Учреждение образования

«Белорусский государственный университет информатики   
и радиоэлектроники»

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

**Т. С. Боброва, М. П. Ревотюк**

**Методические указания к курсовому проектированию**

по дисциплине «Объектно-ориентированное программирование и проектирование»

для студентов специальности 1-53 01 02 «Автоматизированные системы обработки информации» всех форм обучения

Минск 2019

**КАЛЕНДАРНЫЙ ПЛАН МЕРОПРИЯТИЙ ПО ОРГАНИЗАЦИИ  
И ПРОВЕДЕНИЮ КУРСОВОГО ПРОЕКТИРОВАНИЯ В БГУИР**

|  |  |  |
| --- | --- | --- |
| Наименование  мероприятия | Срок исполнения | Ответственные  за исполнение |
| 1 | 2 | 3 |
| 1. Разработка тематики курсовых проектов (работ) | До начала учебного семестра, в котором запланировано курсовое проектирование | Заведующие кафедрами |
| 2. Выдача студентам заданий по курсовому проекту (работе) и представление информации в деканаты | *Дневная и вечерняя формы обучения*  Осенний семестр – до 15.09  Весенний семестр:  1–3 курс – до 24.02  4–5 курс – до15.02 текущего учебного года  *Заочная (в т. ч. дистанционная) форма обучения*  Выдача заданий – на установочной сессии и в первый «день заочника»  Представление информации:  Осенний семестр – до 01.10  Весенний семестр – до 01.03 | Руководители курсовых проектов (работ) |
| 3. Представление информации в деканаты о ходе выполнения студентами заданий по курсовому проектированию:  1-я контрольная точка | *Для всех форм обучения*  Осенний семестр – до 15.10  Весенний семестр – до 15.03 текущего учебного года | Руководители курсовых проектов (работ) |
| 4. 2-я контрольная точка | *Для всех форм обучения*  Осенний семестр – до 15.11  Весенний семестр – до 15.04 текущего учебного года | Руководители курсовых проектов (работ) |
| 5. 3-я контрольная точка | *Для всех форм обучения*  Осенний семестр – до 15.12  Весенний семестр – до 15.05 текущего учебного года | Руководители курсовых проектов (работ) |
| 6. Представление студентами готовых курсовых проектов (работ) руководителям для проверки | *Дневная и вечерняя формы обучения*  Осенний семестр:  1–3 курс – с 10.12 по 26.12  4–5 курс – с 01.12 по 15.12  Весенний семестр:  1–3 курс – с 20.05 по 05.06  4 курс – с 25.04 по 10.05 текущего учебного года  *Заочная форма обучения*  Не позднее чем за одну неделю до лабораторно-экзаменационной сессии  *Дистанционная форма обучения*  Не позднее чем за одну неделю до лабораторно-зачетно-экзаменационной сессии | Студенты первой ступени высшего образования |
| 1 | 2 | 3 |
| 7. Представление информации в деканаты о сдаче студентами готовых курсовых проектов (работ) руководителям для проверки | *Дневная и вечерняя формы обучения*  Осенний семестр:  1–3 курс – по 27.12  4–5 курс – по 16.12  Весенний семестр:  1–3 курс – по 06.06  4 курс – по 11.05 текущего учебного года  *Заочная (в т. ч. дистанционная) форма обучения*  За три дня до начала экзаменационной сессии | Руководители курсовых проектов (работ) |
| 8. Защита студентами курсовых проектов (работ) | *Дневная и вечерняя формы обучения*  Осенний семестр:  1–3 курс – с 13.12 по 02.01  4–5 курс – с 04.12 по 21.12  Весенний семестр:  1–3 курс – с 23.05 по 12.06  4 курс – с 28.04 по 17.05 текущего учебного года  *Заочная форма обучения:*  В ходе семестра во время консультаций и в течение лабораторно-экзаменационной сессии, но не позднее чем за два дня до экзамена по соответствующей учебной дисциплине (при отсутствии экзамена – до окончания лабораторно-экзаменационной сессии)  *Дистанционная форма обучения*  В течение лабораторно-зачетно-экзамена-ционной сессии, за два дня до экзамена по соответствующей учебной дисциплине | Заведующие кафедрами, руководители курсовых проектов (работ) |
| 9. Представление информации в деканаты о защите студентами курсовых проектов (работ) | *Дневная и вечерняя формы обучения*  Осенний семестр:  1–3 курс – по 03.01  4–5 курс – по 22.12  Весенний семестр:  1–3 курс – по 13.01  4 курс – по18.05 текущего учебного года  *Заочная (в т. ч. дистанционная) форма обучения*  За один день до экзамена по соответствующей учебной дисциплине (при отсутствии экзамена – до окончания лабораторно-экзаменационной сессии) | Заведующие кафедрами,  руководители курсовых проектов (работ) |

