю m_{pd_d}
$R_{l,f,s_{g_{ijk}}}^{pd_d}$	Рейтинг $s_{g_{ijk}}$ -го студента за семестр по поддисциплине (дисциплине)

Задания

1. Разработать модель вариантов использования.
2. Разработать спецификации вариантов использования.

Контрольные вопросы

1. Какие бывают уровни представления требований?
2. К какому стандарту относится модель вариантов использования?
3. Что такое вариант использования?
4. Какие вы знаете отношения между вариантами использования?
5. Что показывает связь между актантом и вариантом использования?
6. Опишите структуру спецификации варианта использования.
7. Для чего разрабатывается спецификация вариантов использования?

Работа с литературой:

Рекомендуемые источники информации
(№ источника)

Полное наименование системы – «Автоматизированная информационная система «Научно-исследовательские работы». Краткое наименование АИС НИР.

Сведения о заказчике и исполнителе

Заказчик системы – «Университет» в лице проректора И. И. Иванова

Исполнитель – ООО «ИТ-консалтинг» в лице директора П. П. Петрова.

Основания для выполнения работ, сроки и финансирование

Разработка ведется на основании договора № 1 от 01.01.2018, заключенного между «Университетом» и ООО «ИТ-консалтинг».

Система должна быть разработана в течение 2018 года и сдана в опытную эксплуатацию до 12.12.2018.

Работы по созданию системы финансируются Университетом в соответствии с календарным планом, являющимся неотъемлемой частью договора.

Основные понятия, определения и сокращения

Данный пункт содержит перечень основных понятий, определений и сокращений, используемых в настоящем документе.

Автоматизированная система в защищенном исполнении – автоматизированная система, реализующая информационную технологию выполнения установленных функций в соответствии с требованиями стандартов и/или нормативных документов по защите информации.

Актант (пользователь системы) – субъект (человек, организация, другая АИС), использующий функции или информацию данной системы.

Архитектура системы – высокоуровневая концепция системы и ее окружения.

Архитектура программной системы (в фиксированный момент времени) – организация структуры значимых компонентов системы, взаимодействующих через интерфейсы. Указанные компоненты, в свою очередь, составлены из более мелких компонентов и интерфейсов.

База данных (БД) – совместно используемый набор логически связанных данных (и описание этих данных), предназначенных для удовлетворения информационных потребностей организации.

Вариант использования – функциональный связный блок, выраженный в виде транзакции между актантом и системой. Вариант использования описывает поведение системы как последовательности действий. Любой вариант использования должен приводить к полезному результату для актанта.

Доступность информации – состояние информации, характеризующееся способностью АИС обеспечивать беспрепятственный доступ к информации субъектов, имеющих на это полномочия.

Защита информации – деятельность по предотвращению утечки защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию.

Конфиденциальная информация – информация с ограниченным доступом, не содержащая сведений, составляющих государственную тайну, доступ к которой ограничивается в соответствии с законодательством Российской Федерации.

Конфиденциальность информации – состояние защищенности информации, характеризующееся способностью АИС обеспечивать сохранение в тайне информации от субъектов, не имеющих полномочий на ознакомление с ней.

Модель – абстрактное представление одного или нескольких аспектов системы. Это полное описание системы с некоторой точки зрения. Одной модели всегда недостаточно для описания всех аспектов системы.

Модель вариантов использования – диаграмма, описывающая основные варианты использования системы, актантов и отображающая связи актантов с вариантами использования (распределение функциональности системы между актантами).

Модуль – элементарный компонент программной системы.

Несанкционированный доступ (НСД) – доступ к информации или действия с информацией, нарушающие правила разграничения доступа.

Сервер приложений – специализированное программное обеспечение, предназначенное для централизованного хранения и обработки базы данных.

Система управления базами данных (СУБД) – специализированное программное обеспечение, предназначенное для централизованного хранения и обработки данных в БД, а также управления доступом нескольких пользователей к одним и тем же данным.

Электронная подпись
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Щербук Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Спецификация вариантов использования – документ, описывающий основную последовательность взаимодействия актанта с системой (поток) и все альтернативные потоки одного варианта использования.

Утечка информации – неконтролируемое распространение защищаемой информации.

Целостность информации – состояние защищенности информации, характеризующееся способностью АИС обеспечивать сохранность и неизменность информации при попытках несанкционированных или случайных воздействий на нее в процессе обработки или хранения.

Актуальность разработки системы

Актуальность выполнения разработки связана с увеличением объемов и тематики выполняемых в «Университете» исследований, расширением влияния результатов исследований на экономику региона и широким внедрением автоматизированных систем мониторинга и управления научно-инновационным процессом.

Внедрение разработанной АИС НИР позволит упорядочить учет, облегчить поиск информации по проводимым НИР, обеспечить оперативный контроль выполнения календарных и финансовых планов по проектам НИР и по вузу в целом. АИС НИР позволит в дальнейшем подключить к ней филиалы «Университета», расширив сферу ее применения.

Назначение и цели создания (развития) системы

Данный раздел обычно состоит из четырех подразделов:

- цели создания системы;
- назначение системы;
- задачи, решаемые системой;
- область применения системы.

Цель создания системы формулируется как решение проблемы, описанной в подразделе «Актуальность разработки системы».

Назначение системы – это перечень потребностей заинтересованных лиц, которые будут полностью или частично удовлетворены за счет применения разрабатываемой АИС.

В подразделе «Задачи, решаемые системой» описывают те задачи, которые решаются за счет использования системы и в совокупности позволяют достичь заявленной цели и удовлетворить потребности заинтересованных лиц. Задачи, описываемые в данном подразделе, связаны с функциями пользователей системы, которые они выполняют в своей повседневной работе.

В подразделе «Область применения системы» нужно указать границы действия системы. Описывается, кто и в каком объеме будет использовать систему при решении своих задач.

Пример написания раздела

Цели создания системы

Автоматизированная информационная система разрабатывается с целью повышения эффективности управления научным процессом за счет автоматизации процесса регистрации и мониторинга состояния НИР и их динамики.

Назначение системы

АИС НИР предназначена для:

- обеспечения оперативного доступа к информации о научных разработках и ходе выполнения хоздоговоров и грантов;
- формирования всех видов отчетов, связанных с научной деятельностью «Университета»;
- обеспечения руководителей научных проектов инструментом, автоматизирующим большую часть рутинной работы по оформлению результатов НИР и сопутствующих документов.

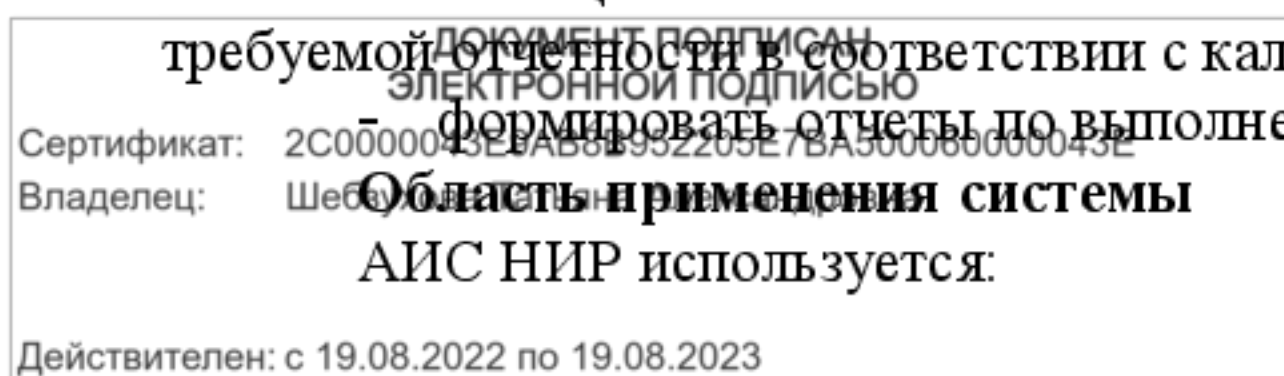
Задачи, решаемые системой

АИС НИР позволяет решать следующие задачи:

- вести учет НИР;
- выполнять обширный поиск информации по НИР;
- вести календарный и финансовый планы;
- сообщать пользователям системы и предупреждать их о сроках выполнения работ и требуемой отчетности в соответствии с календарным и финансовым планами;
- формировать отчеты по выполненным НИР.

Область применения системы

АИС НИР используется:



- проректором по НИР для мониторинга состояния НИР и эффективности использования финансовых средств;
- начальником Управления НИОКР для мониторинга, оперативного управления организацией НИР и подготовки организационных решений для утверждения проректором по НИР;
- экономической службой УНИОКР для контроля исполнения календарных планов, включая контроль получения финансовых средств и исполнения сметы расходов по каждой НИР;
- руководителями НИР для регистрации НИР, ведения календарных планов и сметы расходов, кадрового состава, основного содержания и результатов завершенных НИР.

Характеристики объекта автоматизации

В настоящем разделе ТЗ приводят:

- краткие сведения об объекте автоматизации или ссылки на документы, содержащие такую информацию;
- сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды;
- количественные характеристики потоков данных и данных, подлежащих хранению.

Для описания объекта автоматизации часто применяют различные методологии, такие как IDEF0, IDEF3, DFD или ARIS. С использованием этих технологий моделируются процессы, которые описываются в тексте. Если объект автоматизации несложный, то он описывается текстом без применения каких-либо технологий.

Пример написания раздела

Организация и планирование научно-исследовательской и инновационной деятельности

«Университет» в целях организации эффективной научной и инновационной деятельности осуществляет:

- организацию экспертизы научных тем и инновационных проектов, представляемых для финансирования в рамках заказа Министерства образования и науки РФ и включения их в число участников научных и научно-технических программ;
- создание центров коллективного пользования уникальным научным оборудованием в кооперации с вузами региона в рамках интегрированных учебно-научно-производственных комплексов, создание системы снабжения оборудованием, средствами вычислительной и оргтехники, материалами и комплектующими изделиями, организацию технического обслуживания и ремонта;
- разработку организационной структуры научно-исследовательской и инновационной деятельности, правовых и экономических основ этого вида работ;
- разработку, внедрение и сертификацию системы менеджмента качества в соответствии с требованиями ИСО 9000–2001 при проведении исследований и разработок, получение лицензий на право проведения специальных НИОКР (научно-исследовательские и опытно-конструкторские разработки);
- контроль за выполнением научных исследований и реализацией инновационных проектов, эффективным использованием и развитием научной и экспериментально-производственной базы вуза.

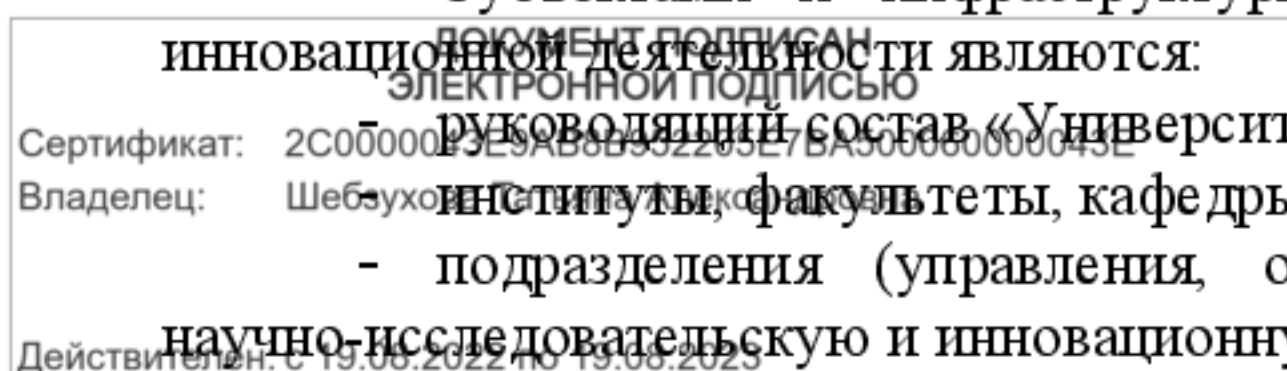
«Университет» планирует свою научно-инновационную деятельность, финансируемую за счет госбюджета и привлеченных средств, в соответствии с утвержденными в установленном порядке научными и научно-техническими программами и (или) договорами, на основе годовых и перспективных тематических планов.

«Университет» ежегодно разрабатывает сводный (из всех источников финансирования) тематический план НИОКР, который утверждается ректором университета по рекомендации НТС.

Утвержденный годовой тематический план НИОКР «Университета» подлежит обязательному исполнению.

Исполнители научно-исследовательских работ

- Субъектами и инфраструктурными подразделениями научно-исследовательской и инновационной деятельности являются:
- руководящий состав «Университета» (ректор, проректоры);
- институты, факультеты, кафедры, филиалы;
- подразделения (управления, отделы, службы), обеспечивающие непосредственно научно-исследовательскую и инновационную деятельность:



- Управление научно-исследовательских и опытно-конструкторских работ и его отделы;
- Управление стандартизации, сертификации и качества;
- Административно-правовое управление;
- библиотека «Университета»;
- информационно-вычислительный центр;
- Управление бухгалтерского учета и финансового контроля;
- Планово-финансовое управление;
- Управление по режиму и безопасности;
- Хозяйственное управление;
- Управление кадров.

Научные работы в «Университете» выполняются:

- профессорско-преподавательским составом в соответствии с индивидуальными планами (написание монографий, статей, тезисов, заявок на объекты интеллектуальной собственности и др.) в основное рабочее время;
- научными, инженерно-техническими работниками, учебно-вспомогательным персоналом, специалистами и рабочими научных и проектно конструкторских подразделений в основное рабочее время;
- докторантами, аспирантами, стажерами в соответствии с индивидуальными планами их подготовки, а также в свободное от работы (учебы) на кафедрах, в научно-инновационных подразделениях за дополнительную плату;
- студентами, бакалаврами, магистрантами в ходе выполнения курсовых и дипломных проектов, других исследовательских работ, предусмотренных учебными планами, в студенческих научных кружках, студенческих конструкторских бюро, центрах НТТМ, молодежных инновационных центрах, а также на кафедрах, в НИИ и в других структурных подразделениях «Университета» в свободное от учебы время за дополнительную плату либо безвозмездно.

