|  |  |
| --- | --- |
| logo ГУМРФ-чб | Федеральное агентство морского и речного транспорта  Федеральное государственное бюджетное образовательное  учреждение высшего образования  **ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  **МОРСКОГО И РЕЧНОГО ФЛОТА**  **имени адмирала С. О. МАКАРОВА** |

**Институт ВОДНОГО ТРАНСПОРТА**

***Кафедра вычислительных систем и информатики***

А. Р. Балса

Методическое обеспечение курсовых

и лабораторных работ дисциплины

«Методы и средства проектирования информационных систем и технологий»

Санкт-Петербург

Изд-во ГУМРФ имени адмирала С. О. Макарова

2017

**УДК**

**ББК**

Рецензент:

КафедраВычислительных систем и информатики Института Водного транспорта, доктор технических наук, профессор А.В. Макшанов

**Балса А.Р.** Методическое обеспечение курсовых и лабораторных работ дисциплины «Методы и средства проектирования информационных систем и технологий»: — СПб.: Изд-во ГУМРФ им. адм. С. О. Макарова, 2017. — 58 с.

В настоящем методическом обеспечении курсовых и лабораторных работ дисциплины «Методы и средства проектирования информационных систем и технологий» раскрываются академические подходы к принципам объектно-ориентированного анализа, проектирования и программирования. Представлены типовые задачи по программированию и варианты индивидуальных заданий. Структура пособия ориентирована на использование в качестве методического указания, содержит порядок выполнения графической и расчетных частей, порядок оформления пояснительной записки.

Данное методическое обеспечение предназначено студентам всех форм обучения по специальности 09.03.02 «Информационные системы и технологии» для практического закрепления основных вопросов дисциплины «Методы и средства проектирования информационных систем и технологий».

|  |  |
| --- | --- |
|  | ©ФГБОУ ВО «ГУМРФ имени адмирала С.О. Макарова», 2016  © Балса А. Р, 2017 |

**Оглавление**

[ВВЕДЕНИЕ 3](#_Toc475036508)

[1. МЕТОДИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ НАД ЗАДАНИЕМ 8](#_Toc475036509)

[2. ПОРЯДОК ОФОРМЛЕНИЯ И ЗАЩИТЫ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ 10](#_Toc475036510)

[3. ТЕМАТИКА КУРСОВЫХ РАБОТ 14](#_Toc475036511)

[3.1 Варианты заданий 20](#_Toc475036514)

[4 УКАЗАНИЯ ПО ОФОРМЛЕНИЮ ЛАБОРАТОРНЫХ РАБОТ 53](#_Toc475036563)

[ПРИЛОЖЕНИЕ 1 56](#_Toc475036564)

[ПРИЛОЖЕНИЕ 2 57](#_Toc475036565)

# ВВЕДЕНИЕ

Целью курса «Методы и средства проектирования информационных систем и технологий» является изучение парадигм и методов программирования, ознакомление с методами разработки программ и основными приемами программирования на языках высокого уровня с помощью инструментов визуальной разработки программ для овладения знаниями в области программирования подсистем информационной безопасности объектов.

Задачей учебной дисциплины является формирование практических навыков разработки алгоритмов, подбора адекватных задаче структур данных по обеспечению информационной безопасности и их реализации на современных программных средствах.

Изучение дисциплины базируется на навыках и знаниях, полученных студентами в курсах «Информатика», «Основы информационной безопасности», «Аппаратные средства вычислительной техники», «Языки программирования».

Дисциплина «Технологии и методы программирования» дает знания, навыки и инструмент, которыми студенты должны пользоваться при выполнении курсовых работ, дипломного проекта, при освоении других дисциплин, а также в рамках профессиональной деятельности, в том числе специализированной на создание программного обеспечения и при работе в специализированных НИИ и фирмах.

В соответствии с требованиями ФГОС ВО по направлениям подготовки 10.03.01«Технологии и методы программирования» (уровень академического и прикладного бакалавриата третьего поколения соответственно) у выпускников должны быть сформированы общекультурные, общепрофессиональные и профессиональные компетенции, что достигается освоением ими программы бакалавриата и соответствует компетентностному подходу в организации образовательного процесса.

Таким образом, в рамках дисциплины «Технологии и методы программирования» Выпускник должен обладать следующими профессиональными компетенциями (ПК):

* способностью понимать сущность и значение информации в развитии современного общества, применять достижения информатики и вычислительной техники, перерабатывать большие объемы информации проводить целенаправленный поиск в различных источниках информации по профилю деятельности, в том числе в глобальных компьютерных системах (ПК-2);
* способностью формировать комплекс мер по информационной безопасности с учетом его правовой обоснованности, административно-управленческой и технической реализуемости и экономической целесообразности (ПК-4);
* способностью организовывать и поддерживать выполнение комплекса мер по информационной безопасности, управлять процессом их реализации с учетом решаемых задач и организационной структуры объекта защиты, внешних воздействий, вероятных угроз и уровня развития технологий защиты информации (ПК-5);
* способностью оформить рабочую техническую документацию с учетом действующих нормативных и методических документов в области информационной безопасности (ПК-14);
* способностью применять программные средства системного, прикладного и специального назначения (ПК-15);
* способностью использовать инструментальные средства и системы программирования для решения профессиональных задач (ПК-16);
* способностью к программной реализации алгоритмов решения типовых задач обеспечения информационной безопасности (ПК-17);
* способностью собрать и провести анализ исходных данных для проектирования подсистем и средств обеспечения информационной безопасности (ПК-18);
* способностью составить обзор по вопросам обеспечения информационной безопасности по профилю своей деятельности (ПК-19);
* способностью проводить анализ информационной безопасности объектов и систем с использованием отечественных и зарубежных стандартов (ПК-21);

Кроме того в результате освоения дисциплины «Технологии и методы программирования» студент должен:

* Знать: один из языков программирования, структуру локальных и глобальных компьютерных сетей;
* Уметь: работать в качестве пользователя персонального компьютера, использовать внешние носители информации для обмена данными между машинами, создавать резервные копии, архивы данных и программ, использовать языки и системы программирования для решения профессиональных задач, работать с программными средствами общего назначения.
* Владеть: методами поиска и обмена информацией в глобальных и локальных компьютерных сетях, техническими и программными средствами защиты информации при работе с компьютерными системами, включая приемы антивирусной защиты.

Для освоения дисциплины ««Технологии и методы программирования»» будет рассмотрен объектно-ориентированный подход к анализу, проектированию и программированию для моделирования сложных процессов и систем.

Объектно-ориентированный подход (ООП) - это метод, при использовании которого главными элементами моделируемой системы являются объекты. В языках программирования и визуального моделирования понятие объекта реализовано как совокупность свойств (структур данных, характерных для данного объекта), методов(операций) их обработки и событий, на которые данный объект может реагировать и, которые приводят, как правило, к изменению свойств объекта. Объединение данных и свойственных им процедур обработки в одном объекте, называется инкапсуляцией и является одним из важнейших принципов ООП.

Другим фундаментальным понятием является класс.

Класс - это шаблон, на основе которого может быть создан конкретный программный объект, он описывает свойства и методы, определяющие поведение объектов этого класса. Каждый конкретный объект, имеющий структуру этого класса, называется экземпляром класса.

Следующими важнейшими принципами ООП являются наследование и полиморфизм.

Наследование - такая организация классов, которая предусматривает создание новых классов на базе существующих и позволяет классу потомку иметь (наследовать) все свойства класса – родителя.

Полиморфизм - означает, что рожденные объекты обладают информацией о том, какие методы они должны использовать в зависимости от того, в каком месте цепочки (дерева классов) они находятся, иными словами это концепция, реализующая "множество методов в одном интерфейсе".

Другим важнейшим принципом ООП является модульность.

Модульность - это такая организация объектов, когда они заключают в себе полное определение их характеристик, никакие определения методов и свойств не должны располагаться вне его, это делает возможным свободное копирование и внедрение одного объекта в другие.

Развитием объектно-орниентированной парадигмы (методологии), стала объектно-событийная парадигма, опирающаяся на понятия объекта и события. Эта парадигма позволяет конструировать, программировать распределенные вычислительные среды, в том числе среды реального времени, [SCADA](http://bourabai.ru/dbt/scada.htm), среды визуального проектирования(Rational Rose) и пр.

Современными языками объектно-ориентированного программирования являются [С++](http://bourabai.ru/C-Builder/cpp/index.htm) и [Java](http://bourabai.ru/alg/classification04.htm#java), а объектно-ориентировванного визуального или графического моделирования – UML(Unified Modeling Language). С середины 90-х годов многие объектно–ориентированные языки реализуются как [системы визуального программирования](http://bourabai.ru/einf/4gl.htm), в которых интерфейсная часть программного продукта создается в диалоговом режиме, практически без написания программных операторов. К объектно – ориентированным системам визуального проектирования относятся [Visual Basic](http://bourabai.ru/einf/vb5/index.htm), [Delphi](http://bourabai.ru/einf/Delphi/index.htm), [C++ Builder](http://bourabai.ru/C-Builder/index.htm), Visual C++. Язык VBA (Visual Basic for Applications) – язык приложений Microsoft Office ([Excel](http://bourabai.ru/einf/Glava6.htm), [Word](http://bourabai.ru/einf/Glava53.htm), [Access](http://bourabai.ru/einf/subd2.htm), [Power Point](http://bourabai.ru/einf/Glava82.htm) и др). [VBA](http://bourabai.ru/einf/vba/index.htm) соблюдает основной синтаксис языка и правила программирования [языков Basic](http://bourabai.ru/alg/basic/index.htm) – диалектов, позволяет создавать макросы для автоматизации выполнения некоторых операций и графический интерфейс пользователя, интеграцию между различными программными продуктами.

Основной задачей курса является подготовка специалистов, владеющих современными методами и средствами разработки алгоритмов и программ, знающих современную [технологию для анализа, проектирования и программирования](http://bourabai.ru/alg/technology.htm) и умеющих применять ее при решении сложных прикладных задач.

# МЕТОДИКА САМОСТОЯТЕЛЬНОЙ РАБОТЫ НАД ЗАДАНИЕМ

Выполнение курсовой работы предполагает самостоятельную практическую работу студента в соответствии с индивидуальным заданием, выдаваемым преподавателем. В ходе работы студент должен разработать программную информационно-поисковую систему (ИПС) для заданной предметной области.

Ниже приводится последовательность этапов выполнения работы, которая предполагает итеративный характер и допускает возврат к любому предыдущему этапу на любом шаге с целью изменения или доработки сформированных ранее решений:

* Анализ поставленной задачи, целей разработки системы, функциональных и иных требований к ней.
* Формирование плана работы над проектом.
* Формализация и документирование требований к разрабатываемой системе.
* Разрешение различных вопросов и неоднозначностей.
* Выработка, формализация и документирование проектных решений, представляющих архитектуру и структуру будущей системы, модель данных (концептуальная, логическая и физическая), алгоритмы обработки и передачи информации, принципы организации пользовательского интерфейса, средства обеспечения требуемого уровня надёжности и т.д.
* Выбор и, при необходимости, изучение инструментария разработки.
* Реализация сформированных проектных решений: программирование отдельных составляющих системы и их интеграция в единое целое.
* Подготовка необходимых тестов и данных для их проведения. Тестирование системы на предмет соответствия требованиям и на наличие ошибок.
* Отладка системы и устранение найденных в ходе тестирования ошибок и несоответствий имеющимся требованиям.
* Подготовка отчетной документации по работе над проектом (пояснительной записки к курсовой работе).

Результаты каждого этапа работ, кроме последнего, следует документировать. Соответствующие материалы должны служить базисом при подготовке пояснительной записки к курсовой работе. При возникновении в ходе работы вопросов, которые не могут быть разрешены самостоятельно, следует проконсультироваться с преподавателем.