# РЕКОМЕНДАЦИИ К ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА

Готовый курсовой проект должен включать в себя спроектированную автоматизированную систему при помощи CASE-средства Rational Rose и печатный вариант записки к курсовому проекту, а также графический материал не менее 2 чертежей в пересчете на формата А1 (4 чертежа формата А2).

Записка курсового проекта имеет следующую структуру

* титульный лист;
* реферат (краткое содержание проекта);
* задание на курсовое проектирование (зеркальные поля, двухсторонняя печать!!!);
* содержание;
* введение;
* основные разделы, содержание которых определяется заданием на проектирование;
* заключение;
* список использованных источников;
* приложения;
* ведомость документов курсового проекта.

Примеры оформления титульного листа, задания по курсовому проекту и ведомости документов приведены в приложениях А, Б и В соответственно.

**Реферат** выполняют по ГОСТ 7.9–95. Слово **РЕФЕРАТ** записывают прописными буквами полужирным шрифтом по центру, страницу не нумеруют, но включают в общее количество страниц пояснительной записки.

В реферате выделяют две составные части: собственно реферативную и заголовочную.

*Заголовочная часть* отражает название темы курсового проекта, фамилию студента с инициалами и выходные данные.

**Пример:**

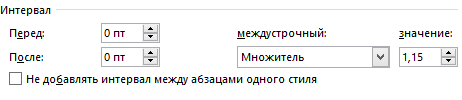
СИСТЕМА ПОЗИЦИОННОГО УПРАВЛЕНИЯ ПОВОРОТНОГО СТОЛА : курсовой проект / В. А. Сергеев. – Минск : БГУИР, 2012, – п.з. – 79 с., чертежей (плакатов) – 2 л. формата А1.

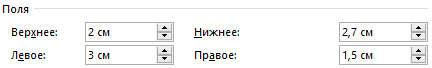
*В реферативной части* кратко излагается содержание дипломного проекта (дипломной работы). Основными аспектами в содержании должны быть: предмет проектирования (исследования); цель работы; данные, относящиеся к методам проектирования; результаты и выводы.

Объем реферата ограничен текстом, который можно разместить на одной странице пояснительной записки. Рекомендуемый объем реферата 850–1200 печатных знаков.

**!!!Пояснительная записка должна быть оформлена в соответствии  
 с СТП 2017 БГУИР!!!**

Шрифт Times New Roman размером 14 пунктов с межстрочным интервалом 1,15.





Номера и названия разделов, подразделов, пунктов и подпунктов следует выделять полужирным шрифтом. **Заголовки** разделов рекомендуется оформлять **полужирным** шрифтом размером 14–16 пунктов, а **подразделов** – полужирным шрифтом 14 пунктов. Заголовки отделяются от текста пробельной строкой.

Абзацы в тексте начинают отступом 1,25.

Слова **СОДЕРЖАНИЕ, ВВЕДЕНИЕ, ЗАКЛЮЧЕНИЕ, СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** записывают прописными буквами **полужирным** шрифтом 14–16 пунктов и располагают по центру строки.

**ВЕДОМОСТЬ ДОКУМЕНТОВ** соответствует составу курсового проекта (работы) и является последним обязательным листом пояснительной записки. Форма ведомости и ее оформление приведена в приложении Д.(СТП -2017).

Основные разделы курсового проекта представлены ниже.