К выполнению научных работ, финансируемых из средств федерального бюджета и по хозяйственным договорам, в том числе на условиях совместительства, привлекаются профессорско-преподавательский состав, научные сотрудники, руководящие и другие работники «Университета», а также сотрудники предприятий, учреждений и организаций, независимо от форм собственности, в свободное от основной работы время по договорам подряда.

Учет и отчетность по научно-исследовательским работам

Все выполняемые в «Университете» НИОКР подлежат государственной регистрации в соответствии с требованиями действующей нормативно-технической документации.

Не подлежат государственной регистрации работы, связанные с обслуживанием научных исследований и предоставлением научно-производственных услуг.

Государственную регистрацию и учет выполняемых и законченных открытых НИР и НИОКР осуществляет Всероссийский научно-технический и информационный центр (ВНТИЦ) Федерального агентства по науке и инновациям.

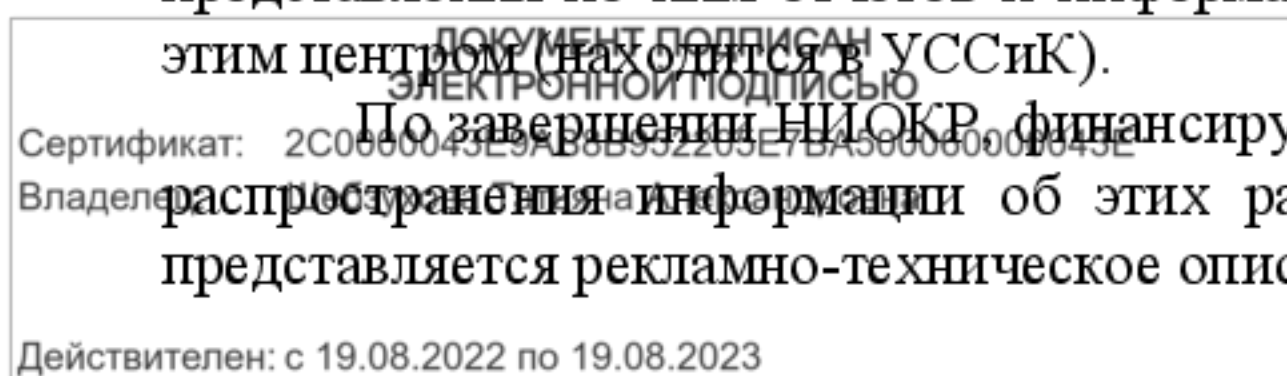
Подразделение-исполнитель НИОКР через Управление стандартизации, сертификации и качества (УССиК) направляет во ВНТИЦ (Всероссийский научно-технический информационный центр) регистрационную карту (РК) установленного образца не позднее 30 дней с момента начала финансирования работы.

Подразделение-исполнитель зарегистрированной НИОКР через УССиК и УНИОКР в срок, не превышающий 30 дней с момента окончания работы и приемки ее заказчиком, направляет во ВНТИЦ информационную карту (ИК) установленного образца и научно-технический отчет о зарегистрированной НИОКР, утвержденный проректором по НИР.

Представление заключительного отчета о зарегистрированной работе является обязательным при выполнении работы, финансируемой из средств федерального бюджета.

Порядок представления информационных материалов, правила составления регистрационной и информационной карт должны соответствовать Инструкции о порядке регистрации и учета открытых научно-исследовательских и опытно-конструкторских работ, представления по ним отчетов и информационных материалов в ВНТИЦ и выдачи информации этим центром (находится в УССиК).

По завершении НИОКР, финансируемых из внебюджетных источников, в целях рекламы и распространения информации об этих работах во ВНТИЦ вместе с информационной картой представляется рекламно-техническое описание или научно-технический отчет о НИОКР.



Материалы информационной карты используются при формировании электронного банка данных завершенных НИР «Университета» и для размещения информации в сети Интернет.

При представлении отчета о НИОКР университетом (УНИОКР) определяются условия его распространения ВНИИЦ в соответствии с действующим законодательством РФ.

Заключительные отчеты о выполненной работе оформляются в соответствии с требованиями ГОСТ 7.32–2001.

При выполнении опытно-конструкторских и технологических работ заказчику может быть передана рабочая документация, опытные образцы, макеты, технологические регламенты.

Полученные результаты по завершенным этапам и НИОКР в целом, выполняемым за счет средств федерального бюджета, подлежат обсуждению на заседаниях кафедры (отдела, лаборатории) с приглашением представителей головных организаций, представителей других кафедр и структурных подразделений «Университета».

Результаты научно-исследовательской и инновационной деятельности подлежат ежегодному обсуждению на Ученом совете или НТС (научно-технический совет) «Университета».

Приемка госбюджетных НИОКР, финансируемых из средств федерального и местного бюджетов, осуществляется НТС «Университета» после прохождения внутренней экспертизы в порядке, установленном приказом по «Университету».

Отчеты по завершенным госбюджетным НИОКР передаются УНИОКР в библиотеку «Университета» в обязательном порядке, а их электронные версии хранятся в подразделениях-исполнителях и в УНИОКР.

Требования к системе

Данный раздел состоит из следующих подразделов:

- требования к системе в целом;
- требования к функциям (задачам), выполняемым системой;
- требования к видам обеспечения.

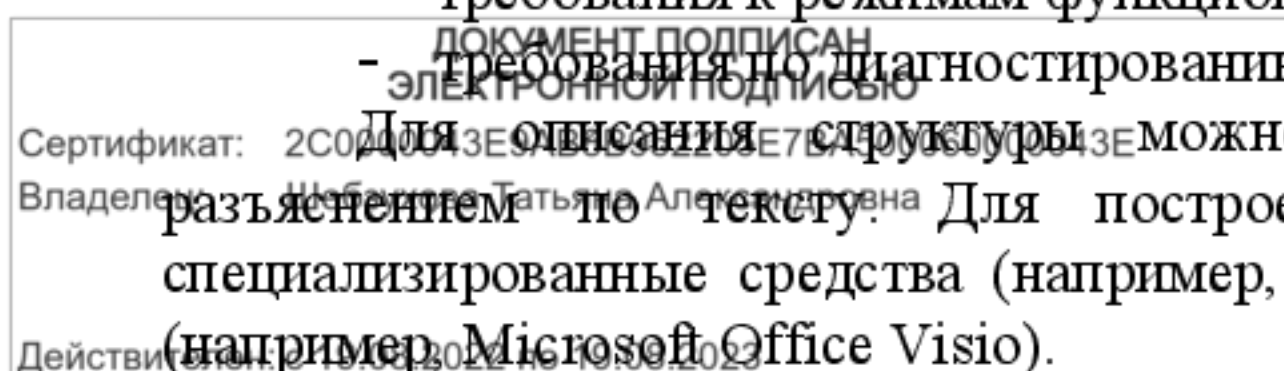
В подразделе «Требования к системе в целом» указывают:

- требования к структуре и функционированию системы;
- требования к численности и квалификации персонала системы и режиму его работы;
- показатели назначения;
- требования к надежности;
- требования безопасности;
- требования к эргономике и технической эстетике;
- требования к транспортабельности для подвижных АС;
- требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы;
- требования к защите информации от несанкционированного доступа;
- требования по сохранности информации при авариях;
- требования к защите от влияния внешних воздействий;
- требования к патентной чистоте;
- требования по стандартизации и унификации;
- дополнительные требования.

Многие пункты данного подраздела можно опускать. Главными являются первые два пункта. В первом из них нужно описать структуру разрабатываемой системы, в которой приводится:

- перечень подсистем, их назначение и основные характеристики, требования к числу уровней иерархии и степени централизации системы;
- требования к способам и средствам связи для информационного обмена между компонентами системы;
- требования к характеристикам взаимосвязей создаваемой системы со смежными системами, требования к ее совместимости, в том числе указания о способах обмена информацией (автоматически, пересылкой документов, по телефону и т. п.);
- требования к режимам функционирования системы;
- требования по диагностированию системы.

Для описания структуры можно приводить схемы и диаграммы с дальнейшим разъяснением по тексту. Для построения схем и диаграмм можно использовать как специализированные средства (например, редакторы UML), так и средства общего назначения (например, Microsoft Office Visio).



В подразделе «Требования к функциям (задачам), выполняемым системой», приводят: по каждой подсистеме перечень функций, задач или их комплексов (в том числе обеспечивающих взаимодействие частей системы), подлежащих автоматизации;

требования к качеству реализации каждой функции (задачи или комплекса задач), к форме представления выходной информации, характеристики необходимой точности и времени выполнения, требования одновременности выполнения группы функций, достоверности выдачи результатов;

перечень и критерии отказов для каждой функции, по которой задаются требования по надежности.

Для описания функций и требований к качеству их реализации можно использовать метод вариантов использования и их спецификаций, рассмотренных в данных методических указаниях по выполнению лабораторных работ.

Пример написания раздела

Требования к системе в целом

АИС НИР должна быть разработана в виде сайта на основе трехуровневой архитектуры. Сервер баз данных и сервер приложений АИС НИР должны быть созданы на базе постреляционной СУБД Cache. АИС НИР должна иметь два типа клиентских мест: первый предназначен для работы руководителей НИР и должен быть реализован в облегченном варианте на базе технологии CSP; второй тип в связи со сложностью реализуемых функций необходимо разработать на базе Java-технологии (applet – servlet) под платформу J2EE.

АИС обрабатывает конфиденциальную информацию (персональные данные преподавателей, сотрудников, аспирантов и студентов) и представляет собой автоматизированную систему в защищенном исполнении.

Требования к структуре и функционированию системы

АИС НИР должна быть реализована с использованием технологии Cache в виде четырех подсистем. Архитектура системы представлена на рисунке 4.1.

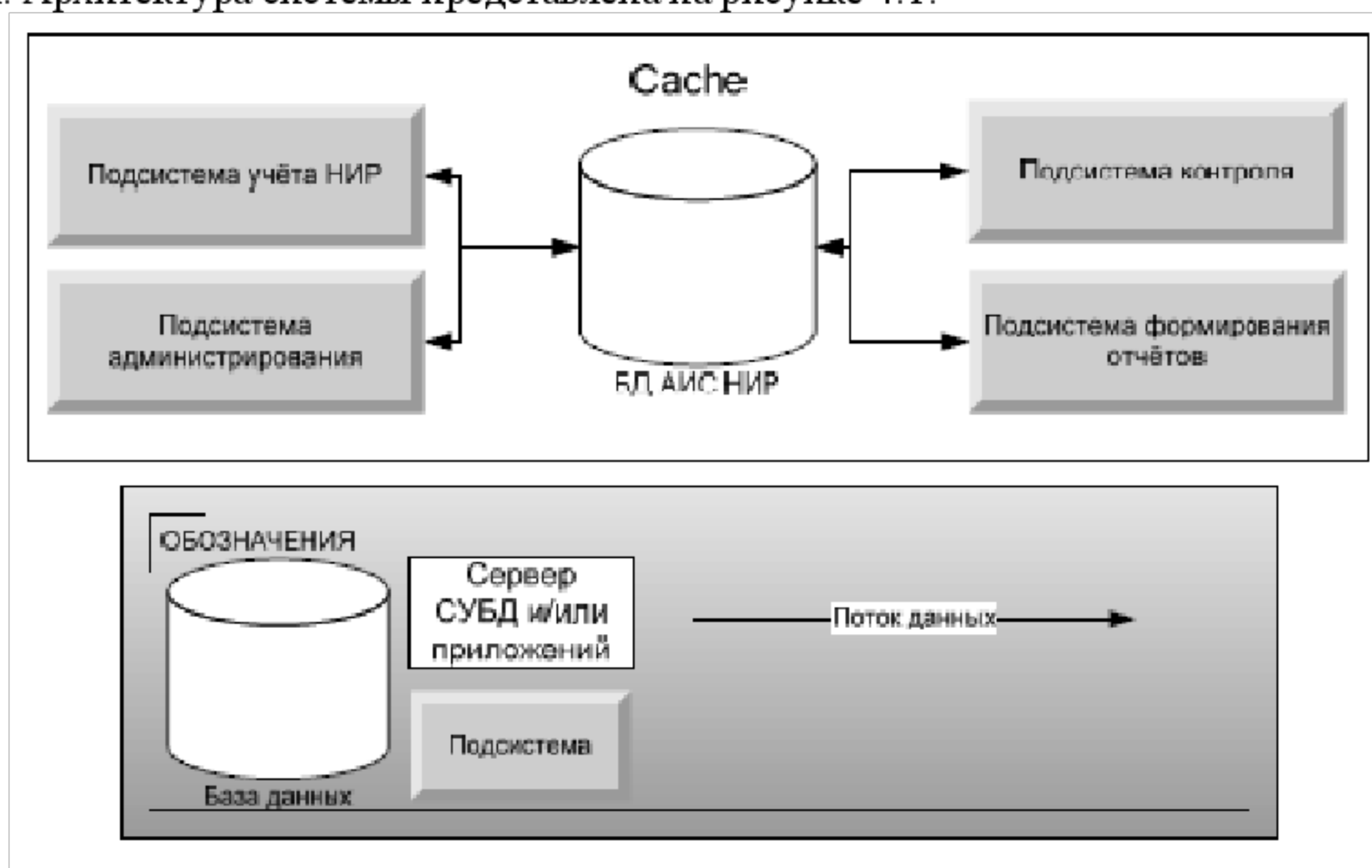


Рисунок 4.1 - Архитектура АИС НИР

Подсистема учета НИР предназначена для осуществления ввода регистрационных данных о сотрудниках и выполняемых НИР, а также информации о результатах выполненных работ. При реализации данной подсистемы необходимо разграничить доступ к данным по НИР в соответствии с установленными уровнями доступа к данным.