Теоретическая часть курсовой работы предполагает самостоятельное изучение студентами следующих вопросов:

* стратегии и принципы проектирования программных систем;
* виды архитектур программных систем;
* этапы проектирования ПО при структурном подходе;
* спецификация ПО и их виды (структурные и функциональные схемы, диаграммы потоков данных, диаграммы сущность связь и др.);
* типы пользовательских интерфейсов и этапы их разработки;
* структурное, функциональное и оценочное тестирование ПО.

Необходимые материалы могут быть найдены, например, в источниках, указанных в списке использованных источников.

Графическая часть работы подразумевает подготовку графических схем спецификации разрабатываемого ПО, которые включаются в пояснительную записку к курсовой работе. Графические схемы должны быть выполнены в электронном виде с использованием программы, поддерживающей инструментарий для их построения.

# ПОРЯДОК ОФОРМЛЕНИЯ И ЗАЩИТЫ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОЙ РАБОТЕ

Пояснительная записка к курсовой работе должна включать в себя:

* Титульный лист.
* Содержание (на отдельном листе с указанием номеров разделов записки).
* Введение.
* Модели и спецификации программного обеспечения.
* Руководство программиста.
* Руководство пользователя.
* Заключение.
* Использованные материалы (при необходимости).
* Список литературы.
* Приложение.

Вид *титульного листа* приведен в конце документа.

*Введение*

В данном разделе необходимо изложить постановку задачи, отметить цели работы, перечислить требования к программе, а также указать её функции. Кроме того, могут быть представлены материалы различного характера, отражающие те или иные аспекты и решения процесса разработки, в том числе указаны (возможно с обоснованием) выбранные методы и инструментальные средства. В данной части записки следует описать проблемы, возникшие в ходе выполнения работы, и обосновать предпочтения одних способов решения возникших трудностей другим. Также во введении может быть приведено неформальное описание наиболее сложных и интересных, на взгляд выполняющего работу, из использованных алгоритмов.

*Модели и спецификации программного обеспечения*

Данная часть записки должна содержать описание различных моделей и схем, отражающих структуру программы, схемы взаимодействия её компонентов, использованные алгоритмы, структуры и файлы данных и т.д. Для оформления могут использоваться любые из известных наглядных средств и диаграмм, например, блок-схемы, схемы Насси-Шнейдермана, структурные и функциональные схемы, диаграммы Варнье-Орра, диаграммы UML и т.д.

Если работа связана с базой данных, то должно быть приведено описание её логической модели: выделенные сущности, их состав (включая пояснения назначение отдельных атрибутов, их типы и длины, ключевые атрибуты) и отношения между ними. Модель может быть представлена графически.

*Руководство программиста*

В данном разделе записки должно быть представлено назначение и краткое описание составляющих программу компонентов: модулей, подпрограмм, классов, таблиц и файлов данных, используемых сторонних библиотек, не входящих в стандартную поставку средства разработки и т.д.

Для каждого модуля\подпрограммы следует привести: название, тип (для подпрограмм), назначение (в том числе выполняемые функции), список входных и выходных параметров (если есть), связи и зависимости от других модулей\подпрограмм, ограничения и особенности использования (если есть). Для модулей может быть приведено описание введенных типов и глобальных переменных.

Для классов следует представить их структуру, а именно поля, методы, свойства и обработчики событий. Для новых классов следует описать наиболее значимые компоненты, для порожденных от стандартных классов – только те, которые были введены в процессе разработки. Методы следует описывать как подпрограммы. Для полей, свойств глобальных переменных и параметров подпрограмм должны быть представлены их название, назначение, тип и возможные особенности и ограничения по использованию. Для параметров, кроме то!о, следует указать способ передачи в подпрограмму: по значению, по результату, по значению и результату и т.д. Описание сложных типов (если они есть) должно быть приведено отдельно.

Если программа предназначена для работы базой данных, то должно быть приведено описание её структуры (таблиц). Для каждой таблицы должны быть указаны название, назначение, список полей, ключи, индексы и т.д. Поля следует описывать так же, как переменные. Для файлов данных также должна быть приведена их структура.

Для сторонних библиотек должны быть приведены их название и назначение, а также список использованных подпрограмм\объектов с указанием их назначения и формальных параметров (для подпрограмм).

*Руководство пользователя*

В данной части следует представить материалы, касающимся способов и особенностей использования разработанной программы. Приводимые сведения должны содержать пояснения касательно пользовательского интерфейса и рисунки рабочих форм (окон) программы. Количества скриншотов следует ограничить разумными пределами.

*Заключение*

Следует отметить полученные в ходе работы результаты и подчеркнуть особенности и достоинства созданной программы. Должны быть указаны существующие ограничения, возможные варианты их устранения, а также перспективы\направления развития и улучшения программы.

*Список использованных источников*

Список литературы должен содержать не менее трех источников с указанием авторов, названия, издательства, года издания, количества страниц.

*Приложение*

Включает в себя листинг программы и, при необходимости, другие материалы. Листинг должен содержать наиболее интересные с точки зрения автора фрагменты программы, например, реализации представленных в основной части записки алгоритмов, моделей и схем. В случае небольшой программы её листинг может быть приведен полностью. Служебный код, автоматически сгенерированный средой разработки и неизменяемый разработчиком, по возможности следует исключить из листинга.

Пояснительная записка должна быть оформлена в соответствии со следующими требованиями. Шрифт текста абзацев Times New Roman, 14 пт., отступ «красной строчки» 1,25 см. Выравнивание текста по ширине; межстрочный интервал 1,5. Заголовки разделов в полужирном начертании. Для листинга программы использовать шрифт Courier (New), 10 пт., выравнивание по левому краю, межстрочный интервал 1. Страницы отчета, за исключением титульного листа, должны быть пронумерованы. Номера должны быть указаны в правом нижнем углу. Титульный лист считается 1-ой страницей. Поля страниц: верхнее и нижнее – 2 см., левое – 3 см., правое – 1,5 см. Страницы отчета должны быть скреплены с левой стороны как минимум двумя скобами (например, с помощью стиплера) или в случае большого объёма прошиты таким же образом нитками.

Курсовая работа выполняется студентом в течение семестра в свободное от основной учёбы время, начиная с момента получения задания. Промежуточные результаты работы не менее чем раз в месяц демонстрируются преподавателю для контроля и оценки выполняемого процесса разработки. Защита курсовой работы проводится на последних неделях семестра и в качестве предварительного этапа включает в себя демонстрацию преподавателю работоспособности созданной программной системы и её функций. Защита возможна только в том случае, если программа полностью готова и удовлетворяет всем сформулированным в задании требованиям, а также при наличии пояснительной записки, включающей все необходимые пункты содержания. На защите студент должен представить преподавателю пояснительную записку и выступить с кратким (около 5 минут) докладом о проделанной работе и полученных в ходе неё результатах, иллюстрируя наиболее значимые положение материалов пояснительной записки. Студент должен быть готов к вопросам и дискуссии относительно различных аспектов и особенностей разработанной им системы.

# **ТЕМАТИКА ЛАБОРАТОРНЫХ И КУРСОВЫХ РАБОТ**

В ходе курсовой работы студент должен разобраться ИПС для заданной предметной области, удовлетворяющую представленным требованиям. В предметной области в соответствии с задачей выделяется несколько информационных сущностей, каждой из которых ставится в соответствии одна или несколько таблиц данных, образующих базу данных, с которой работает система. Разрабатываемая ИПС должна обеспечивать реализацию следующих базовых функций и требований:

* вывод на экран содержимого каждой из таблиц данных с возможностью перемещения по отдельным записям в прямом и обратном направлении, а также индикацией номера текущей записи и общего их количества;
* быстрый переход к записи с указанным порядковым номером;
* добавление новой записи в таблицу;
* редактирование или удаление произвольной выбранной пользователем записи;
* запрос на подтверждение изменений при добавлении, редактирование или удалении данных;
* контроль правильности ввода данных в соответствии с их типами и форматами (определяются условиями задачи или исходя из рациональных соображений);
* контроль непротиворечивости и целостности структур данных в соответствии с условиями задачи;
* главное управляющее меню, обеспечивающее быстрый доступ к основным функциям программы.

В зависимости от спецификации задания ИПС должна обеспечивать сортировку, фильтрацию и поиск данных по заданным признакам и критериям, а также реализовывать требуемую обработку данных и выдачу информации по определенным запросам. Типовые задания:

* Простейшая видеоигра. Программа содержит два спрайта в виде человечков, один из которых убегает, другой догоняет. Игра выполняется на игровом поле, имеющем некоторые препятствия. Одним спрайтом управляет компьютер, другим «игрок». При соприкосновении убегающий и догоняющий меняются ролями. Победитель тот, который дольше убегал. Перемещение по горизонтали, вертикали и диагоналям.

## Табло на станции метро:

Требуется разработать средствами Rational Rose модель программного обеспечения табло для информационной службы метрополитена.

Табло расположены на каждой станции метро. Они работают под управлением единого пункта управления (ПУ) информационной службы метро. Табло отображает текущее время (часы, минуты, секунды) и время, прошедшее с момента отправления последнего поезда (минуты, секунды). Момент прибытия и отправления поезда определяется при помощи датчиков, устанавливаемых на путях. Все табло метро синхронизованы, текущее время отсчитывается и устанавливается из центральной службы времени, находящейся на ПУ. На табло высвечивается конечная станция назначения прибывающего поезда. Эти данные содержатся в расписании движения поездов, которое хранится в памяти табло и периодически обновляется с ПУ.В «бегущей строке» табло отображается рекламная информация. Память табло хранит до 10 рекламных сообщений. Сообщения отображаются друг за другом с небольшими паузами, циклически. Содержание рекламных сообщений поступает с ПУ. Дополнительная функция табло – по запросу с ПУ оно пересылает данные о нарушениях расписания (преждевременных отправлениях поездов или опозданиях). В ходе выполнения задания должна быть создана схема базы данных для хранения рекламных сообщений, расписания и сведений о нарушении расписаний. Пояснение: в задании требуется разработать модель ПО только для табло, но не для пункта управления информационной службы.

По индивидуальной договорённости со студентом тема курсовой работы может быть изменена на разработку программной системы другого прикладного назначения. Например:

* Разработать графическую оболочку для заданного архива;
* Разработать программу, расширяющую возможности стандартного буфера обмена Windows, а именно: отслеживание и сохранение элементов, помещаемых в буфер, с возможностью их классификации и последующей вставки в любое приложение, поддерживающее буфер обмена, в том числе с использованием горячих клавиш;
* Разработать программу построения графиков элементарных математических функций, задаваемых пользователем аналитически или таблично с возможностью увеличения и уменьшение исходного масштаба отображения, изменения области определения и сохраняя графика в файле формата bmp/jpg/gif или другом.
  1. Задание для курсовой работы

# Требуется разработать ИПС для работы с базой данных (БД) некоторой предметной области. БД включается в себя несколько взаимосвязанных таблиц, организованных в виде файлов, в которых хранятся требуемые сведения. Взаимосвязь между двумя таблицами в данном случае означает, что значения некоторого поля другой таблицы. Для каждой таблицы (и, соответственно, информационного объекта, подлежащего учету) приведен минимальный набор атрибутов-полей, которые она должна включать. Этот набор может быть дополнен и уточнен студентом для организации взаимосвязей между таблицами, поддержки целостности БД, сохранения необходимых сведений и реализации функций системы. Разрабатываемая ИПС должна обеспечивать реализацию следующих базовых функций и требований:

* вывод на экран содержимого каждой из таблиц данных с возможностью перемещения по отдельным записям в прямом и обратном направлении, а также индикацией номера текущей записи и общего их количества;
* для таблиц, содержащих большое количество полей, должно быть реализовано 2 режима просмотра данных: табличный, с отображением наиболее значимых полей, и по отдельным записям, предназначенный для вывода всех полей записи; при этом все базовые функции работы с данными (добавление/редактирование/удаление, переход к записи и т.д.) должны быть реализованы для обоих режимов;
* быстрый переход к записи с указанным порядковым номером;
* добавление новой записи в таблицу;
* редактирование или удаление произвольно выбранной пользователем записи;
* запрос на подтверждение изменений при добавлении, редактировании или удалении данных;
* контроль правильности ввода данных в соответствии с их типами и форматами (определяются условиями задачи или исходя из рациональных соображений);
* для каждой таблицы (кроме тех, для которых в задании указано требование реализации произвольного фильтра) возможность установки и отмены фильтра на равенство значению для наиболее значимых полей; установка фильтра представляет собой временное (до его отмены) сокрытие записей таблицы, не удовлетворяющих условию фильтра; при этом остаются доступными все остальные базовые функции для работы с таблицей (перемещение по записям, добавление/редактирование /удаление записей, сортировка и т.д.);
* контроль непротиворечивости и целостности структур данных в соответствии с условиями задачи;
* главное управляющее меню, обеспечивающие быстрый доступ к основным функциям программы.