# 1 АНАЛИЗ ТЕХНИЧЕСКОГО ЗАДАНИЯ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

* 1. **Постановка задачи**

Приводится текст задания на курсовое проектирование с формулировкой условия из варианта, анализ задания и формулировку целей на курсовое проектирование.

* 1. **Составление глоссария проекта**

Глоссарий предназначен для описания терминологии предметной области. Он может быть использован как неформальный словарь данных системы. Пример глоссария приведен в таблице 1.

Таблица 1 – Глоссарий проекта

|  |  |
| --- | --- |
| Курс | Учебный курс, предлагаемый университетом |
| Предлагаемый курс (Course Offering) | Предлагаемое чтение данного курса в конкретном семестре (один и тот же курс может вестись в нескольких параллельных сессиях). Включает точные дни недели и время. |
| Каталог курсов | Полный каталог всех курсов, предлагаемых университетом |
| Расчетная система | Система обработки информации об оплате за курсы. |
| Оценка | Оценка, полученная студентом за конкретный курс. Профессор Преподаватель университета. |
| Табель успеваемости (Report Card) | Все оценки за все курсы, полученные студентом в данном семестре. |
| Список курса (Roster) | Список всех студентов, записавшихся на предлагаемый курс. |
| Студент | Личность, проходящая обучение в университете. |
| Учебный график (Schedule) | Курсы, выбранные студентом в текущем семестре |

* 1. **Описание дополнительных спецификаций к проекту**

Назначение дополнительных спецификаций – определить требования к системе регистрации курсов, которые не охватывает модель вариантов использования. Вместе они образуют полный набор требований к системе. Дополнительные спецификации определяют нефункциональные требования к системе, такие, как надежность, удобство использования, производительность, сопровождаемость, а также ряд функциональных требований, являющихся общими для нескольких вариантов использования.

**Функциональные возможности** Система должна обеспечивать многопользовательский режим работы. Если предлагаем курс оказывается заполненным в то время, когда студент формирует свой учебный график, включающий данный курс, то система должна известить его об этом.

**Удобство использования** Пользовательский интерфейс должен быть Windows ХР, 7-10-совместимым.

**Надежность** Система должна быть в работоспособном состоянии 24 часа в день 7 дней в неделю, время простоя – не более 10%.

**Производительность** Система должна поддерживать до 2000 одновременно работающих с центральной базой данных пользователей, и до 500 пользователей, одновременно работающих с локальными серверами.

**Безопасность** Система не должна позволять студентам изменять любые учебные графики, кроме своих собственных, а также не должна позволять профессорам модифицировать конкретные курсы, выбранные другими профессорами.

Только профессора имеют право ставить студентам оценки.

Только регистратор может изменять любую информацию о студентах.

**Проектные ограничения** Система должна быть интегрирована с существующей системой каталога курсов, функционирующей на основе реляционной СУБД.

# ПРОЕКТИРОВАНИЕ СИСТЕМЫ

**2.1 Создание диаграммы вариантов использования**

**Исходя из анализа технического задания**

Действующие лица:

* Student (Студент) – записывается на курсы;
* Professor (Профессор) – выбирает курсы для преподавания;
* Registrar (Регистратор) – формирует учебный план и каталог курсов, ведет все данные о курсах, профессорах и студентах и т.д.;

Исходя из потребностей действующих лиц, выделяются следующие варианты использования:

* Login (Войти в систему);
* Register for Courses (Зарегистрироваться на курсы);
* View Report Card (Просмотреть табель успеваемости);

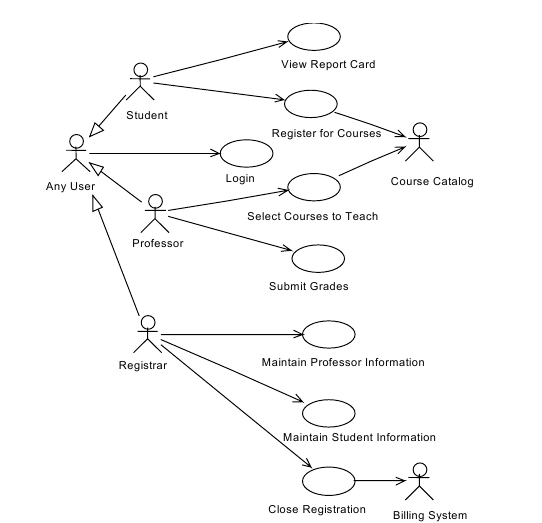


Рисунок 1 – Диаграмма вариантов использования системы

* 1. **Добавление описаний к вариантам использования**

В данном подразделе необходимо описать подробно все варианты использования в пояснительной записке и прикрепить в отдельных файлах к вариантам использования. Пример описания приведен ниже.