Подсистема контроля предназначена для мониторинга выполнения календарного плана по темам, а также выполнения финансового плана поквартально с выдачей информации о невыполнении плана по срокам и объемам по темам.

Подсистема формирования отчётов предназначена для создания отчетов о состоянии квартальных сводок выполнения финансового и календарного планов, а также отчетов о выполнении НИР.

Подсистема администрирования предназначена для регистрации пользователей системы, назначения прав пользователям и редактирования справочных данных.

Электронная подпись
Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шибзукова Татьяна Александровна
Действителен: с 19.08.2022 по 19.08.2023

Пользователями АИС НИР являются:

- Пользователи АИС НИР должны:

- Администратор АИС НИР должен иметь высшее образование со специализацией в области разработки информационных систем и баз данных, обладать навыками администрирования современных SQL-серверов и серверов приложений, пройти обучение основам работы с АИС НИР в объеме технической (эксплуатационной) документации (Руководство администратора) на систему.

В АИС НИР должны быть реализованы функции, представленные ниже на диаграммах вариантов использования. Каждая диаграмма описывает функциональность одной подсистемы (рисунок 4.2).



Описание вариантов использования

Зарегистрировать НИР. Научно-исследовательская работа регистрируется в соответствии с учетной карточкой НИР и регистрационной картой (РК), формы которых приведены ниже.

- наименование НИР: _____

Сертификат: 2C0600040E9498E9522031

Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

банков, другие средства; номер государственной регистрации; характер работы (фундаментальная, поисковая, прикладная, ОКР, разработка);

- сведения о наличии интеллектуальной собственности;
- заказчик (полное наименование и юридический адрес). Заказчик должен выбираться из справочника. Если его там нет, то предлагать форму для
- ввода данных в справочник;
- факультет (из справочника);
- кафедра (из справочника);
- научный руководитель (ФИО, ученая степень и звание, стаж работы). Руководитель

выбирается из справочника сотрудников вуза;

- кадровый состав НИР. Данные о работниках вводятся из справочника путем прикрепления существующих в справочнике работников к данной НИР. В случае отсутствия работников в справочнике система должна предложить внести работников в справочник;

- организации-соисполнители;
- код ГРНТИ (из справочника).

Ввести результаты работы по НИР. По окончании работы руководителем НИР вводятся результаты в систему. Форма ввода результатов должна содержать следующие поля:

- полученные научные и/или научно-технические результаты (текстовое поле на 2000 знаков);

- полученная научная и/или научно-техническая продукция (текстовое поле на 2000 знаков);

- ключевые слова и словосочетания, характеризующие результаты (продукцию). Текстовое поле на 500 знаков;

- наличие аналогов для сопоставления результатов (продукции) (Да/Нет);

- преимущества полученных результатов (продукции) по сравнению с результатами отечественных и зарубежных НИР (текстовое поле на 2000 знаков);

- степень готовности полученных результатов к практическому использованию (текстовое поле на 500 знаков);

- перспективы, формы и сроки коммерческого использования результатов и продукции (текстовое поле на 2000 знаков);

- форма представления результатов НИР (научно-технические отчеты, учебники, доклады, патенты, экспонаты выставок). Поле заполняется из справочника;

- библиографический список публикаций, отражающих результат работы. Система должна выводить подсказку о правильности заполнения библиографического списка;

- использование результатов в учебном процессе (Да/Нет). Если Да, то вывести текстовое поле для ввода описания на 100 знаков;

- количество сотрудников из числа ППС, принимавших участие в выполнении НИР и указанных научно-технических отчетах в качестве исполнителей;

- количество студентов, принимавших участие в выполнении НИР, в том числе: указанных в научно-технических отчетах в качестве исполнителей; являющихся авторами публикаций по результатам НИР; с оплатой труда за счет средств НИР.

Состав и содержание работ по созданию (развитию) системы

Раздел должен содержать перечень стадий и этапов работ по созданию системы в соответствии с ГОСТ 34.601–90, сроки их выполнения, перечень организаций-исполнителей работ, ссылки на документы, подтверждающие согласие этих организаций на участие в создании системы, или запись, определяющую ответственного (заказчик или разработчик) за проведение этих работ.

В данном разделе также приводят:

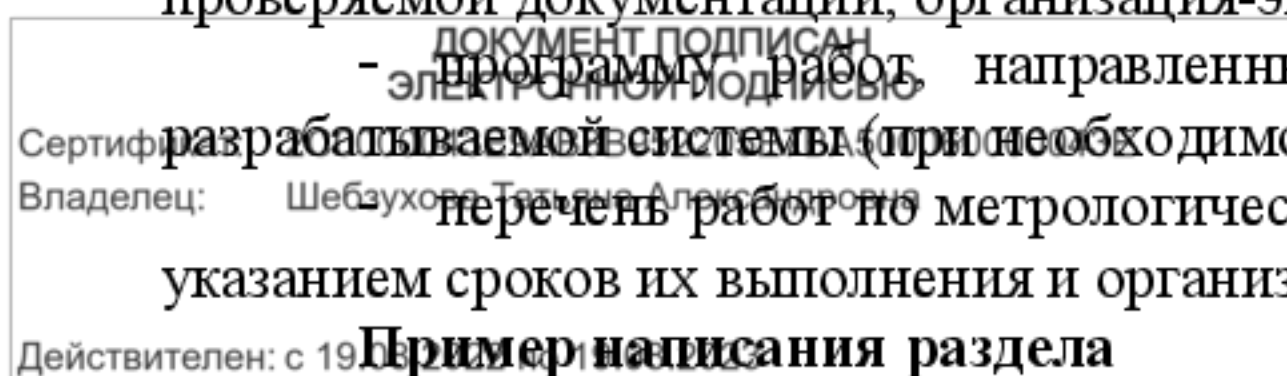
- перечень документов по ГОСТ 34.201, предъявляемых по окончании соответствующих стадий и этапов работ;

- вид и порядок проведения экспертизы технической документации (стадия, этап, объем проверяемой документации, организация-эксперт);

- программу работ, направленных на обеспечение требуемого уровня надежности разрабатываемой системы (при необходимости);

- перечень работ по метрологическому обеспечению на всех стадиях создания системы с указанием сроков их выполнения и организаций исполнителей (при необходимости).

Пример написания раздела



Разработка системы должна выполняться на основе архитектурно-ориентированного подхода. Выбранная модель жизненного цикла должна позволять выполнять итеративную и инкрементную разработку системы.

Таблица 4.1 - Перечень работ по созданию АИС

Наименование работы	Результат
Разработка спецификаций вариантов использования (описание последовательностей действий пользователей и системы в рамках каждого варианта использования)	Документы спецификаций
Разработка архитектуры программной системы	Модели архитектуры системы для каждого выбранного архитектурного представления
Уточнение логической структуры АИС (детальное проектирование)	Спецификация логической архитектуры АИС
Разработка модели данных для проектируемой подсистемы или системы в целом и создание БД	Объектная или реляционная модель данных и БД
Разработка проектных моделей пользовательского интерфейса	Модель пользовательского интерфейса модулей АИС в среде разработки
Проектирование, разработка компонентов системы и их тестирование	Действующий образец АИС, функционирующий на программно-аппаратном комплексе разработчика. Сценарии тестов
Интеграционное тестирование функций АИС, исправление кода	Действующий образец АИС, удовлетворяющий требованиям ТЗ
Разработка документации	Комплект пользовательской документации АИС
Установка системы и приемочное тестирование	АИС, соответствующая требованиям ТЗ, установленная на программно-аппаратном комплексе заказчика и готовая к опытной эксплуатации
Обучение пользователей	Пользователи обладают практическими навыками работы с системой
Внедрение в опытную эксплуатацию	Акт сдачи-приемки системы в опытную эксплуатацию
Сопровождение системы (работа по замечаниям пользователей) во время опытной эксплуатации	Список дефектов и предложений по развитию и/или изменению системы

Основной перечень работ по созданию АИС, их содержание и результаты приведены в таблице 4.1. Здесь приведен перечень работ, соответствующий одной итерации жизненного цикла. Предполагается, что все перечисленные работы будут повторяться на каждой итерации при реализации подсистемы или отдельных вариантов использования.

Порядок контроля и приемки системы

В данном разделе указывают:

виды, состав, объем и методы испытаний системы и ее составных частей (виды испытаний в соответствии с действующими нормами, распространяющимися на разрабатываемую систему);

общие требования к приемке работ по стадиям (перечень участвующих предприятий и организаций, место и сроки проведения), порядок согласования и утверждения приемочной документации;

статус приемочной комиссии (государственная, межведомственная, ведомственная).

Пример написания раздела

Для взаимодействия Исполнителя и Заказчика в организации Заказчика определяется эксплуатационная служба и назначается сотрудник, ответственный за приемку системы.

Разработанная система принимается в опытную эксплуатацию. Готовые компоненты системы могут передаваться поочередно. Сдача и приемка автоматизированной информационной системы осуществляется на основе результатов тестирования, проводимого представителями Заказчика и Исполнителя в соответствии с программой испытания, которая формируется

Документ подписан
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 2C0000043E9AB8B952205E7BA50006000043E
Владелец: 2C0000043E9AB8B952205E7BA50006000043E
Действителен: с 19.08.2022 по 19.08.2023

совместно. В программе испытания должны быть указаны виды, состав, объем и методы проверки правильности получения выходных данных и соответствия системы требованиям данного ТЗ.

Для проверки работоспособности системы проводится выполнение контрольных примеров. Составление контрольных примеров с последующей их передачей комиссии производится эксплуатационной службой и разработчиками совместно. Для выполнения контрольного примера должен быть предоставлен программно-аппаратный комплекс, удовлетворяющий требованиям, изложенным в подразделе «Требования к видам обеспечения» настоящего документа. По результатам выполнения тестов комиссией составляется перечень замечаний, который рассматривается разработчиком в течение трех дней.

Опытная эксплуатация призвана выявить ошибки и собрать замечания и проводится в обязательном порядке. Для обеспечения проведения опытной эксплуатации формируется комиссия по приемке системы, в состав которой входят эксплуатационная служба и разработчики.

По окончании опытной эксплуатации эксплуатационная служба передает в комиссию по приемке системы перечень замечаний по работе системы. Комиссия рассматривает замечания и принимает решение о готовности системы к промышленной эксплуатации. В случае подтверждения комиссией готовности системы к промышленной эксплуатации в течение семи дней подписывается акт сдачи-приемки системы в промышленную эксплуатацию.

В противном случае комиссия передает разработчикам согласованный протокол замечаний. После устранения замечаний проводится повторная опытная эксплуатация на усеченном временном интервале.

Система считается сданной в промышленную эксплуатацию после подписания акта сдачи-приемки системы в промышленную эксплуатацию должностным лицом, ответственным за приемку системы. При выявлении существенных несоответствий характеристик системы требованиям ТЗ Заказчиком составляется обоснованный перечень замечаний, который подписывается ответственным лицом Заказчика и передается разработчикам для доработки системы.

Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

В настоящем разделе необходимо привести перечень основных мероприятий (и их исполнителей), которые следует выполнить при подготовке объекта автоматизации к вводу АИС в действие.

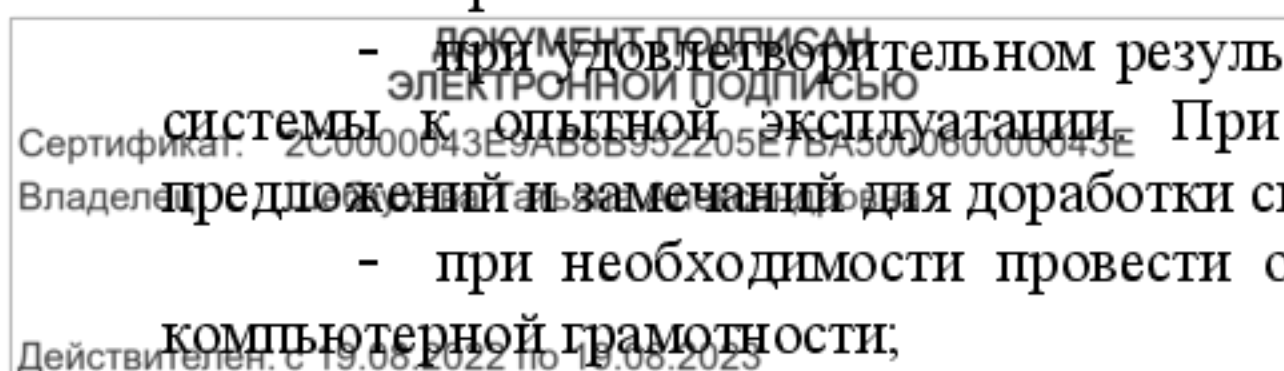
В перечень основных мероприятий включают:

- приведение поступающей в систему информации (в соответствии с требованиями к информационному обеспечению) к виду, пригодному для ввода и обработки в АИС;
- изменения, которые необходимо осуществить в объекте автоматизации;
- создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ;
- создание необходимых для функционирования системы подразделений и служб;
- сроки и порядок комплектования штатов и обучения персонала.