В ряде заданий требуется реализовать функцию поиска по заданным условиям. Поиск должен быть реализован следующим образом. Находится первая запись среди доступных (т.е., если установлен фильтр, то среди оставшихся после него записей таблицы), удовлетворяющая условиям поиска, или выдается сообщение о том, что такой записи нет. Найденная запись должна быть сделана текущей (выведена на экран и/или подсвечена в случае необходимости). Если запись найдена, выдается запрос на продолжение поиска среди последующих записей. Если на этот запрос получен положительный ответ, то поиск возобновляется, начиная с записи, следующей после текущей, и т.д.

В некоторых заданиях присутствует требование реализации произвольного фильтра на таблицу. Если не оговорено особо, то это означает, что фильтр может устанавливаться по произвольному набору полей на равенство значений с выбором логической связки И/ИЛИ для объединения частей фильтра. Используемые в фильтре поля, их значения, а также логическая связка задаются пользователем системы. Например, если в таблице есть 3 поля, то могут быть заданы в частности, следующие фильтры: поле2=знач2; поле1=знач1 И поле3=знач3; поле1=знач1 И поле3=знач3; поле1=знач1 ИЛИ поле2=знач2 ИЛИ поле3=знач3. То же самое касается функции произвольного поиска.

Если для поля или параметра указано, что оно выбирается из справочника, то это означает, что значения поля/параметра/ должны выбираться из списка, данные для которого извлекаются из отдельной таблицы-справочника. Если для поля или параметра указано, что оно выбирается из списка, то это означает, что значения поля/параметра должны выбираться из предопределенного, фиксированного набора значений, определяемого условиями задания или исходя из рациональных соображений. Если указано, что поле является вычисляемым, то это означает, что соответствующие данные не хранятся в таблице, а автоматически вычисляются системой (например, при выводе данных) для каждой записи на основании значений других полей.

В формулировке некоторых функций имеется ссылка на период или отрезок значений (обозначим левую границу отрезка символом С, а правую – символом D). Если не оговорено особо, то это означает, что в качестве параметра запроса могут быть заданы: а) обе границы отрезка; в этом случае значение соответствующего поля должно попадать в указанный отрезок, т.е. C ≤ поле ≤ D; б) левая граница отрезка; в этом случае значение соответствующего поля не должно быть меньше этой границы, т.е. C ≤ поле; в) правая граница отрезка; в этом случае значение соответствующего поля не должно быть больше этой границы, т.е. поле ≤ D.

В ряде заданий требуется реализовать поиск или фильтр по шаблону. Если не оговорено особо, то это означает что значение определяющие условие поиска или установки фильтра, может включать любую комбинацию следующих символов, интерпретируемых специальным образом: # (символ решетки) – ноль или более алфавитно-цифровых символов, ! (восклицательный знак) – один алфавитно-цифровой символ. Таким образом, для того чтобы запись таблицы удовлетворяла условию поиска или фильтра, необходимо чтобы значение поля соответствовало заданному шаблону. Например, пусть задан шаблон «c!o» (кавычки не входят в шаблон). Ему будут соответствовать следующие значения поля: «стол», «столешница», «стоимость» и т.д. Значения «сок», «собака», «строитель» не соответствуют шаблону.

Функция экспорта данных означает, что данные из таблицы следует сохранить в выбранном пользователем текстовом файле, при этом должны быть учтены установленные фильтр и сортировка. Функция импорта данных подразумевает, что в таблицу добавляются данные из выбранного пользователем текстового файла. Если специально не оговорено, внутренний формат файла экспорта/импорта может быть любым, однако при его разработке следует руководствоваться требованием простоты изменения и чтения содержимого файла. При импорте следует учитывать связи обрабатываемой таблицы с другими (например, со справочники) и, при необходимости, обновлять данные и в них для поддержания целостности и непротиворечивости БД. Например, если импортируются данные в таблицу товаров и производитель отсутствует в соответствующем справочнике, то следует добавить в него сведения об этом производителе. Если при описании функции экспорта/импорта указано «с возможностью выбора набора полей», то это означает, что пользователь должен иметь возможность форматирования любой комбинации из имеющихся информационных атрибутов (полей) с целью определения данных, которые следует сохранить/добавить.

Система должна быть реализована в виде консольного (MS DOS) приложения без использования СУБД языка SQL и средств визуального конструирования интерфейсов. В заданиях приведены минимальные требования к БД и функциям, которые должна выполнять система.

К моменту демонстрации перед защитой работоспособности системы БД должна быть заполнена таким образом, чтобы обеспечить нормальную проверку функций. В каждой основной таблице должно быть не менее 50 записей, в каждом справочнике не менее 15 записей.

Задания для курсовой работы делятся на два вида. Первый вид задания – это объектно-ориентированный подход к программированию, обобщенное программирование и технология .NET (ASP.NET, C++, C#) и объектно-ориентированный подход к программированию, обобщенное программирование и Java-технологии (JAVA SE).

**ПРИМЕР ВЫПОЛНЕНИЯ УЧЕБНОГО ПРОЕКТА**

**СЮДА ВСТАИТЬ ВСЕ ЛАБОРАТОРНЫЕ РАБОТЫ**

# Варианты заданий

В каждом из предложенных вариантов требуется при помощи CASE средства Rational Rose построить модель программного обеспечения.

Должны быть выполнены следующие действия:

1. составление глоссария проекта;
2. создание модели вариантов использования;
3. анализ вариантов использования;
4. проектирование системы;
5. реализация системы.

После выполнения третьего этапа модель должна удовлетворять перечисленным ниже требованиям. Глоссарий проекта должен иметь вид таблицы и храниться в отдельном файле. На диаграммах вариантов использования каждое действующее лицо (actor) и вариант использования должны сопровождаться описанием. Эти описания должны быть составлены на русском языке. Описание действующего лица должно коротко (в одну-две строки) сообщать о роли данного лица. Описание варианта использования должно включать в себя пояснение, предусловие, потоки событий (основной и альтернативные, если таковые есть) и постусловие. Описания представляют собой либо присоединенные текстовые файлы, либо текст, введенный в поле Documentation спецификации соответствующего элемента диаграммы. Диаграммы взаимодействия, соответствующие потокам событий вариантов использования, должны содержать необходимые пояснения.

При проектировании системы требуется:

* создать иерархию классов системы;
* разместить классы по пакетам (использовать деление: пользовательский интерфейс – управление – данные; или другое в зависимости от постановки задачи);
* связать объекты с классами, сообщения на диаграммах взаимодействия – с операциями;
* каждый класс снабдить описанием, которое должно включать в себя краткое описание (ответственность класса), описание атрибутов в виде таблицы (имя, описание, тип), таблицу с описанием операций (имя, описание, сигнатура);
* для классов указать стереотипы;
* построить диаграммы классов системы, отображающие связи между классами;
* для описания поведения экземпляров отдельных классов построить диаграммы состояний;
* разработать (если это требуется вариантом задания) схему базы данных и отобразить ее на диаграмме «сущность – связь».

При реализации системы необходимо построить диаграммы компонентов для каждого пакета и для системы в целом. Также следует разработать диаграмму размещения. В зависимости от варианта задания диаграмма размещения должна показывать расположение компонентов в распределенном приложении или связи между встроенным процессором и устройствами. Должна быть произведена проверка корректности модели и автоматическая генерация кода средствами Rational Rose.

Ниже перечислены варианты заданий.

I

1. Цифровой диктофон.

Требуется разработать средствами Rational Rose модель программного обеспечения, управляющего работой цифрового диктофона. Цифровой диктофон – это бытовое электронное устройство, предназначенное для записи и воспроизведения речи. Звуковые сообщения записываются через встроенный микрофон и сохраняются в памяти устройства. Сообщения воспроизводятся через встроенный громкоговоритель. Работа устройства осуществляется под управлением центрального процессора.

Примерный внешний вид устройства изображен на рисунке 6.2.1.

Да

Нет

Рис.6.2.1. Внешний вид диктофона.

Диктофон хранит до 10 звуковых сообщений. Длина каждого сообщения ограничена размером свободной памяти. Диктофон осуществляет прямой (по номеру сообщения) доступ к любому сообщению из памяти. Пользователь имеет возможность воспроизводить сообщения, хранящиеся в памяти диктофона, стирать их, записывать новые. Исполнителем должна быть разработана схема базы данных для хранения сообщений в памяти диктофона. Интерфейс с пользователем осуществляется при помощи экранного меню и управляющих кнопок на корпусе диктофона. При помощи кнопокстрелок осуществляется навигация по пунктам меню. Кнопки «Да», «Нет» служат для подтверждения или отмены пользователем выбора той или иной опции меню (структуру меню исполнитель должен разработать самостоятельно). Имеются также кнопки «Воспроизведение», «Пауза» и «Запись» для работы со звуковыми сообщениями. Во время записи сообщения на экране отображается время, в течение которого ведется запись, при воспроизведении – длительность воспроизведенной части сообщения. Если диктофон не используется, через 30 секунд он автоматически переходит в режим сбережения энергии. В этом режиме никакие операции над звуковыми сообщениями не возможны. Энергия расходуется только на сохранение памяти диктофона в неизменном состоянии. Переход из режима сбережения энергии в обычный режим осуществляется при нажатии пользователем любой кнопки. В диктофоне имеется датчик уровня заряда батарей. При падении уровня заряда ниже установленного предела диктофон автоматически переходит в режим сбережения энергии (независимо от того используется он в данный момент или нет). Переход в обычный режим становится возможным только после восстановления нормального уровня заряда батарей.

## 2. Торговый автомат.

Требуется разработать средствами Rational Rose модель программного обеспечения встроенного процессора универсального торгового автомата.

20.00

ру

б

Кнопки

вы

д

ачи

това

р

а

Сигнальные

лампочки

Возврат

денег

Лотки с товаром

Рис.6.2.2. Лицевая панель торгового автомата

Внешний вид автомата изображен на рисунке 6.2.2. В автомате имеется пять лотков для хранения и выдачи товаров. Загрузка товаров на лотки осуществляется обслуживающим персоналом. Автомат следит за наличием товара. Если какой-либо товар распродан, автомат отправляет сообщение об этом на станцию обслуживания и информирует покупателей (зажигается красная лампочка рядом с лотком данного товара). Автомат принимает к оплате бумажные купюры и монеты. Специальный индикатор высвечивает текущую сумму денег, принятых автоматом к оплате. После ввода денег клиент нажимает на кнопку выдачи товара. Выдача товара производится только в том случае, если введенная сумма денег соответствует цене товара. Товар выдается поштучно. При нажатии на кнопку «Возврат» клиенту возвращаются все принятые от него к оплате деньги. Возврат денег не производился после выдачи товара. Автомат должен корректно работать при одновременном нажатии на кнопки выдачи товара и возврата денег. В специальном отделении автомата, закрываемом замком, есть «секретная кнопка», которая используется обслуживающим персоналом для выемки выручки. При нажатии на эту кнопку открывается доступ к ящику с деньгами. Автомат получает со станции обслуживания данные о товарах и хранит их в своей памяти. Данные включают в себя цену, наименование товара, номер лотка, на котором находится товар и количество товара на лотке. Вариант задания включает в себя разработку схемы базы данных о товарах.