**Вариант использования «Login»:**

**Краткое описание**

Данный вариант использования описывает вход пользователя в систему регистрации курсов.

**Основной поток событий**

Данный вариант использования начинает выполняться, когда пользователь хочет войти в систему регистрации курсов.

1 Система запрашивает имя пользователя и пароль.

2 Пользователь вводит имя и пароль.

3 Система проверяет имя и пароль, после чего открывается доступ в систему.

**Альтернативные потоки**

*Неправильное имя/пароль*

Если во время выполнения Основного потока обнаружится, что пользователь ввел неправильное имя и/или пароль, система выводит сообщение об ошибке. Пользователь может вернуться к началу Основного потока или отказаться от входа в систему, при этом выполнение варианта использования завершается.

**Предусловия**

Отсутствуют.

**Постусловия**

Если вариант использования выполнен успешно, пользователь входит в систему. В противном случае состояние системы не изменяется.

* 1. **Создание диаграмм взаимодействия**

В данном подразделе описывается создание **диаграммы последовательности** и **кооперативной диаграммы** для основного потока событий варианта использования. Приводятся скриншоты диаграмм.

* + 1. **Диаграмма последовательности**
    2. **Кооперативная диаграмма**
  1. **Создание диаграммы классов**

В данном подразделе описывается создание диаграммы классов, которое включает проектирование классов системы (добавление атрибутов и операций), размещение классов по пакетам («границы», «сущности», «управление»), создание общей диаграммы классов и установление связей между классами, присвоение классам стереотипов.

* 1. **Создание диаграммы состояний для объектов**

Если некоторый объект всегда одинаково реагирует на событие, то он считается *независимым* от состояния по отношению к этому событию. В отличие от него, *зависимые от состояния объекты* по-разному реагируют на одно и то же событие в зависимости от своего состояния.

Если в системе присутствуют зависимые от состояния объекты со сложной динамикой поведения, то для них можно построить модель, описывающую состояния объектов и переходы между ними. Эта модель представляется в виде диаграмм состояний.

* 1. **Создание диаграммы активности**

Сложные потоки событий (с ветвлениями или альтернативами, циклами) должны быть смоделированы с помощью диаграмм деятельности. Строится на основе диаграммы последовательности.

* 1. **Проектирование баз данных**

Если того требует задание, то в данном подразделе описывается Проектирование реляционных баз данных выполняется с использованием средства ***Data Modeler.*** Его работа основана на известном механизме отображения объектной модели в реляционную. Результатом является построение диаграммы «сущность-связь» и последующая генерация описания БД на SQL.

Проектирование БД состоит из следующих шагов:

**Создание нового компонента – базы данных:**

1. Щелкните правой кнопкой мыши на представлении компонентов.

2. В открывшемся меню выберите пункт Data Modeler > New > Database.

3. Откройте окно спецификации вновь созданного компонента DB\_0 и в списке Target выберите Oracle 8.x.

**Определение устойчивых (persistent) классов:**

1. Откройте окно спецификации класса Student в пакете University

Artifacts.

2. Перейдите на вкладку Detail.

3. Установите значение переключателя Persistence в Persistent.

4. Проделайте такие же действия для классов Classification, FulltimeClassification и ParttimeClassification.

5. Откройте класс Student в браузере, нажав « + ».

6. Щелкните правой кнопкой мыши на атрибуте studentID.

7. В открывшемся меню выберите пункт Data Modeler > Part of Object Identity (указание атрибута в качестве части первичного ключа).

Примечание. Шаги 5, 6 и 7 можно выполнять в Rational Rose, начиная с версии 2001.