Пример написания раздела

Для подготовки АИС к вводу в эксплуатацию необходимо:

- назначить должностное лицо в организации Заказчика, ответственное за приемку системы;
- установить комплекс технических средств, удовлетворяющих требованиям соответствующего ТЗ, на рабочие места сотрудников организации Заказчика, которые должны участвовать в эксплуатации АИС;
- совместно с Исполнителем выполнить инсталляцию системного ПО в соответствии с Руководством администратора;
- провести ввод данных справочной информации и настройку системы в соответствии с Руководством администратора;
- совместно с Исполнителем составить документ «Программа испытаний»;
- провести испытания в соответствии с документом «Программа испытаний»;
- при удовлетворительном результате испытаний подписать акт технической готовности системы к опытной эксплуатации. При наличии замечаний составить документ «Перечень предложений и замечаний для доработки системы»;
- при необходимости провести обучение потенциальных пользователей АИС основам компьютерной грамотности;



- провести обучение потенциальных пользователей работе с АИС в объеме Руководства пользователя.

Для обеспечения функционирования системы необходимо разработать регламент эксплуатации, предусматривающий работу пользователей и служб сопровождения.

Создание служб, необходимых для функционирования системы

Функционирование АИС должна обеспечивать эксплуатационная служба – структурное подразделение или системный администратор, отвечающие за поддержку работы системы и контроль выполнения требований, изложенных в настоящем документе.

В целях планирования развития системы данная служба должна собирать заявки пользователей, подписанные руководителем соответствующих организационных подразделений, обобщать их и передавать разработчику системы. Для решения этих задач служба сопровождения должна выполнять следующие функции:

- проводить диагностику АИС;
- своевременно проводить резервное копирование баз;
- при возникновении аварийных ситуаций ликвидировать их последствия и восстанавливать технологический режим функционирования АИС;
- регистрировать ошибки, выявленные пользователями в процессе работы с системой, и оперативно передавать их разработчику системы;
- выполнять требования к эксплуатации и техническому обслуживанию АИС;
- проводить настройку автоматизированных рабочих мест пользователей в соответствии с их должностными обязанностями.

Для качественного выполнения перечисленных выше функций все сотрудники рассматриваемого подразделения должны пройти обучение и быть аттестованы разработчиком АИС. Сотрудники, не прошедшие аттестацию, не должны допускаться к выполнению администрирующих функций АИС.

Функциональные этапы внедрения системы

Внедрение системы осуществляется по мере реализации отдельных подсистем или функций системы. В таблице 4.2 представлен порядок внедрения первоочередных функций системы.

Таблица 4.2 Порядок внедрения первоочередных функций системы

№ п/п	Действия с системой и функции системы, подлежащие тестированию	Объем контрольных испытаний	Исполнители
1	Установка базы данных и сервера приложений	Первоначальная и обновления по мере модификаций	ИВЦ
2	Установка клиентских мест	Не менее 60 мест	ИВЦ
3	Генерация ведомостей	Ведомости по всем аттестациям для студентов первого курса	ИВЦ
4	Внесение данных о текущей трудоемкости модулей, внесение результатов по всем видам аттестаций	Все оценки для всех аттестаций студентов первого курса	Кафедры, ведущие занятия на первом курсе
5	Управление статусом ведомостей	Все ведомости должны пройти жизненный цикл от «новых» до «утвержденных»	Кафедры, ведущие занятия на первом курсе
6	Перевод результатов аттестации за семестр в пятибалльную систему и экспорт данных в АИС «Сессия»	В полном объеме для формирования протоколов стипендиальных комиссий для студентов первого курса.	ИВЦ
7	Функции обмена информацией	—	Все участники

Требования к документированию

В разделе приводят:

- согласованный разработчиком и заказчиком системы перечень подлежащих разработке комплектов и видов документов, соответствующих требованиям ГОСТ 34.201 и нормативной технической документации отрасли заказчика; перечень документов, выпускаемых на машинных носителях; требования к микрофильмированию документации;
- требования по документированию комплектующих элементов межотраслевого применения в соответствии с требованиями ЕСКД и ЕСПД;
- при отсутствии государственных стандартов, определяющих требования к документированию элементов системы, дополнительно включают требования к составу и содержанию таких документов.

Пример написания раздела

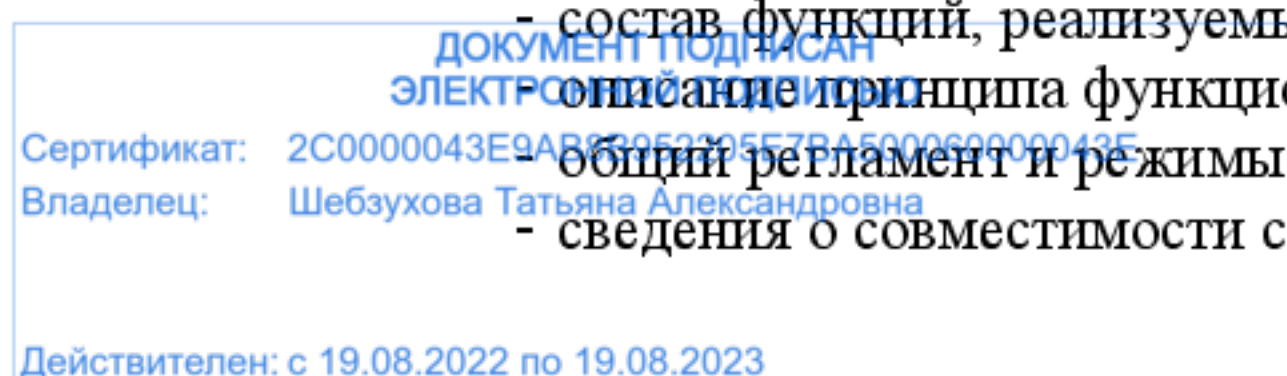
Комплект сопровождающей документации должен состоять из следующих документов:

- Паспорт системы;
- Общее описание системы;
- Руководство пользователя;
- Руководство администратора;
- Руководство программиста;
- Регламент эксплуатации.

Паспорт системы

Документ «Паспорт системы» должен описывать состав и краткое назначение основных элементов системы, передаваемой Заказчику, и включать следующие разделы:

- *общие сведения*:
 - наименование АИС, ее обозначение, присвоенное разработчиком;
 - наименование организации-разработчика;
- *основные характеристики АИС*:
 - состав функций, реализуемых АИС;
 - описание принципа функционирования;
 - общий регламент и режимы функционирования;
 - сведения о совместимости с другими системами;



- *комплектность* – перечень всех непосредственно входящих в состав АИС комплексов программных средств, в том числе носителей данных и эксплуатационных документов.

Общее описание системы

Документ «Общее описание системы» должен содержать следующие разделы:

- *назначение системы*:
 - вид деятельности, для информатизации которой предназначена система;
 - перечень объектов автоматизации, на которых будет использоваться система;
- *описание системы*:
 - структура системы и назначение ее частей;
 - сведения о АИС в целом и его составных частях;
 - описание функционирования системы;
 - описание взаимосвязи АИС с другими системами;
 - перечень функций, реализуемых системой.

Руководство администратора

Документ «Руководство администратора» должен содержать всю необходимую информацию, достаточную для работы системного администратора с данной АИС:

- функции администрирования при применении данной АИС;
- процедуры по установке и подготовке АИС к эксплуатации;
- инструкции по тестированию и описание тестового примера;
- инструкции по сохранению и восстановлению данных;
- техническое описание структуры системы и модели данных.

Руководство пользователя

Документ «Руководство пользователя» должен содержать описание пользовательского интерфейса и действий пользователя, достаточное для работы специально обученного пользователя. Данный документ должен содержать следующие разделы:

- введение;
- область применения;
- краткое описание возможностей;
- требования к уровню подготовки пользователя;
- перечень эксплуатационной документации, с которой необходимо ознакомиться пользователю;
- описание пользовательского интерфейса;
- описание процесса импорта данных из смежных систем.

Регламент эксплуатации

Документ «Регламент эксплуатации» должен содержать всю необходимую информацию об использовании системы в работе отделов и отдельных сотрудников в рамках их основной деятельности. В документе должны быть отражены все процессы деятельности отделов, в которых используется АИС, и описан порядок действий сотрудников с использованием АИС.

При проведении сертификации техническая документация должна отвечать действующим государственным стандартам (ГОСТ 34.602–89, ГОСТ 19.201–78, ГОСТ 19.202–78, ГОСТ 19.402–78, ГОСТ 19.502–78, ГОСТ 19.504–79, РД 50-34.698–90, ГОСТ 19.301–79).

Источники разработки

В данном разделе должны быть перечислены документы и информационные материалы (технич.-экономическое обоснование, отчеты о завершенных научно-исследовательских работах, информационные материалы на отечественные, зарубежные системы-аналоги и др.), на основании которых разрабатывалось ТЗ и которые должны быть использованы при создании системы.

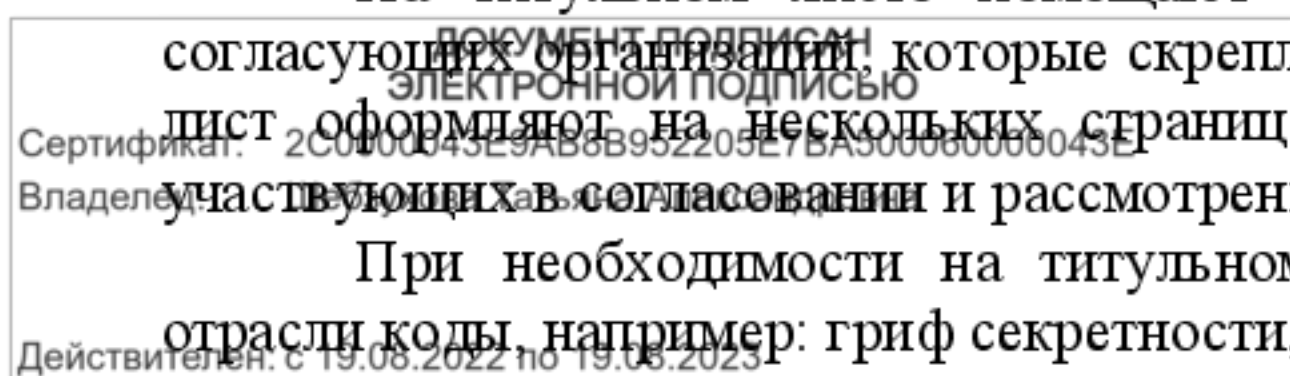
Правила оформления ТЗ

Разделы и подразделы ТЗ должны быть размещены в порядке, установленном в ГОСТ 34.602–89.

Номера листов (страниц) проставляют начиная с первого листа, следующего за титульным листом, в верхней части листа (над текстом, посередине) после обозначения кода ТЗ.

На титульном листе помещают подписи Заказчика, Разработчика и представителей согласующих организаций, которые скрепляют гербовой печатью. При необходимости титульный лист оформляют на нескольких страницах. Подписи разработчиков ТЗ и должностных лиц, участвующих в согласовании и рассмотрении проекта ТЗ, помещают на последнем листе.

При необходимости на титульном листе ТЗ допускается помещать установленные в отрасли коды, например: гриф секретности, код работы, регистрационный номер ТЗ и др.



Титульный лист дополнения к ТЗ оформляют аналогично титульному листу ТЗ. Вместо наименования «Техническое задание» пишут «Дополнение № ... к ТЗ на АИС ...».

На последующих листах дополнения к ТЗ помещают основание для изменения, содержание изменения и ссылки на документы, в соответствии с которыми вносятся эти изменения.

При изложении текста дополнения к ТЗ следует указывать номера соответствующих пунктов, подпунктов, таблиц основного ТЗ и т. п. и применять слова «заменить», «дополнить», «исключить», «изложить в новой редакции».

Задание

Написать разделы ТЗ и оформить его в соответствии с ГОСТ 34.602–89.

Контрольные вопросы

1. Структура технического задания по ГОСТу.
2. Какие допущения регламентирует ГОСТ при написании ТЗ?
3. В каких разделах ТЗ используется материал предыдущих лабораторных работ?
4. Какими ГОСТами и руководящими документами нужно руководствоваться при написании раздела «Требования к документированию»?
5. Какой ГОСТ регламентирует оформление ТЗ? __

Работа с литературой:

Рекомендуемые источники информации (№ источника)			
Основная	Дополнительная	Методическая	Интернет-ресурсы
1-3	1-3	1-3	1-2

Оценочные средства: контрольные вопросы

Тема 4. Архитектуры программных систем.

Лабораторная работа № 5. Реализация архитектуры на базе объектно-реляционного отображения с типизированными объектами

Форма проведения: Практическое выполнение задания

Цель работы – сформировать навыки:

- проектирования архитектуры клиент-серверных приложений;
- разработки системы на примере трехслойной архитектуры с использованием объектно-реляционного отображения;
- работы в Borland Developer Studio.

Краткие теоретические сведения

Трехслойная архитектура на базе объектно-реляционного отображения с типизированными объектами

Трехслойная архитектура реализуется на стороне клиента и имеет следующие слои: графический пользовательский интерфейс, бизнес-логику, объектно-реляционное отображение (рисунок 5.1). Взаимодействие слоев осуществляется посредством интерфейса и только в одностороннем порядке, т. е. классы верхнего слоя могут посылать сообщения классам нижнего слоя, но не наоборот.

Графический пользовательский интерфейс

Графический пользовательский интерфейс содержит классы интерфейсных объектов и классы управления интерфейсом пользователя. В слое бизнес-логики содержатся классы, управляющие логикой приложения и реализующие всю функциональность, связанную с обработкой данных и вычислениями. В слое объектно-реляционного отображения находятся классы, выполняющие функции преобразования данных из реляционного вида в объектный и наоборот, а также функции по работе с реляционной базой данных.