## 3. Табло на станции метро.

Требуется разработать средствами Rational Rose модель программного обеспечения табло для информационной службы метрополитена. Табло расположены на каждой станции метро. Они работают под управлением единого пункта управления (ПУ) информационной службы метро. Табло отображает текущее время (часы, минуты, секунды) и время, прошедшее с момента отправления последнего поезда (минуты, секунды). Момент прибытия и отправления поезда определяется при помощи датчиков, устанавливаемых на путях. Все табло метро синхронизованы, текущее время отсчитывается и устанавливается из центральной службы времени, находящейся на ПУ. На табло высвечивается конечная станция назначения прибывающего поезда. Эти данные содержатся в расписании движения поездов, которое хранится в памяти табло и периодически обновляется с ПУ. В «бегущей строке» табло отображается рекламная информация. Память табло хранит до 10 рекламных сообщений. Сообщения отображаются друг за другом с небольшими паузами, циклически. Содержание рекламных сообщений поступает с ПУ. Дополнительная функция табло – по запросу с ПУ оно пересылает данные о нарушениях расписания (преждевременных отправлениях поездов или опозданиях). В ходе выполнения задания должна быть создана схема базы данных для хранения рекламных сообщений, расписания и сведений о нарушении расписаний.

Пояснение: в задании требуется разработать модель ПО только для табло, но не для пункта управления информационной службы.

## 4. Система автоматизации для пункта проката видеокассет.

Требуется разработать средствами Rational Rose модель программной системы автоматизации работы пункта проката видеокассет (далее в тексте – системы). Пункт проката содержит каталог кассет, имеющихся в наличии в данный момент времени. Система поддерживает работу каталога, позволяя служащим проката добавлять новые наименования кассет, удалять старые и редактировать данные о кассетах. Клиент, обратившийся в пункт, выбирает кассету по каталогу, вносит залог и забирает ее на определенный срок. Срок проката, измеряемый в сутках, оговаривается при выдаче кассеты. Стоимость проката вычисляется системой исходя из тарифа за сутки и срока проката. Клиент возвращает кассету и оплачивает прокат. Если кассета не повреждена, клиенту возвращается залог. Служащий пункта проката регистрирует сдачу кассеты клиенту и ее возврат в системе. Если клиент повредил кассету, то кассета удаляется из каталога, а залог остается в кассе проката. При необходимости служащий может запросить у системы следующие данные:

* имеется ли в наличии кассета с данным названием;
* когда будет возвращена какая-либо кассета из тех, что сданы в прокат;
* является ли данный клиент постоянным клиентом пункта проката (пользовался ли прокатом 5 или более раз).

Постоянным клиентам предоставляются скидки, а также от них принимаются заявки на пополнение ассортимента кассет. Заявки регистрируются в системе. По ним готовится итоговый отчет, руководствуясь которым, служащие пункта проката обновляют ассортимент кассет. Необходимо разработать схему базы данных для хранения каталога, учетных записей о прокате кассет и заявок на пополнение ассортимента.

## 5. Мини-АТС.

Требуется разработать средствами Rational Rose модель программного обеспечения встроенного микропроцессора учрежденческой мини-АТС (автоматической телефонной станции). Мини-АТС осуществляет связь между служащими учреждения. Каждый абонент подключен к ней линией связи. Мини-АТС соединяет линии абонентов (осуществляет коммутацию линий). Абоненты имеют номера, состоящие из трех цифр. Специальный номер 9 зарезервирован для внешней связи. Телефонное соединение абонентов производится следующим образом. Абонент поднимает трубку телефона, и мини-АТС получает сигнал «Трубка». В ответ мини-АТС посылает сигнал «Тон». Приняв этот сигнал, абонент набирает телефонный номер (посылает три сигнала «Цифра»). Мини-АТС проверяет готовность вызываемого абонента. Если абонент не готов (его линия занята), мини-АТС посылает вызывающему абоненту сигнал «Занято». Если абонент готов, мини-АТС посылает обоим абонентам сигнал «Вызов». При этом телефон вызываемого абонента начинает звонить, а вызывающий абонент слышит в трубке длинные гудки. Вызываемый абонент снимает трубку, и мини-АТС получает от него сигнал «Трубка», после чего осуществляет коммутацию линии. Абоненты обмениваются сигналами «Данные», которые мини-АТС должна передавать от одного абонента к другому. Когда один из абонентов опускает трубку, мини-АТС получает сигнал «Конец» и посылает другому абоненту сигнал «Тон». В любой момент абонент может положить трубку, при этом мини-АТС получает сигнал «Конец». После получения этого сигнала сеанс обслуживания абонента завершается. Если абонент желает соединиться с абонентом за пределами учреждения, то он набирает номер «9». Мини-АТС посылает по линии, соединяющей с внешней (городской) АТС, сигнал «Трубка» и в дальнейшем служит посредником между телефоном абонента и внешней АТС. Она принимает и передает сигналы и данные между ними, не внося никаких изменений. Единственное исключение касается завершения сеанса. Получив от городской АТС сигнал «Конец», мини-АТС посылает абоненту сигнал «Тон», и ждет сигнала «Конец» для завершения обслуживания абонента. Если же вызывавший абонент первым вешает трубку, то мини-АТС получает сигнал «Конец», передает его городской АТС и завершает сеанс. Мини-АТС может получить сигнал «Вызов» от городской АТС. Это происходит, когда нет соединений с внешними абонентами. Сигнал «Вызов» от городской АТС передается абоненту с кодом «000». Только этот абонент может отвечать на внешние звонки.

## 6. Телефон.

Требуется разработать средствами Rational Rose модель программного обеспечения встроенного микропроцессора для аппарата учрежденческой телефонной сети. Аппарат подключен к линии связи, ведущей к мини-АТС. В его задачу входит прием и передача сигналов (в том числе и голосовых данных) мини-АТС. Аппарат имеет кнопочную панель управления, экран для отображения набираемых номеров, звонок и трубку, в которую встроены микрофон и громкоговоритель. В начальном состоянии трубка телефона повешена, телефон не реагирует на нажатия кнопок. Телефон реагирует только на сигнал «Вызов» от мини-АТС, при этом включается звонок. При снятии трубки на АТС подается сигнал «Трубка». При получении ответного сигнала «Тон» от АТС телефон воспроизводит звуковой тон «Готов» (длинный непрекращающийся гудок) в трубку. При получении сигнала «Занято», в трубке воспроизводится тон «Занято» (частые короткие гудки). Пользователь, слыша в трубке тон «Готов», набирает трехзначный номер. Номер может быть набран при помощи кнопок с цифрами или нажатием на специальную кнопку **« # »**. При нажатии на кнопку с цифрой соответствующий ей сигнал «Цифра» передается АТС. Нажатия на кнопки с цифрами после третьего игнорируются. Во время набора номера введенные цифры отображаются на экране. Последний полностью набранный номер запоминается в памяти аппарата для того, чтобы можно было его воспроизвести при нажатии на кнопку **« # »**. При нажатии на эту кнопку номер из памяти аппарата высвечивается на экране, и АТС передается последовательность из трех сигналов «Цифра». В ответ на набранный номер от АТС приходит либо сигнал «Занято», либо сигнал «Вызов». При получении сигнала «Вызов» телефон воспроизводит в трубку длинные гудки до того момента, когда АТС осуществит коммутацию и передаст сигнал «Данные». Телефон воспроизводит данные, передаваемые с сигналом, в трубку. Ответ пользователя воспринимается микрофоном трубки, преобразуется в сигнал «Данные» и передается АТС. Обмен данными прерывается, если повешена трубка одного из телефонов, участвующих в обмене. О том, что трубку повесил вызываемый абонент, сообщает сигнал «Занято», посылаемый АТС. После того, как трубка аппарата была повешена, телефон посылает АТС сигнал «Конец», и телефон переходит в начальное состояние.

## 7. Стиральная машина.

Требуется разработать средствами Rational Rose модель программного обеспечения встроенного микропроцессора стиральной машины. Машина предназначена для автоматической стирки белья. Машина включает в себя следующие устройства: бак для белья, клапаны для забора и слива воды, мотор, устройство подогрева воды, термометр, таймер, дверца для доступа в бак, несколько емкостей для различных моющих средств, панель управления с кнопками и индикатором. В памяти машины хранятся 5 программ стирки, заданные изготовителем. Пользователи не могут вносить в них изменения. Каждая программа определяет температуру воды, длительность стирки, используемые моющие средства (номер емкости и время подачи), скорость вращения бака во время стирки и отжима. Вариант задания предусматривает разработку схемы базы данных для хранения программ стирки в памяти машины. Для использования машины необходимо открыть дверцу, поместить белье в бак, поместить моющие средства в емкости, закрыть дверцу, выбрать программу стирки и нажать на кнопку «Пуск». Перед тем как приступить к стирке машина открывает клапан для забора воды, набирает необходимое количество воды, после чего закрывает клапан. Далее, машина действует по выбранной пользователем программе:

1. Подогревает, если необходимо, воду до нужной температуры.
2. Включает таймер и запускает вращение бака для стирки.
3. По таймеру подает в бак моющие средства, предусмотренные программой.
4. По окончании стирки сливает воду и запускает отжим.

Во время работы машины на индикаторе высвечивается время, прошедшее с момента запуска (минуты и секунды), текущий режим работы (стирка или отжим), номер текущей программы стирки. В целях безопасности дверца бака блокируется до окончания стирки. Машина не воспринимает нажатий на кнопки, за исключением одной – пользователь имеет возможность в любой момент нажать на кнопку «Останов», чтобы принудительно остановить стирку и слить воду.

## 8. Таксофон.

Требуется разработать средствами Rational Rose модель встроенной системы управления работой таксофона городской телефонной сети. Таксофон предназначен для оказания платных услуг телефонной связи. Он подключен к линии связи. В нем имеется кнопочная панель, дисплей, трубка со встроенным микрофоном и громкоговорителем, приемник карт – устройство для считывания телефонных карт, используемых для оплаты разговора. В начальном состоянии трубка таксофона повешена, дисплей потушен, таксофон не реагирует на нажатия кнопок и какие-либо сигналы из линии. При снятии трубки таксофон выдает на дисплей сообщение «Вставьте карту» и ожидает, когда пользователь вставит карту в приемник. Дальнейшее функционирование таксофона осуществляется только при вставленной карте. Если карту вынимают, таксофон возвращается к началу и выдает сообщение о необходимости вставить карту. При попадании карты в приемник производится считывание информации с карты. Если кредит исчерпан или карта не пригодна (не удается узнать кредит), то таксофон выдает соответствующее сообщение на дисплей таксофона. Если карта может быть использована для оплаты, то на дисплей выдается количество «единиц» на карте, и на телефонную станцию (АТС) подается сигнал «Трубка». При получении ответного сигнала «Тон» из линии таксофон воспроизводит звуковой тон «Готов» (длинный непрекращающийся гудок) в трубку. При получении сигнала «Занято», в трубке воспроизводится тон «Занято» (короткие гудки).После получения от АТС сигнала «Тон» от пользователя принимаются семизначный номер вызываемого абонента, остальные нажатия на кнопки игнорируются. Когда пользователь нажимает на кнопку с цифрой соответствующий ей сигнал «Цифра» передается АТС. Во время набора номера введенные цифры отображаются на дисплее. В ответ на набранный номер от АТС приходит либо сигнал «Занято», либо сигнал «Вызов». При получении сигнала «Вызов» таксофон воспроизводит в трубку длинные гудки до того момента, когда АТС осуществит коммутацию и передаст сигнал «Данные». Таксофон воспроизводит данные, передаваемые с сигналом, в трубку. При получении данных из трубки, аппарат преобразует их в сигнал «Данные» и передает их АТС. Во время разговора на дисплее ведется отсчет времени и уменьшается кредит на телефонной карте – каждые 15 секунд вычитается четверть «единицы». Обмен данными прерывается, в следующих случаях:

* исчерпан кредит;
* карта вынута из приемника;
* от АТС пришел сигнал «Занято»; – повешена трубка таксофона.