**Создание схемы БД:**

1. Щелкните правой кнопкой мыши на пакете University Artifacts.

2. В открывшемся меню выберите пункт Data Modeler > Transform to Data Model.3. В появившемся окне в списке Target Database укажите DB\_0 и нажмите ОК. В результате в логическом представлении появится новый пакет Schemas.

4. Откройте пакет Schemas и щелкните правой кнопкой мыши на пакете <<Schema>> S\_0.

5. В открывшемся меню выберите пункт Data Modeler > New > Data Model Diagram.

6. Откройте пакет, затем откройте вновь созданную диаграмму «сущность-связь» NewDiagram и перенесите на нее все классы- таблицы, находящиеся в пакете <<Schema>> S\_0. Получившаяся диаграмма показана на рисунке 2.

**Генерация описания БД на SQL:**

1. Щелкните правой кнопкой мыши на пакете <<Schema>> S\_0.

2. В открывшемся меню выберите пункт Data Modeler > Forward Engineer.

3. В открывшемся окне мастера Forward Engineering Wizard нажмите Next.

4. Отметьте все флажки генерации DDL и нажмите Next.

5. Укажите имя и расположение текстового файла с результатами генерации и нажмите Next.

6. После завершения генерации откройте созданный текстовый файл и просмотрите результаты.

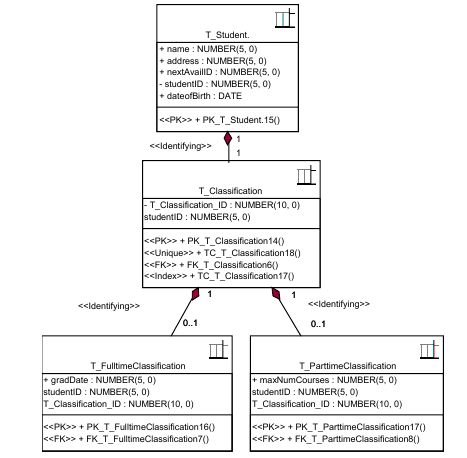


Рисунок 2 – Диаграмма «сущность-связь»

# ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ

**3.1 Создание диаграммы компонентов**

Построить диаграммы компонентов для каждого пакета и для системы в целом.

* 1. **Создание диаграммы размещения**

В данном подразделе разработать диаграмму размещения. В зависимости от варианта задания диаграмма размещения должна показывать расположение компонентов в распределенном приложении или связи между встроенным процессором и устройствами.

* 1. **Генерация кода**

Описать генерацию кода на языке С++. Привести скриншот сгенерированных файлов. Содержимое самих файлов с кодом представить в виде ПРИЛОЖЕНИЯ.

Министерство образования Республики Беларусь

Учреждение образования

БелорусскиЙ государственный университет

информатики и радиоэлектроники

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Дисциплина: Объектно-ориентированное программирование и проектирование

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

Пояснительная записка

к курсовому проекту

на тему

**НАЗвание ТЕМЫ ПРОЕКТА**

БГУИР КП 1-53 01 02 06 номер задания на кп (например, 008) ПЗ

|  |  |  |
| --- | --- | --- |
| Студент |  | В. М. Сидоренко |
| Руководитель |  | Т. С. Боброва |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Минск 2019

**РЕФЕРАТ**

АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА УПРАВЛЕНИЯ ПРОДАЖАМИ БИЛЕТОВ НА ТРАНСПОРТ: курсовой проект / К. Ю. Рыжова. – Минск : БГУИР, 2018, – п.з. –75 с., чертежей (плакатов) – 2 л. формата А1.

Курсовой проект на тему «Автоматизированная информационная система управления продажами билетов на транспорт» разработан с целью повышения степени удовлетворенности клиентов за счет анализа накопленной информации о клиентском поведении, правильной настройке инструментов маркетинга, оптимизации работы сотрудников компании.

Пояснительная записка к курсовому проекту состоит из введения, 4 разделов, включающих литературный обзор по теме дипломного проекта, сравнительный анализ существующих аналогов системы, описание процессов проектирования и программной реализации, а также технико-экономического обоснования разработки и внедрения системы, заключение, список использованных источников и приложение, содержащее листинг кода отдельных классов.