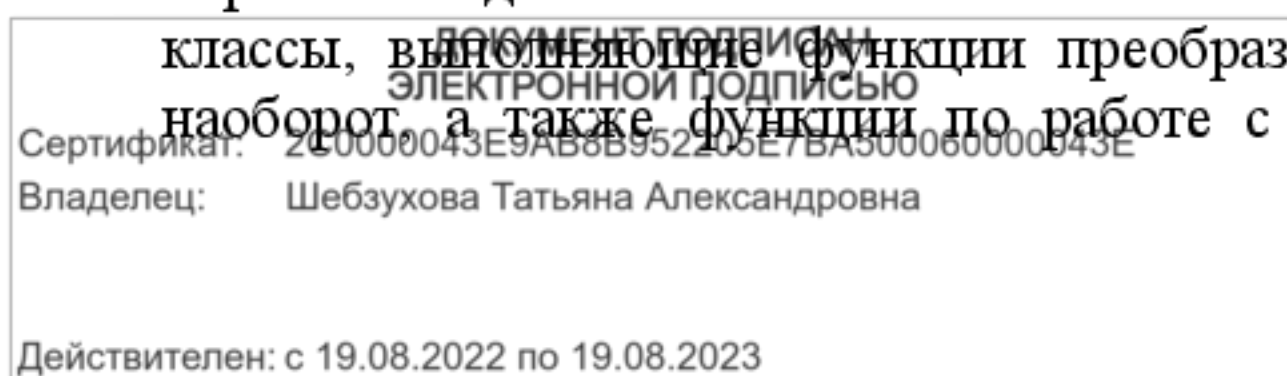




Рисунок 5.1 - Трехслойная архитектура

Понятие «типизированные объекты» связано со слоем графического интерфейса. Классы этого слоя напрямую могут работать с объектами предметной области, ссылки на которые они получают от классов бизнес-логики.

В интерфейсе таким образом может выполняться привязка компонентов формы к атрибутам класса предметной области.

Бизнес-логика. Данный слой должен содержать классы, реализующие функции обработки данных и вычислений.

Приведем пример базового (основного) класса, который может создаваться в этом слое.

Таблица 5.1 - Поля класса TDataPrepare

Поле	Описание
fExecute: TExecuteObject	Объект, используемый для выполнения запросов, не возвращающих набор данных
fSelect: TExecuteObject	Объект, используемый для выполнения запросов, возвращающих набор данных

Таблица 5.2 - Свойства класса TDataPrepare

Свойство	Описание
DataSource: TDataSource	Источник данных, используемый для отображения списка объектов
id: integer	Идентификатор объекта (совпадает с Id записи в таблице базы данных). Используется для хранения идентификатора текущего объекта

Таблица 5.3 - Методы класса TdataPrepare

Метод	Описание
Add()	Добавляет объект в БД
Update()	Изменяет атрибуты объекта в БД
Delete()	Удаляет объект из БД
Select()	Выбирает список объектов для отображения
LoadCurrent()	Загружает в поля класса атрибуты текущего объекта
Refresh()	Обновляет список объектов
Create(Connection: TConnection; Transaction: TTransactionObject=nil)	Конструктор класса

Сертификат: 2C70100043E9AB8952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Назовем этот класс TDataPrepare. Он должен быть порожден от класса TDBObject, поскольку должен уметь работать с соединениями и транзакциями. Атрибуты класса представлены в таблице 5.1, свойства класса – в таблице 5.2, виртуальные методы класса – в таблице 5.3.

Объектно-реляционное отображение. Данный слой состоит из классов, которые с помощью драйверов доступа к БД осуществляют взаимодействие с СУБД.

На рисунке 5.2 приведен пример базовых классов, на основе которых можно строить слой объектно-реляционного отображения. В таблице 5.4 представлено описание классов, изображенных на рисунке 5.2.

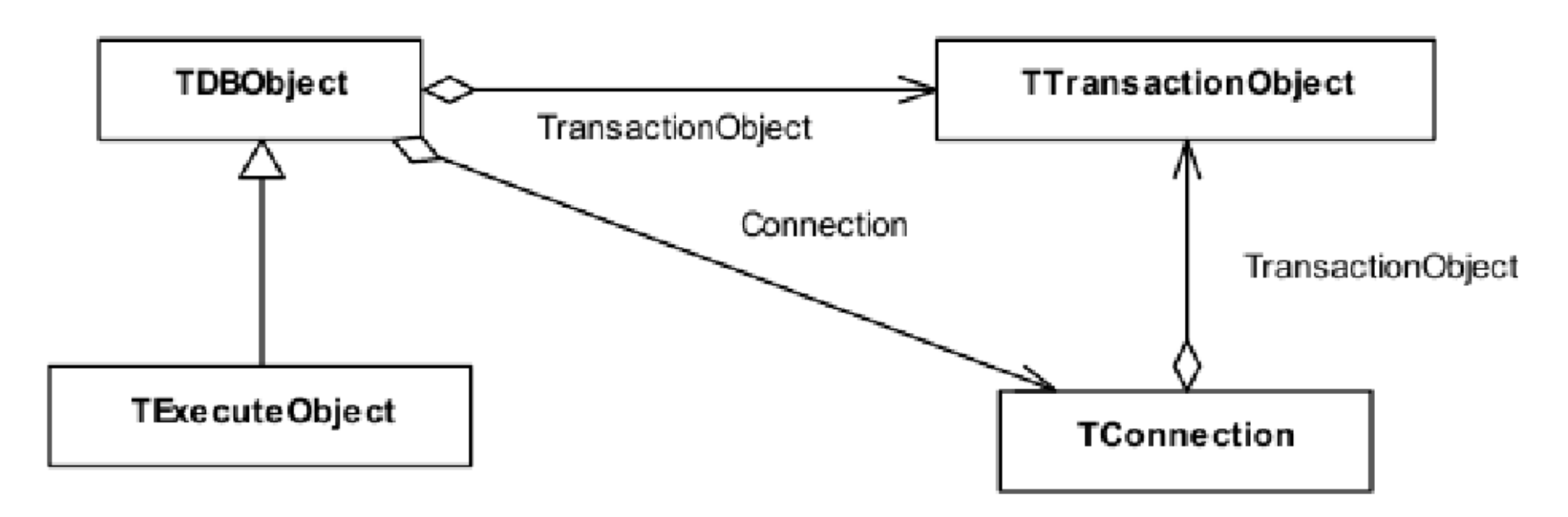


Рисунок 5.2 - Классы объектно-реляционного отображения

Таблица 5.4 - Описание базовых классов объектно-реляционного отображения

Имя класса	Описание
TConnection	Предназначен для управления соединением к базам данных и настройки параметров соединения
TTransactionObject	Предназначен для управления транзакцией (открытие, закрытие, откат)
TDBObject	Предназначен для работы с базой данных. Использует классы TConnection и TTransactionObject
TExecuteObject	Обеспечивает работу с SQL

Характеристики класса TConnection. Свойства класса TConnection представлены в таблице 5.5, методы этого класса – в таблице 5.6.

Таблица 5.5 - Свойства класса TConnection

Свойство	Описание
ServerName: string	Сетевое имя сервера СУБД
DataBaseName: string	Полный локальный путь к файлу БД
UserName: string	Имя пользователя СУБД, используемое для авторизации
Password: string	Пароль доступа к СУБД
DataBase: TIBDatabase	Компонент доступа к базе данных, с которой работает объект
TransactionObject: TTransactionObject	Транзакция, с которой работает объект

Таблица 5.6 - Методы класса TConnection

Метод	Описание
Connect()	Открытие соединения с базой данных
Disconnect()	Закрытие соединения с базой данных

Характеристики класса TTransaction Object.

Класс TTransactionObject является оболочкой к компоненту TIBTransaction, реализующему функции по работе с транзакциями.

Свойства класса TTransactionObject представлены в таблице 5.7, методы данного класса приведены в таблице 5.8.

Таблица 5.7 - Свойства класса TTransactionObject

Свойство	Описание
Transaction: TIBTransaction	Определяет, с какой транзакцией работает объект
InTransaction: Boolean	Определяет, запущена ли транзакция у объекта

Таблица 5.8 - Методы класса TTransactionObject

Метод	Описание
StartTransaction()	Запуск транзакции
Commit()	Подтверждение транзакции
Rollback()	Отмена транзакции

Характеристики класса TDBObject. Поля класса TDBObject представлены в таблице 5.9, свойства данного класса – в таблице 5.10.

Класс TDBObject содержит те же методы, что и класс TTransactionObject, для реализации возможности работы со своей собственной транзакцией. Соответственно если он использует чужую транзакцию, он ничего не может с ней делать. Методы класса TDBObject представлены в табл. 5.11.

Таблица 5.9 - Поле класса TDBObject

Поле	Описание
fOwn_transaction: Boolean	Определяет, использует объект свою транзакцию или чужую, переданную при создании класса

Таблица 5.10 - Свойства класса TDBObject

Свойство	Описание
Connection: TConnection	Подключение, используемое данным объектом
TransactionObject: TTransactionObject	Транзакция, используемая данным объектом

Таблица 5.11 - Методы класса TDBObject

Метод	Описание
StartTransaction()	Запуск транзакции
Commit()	Подтверждение транзакции
Rollback()	Отмена транзакции

Свойства класса TExecuteObject представлены в таблице 5.12, методы данного класса приведены в таблице 5.13.

Таблица 5.12 - Свойства класса TExecuteObject

Свойство	Описание
nfi[name: string]: Integer	Свойство доступа к полям набора данных. Используется, когда поле является целочисленным
nfs[name: string]: string	Свойство доступа к полям набора данных. Используется, когда поле является строковым
nff[name: string]: Extended	Свойство доступа к полям набора данных. Используется, когда поле является дробным

Таблица 5.13 - Методы класса TExecuteObject

Метод	Описание
ExecuteQuery(sql: string; params: array of variant)	Выполняет запрос к базе данных и не возвращает набор данных
SelectQuery(sql: string; params: array of variant; FetchAll: boolean=false)	Выполняет запрос к базе данных и возвращает набор данных
SelectStrings(sql: string; params: array of variant): TStringList	Выполняет запрос к базе данных и возвращает коллекцию строк в классе TStringList из первого поля запроса

Структура БД. В рассматриваемом в лабораторной работе примере будет использоваться база данных, реляционная модель которой приведена на рисунке 5.3. Описание таблиц базы данных приведено в таблице 5.14.

Таблица 5.14 - Описание таблиц базы данных

Таблица	Описание
Books	Хранит информацию о книгах
Groups	Хранит информацию о группах, в которых учатся студенты
Students	Хранит информацию о студентах
Students_Book	Хранит информацию о выданных студенту книгах

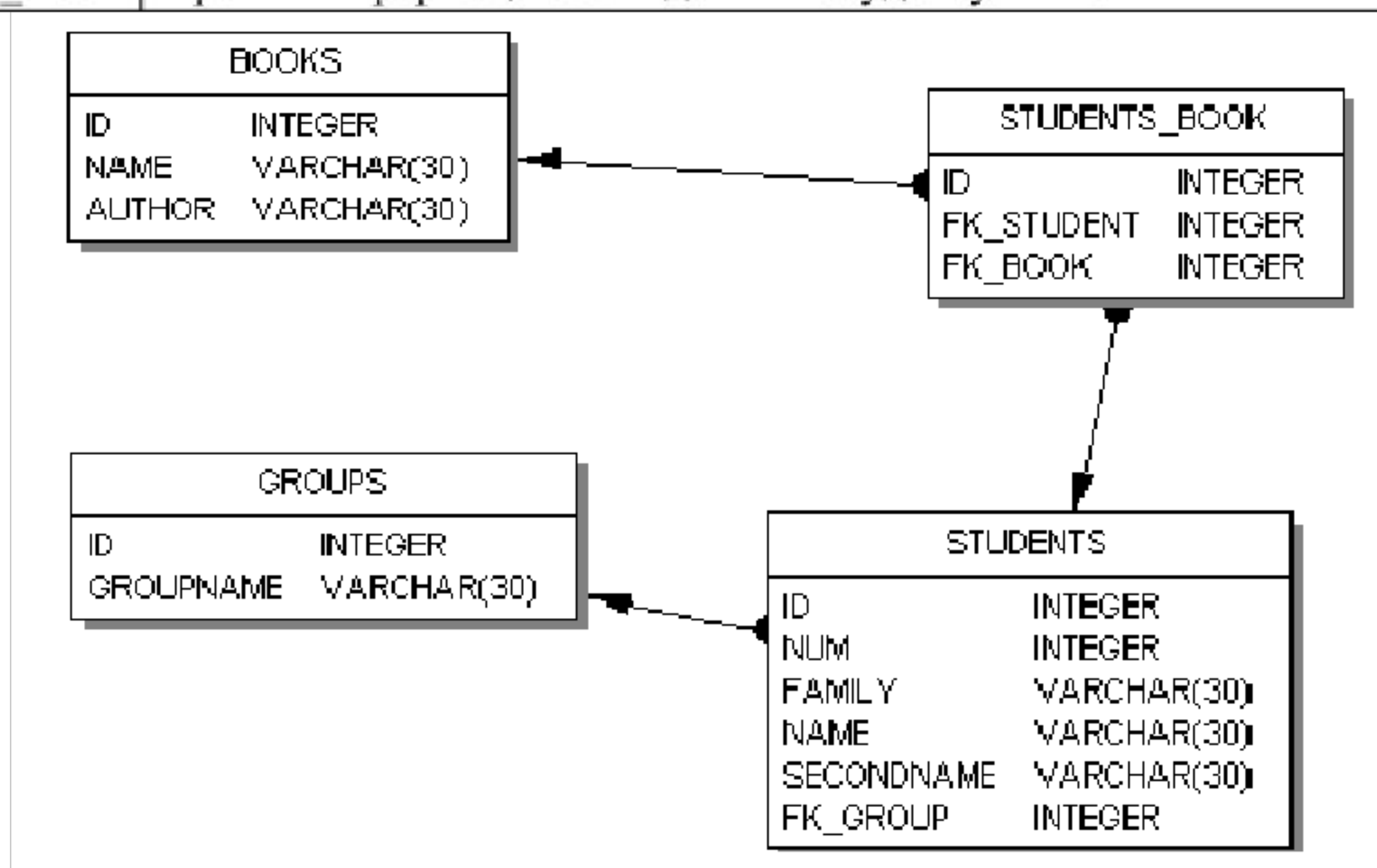


Рисунок 5.3 - Структура базы данных выдачи книг в библиотеке

Создание проекта в Borland Developer Studio

Единицей разработки в Borland Developer Studio 2006 является проект, представляющий файл, в котором хранятся ссылки на все формы и модули.