Если трубка была повешена, аппарат посылает в линию сигнал «Конец» и выдает на дисплей сообщение «Выньте карту». После извлечения карты из приемника таксофон переходит в начальное состояние.

## 9. Холодильник.

Требуется разработать средствами Rational Rose модель программного обеспечения встроенного процессора холодильника. Холодильник состоит из нескольких холодильных камер для хранения продуктов. В каждой холодильной камере имеется регулятор температуры, мотор, термометр, индикатор, таймер, датчик открытия двери камеры и устройство для подачи звуковых сигналов. При помощи терморегулятора устанавливается максимально допустимая температура в данной камере. Мотор предназначен для поддержания низкой температуры. Термометр постоянно измеряет температуру внутри камеры, а индикатор температуры, расположенный на дверце, постоянно высвечивает ее значение. При повышении температуры выше предела, определяемого текущим положением регулятора, включается мотор. При снижении температуры ниже некоторого другого значения, связанного с первым, мотор отключается. Доступ в камеру осуществляется через дверцу. Если дверь холодильной камеры открыта в течение слишком долгого времени, подается звуковой сигнал. Звуковой сигнал также подается в любых нештатных ситуациях (например, при поломке мотора).

Холодильник ведет электронный журнал, в котором отмечаются все происходящие события:

* изменение положения терморегулятора камеры;
* включение и отключение мотора;
* доступ в камеру; – внештатные ситуации.

Вариантом задания предусмотрена разработка схемы базы данных для хранения журнала событий холодильника. Содержимое журнала может быть передано в компьютер, подсоединенный к специальному гнезду на корпусе холодильника.

## 10. Кодовый замок.

Требуется разработать средствами Rational Rose модель программного обеспечения встроенного микропроцессора для кодового замка, регулирующего доступ в помещение.Кодовый замок состоит из панели с кнопками (цифры «0»...«9», кнопка «Вызов», кнопка «Контроль»), цифрового дисплея, электромеханического замка, звонка. Панель с кнопками устанавливается с наружной стороны двери, замок устанавливается с внутренней стороны двери, звонок устанавливается внутри охраняемого помещения.В обычном состоянии замок закрыт. Доступ в помещение осуществляется после набора кода доступа, состоящего из четырех цифр. Во время набора кода введенные цифры отображаются на дисплее. Если код набран правильно, то замок открывается на некоторое время, после чего дверь снова закрывается. Содержимое дисплея очищается. Кнопка «Вызов» используется для подачи звукового сигнала внутри помещения. Кнопка «Контроль» используется для смены кодов. Смена кода доступа осуществляется следующим образом. При открытой двери нужно набрать код контроля, состоящий из четырех цифр, и новый код доступа. Для смены кода контроля нужно при открытой двери и нажатой кнопке «Вызов» набрать код контроля, после чего – новый код контроля.

## 11. Турникет метро.

Требуется разработать средствами Rational Rose модель программного обеспечения встроенного процессора турникета для метрополитена. При помощи турникета контролируется проход пассажиров в метро и взимается входная плата. Турникет имеет приемник карт, устройство для перекрывания доступа, таймер, три оптических датчика для определения прохода пассажира, устройство подачи звуковых сигналов, индикаторы «Проход» и «Стоп». В начальном состоянии турникета зажжен индикатор «Стоп», индикатор «Проход» потушен. Если один из датчиков посылает сигнал, то проход через турникет сразу же перекрывается, и подается предупредительный звуковой сигнал. Для прохода пассажир должен поместить карту в приемник карт. Турникет считывает с нее данные: срок годности карты и количество «единиц» на ней. Если данные не удается считать, или карта просрочена, или заблокирована, то карта возвращается пассажиру, и турникет остается в исходном состоянии. В другом случае с карты списывается одна «единица», карта возвращается из приемника, индикатор «Стоп» гаснет, зажигается индикатор «Проход», и пассажир может пройти через турникет. Получив от одного из датчиков сигнал, турникет ожидает время, отведенное на проход пассажира (5 секунд), после чего он возвращается в начальное состояние. Наличие трех датчиков в турникете гарантирует, что при проходе пассажира хотя бы один из них подаст сигнал (датчики невозможно перешагнуть, перепрыгнуть и т.д.). Во время прохода пассажира возможна ситуация, когда все три датчика посылают сигналы. В этом случае принимается только первый сигнал и от момента его приема отсчитывается положенное время. Остальные сигналы игнорируются. Турникет заносит в свою память время всех оплаченных проходов. В конце рабочего дня он передает всю информацию, накопленную за день, в АСУ метрополитена. В ходе выполнения этого варианта задания должна быть разработана схема базы данных о проходах через турникет.

## 12. Система учета товаров.

Требуется разработать средствами Rational Rose модель системы поддержки заказа и учета товаров в бакалейной лавке.В бакалейной лавке для каждого товара фиксируется место хранения (определенная полка), количество товара и его поставщик. Система поддержки заказа и учета товаров должна обеспечивать добавление информации о новом товаре, изменение или удаление информации об имеющемся товаре, хранение (добавление, изменение и удаление) информации о поставщиках, включающей в себя название фирмы, ее адрес и телефон. При помощи системы составляются заказы поставщикам. Каждый заказ может содержать несколько позиций, в каждой позиции указываются наименование товара и его количество в заказе. Система учета по требованию пользователя формирует и выдает на печать следующую справочную информацию:

* список всех товаров;
* список товаров, имеющихся в наличии;
* список товаров, количество которых необходимо пополнить; – список товаров, поставляемых данным поставщиком.

В ходе выполнения этого варианта задания должна быть разработана схема базы данных, хранящей информацию о товарах, заказах и поставщиках.

## 13. Библиотечная система.

Требуется разработать средствами Rational Rose модель системы автоматизирующей деятельность библиотеки. Система поддержки управления библиотекой должна обеспечивать операции (добавление, удаление и изменение) над данными о читателях. В регистрационном списке читателей хранятся следующие сведения: фамилия, имя и отчество читателя; номер его читательского билета и дата выдачи билета. Наряду с регистрационным списком системой должен поддерживаться каталог библиотеки, где хранится информация о книгах: название, список авторов, библиотечный шифр, год и место издания, название издательства, общее количество экземпляров книги в библиотеке и количество экземпляров, доступных в текущий момент. Система обеспечивает добавление, удаление и изменение данных каталога, а также поиск книг в каталоге на основании введенного шифра или названия книги. В системе осуществляется регистрация взятых и возвращенных читателем книг. Про каждую выданную книгу хранится запись о том, кому и когда была выдана книга, и когда она будет возвращена. При возврате книги в записи делается соответствующая пометка, а сама запись не удаляется из системы. Система должна выдавать следующую справочную информацию:

* какие книги были выданы за данный промежуток времени;
* какие книги были возвращены за данный промежуток времени; – какие книги находятся у данного читателя; – имеется ли в наличии некоторая книга.

Вариант задания предусматривает разработку схемы базы данных, хранящей список читателей, каталог книг и записи о выдаче книг.

14. WWW-конференция.

Требуется разработать средствами Rational Rose модель программного обеспечения WWW-конференции. WWW-конференция представляет собой хранилище сообщений в сети Интернет, доступ к которому осуществляется при помощи браузера. Для каждого сообщения конференции хранятся значения следующих полей: номер сообщения, автор, тема, текст сообщения, дата добавления сообщения, ссылка на родительское сообщение. Начальной страницей конференции является иерархический список сообщений. Верхний уровень иерархии составляют сообщения, открывающие новые темы, а подуровни составляют сообщения, полученные в ответ на сообщения верхнего уровня. Сообщение-ответ всегда имеет ссылку на исходное сообщение. В списке отображаются только темы сообщений, их авторы и даты добавления. Просматривая список, пользователь выбирает сообщение и по гиперссылке открывает страницу с текстом сообщения. Помимо текста на этой странице отображается список (иерархический) сообщений, являющихся ответами, ответами на ответы и т.д. Для удобства пользователей необходимо предусмотреть поиск сообщений по автору или, по ключевым словам, в теме или тексте сообщения. Сообщения добавляются в конференцию зарегистрированными пользователями, которые при отправке сообщения должны указать своё имя и пароль. Регистрирует новых пользователей модератор конференции – её ведущий. При регистрации пользователь заполняет специальную форму, содержимое которой затем пересылается модератору и запоминается в базе пользователей. Модератор решает, регистрировать пользователя или нет, и отправляет свой ответ. При добавлении сообщений пользователь имеет возможность начать новую тему или ответить на ранее добавленные сообщения. После добавления сообщения оно доступно для чтения всем пользователям (даже незарегистрированным), и список сообщений обновляется. Модератор имеет право по тем или иным причинам удалять сообщения любых авторов. Он также может наказывать пользователей, нарушающих правила поведения в конференции, лишая на некоторое время пользователя возможности добавлять и редактировать сообщения. Вариант задания включает в себя разработку схемы базы данных для хранения сообщений конференции и информации об её участниках. Выполняющим это задание полезно ознакомиться с заключительным замечанием к варианту «Интернет-магазин». Наиболее подходящей архитектурой для WWW-конференции является «тонкий клиент», поскольку клиентская часть практически не содержит «бизнес-логики». Единственным её элементом, который может выполняться на стороне клиента, является проверка правильного заполнения полей формы, перед отправкой её содержимого на сервер.

## 15. Каталог ресурсов Интернет.

Требуется разработать средствами Rational Rose модель программного обеспечения каталога ресурсов сети Интернет. В каталоге хранится следующая информация о ресурсах: название ресурса, уникальный локатор ресурса (URL), раздел каталога, в котором содержится ресурс, список ключевых слов, краткое описание, дата последнего обновления, контактная информация. Доступ пользователей к каталогу осуществляется при помощи браузера. Пользователи каталога могут добавлять новые ресурсы, информация о которых не была внесена ранее. Ресурсы в каталоге классифицируются по разделам. Полный список ресурсов каждого раздела должен быть доступен пользователям. Пользователям каталога должны быть предоставлены возможности по поиску ресурсов. Поиск осуществляется по ключевым словам. Если пользователь не доволен результатами поиска, он может уточнить запрос (осуществить поиск среди результатов предыдущего поиска). Должна быть возможность выдавать результаты поиска в разной форме (вывод всей информации о ресурсах или частичной). Пользователь может отсортировать список ресурсов по релевантности (соответствию ключевым словам из запроса) или по дате обновления. Поскольку содержание ресурсов Интернет со временем изменяется необходимо следить за датой последнего обновления, периодически опрашивая Web-сайты, URL которых хранятся в каталоге. Вариант задания включает в себя разработку схемы базы данных для хранения сообщений конференции и информации об её участниках. Выполняющим это задание полезно ознакомиться с заключительным замечанием к варианту «Интернет-магазин». Как и в варианте «WWW-конференция» самой подходящей архитектурой для каталога является «тонкий клиент», поскольку клиентская часть практически не включает в себя функций «бизнес-логики» кроме проверки содержимого форм перед пересылкой на сервер.