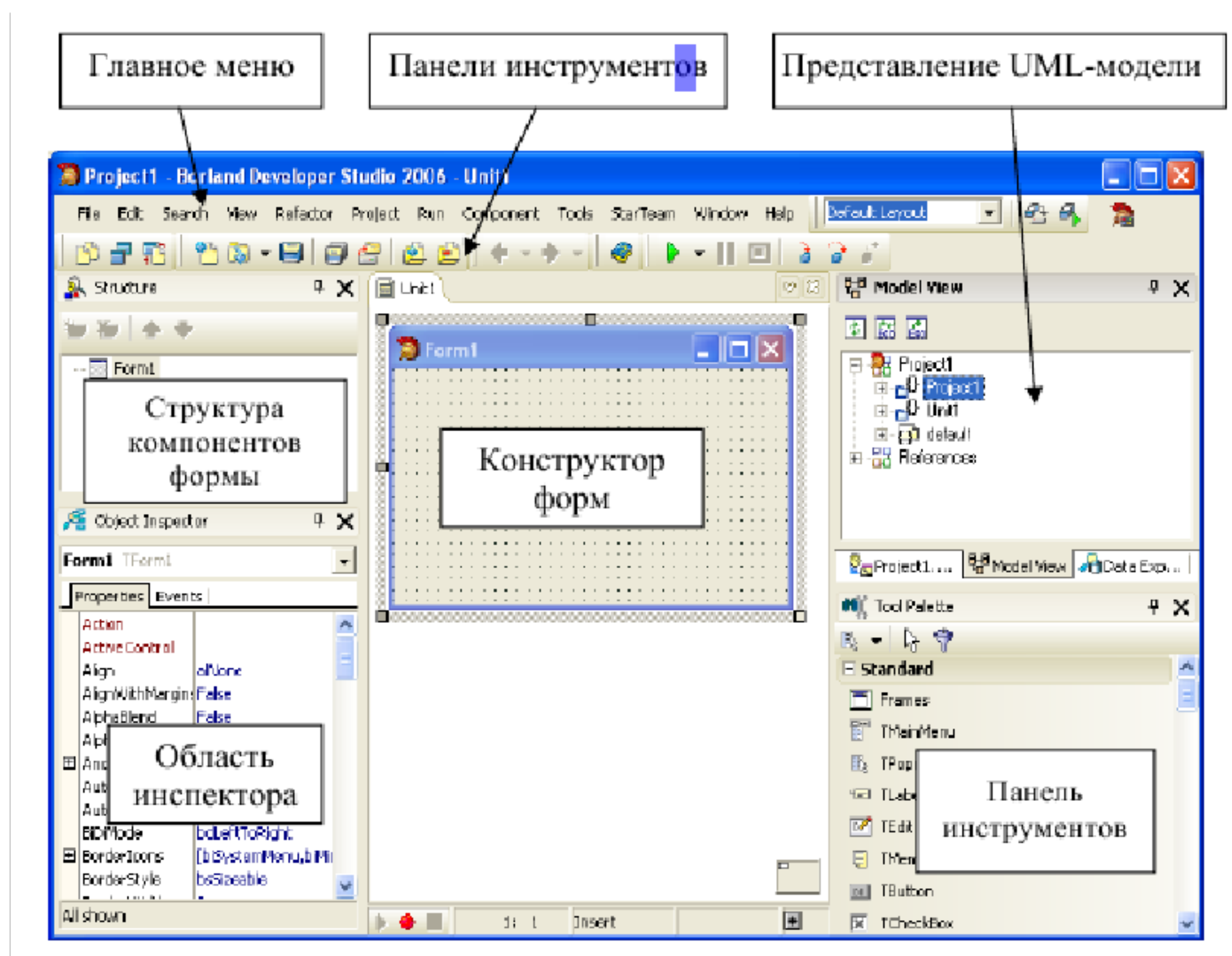


Рисунок 5.4 - Внешний вид Borland Developer Studi

Чтобы создать проект, необходимо выбрать в главном меню пункт File → New → VCL Forms Applications – Delphi. При этом будет создан проект с простым интерфейсом. Вид окна Borland Developer Studio представлен на рисунке 5.4.

Добавление нового модуля в проект

Модули представляют собой структурные единицы, из которых состоит проект. В одном модуле может содержаться несколько классов с описанием их интерфейсов и реализаций.

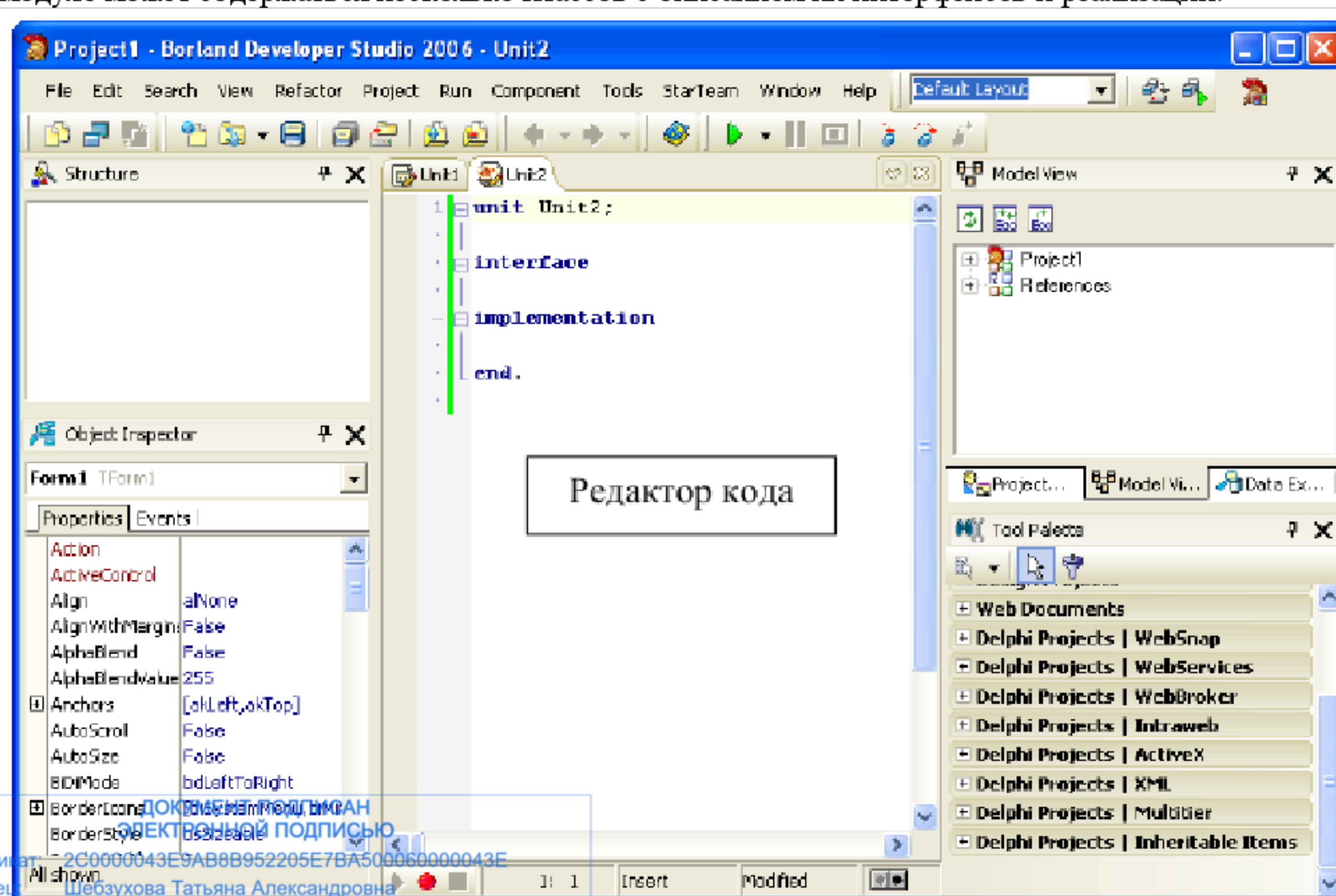


Рисунок 5.5 - Новый модуль

Добавление нового модуля осуществляется путем выбора в главном меню пункта File → New → Unit – Delphi. В правой части редактора в Дереве проекта появится новый элемент (Unit). Модификация кода выполняется в редакторе кода для выбранного модуля (рисунок 5.5).

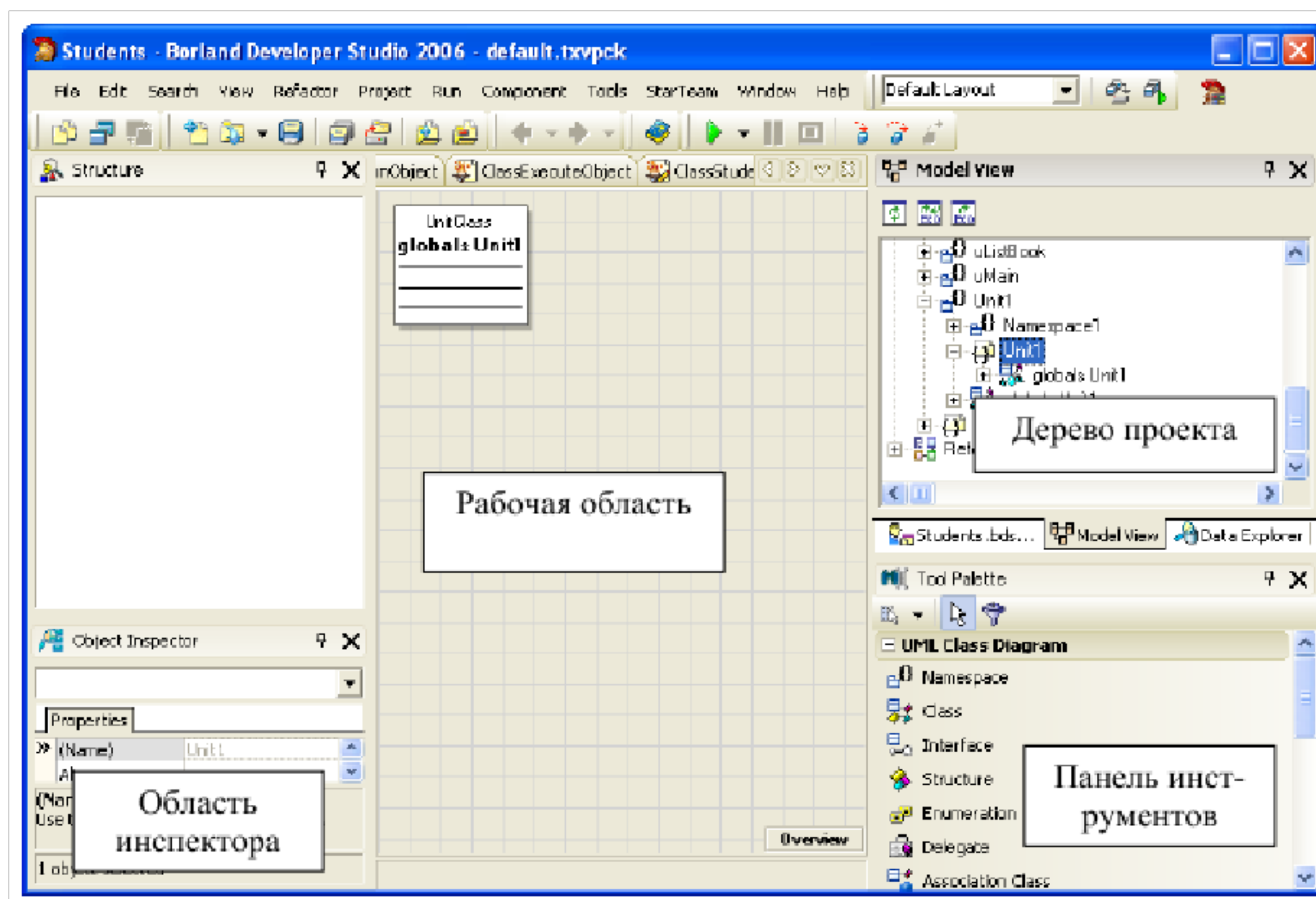


Рисунок 5.6 - Представление UML

Создание классов с помощью диаграммы UML

Для создания класса с помощью диаграммы UML переключитесь на закладку Model View и дважды щелкните по модулю, в который собираетесь добавить класс или несколько классов (рисунок 5.6). Далее на панели инструментов выберите Class и щелкните в Рабочей области. Так же на Рабочую область можно переносить классы, расположенные в других модулях проекта. Для этого необходимо выбрать класс в Дереве проекта и перетащить его на Рабочую область. Перенос других классов на Рабочую область позволяет использовать наследование и агрегацию классов.

Добавление полей. Для добавления поля в класс необходимо щелкнуть правой кнопкой мыши по классу. Появится контекстное меню, в котором нужно выбрать пункт Add и подпункт Field или нажать сочетание клавиш Ctrl + W. При этом в Рабочей области у класса, для которого вы добавили поле, появится новая строка, в которой указывается название поля и его тип. По-другому вы можете настроить свойства поля в Области инспектора.