16. Будильник.

Требуется разработать средствами Rational Rose модель программного обеспечения встроенного микропроцессора для будильника. На экране будильника постоянно отображается текущее время (часы и минуты, например: *12 : 00*), двоеточие между числом часов и числом минут зажигается и гаснет с интервалом в полсекунды.

Управление будильником осуществляется следующими кнопками:

* кнопкой режима установки времени,
* кнопкой режима установки времени срабатывания, – двумя отдельными кнопками для установки часов и минут, – кнопкой сброса сигнала «СБРОС».

На будильнике имеется переключатель режима работы со следующими положениями: «ВЫКЛ», «ВКЛ», «РАДИО» и «ТАЙМЕР». Для установки текущего времени нужно нажать на кнопку режима установки и, при нажатой кнопке, нажимать на кнопки установки часов и минут. При каждом нажатии на кнопки, устанавливаемое значение увеличивается на одну единицу (один час или одну минуту соответственно). При достижении максимального значения производится сброс. Для установки времени срабатывания будильника нужно нажать на кнопку режима установки времени срабатывания и, держа кнопку нажатой, нажимать на кнопки установки часов и минут. Когда переключатель режима работы находится в положении «ВКЛ», при достижении времени срабатывания происходит подача звукового сигнала в течение одной минуты. Сигнал можно прервать, нажав на кнопку «СБРОС». При этом сигнал должен быть возобновлен через пять минут. При установке переключателя в положение «ВЫКЛ» звуковой сигнал не подается. Когда переключатель находится в положении «РАДИО» работает радиоприемник. При переводе переключателя в положение «ТАЙМЕР» включается радиоприемник на тридцать минут, а затем часы переходят в состояние будильника (аналогично положению «ВКЛ»). При нажатии на кнопку режима установки времени, будильник должен отображать время срабатывания.

17. Генеалогическое дерево.

Требуется разработать средствами Rational Rose модель системы для поддержки генеалогических деревьев. Система хранит сведения о персонах (Ф.И.О., пол, дата рождения, дата смерти, биография) и о родственных связях между ними. Связи бывают только трех видов: «мужья-жены», «дети-родители» и «братьясестры». Система обеспечивает возможность добавления данных о новых персонах и родственных связях, изменение введенных данных и удаление ненужных данных. Система следит за непротиворечивостью вводимых данных. Например, недопустимо, чтобы человек был собственным предком или потомком. Разработанная модель должна содержать схему базы данных для хранения генеалогических деревьев. Пользователи системы могут осуществлять поиск полезной информации по дереву:

* находить для указанного члена семьи его детей;
* находить для указанного члена семьи его родителей;
* находить для указанной персоны братьев и сестер, если таковые есть;
* получать список всех предков персоны;
* получать список всех потомков персоны;
* получать список всех родственников персоны;
* прослеживать цепочку родственных связей от одной персоны до другой (например, если Петр является шурином Ивана, то на запрос о родственных связях между Петром и Иваном выдается такой результат: «Петр – брат Ольги, Ольга – жена Ивана»).

## 18. Телевизор.

Требуется разработать средствами Rational Rose модель встроенной системы управления работой телевизора. В телевизоре имеются следующие устройства: приемник телевизионного сигнала, устройство отображения картинки, память каналов, память настроек, управляющие кнопки, пульт дистанционного управления (ДУ). Управление телевизором осуществляется при помощи кнопок на корпусе (их четыре: «ВКЛ / ВЫКЛ», « – », « + », кнопка начальной установки) и пульта ДУ. Кнопка «ВКЛ / ВЫКЛ» позволяет включать и выключать телевизор. После включения телевизора на экран отображается передача, идущая по каналу №1, при этом используются параметры изображения и значение громкости, сохраненные в памяти настроек. Память каналов телевизора хранит до 60 каналов. Каналы нумеруются, начиная с нуля. Последовательное переключение каналов осуществляется при помощи кнопок «–» и «+». Нажатие на «+» переключает телевизор на канал с номером, на единицу большим (с 59-го канала телевизор переключается на 0-ой). Нажатие на «–» переключает телевизор на канал с номером, на единицу меньшим (с 0-го канала телевизор переключается на 59-ый). При нажатии на кнопку начальной установки очищается память каналов телевизора, после чего осуществляется поиск передач и сохранение их частот в памяти каналов. Поиск начинается с нижней границы рабочего диапазона телевизора. На экран телевизора выводится «синий экран». Рабочая частота постепенно увеличивается до тех пор, пока приемник не обнаружит телевизионный сигнал. Найденная передача выводится на экран в течение 10 секунд. Также отображается номер, под которым найденный канал будет сохранен в памяти (начиная с 1). Затем поиск продолжается до тех пор, покане достигнута верхняя граница диапазона, или пока не заполнена вся память каналов. Телевизор принимает управляющие сигналы с пульта ДУ. На пульте ДУ расположены следующие кнопки:

* кнопки с цифрами «0»...«9» для прямого переключения канала (по номеру);
* кнопки «П–» и «П+» для последовательного переключения каналов;
* кнопки «Г–» и «Г+» для изменения громкости;
* кнопки «МЕНЮ», « < » и « > » для доступа к экранному меню.

Для прямого переключения на нужный канал его номер набирается с помощью кнопок с цифрами. После нажатия первой цифры в течение 5 секунд ожидается нажатие второй. Если вторая цифра не была нажата, то номер канала считается состоящим из одной цифры и осуществляется переключение на него. Кнопки «П–» и «П+» на пульте имеют те же функции, что и кнопки «-» и «+» на корпусе телевизора. Кнопки «Г–» и «Г+» позволяют увеличивать или уменьшать громкость. Каждое нажатие на «Г–» уменьшает громкость на одну единицу, пока она больше нуля, «Г+» увеличивает громкость на единицу, пока не достигнуто максимальное значение. Текущее значение громкости сохраняется в памяти настроек. Кнопки «МЕНЮ», «<» и «>» позволяют устанавливать значения настоек, хранящихся в памяти телевизора. При нажатии на кнопку «МЕНЮ» внизу экрана возникает надпись «ЯРКОСТЬ» и полоса, отображающая текущее значение яркости. Кнопками «<» и «>» яркость можно уменьшить или увеличить. При работе с меню нажатия на все остальные кнопки игнорируются. После повторного нажатия на кнопку «МЕНЮ» значение яркости запоминается в памяти настроек, и осуществляется переход к настройке контрастности. Настройка контрастности и остальных параметров (четкости, цветовой гаммы) происходит аналогично. Нажатие на кнопку «МЕНЮ» по окончании настройки цветовой гаммы (последнего пункта меню) приводит к окончанию работы с меню. Выход из меню также осуществляется в том случае, если в течение 15 секунд не была нажата ни одна кнопка.

## 19. Система поддержки составления расписания занятий.

Требуется разработать средствами Rational Rose модель системы поддержки составления расписания занятий. Система обеспечивает составление расписания некоторого учебного заведения, внесение в расписание изменений, выдачу полного расписания и дополнительной информации (например, по итоговому расписанию составляется расписание указанной группы на заданный день или неделю). В расписании фиксируются время и место проведения занятия, предмет и преподаватель, проводящий занятие, а также номер группы, для которой это занятие проводится. Расписание не должно содержать коллизий (например, разные занятия не должны пересекаться друг с другом по месту и времени их проведения, один преподаватель не может вести одновременно два разных занятия, в одно и то же время у одной и той же группы не может быть два различных занятия и т. д.).При работе над этим вариантом задания необходимо разработать схему базы данных для хранения расписания.

## 20. Домофон.

Требуется разработать средствами Rational Rose модель программного обеспечения встроенного микропроцессора домофона. Домофон регулирует доступ в подъезд многоквартирного дома. В подъезде имеется дверь с замком. С наружной стороны двери установлена внешняя панель домофона, на которой находятся кнопки для связи с каждой квартирой, микрофон и динамик. В каждой квартире находится внутренняя панель домофона с кнопками: «СВЯЗЬ», «БЛОКИРОВКА» и «ОТКРЫТЬ». Кроме того, на внутренней панели имеется микрофон и динамик. Жильцы могут открывать дверь ключом. Посетитель может нажать кнопку квартиры на внешней панели. При этом в квартире раздается звонок (если подача звонка в квартиру не заблокирована). Услышав звонок, жилец квартиры нажимает на кнопку «СВЯЗЬ» внутренней панели домофона, после чего домофон устанавливает звуковое сообщение между жильцом и посетителем. Звуки, произносимые посетителем в микрофон, установленный на внешней панели, воспроизводятся в динамике, установленном в квартире. Звуки из микрофона в квартире, передаются в динамик на внешней панели. После сеанса связи жилец может нажать на кнопку «ОТКРЫТЬ», чтобы замок на двери в подъезд открылся, и посетитель смог войти. По истечении минуты замок должен снова заблокировать вход в подъезд. Жилец, который желает, чтобы его не беспокоили, может отключить подачу звонка в свою квартиру, нажав на кнопку «БЛОКИРОВКА». Повторное нажатие на эту кнопку вновь включает подачу звонка.

II

1. ИПС для учета и контроля конкурса в ВУЗы. Таблица ВУЗов: наименование, адрес, специализация (выбирается из справочника), форма собственности (выбирается из справочника). Таблица конкурсов: ВУЗ (выбирается из таблицы ВУЗов), специальность (выбирается из справочника), год, форма обучения (выбирается из списка: дневная, вечерняя, заочная), средний балл, размер оплаты на коммерческом отделении. Функции системы: а) подсчет количества специальностей для каждого ВУЗа на указанный год; б) вывод списка ВУЗов, обучавших по данной специальности в заданный период лет; в) построение диаграммы изменения среднего балла для заданных ВУЗа и специальности.
2. ИПС для службы подбора кадров. Таблица безработных: фамилия, имя, отчество, дата рождения (в формате дд.мм.гггг), место рождения, профессия (выбирается из справочника), образование (выбирается из списка, например: среднее, среднее специальное, высшее, ученая степень), место последней работы (фирма; выбирается из справочника), должность на последней работе, причина увольнения, семейное положение (логический флаг — состоит или нет в браке), жилищные условия, контактный телефон, минимальный уровень зарплаты. Таблица вакансий: фирма (выбирается из справочника фирм), должность, условия труда, размер зарплаты, требования к специалисту. Функции системы: а) поиск работников в соответствии с заданными условиями (по возрасту — граничные значения от и до, профессии, образованию, семейному положению, уровень зарплаты — граничные значения от и до); б) поиск вакансий по заданным условиям (по должности, условиям труда, размеру зарплаты — граничные значения от и до); в) подсчет и вывод количества вакансий для каждой фирмы; подсчет и вывод среднего уровня зарплаты для каждой должности (по вакансиям). Поиск должен быть реализован таким образом, чтобы его можно было осуществлять по любой комбинации указанных условий.
3. ИПС для склада товаров. Таблица товаров: наименование, категория (выбирается из справочника), фирма-производитель (выбирается из справочника), единица измерения (выбирается из справочника). Таблица поставок товаров: товар (выбирается из таблицы товаров), дата поступления (в формате дд.мм.гггг), количество единиц, цена за единицу, фирма-поставщик (выбирается из справочника). Таблица отгрузок товаров: товар (выбирается из таблицы товаров), дата отгрузки, количество единиц. Функции системы: а) вывод состояния склада на указанную дату (список имеющихся товаров с указанием их количества); б) вывод списка поставщиков, поставлявших товары указанной категории в заданный период дат с возможностью просмотра поставок за этот период для выбранного из списка поставщика; в) фильтр по наименованию товаров на вхождение произвольной подстроки.
4. ИПС для учета продаж билетов на авиарейсы. Таблица авиарейсов: номер рейса, количество посадочных мест. Таблица маршрутов: рейс (выбирается из таблицы рейсов), город (выбирается из справочника; выбранное значение должно проверяться на дублирование, т.е. для одного и того же рейса город не может быть указан дважды), время полета в часах (вещественное число) до данного города от предыдущего города, порядковый номер (1 — место первой посадки, 2 — место второй посадки, …, n — место назначения; автоматически формируется системой при выборе рейса). Таблица вылетов: номер рейса (выбирается из таблицы рейсов), дата вылета (в формате дд.мм.гггг), время вылета (чч:мм). Таблица проданных билетов: рейс (выбирается из таблицы рейсов), дата вылета, ФИО пассажира. Функции системы: а) нахождение ближайшего вылета для указанной даты, позволяющего попасть в заданный город (выбирается из таблицы городов); б) вывод списка рейсов, отсортированного по убыванию времени перелета; в) вывод списка рейсов, с помощью которых можно добраться до указанного города (выбирается из таблицы городов), отсортированного по возрастанию времени перелета; г) подсчет количества билетов, проданных на указанный вылет.
5. ИПС для работы с реестром организаций. Таблица организаций: название, адрес, e-mail, адрес сайта, специализация (выбирается из справочника), форма собственности (выбирается из справочника), время начала работы (в формате чч:мм), время окончания работы, описание. У одной организации может быть несколько телефонов. Таблица телефонов: организация (выбирается из таблицы организаций), номер телефона, категория (выбирается из справочника), факс (логический флаг). Функции системы: а) при просмотре сведений об организации должен отображаться список её телефонов; б) произвольный фильтр на таблицу организаций; в) вывод списка специализаций с указанием количества соответствующих организаций; г) вывод списка форм собственности с указанием количества соответствующих организаций; д) поиск организаций, у которых в названии встречается заданная подстрока.
6. ИПС для составления меню и учета запасов продуктов питания. Таблица блюд: название, рецепт приготовления, примерное время приготовления (в формате чч:мм), количество персон. Каждое блюдо содержит несколько ингредиентов. Таблица ингредиентов: блюдо (выбирается из таблицы блюд), продукт (выбирается из таблицы продуктов), количество для приготовления в единицах измерения. Таблица продуктов: название, единица измерения (выбирается из справочника), имеющееся количество единиц. Функции системы: а) вывод количества продуктов, необходимого для приготовления указанного меню, задаваемого путем выбора из таблицы блюд нескольких блюд; для каждого продукта должно быть указано также его имеющееся количество и, в случае нехватки, какое количество надо докупить; б) экспорт результатов работы предыдущей функции; в) найти блюда, в которых используется указанный продукт (выбирается из таблицы продуктов) в количестве не менее/более заданного (реализовать в виде фильтра на таблицу блюд); г) вывести блюда, время приготовления которых не превышает заданного количества минут (реализовать в виде фильтра на таблицу блюд); д) вывести блюда, которыми можно накормить заданное количество персон (реализовать в виде фильтра на таблицу блюд).
7. ИПС для учета журналов. Таблица журналов: название, издатель (выбирается из справочника), тематика (выбирается из справочника). Таблица номеров журналов: журнал (выбирается из таблицы журналов), номер, год, объем в страницах, дата выпуска (в формате дд.мм). Таблица статей: номер журнала (выбирается из таблицы номеров), название статьи, авторы, номер начальной страницы, номер конечной страницы, тематика (выбирается из справочника тематик). Функции системы: а) вывод статей, в названии которых встречается указанная подстрока (реализовать в виде фильтра на таблицу статей); б) вывод упорядоченного по алфавиту списка авторов с указанием сведений об опубликованных ими статьях и их суммарного объема; в) вывод списка издателей с указанием количества издаваемых ими журналов; г) подсчет количества выпущенных за указанный период дат (даты в формате дд.мм.гггг) журналов заданной категории (выбирается из справочника) с выводом их суммарного объема.
8. ИПС для фирмы, занимающейся продажей автомобилей. Таблица автомобилей: производитель (выбирается из справочника), марка, страна производства (выбирается из справочника), год выпуска, технические характеристики (для этих сведений выбрать и ввести несколько полей, например: объем двигателя, количество цилиндров, коробка передач, максимальная скорость и т.д.), особенности исполнения (выбирается из списка: седан, хэтчбэк, купе и т.д.), техническое состояние, пробег, запрашиваемая цена. Таблица покупателей: имя, телефон, требования к марке, техническим характеристикам и техническому состоянию, минимальная цена, максимальная цена. Функции системы: а) вывод автомобилей, удовлетворяющих требованиям указанного клиента (реализовать в виде фильтра на таблицу автомобилей); б) сохранение полученных в результате предыдущего запроса данных в выбираемый пользователем текстовый файл; в) вывод списка производителей с указанием для каждого из них общего количества продаваемых автомобилей, их марок и количества по каждой марке; г) произвольный фильтр на таблицу автомобилей.
9. ИПС для ведения реестра населенных пунктов. Таблица стран: название, материк (выбирается из списка), площадь, численность населения, форма государственного правления (выбирается из списка), столица. Страна может включать несколько регионов. Таблица регионов: название, тип (выбирается из списка, например: область, провинция, штат и т.д.), страна (выбирается из таблицы стран), численность населения, столица. Каждый регион включает несколько населенных пунктов. Таблица населенных пунктов: название, тип (выбирается из списка: город, деревня, поселок и т.д.), регион (выбирается из таблицы регионов), численность населения, долгота, широта. Функции системы: а) вывод таблицы населенных пунктов, имеющих одинаковое название, с указанием для каждого пункта всех его данных и страны, а также возможностью редактирования выбранной записи или её удаления; б) вывод всех населенных пунктов указанной страны, численность населения которых попадает в заданный отрезок (реализовать в виде фильтра на таблицу населенных пунктов); в) вывод всех населенных пунктов, которые лежат в заданном географическом прямоугольнике (реализовать в виде фильтра); г) вывести таблицу материков, с указанием для каждого из них списка форм государственного правления расположенных там стран; для каждой формы правления должна быть указана общая площадь соответствующих стран, расположенных на данном материке.
10. ИПС для работы с реестром почтовых отделений. Таблица административных единиц: название, тип (выбирается из списка, например: республика, область, край, район и т.д.), ссылка на административную единицу, в которую входит данная единица (если данная единица не является образованием «верхнего уровня»). Административная единица может включать несколько населенных пунктов. Таблица населенных пунктов: название, административная единица (выбирается из таблицы административных единиц). Населенный пункт может включать несколько почтовых отделений. Таблица почтовых отделений: населенный пункт (выбирается из таблицы населенный пунктов), индекс, количество обслуживаемых жителей. Функции системы: а) вывод списка населенных пунктов, название которых содержит указанную подстроку (реализовать в виде фильтра); б) вывод списка населенных пунктов, входящих в заданную административную единицу (здесь и далее с учетом иерархии единиц; реализовать в виде фильтра); в) подсчет количества почтовых отделений в заданном населенном пункте и в заданной административной единице; г) вывод списка населенных пунктов, индекс почтовых отделений которых включает указанную подстроку (реализовать в виде фильтра); д) найти почтовое отделение, обслуживающее максимальное количество жителей в указанной административной единице.
11. ИПС учета состояния автотранспортных средств. Таблица автотранспортных средств: тип (выбирается из списка, например: автомобиль, мотоцикл, мопед, прицеп и т.д.), марка, производитель (выбирается из справочника), цвет (выбирается из справочника), заводской номер, бортовой номер, дата выпуска, особенности конструкции (выбирается из списка, например: седан, хэтчбэк, купе и т.д.), особенности окраски, дата последнего техосмотра, ФИО владельца, номер водительского удостоверения, паспортные данные владельца. Функции системы: а) вывод списка транспортных средств, которые должны пройти техосмотр в течение следующего месяца; такие средства определяются на основании текущей даты, даты выпуска и даты последнего техосмотра (реализовать в виде фильтра); б) вывод всех транспортных средств указанной категории, у которых номер совпадает с заданным шаблоном (реализовать в виде фильтра); в) вывод всех транспортных средств указанной марки, выпущенных в заданный период времени (реализовать в виде фильтра); г) вывод списка производителей с указанием для каждого из них списка марок и количества транспортных средств по маркам.
12. ИПС для учета коллекции книг. Таблица книг: авторы, название, издательство (выбирается из таблицы издательств), год издания, количество страниц, категория (выбирается из списка, например: детектив, фантастика, публицистика, справочник и т.д.), происхождение (выбирается из списка, например: куплена, подарена и т.д.), когда отдана (дата в формате дд.мм.гггг, когда книга отдана почитать), кому отдана (сведения о человеке, которому книга отдана на чтение; выбирается из справочника), субъективная оценка книги. Таблица издательств: название, город. Функции системы: а) произвольный поиск по следующему набору полей: авторы (по шаблону), название (по шаблону), год издания (на вхождение в отрезок), категория (на совпадение); б) вывод всех книг, которые отданы почитать более чем месяц назад (реализовать в виде фильтра); в) вывод списка людей, которым отдано на чтение более одной книги (реализовать в виде фильтра); г) вывод списка категорий, с указанием для каждой из них общего количества книг, количества издательств и суммарного количества страниц.
13. ИПС для службы знакомств. Таблица клиентов: регистрационный номер, дата регистрации, фамилия, имя, отчество, пол (выбирается из двух значений), дата рождения, состоял ли в браке (логический флаг), количество детей, рост, вес, цвет волос (выбирается из списка), цвет глаз (выбирается из списка), профессия (выбирается из справочника), увлечения, требования к партнеру (для этих сведений выбрать и ввести несколько полей, например: границы для возраста, роста и веса, отсутствие детей и т.д.), контактная информация. Функции системы: а) вывод списка подходящих кандидатур, для выбранного клиента в соответствии с его требованиями (реализовать в виде фильтра); б) сохранение данных, полученных при выполнении предыдущего запроса, в указанном текстовом файле; в) вывод статистических данных: количество клиентов каждого пола, количество клиентов для каждой возрастной группы (до 25 лет, 25‑35, 35‑45, 45‑55, от 55 лет), количество клиентов для каждой профессии; г) произвольный поиск по следующим параметрам: пол, возраст (на вхождение в отрезок), состоял ли в браке (выбирается из списка: нет, да, всё равно), наличие детей (выбирается из списка: нет, да, всё равно), рост (на вхождение в отрезок), вес (на вхождение в отрезок).
14. ИПС для учета контактных данных. Таблица персон: фамилия, имя, отчество, пол (выбирается из двух значений), дата рождения, тип отношений (выбирается из списка, например: родственник, друг, знакомый, коллега и т.д.), место работы или учебы, профессия (выбирается из справочника), должность, деловые качества, e-mail, примечание. Каждая персона может иметь несколько контактных телефонных номеров. Таблица телефонов: персона (выбирается из таблицы персон), номер телефона, тип (выбирается из списка, например: рабочий, домашний, мобильный), примечание. Функции системы: а) вывод списка всех телефонов указанной персоны (реализовать в виде фильтра на таблицу телефонов); б) при выводе таблицы персон для каждой записи предусмотреть отображение ссылки «телефоны», позволяющей перейти к просмотру телефонов соответствующей персоны путем выполнения предыдущей функции; в) вывод списка персон, у которых будет день рождения (на выбор пользователя): сегодня, в течение указанного количества последующих дней, в течение текущей недели или текущего месяца (реализовать в виде фильтра); г) произвольный поиск.
15. ИПС для работы с реестром товаров. Таблица товаров: название, категория (выбирается из справочника), фирма-производитель (выбирается из справочника), единица измерения (выбирается из справочника), цена единицы, количество, дата последнего завоза. Функции системы: а) импорт данных; дата последнего завоза для импортированных товаров устанавливается равной значению, заданному пользователем; б) вывод списка категорий с указанием для каждой из них: количества наименований товаров, суммарного количества единиц товаров, средней цены, товара с минимальной и максимальной ценой; в) построение круговой диаграммы распределения суммарного количества единиц товаров по категориям; г) вывод списка фирм-производителей с указанием для каждого из них: суммарного количества единиц товаров, категорий поставляемых товаров, а также количества и процента товаров этой категории от приведенного суммарного количества.
16. ИПС для отдела кадров. Таблица сотрудников: фамилия, имя, отчество, дата рождения, место рождения (выбирается из справочника), серия паспорта, номер паспорта, дата выдачи паспорта, место выдачи паспорта, ИНН, номер страхового свидетельства пенсионного фонда, домашний адрес, образование (выбирается из списка, например: среднее, среднее специальное, высшее, ученая степень), специальность (выбирается из справочника), подразделение (выбирается из справочника), должность (выбирается из справочника), оклад, дата поступления в организацию. Функции системы: а) произвольный фильтр; б) вывод списка подразделений с указанием для каждого из них количества сотрудников; в) вывод списка сотрудников предпенсионного и пенсионного возраста (реализовать в виде фильтра); г) экспорт данных с возможностью выбора набора полей.
17. ИПС для кассы автовокзала. Таблица рейсов автобусов: номер рейса, время отправления, количество свободных мест. Таблица маршрутов: номер рейса (выбирается из таблицы рейсов), пункт остановки (выбирается из справочника; выбранное значение должно проверяться на дублирование, т.е. для одного и того же рейса пункт остановки не может быть указан дважды), время движения в минутах (вещественное число) до данной остановки от предыдущей остановки, порядковый номер (1 — место первой остановки, 2 — место второй остановки, …, n — конечный пункт маршрута; автоматически формируется системой при выборе рейса). Функции системы: а) при выборе записи в таблице рейсов для данного рейса должен отображаться его маршрут с указанием общей продолжительности движения и количества промежуточных остановок; б) вывод списка рейсов, с помощью которых можно добраться до заданного пункта, отсортированного по возрастанию времени отправления; в список включаются только те рейсы, время отправления которых позже текущего времени и на которые есть свободные места; реализовать в виде фильтра на таблицу рейсов; в) вывод списка рейсов, отсортированного на выбор пользователя по общей продолжительности движения или количеству промежуточных остановок; г) экспорт данных о рейсах с указанием их маршрутов.
18. ИПС для гостиницы. Таблица номеров: номер, класс номера (выбирается из списка), этаж, число мест, количество комнат, наличие ТВ, наличие телефона, наличие кондиционера, наличие холодильника, тип санузла (выбирается из списка: совмещенный, раздельный), душ/ванна, наличие балкона, номер для некурящих (логический флаг), дополнительные удобства, стоимость за сутки. Таблица постояльцев: номер (выбирается из таблицы номеров), фамилия, имя, отчество, страна проживания (выбирается из справочника), место проживания (выбирается из справочника), серия паспорта, номер паспорта, дата выдачи паспорта, место выдачи паспорта, дата приезда, дата отъезда. Таблица брони: номер (выбирается из таблицы номеров), дата начала использования, дата окончания использования, на кого забронирован, код брони (автоматически генерируется системой при добавлении записи о брони). Функции системы: а) произвольный фильтр на таблицу номеров; б) вывод списка свободных номеров на указанную дату с учетом брони (реализовать в виде фильтра); в) вывод всех постояльцев, отъезжающих указанного числа (реализовать в виде фильтра); г) проверка возможности бронирования номера на указанный период времени (задаются две даты); д) проверка возможности продления проживания в заданном номере на указанное количество дней.
19. ИПС для работы с реестром терминов. Таблица терминов: термин, область науки или хозяйства (выбирается из справочника), толкование/определение термина, используемые термины (выбираются из таблицы терминов). Каждый термин может иметь различные толкования для разных областей науки/хозяйства. Функции системы: а) реализовать вывод используемых терминов в виде ссылок, т.е. чтобы можно было выбрать некоторый термин и перейти к соответствующей записи; б) произвольный поиск; для полей термина и определения по шаблону; для поля используемых терминов на наличие заданных терминов; в) вывод списка областей науки/хозяйства с указанием для каждой из них количества терминов; г) построение диаграммы распределения количества терминов по областям науки/хозяйства.