Добавление свойств. Добавить свойство можно посредством контекстного меню и выбора пункта Add. Далее выберите подпункт Property.

Добавление процедуры. Для добавления процедуры щелкните правой кнопкой мыши по классу и в появившемся контекстном меню выберите пункт Add и подпункт Procedure. Для перехода к определению процедуры нужно выбрать элемент и вызвать контекстное меню, в котором выбирается пункт Go to definition.

В таблице 5.15 представлены свойства, которые можно настроить в Области инспектора для процедуры.

Таблица 5.15 - Свойства процедуры

Свойство	Описание
Override	Определяет, является ли эта процедура переопределенной
Params	Определяет список формальных параметров процедуры
Virtual	Определяет, является ли процедура виртуальной
Visibility	Определяет директиву видимости процедуры

Сертификат: 2043E9AB8B9C7D05E7BA50669000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Добавление функции. Для добавления функции в класс щелкните правой кнопкой мыши по классу. Появится контекстное меню, в котором нужно выбрать пункт Add и подпункт Function или нажать сочетание клавиш Ctrl + M. Свойства функции, которые можно настроить в Области инспектора представлены в таблице 5.16.

Таблица 5.16 - Свойства функции

Свойство	Описание
Override	Определяет, является ли эта функция переопределенной
Params	Определяет список формальных параметров функции
Returns	Определяет тип возвращаемого значения функции
Virtual	Определяет, является ли функция виртуальной
Visibility	Определяет директиву видимости функции

Создание отношений между классами

Ассоциация. Между двумя классами устанавливается отношение ассоциации, если между этими классами есть связь. У ассоциации указываются роли и мощности.

Для установки ассоциации между двумя классами поместите на диаграмму эти классы из Дерева проекта, выберите на панели инструментов пункт Association, щелкните мышкой по одному, а затем по второму классу.

В свойствах ассоциации по умолчанию отсутствует направленность, что подразумевает двунаправленную связь между классами. Для установки однонаправленной ассоциации нужно установить свойство Directed в положение true. Мощность и роль устанавливаются в двунаправленной ассоциации на обоих полюсах, в однонаправленной ассоциации – на одном полюсе по направлению стрелки. Для установки мощности выберите из списка значение для свойства Client Cardinality и/или для свойства Supplier Cardinality.

Для установления ролей заполните свойства Client Role и/или Supplier Role соответственно.

Агрегация. Между двумя классами устанавливается отношение агрегации, если один объект одного класса включает в себя объекты другого класса. Для установки типа ассоциации «агрегация» укажите в свойстве ассоциации Type значение Aggregation.

Наследование. Между классами устанавливается отношение наследования, если один класс является базовым (родитель), а второй – дочерним (потомком). При этом дочерний класс наследует свойства и методы родительского класса и реализует свои собственные.

Для установки отношения наследования между классами выберите на панели инструментов компонент Generalization и установите эту связь между классами.

Пример создания классов

Создание классов и отношений между ними слоя объектно-реляционного отображения

В качестве примера создания отношений между классами возьмем классы слоя объектно-реляционного отображения (рисунок 5.2).

Для создания этих классов необходимо выполнить следующие действия:

1. Создать новый модуль. Для этого в главном меню выберите File → New → Unit – Delphi.
2. Сохраните его, выбрав в главном меню File → Save или нажав сочетание клавиш Ctrl + S, указав при этом имя модуля ClassConnection.

3. Выберите в верхнем правом окне вкладку Model View. Найдите там только что созданный модуль и два раза щелкните по нему левой кнопкой мышки. Окно Borland Studio будет иметь вид, представленный на рисунке 5.6.

4. Перетащите в Рабочую область пункт Class из панели инструментов.

5. Переименуйте только что созданный класс. Для этого щелкните один раз по заголовку класса и введите новое название, либо выберите класс на Рабочей области и настройте свойства Name в Области инспектора. Назовем только что созданный класс TConnection.

6. Для добавления свойства нужно щелкнуть по классу правой кнопкой мыши. При этом появится контекстное меню, в нем выберите пункт Add, а затем пункт Property – появится новое поле. Для его переименования нужно выбрать появившееся свойство и щелкнуть по нему один раз – появится поле ввода, в котором можно указать название и тип свойства. По умолчанию поля

создаются с доступом к записи и чтению через процедуры. Если это не нужно, можно вручную отредактировать свойства в редакторе кода. Создадим свойства согласно таблице 5.5.

7. Добавьте процедуры в соответствии с таблицей 5.6. Для добавления процедуры щелкните правой кнопкой мыши по классу и в появившемся контекстном меню выберите пункт Add и подпункт Procedure.

8. Для добавления функции в классе щелкните правой кнопкой мыши по нему. Появится контекстное меню, в котором нужно выбрать пункт Add и подпункт Function или нажать сочетание клавиш Ctrl + M.

Остальные классы создаются аналогично. Далее установим наследование между классами TExecuteObject и TDBObject.

1. Для установления отношения наследования у класса TExecuteObject нужно найти класс TDBObject во вкладке Model View и перенести на диаграмму класса TExecuteObject. На диаграмме появится ссылка на класс.

2. Выберите на панели инструментов пункт Generalization.

3. Щелкните мышкой по классу TExecuteObject.

4. Щелкните мышкой по классу TDBObject.

Далее установим ассоциацию между классами TDBObject и TTransactionObject.

1. Для настройки ассоциации между классами TDBObject и TTransactionObject нужно перенести класс TDBObject на диаграмму класса TTransactionObject.

2. Выберите на панели инструментов пункт Association.

3. Щелкните по классу TDBObject.

4. Щелкните по классу TTransactionObject.

5. В Области инспектора установите у свойства Type значение Aggregation.

Создание классов слоя бизнес-логики

Структура классов бизнес-логики представлена на рисунок 5.7.

Класс TDataPrepare является базовым классом слоя бизнес-логики. Поля класса представлены в таблице 5.1, свойства класса – в таблице 5.2. Данный класс имеет виртуальные методы, приведенные в таблице 5.3.

Классы, порожденные от класса TDataPrepare. В этих классах необходимо переопределить все виртуальные методы класса TDataPrepare и создать свойства согласно таблицам 5.17 - 5.19.

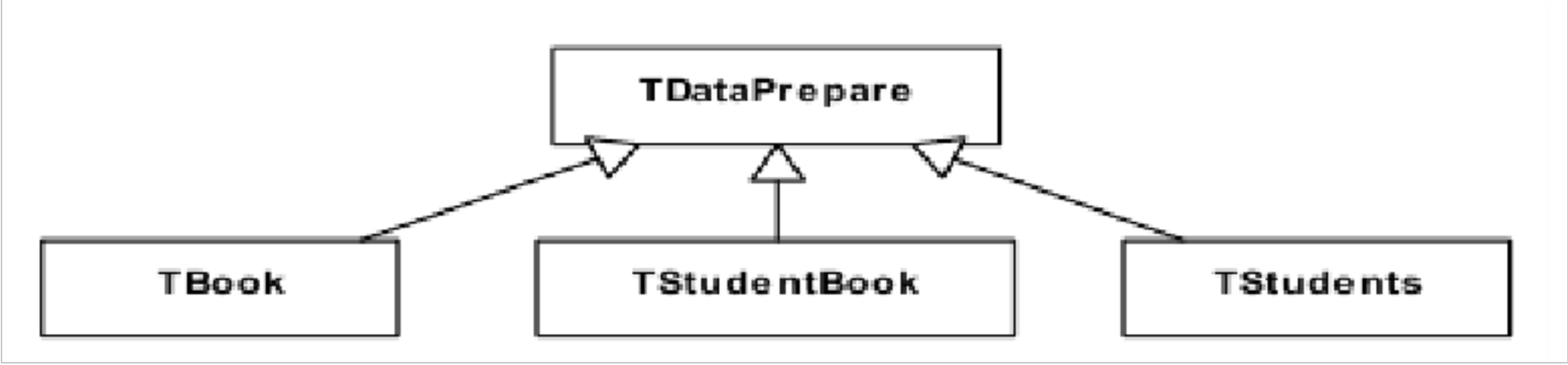


Рис. 5.7. Структура классов слоя «Бизнес-логика»

Таблица 5.17 - Свойства класса TBook

Свойство	Описание
Author: string	Автор книги
Name: string	Название книги

Таблица 5.18 - Свойства класса TStudentBook

Свойство	Описание
Fk_book: integer	Ссылка на книгу, выданную студенту
Fk_student: integer	Ссылка на студента, которому выдали книгу

Таблица 5.19 - Свойства класса TStudents

Свойство	Описание
Family: string	Фамилия студента
Name: string	Имя студента
SecondName: string	Отчество студента
Num: integer	Номер зачетной книжки студента
Group: string	Группа студента

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Юрия Владимировна
Действителен: с 19.08.2022 по 19.08.2023

Невизуальные компоненты интерфейса, используемые в примере

TImageList предназначен для хранения и последующего использования различных изображений.

Для использования этого компонента поместите его на форму и двойным щелчком мыши запустите Мастер создания изображений (рисунок 5.8).

В Мастере создания изображений можно добавлять, редактировать удалять изображения формата иконок.