ИПС для медицинского работника. Таблица болезней: название, симптомы (выбираются из справочника), лечебные процедуры (выбираются из справочника). Таблица медикаментов: название, категория (выбирается из справочника; например: для лечения простуды, для лечения болезней ЖКТ, сердечно-сосудистое средство, БАД и т.д.), взаимозаменяемые средства (выбираются из таблицы медикаментов), показания к применению, противопоказания, побочные действия, форма выпуска (выбирается из справочника). Для лечения каждой болезни может использоваться несколько медикаментов. Таблица курсов лечения: болезнь (выбирается из таблицы болезней), медикамент (выбирается из таблицы медикаментов), режим приема лекарственного средства, продолжительность приема. Функции системы: а) при выборе записи в таблице болезней вывод соответствующих сведений о курсе лечения; б) произвольный поиск по таблице болезней; для поля названия по шаблону; для полей симптомов и процедур на наличие соответствующих элементов, выбираемых из списков; в) произвольный фильтр на таблицу медикаментов; г) вывод списка категорий медикаментов, с указанием для каждой из них общего количества медикаментов, а также списка названий болезней, для лечения которых применяются соответствующие средства.

# УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ

***Содержание отчёта по лабораторной работе***

Отчёт по выполнению лабораторной работы должен включать в себя:

* титульный лист;
* постановка задачи;
* метод решения задачи;
* алгоритм решения задачи;
* листинг программы;
* текстовые примеры;
* заключение;

Требования к оформлению и вид ***титульного листа*** приведены в конце документа.

***Постановка задачи***

В данном разделе должна быть представлена формулировка задачи, которая была решена в данной лабораторной работе.

***Метод решения задачи***

В словесной форме описывается способ или подход, применяемый для решения задачи. При необходимости приводятся требуемые теоретические сведения, а также формулы и преобразования. При составлении описания следует учитывать, что оно используется в качестве основы при построении алгоритма решения и поэтому должно максимально точно и однозначно указывать, как решать данную задачу.

***Алгоритм решения задачи***

В формальной форме, например, в виде блок-схемы или схемы Насси-Шнейдермана, приводится алгоритм решения задачи.

***Листинг-программы***

В данном разделе должен быть представлен полный текст программы, предназначенной для решения задачи и созданной в соответствии с разработанными ранее методом и алгоритмом решения.

***Тестовые примеры***

Приводится как минимум два характерных варианта исходных данных и соответствующие выходные результаты, полученные в ходе работы программы.

***Заключение***

Следует отметить полученные в ходе лабораторной результаты (это не результаты из предыдущего раздела) и подчеркнуть особенности метода/алгоритма решения и/или программной реализации. Должны быть указаны существующие ограничения, возможные варианты их устранения, а также перспективы/направления развития и улучшения программы.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

|  |  |  |  |
| --- | --- | --- | --- |
| №  п/п | Наименование работы, ее вид | Выходные данные | Автор(ы) |
| 1. | Объектно-ориентированные методы анализа, программирования и проектирования. Учебное пособие. | М. : ТУСУР, 2012. — 25 с. | Катаев М.Ю. |
| 2. | Применение UML и шаблонов проектирования: Введение в объектно-ориентированный анализ, проектирование и итеративную разработку. Монография. | МСК-СПб\_Киев: Вильямс, 2013. – 736 с. | Крэг Ларман |
| 3. | Объектно-ориентированный анализ и проектирование с примерами приложений.  Монография. | МСК-СПб-Киев.: Вильямс, 2008. – 720 с. | Гради Буч, Роберт А. Максимчук и др. |

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Автор | Вид издания (учебник, учебное  пособие) | Место издания, издательство, год издания, кол-во страниц |
| **Основная литература** | | | |
| Применение UML и шаблонов проектирования: Введение в объектно-ориентированный анализ, проектирование и итеративную разработку. | Крэг Ларман | Монография. | МСК-СПб\_Киев: Вильямс, 2013. – 736 с. |
| Объектно-ориентированный анализ и проектирование с примерами приложений. | Гради Буч, Роберт А. Максимчук и др. | Монография. | МСК-СПб-Киев.: Вильямс, 2008. – 720 с. |
| **Дополнительная литература** | | | |
|  |  |  |  |
| Проектирование программного обеспечения экономических информационных систем. | Вендров А.М. | Учебное пособие | МСК.: Финансы и статистика, 2007. – 544с. |
| Практикум по проектированию программного обеспечения экономических информационных систем | Вендров А.М. | Практикум | МСК.: Финансы и статистика, 2007. – 192с. |
| Моделирование на UML | Д. Иванов,  Ф. Новиков. | Учебное пособие. | СПб. : НИУ ИТМО, 2010. — 200 с. |
| ПРОГРАММНАЯ ИНЖЕНЕРИЯ. ПАРАДИГМЫ, ТЕХНОЛОГИИ И CASE-СРЕДСТВА. | Лаврищева Е.М. | Учебное пособие. | М.: ИЗДАТЕЛЬСТВО ЮРАЙ, 2016. 280 с. |
| БАЗЫ ДАННЫХ: ПРОЕКТИРОВАНИЕ. | Стружкин Н.П., Годин В.В. | Учебное пособие. | М.: ИЗДАТЕЛЬСТВО ЮРАЙ, 2016. —291с. |
| 11. Объектно-ориентированные методы анализа, программирования и проектирования. Учебное пособие. | М. : ТУСУР, 2012. — 25 с. | Катаев М.Ю. | Катаев М.Ю. |

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование информационного ресурса | Ссылка на информационный ресурс |
|  | Сайт Унифицированного языка визуального моделирования UML | http://www.uml.org/ |
|  | Сайт Института развития информационного общества | http://www.iis.ru |
|  | Сайт научно-аналитического журнала «Информационное общество» | http://www.infosoc.iis.ru |
|  | Национальный открытый университет «ИНТУИТ» | http://www.intuit.ru/studies/courses/ |

# ПРИЛОЖЕНИЕ 1

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА**

**ФЕДЕРАЛЬНОЕ бюджетное ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ БЮДЖЕТНЫЙ УНИВЕРСИТЕТ ВОДНЫХ КОММУНИКАЦИЙ»**

**Кафедра вычислительных систем и информатики**

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Методы и средства проектирования информационных систем и технологий»

Выполнил: ст.гр. ИС-31

Иванов И.И.

Проверил: Балса А.Р.

Санкт-Петербург

2017

# ПРИЛОЖЕНИЕ 2

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО МОРСКОГО И РЕЧНОГО ТРАНСПОРТА**

**ФЕДЕРАЛЬНОЕ бюджетное ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ БЮДЖЕТНЫЙ УНИВЕРСИТЕТ ВОДНЫХ КОММУНИКАЦИЙ»**

**Кафедра вычислительных систем и информатики**

Пояснительная записка к курсовой работе

по дисциплине «Методы и средства проектирования информационных систем и технологий» тема работы

«Разработка информационно – поисковой системы»

Выполнил: ст.гр. ИС-31

Иванов И.И.

Проверил: Балса А.Р.

Санкт-Петербург

2017

**Балса Алдрин Раульевич, к.г.н, доцент**

Методы и средства проектирования

информационных систем и технологий

**Методическое обеспечение курсовых и лабораторных работ**



198035, Санкт-Петербург, Межевой канал, 2

Тел. 812-748-97-19, 748-97-23

e-mail:izdat@gumrf.ru

|  |  |  |
| --- | --- | --- |
|  |  |  |