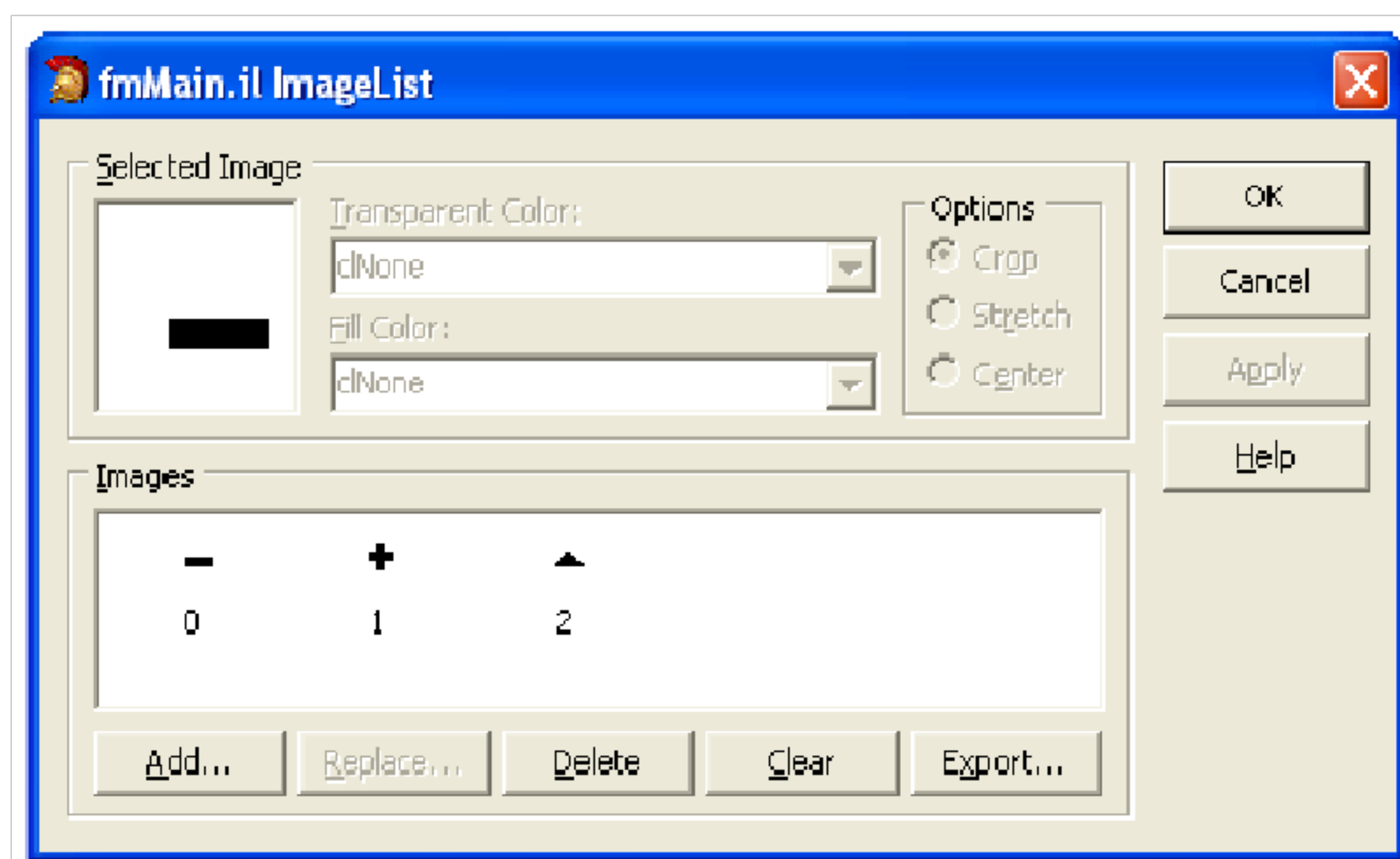


Рисунок 5.8 - Мастер создания изображений в TImageList

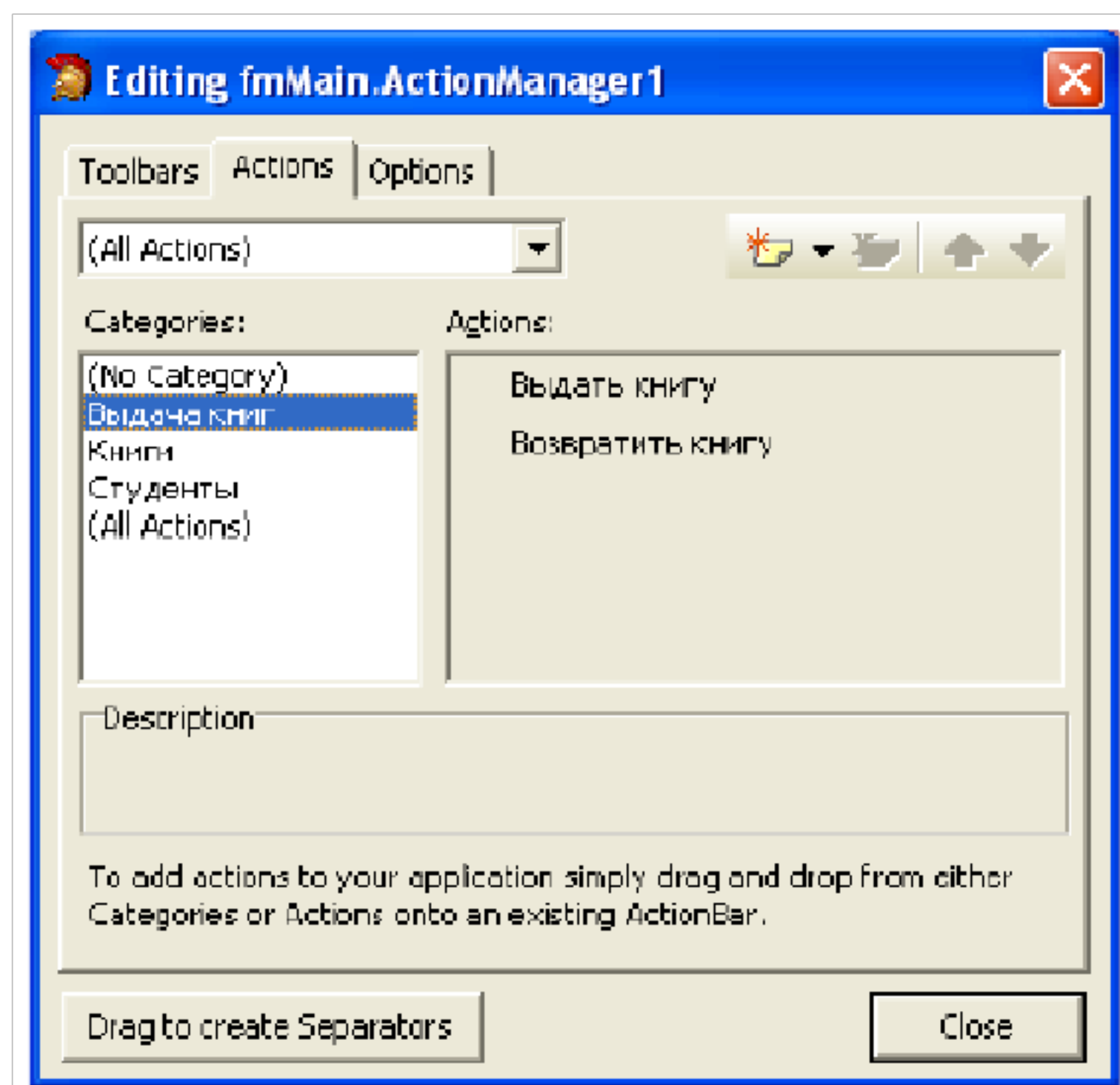


Рисунок 5.9 - Окно настройки действий компонента TActionManager

ТActionManager - менеджер действий. Применяется для управления действиями, повторно используемыми в проекте. Для установки изображений, отображаемых в действиях, нужно настроить свойство Images. Для этого щелкните два раза по компоненту (рисунок 5.9). Для добавления действия нажмите клавишу Ins. В списке Actions появится новое действие. Для настройки свойств и событий действия используйте Область инспектора. Свойства, которые

можно настраивать у действия, представлены в таблице 5.20. События действий описаны в таблице 5.21.

Таблица 5.20 - Свойства действий

Свойство	Описание
Caption	Отображаемое название действия
Name	Имя действия
ImageIndex	Номер рисунка, ассоциируемого с действием
Visible	Видимость действия

Таблица 5.21 - События действий

Событие	Описание
OnExecute	Событие, происходящее при выполнении действия
OnHint	Событие, происходящее при отображении всплывающей подсказки

Визуальные компоненты, используемые в примере

TBitBtn представляет собой кнопку с изображением. Свойства компонента представлены в таблице 5.22, событие компонента – в таблице 5.23.

TDBGrid – компонент, отображающий сетку строк. Предназначен для отображения набора данных. События TDBGrid представлены в таблице 5.24.

Таблица 5.22 - Свойства компонента TBitBtn

Свойства	Описание
Caption	Надпись, отображаемая на кнопке
ModalResult	Значение, которое записывается в результат отображения модальной формы
Glyph	Изображение, отображаемое на кнопке

Таблица 5.23 - Событие компонента TBitBtn

Событие	Описание
OnClick	Событие, происходящее по нажатию на кнопку

Таблица 5.24 - События компонента TDBGrid

Событие	Описание
Align	Выравнивание компонента по форме
Columns	Настройка визуальных свойств (название, шрифт и т. д.) полей, отображаемых в сетке строк
DataSource	Источник данных для отображения данных в сетке таблицы

Таблица 5.25 - Свойства компонента TLabel

Свойство	Описание
Caption	Отображаемая надпись
Font	Шрифт отображаемой надписи
WordWrap	Перенос по слогам

Таблица 5.26 - Свойства компонента TEdit

Свойство	Описание
Text	Отображаемый текст

Таблица 5.27 - События компонента TEdit

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебаукова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023

Событие	Описание
OnChange	Изменение надписи в компоненте
OnClick	Щелчок левой кнопкой мышки по компоненту

Таблица 5.28 - Свойства компонента TComboBox

Свойство	Описание
Text	Отображаемое значение
Items	Содержит значения выпадающего списка
ItemIndex	Номер отображаемого значения из выпадающего списка
Style	Стиль отображения значения выпадающего списка

Таблица 5.29 - События компонента TComboBox

Событие	Описание
OnChange	Изменение надписи в компоненте
OnClick	Щелчок левой кнопкой мышки по компоненту
OnSelect	Выбор значения из выпадающего списка

Таблица 5.30 - Свойства компонента TPageControl

Свойство	Описание
Align	Способ расположения компонента на форме
ActivePage	Свойство определяет текущую страницу
ActivePageIndex	Свойство определяет номер текущей страницы
Caption	Название листа

Таблица 5.31 - Событие компонента TPageControl

Событие	Описание
OnChange	Изменение текущей страницы

TLabel – компонент, отображающий нередактируемую строку. Свойства TLabel представлены в таблице 5.25.

TEdit – компонент, отображающий редактируемую надпись. Свойство TEdit представлено в таблице 5.26, события данного компонента приведены в таблице 5.27.

TComboBox – компонент, представляющий собой комбинированный выпадающий список. Свойства компонента представлены в таблице 5.28, его события – в таблице 5.29.

TPageControl – компонент, содержащий список страницы и закладок, с помощью которых можно переходить с одной страницы на другую. После добавления компонента на форму для добавления новой страницы нужно вызвать контекстное меню, в котором выбрать пункт New Page. Свойства компонента TPageControl представлены в таблице 5.30, событие компонента TPageControl приведены в табл. 5.31.

TPanel – компонент, предназначенный для организации расположения других визуальных компонентов на форме. Выравнивание компонента TPanel настраивается в свойстве Align.

Разработка интерфейса

Добавление новой формы осуществляется путем выбора в главном меню пункта File → New → Form – Delphi. При этом будет создана новая форма, которая появится на экране. При добавлении новой формы в проект она встает в список автоматически создаваемых форм.

Для удаления формы из этого списка нужно в главном меню выбрать пункт Project → Options. Появится диалог, представленный на рисунке 5.10. Для удаления формы из списка автоматически создаваемых форм нужно выбрать ее в списке Auto-create form и нажать кнопку >.

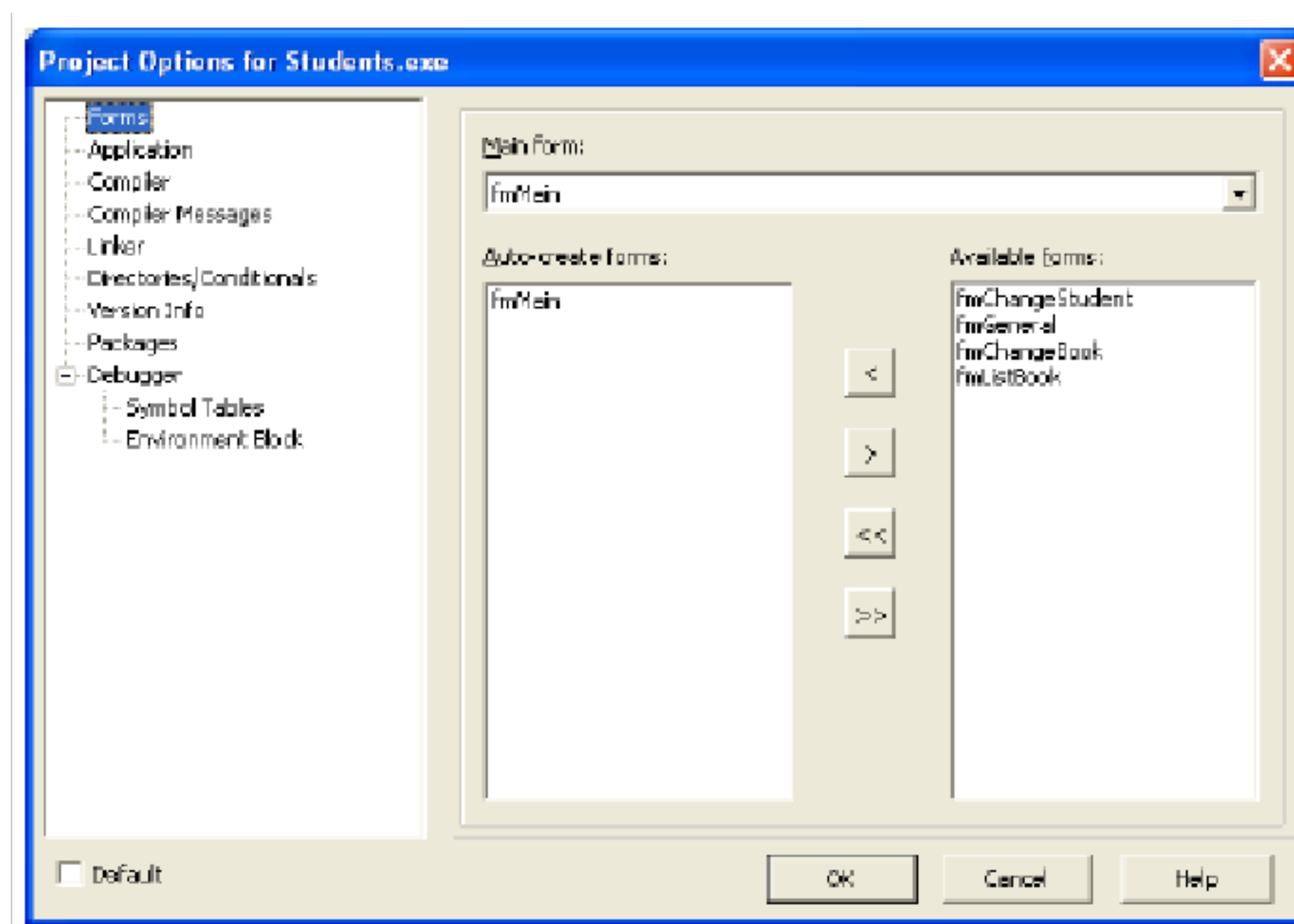


Рисунок 5.10 - Окно свойства проекта

Пример создания интерфейса

Создадим интерфейс, обладающий следующей функциональностью:

- редактирование списка студентов;
- редактирование списка книг;
- выдача книг студентам.

Для этого создадим четыре формы.

Главная форма

Главная форма является основной рабочей формой всей программы.

Пример интерфейса представлен на рисунок 5.11 и рисунок 5.12.

Для создания главной формы выполните следующие шаги:

1. Поместите на форму компонент `TPageControl`, настройте у него свойство `Align` равным `alClient` – компонент займет все свободное место на форме.
2. Щелкните правой клавишей мышки по компоненту `TPageControl`. Появится контекстное меню, в котором выберите пункт `New Page`. Будет создана новая страница.
3. Повторите действие п. 2 для создания еще одной страницы.
4. Добавьте на форму компонент `TDBGrid`, настройте у него свойство `Align` равным `alTop` – компонент займет верхнюю часть страницы. Так же необходимо настроить свойство `Height` – высота компонента.
5. Щелкните два раза по компоненту `TDBGrid` – появится новое окно (рисунок 5.13).
6. Для добавления поля нужно вызвать контекстное меню, в котором следует выбрать пункт `Add` или нажать кнопку `Ins`.
7. Выберите созданное поле и в Области инспектора настройте свойства `Caption` (название столбца) и `Fieldname` (название отображаемого поля).
8. Повторите п. 6–7 столько раз, сколько необходимо создать полей.
9. Добавьте на форму компонент `TPanel`, настройте свойство `Align` равным `alTop` – компонент займет место сразу за компонентом `TDBGrid`. Очистите свойство `Caption` у компонента `TPanel`.
10. Добавьте на форму компонент `TPanel`, настройте свойство `Align` равным `alBottom` – компонент займет место внизу страницы. Очистите свойство `Caption` у компонента `TPanel`.
11. Добавьте еще один компонент `TDBGrid`, настройте свойство `Align` равным `alClient` – компонент займет все свободное место на форме.
12. Настройте у компонента, добавленного в п. 11, столбцы так же, как описано в п. 5–8.

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 2C0000043E9AB8B952205E7BA500060000043E
Владелец: Шебзухова Татьяна Александровна

Действителен: с 19.08.2022 по 19.08.2023