Для разработки автоматизированной информационной системы был выбран объектно-ориентированный язык программирования Java. Система представляет собой веб-приложение доступ, к которому, пользователи смогут получать посредствам веб-браузера. Она предназначена для использования в локальной сети либо сети Интернет.

В результате работы над дипломным проектом была спроектирована и разработана автоматизированная информационная система управления продажами билетов на транспорт, подготовлено руководство пользователя и выполнено экономическое обоснование разработки и внедрения системы.

Результаты, полученные в ходе дипломного проектирования, могут использоваться участниками финансового рынка для анализа рынка производных финансовых инструментов, выбора стратегии и принятия решения при проведении операций хеджирования рисков на рынке ценных бумаг. (здесь про практическую значимость и возможную область применения)

Учреждение образования

«Белорусский государственный университет информатики   
и радиоэлектроники»

Факультет информационных технологий и управления

УТВЕРЖДАЮ

Заведующий кафедрой

*––––––––––––––––––––––––*

(подпись)

–––––––––––––––––––2019г.

ЗАДАНИЕ

по курсовому проектированию

Студенту *Успенскому Эдуарду Алексеевичу–––––––––––––––––––*

1. Тема проекта  Система поддержки составления расписания занятий––––––––––––––––––––– ––  ––––

2. Срок сдачи студентом законченного проекта– –––*20 мая 2019 г.––––*

3. Исходные данные к проекту Система обеспечивает составление расписания некоторого учебного заведения, внесение в него изменений, выдачу полного расписания и дополнительной информации (например, по итоговому расписанию составляется расписание указанной группы или указанного преподавателя 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. Некоторые занятия проводятся для нескольких групп: потока или всего курса.

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 допускается одновременно два различных занятия и т. д.).

Система по требованию пользователя должна осуществлять поиск свободных аудиторий на заданную дату и время.

При работе над этим вариантом задания необходимо разработать схему базы о расписании, курсах, преподавателях.*–––––––––––––––––––––––––––––*

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке):

*Введение*

1. *Анализ технического задания н акурсовое проектирование*
2. *Проектирование системы поддержки составления расписания занятий*
3. *Программная реализация системы поддержки составления расписания занятий*

*Заключение*

*Список использованных источников*

*Приложения*

*Ведомость курсового проекта*

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

*1. Диаграмма вариантов использования (ПЛ, формат А2)–––––––––––––––––*

*2. Диаграмма классов (ПЛ, формат А2) ––––––––––––––––––––––––––––*

*3. Диаграмма активности (ПД, формат А2)––––––––––––––––––––––––––––*

*4. Диаграмма размещения (ПД, формат А2) ––––––––––––––––––––––––––––*

6. Консультант по проекту (с обозначением разделов проекта) *Т. С. Боброва–––––––– ––––––––––––––––––––––––––––––––––––––––––––––––––––– –*

7. Дата выдачи задания –––– –*17 февраля 2019 г.––––––––––––––––––––––  –*

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):

*раздел 1 к 15.03 – 15 %; –– –––––––––––––– –––––––––––––––– –*

*раздел 2 к 15.04 – 50 %;–––––––––––––––– ––––––––––––––––––  ––––*

*раздел 3 к 15.05 – 15 %;––––––––––––––––––––– –––––––––––––––––*

*оформление пояснительной записки и графического   
материала к 20. 05 – 20 %*

*Защита курсового проекта с* ***23.05*** *по* ***05.06.2019****г.––––––––––––––––––––––*

Руководитель*– –––––––––– Т. С. Боброва*

(подпись)

Задание принял к исполнению *–––––––\_\_\_\_\_\_\_\_\_\_\_\_–– Э. А. Успенский*

(дата и подпись студента)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Обозначение* | | | | | *Наименование* | *Дополнительные сведения* | | | | | |
|  | | | | | *Текстовые документы* |  | | | | | |
|  | | | | |  |  | | | | | |
| *БГУИР ДП 1-53 01 02 06 008 ПЗ* | | | | | *Пояснительная записка* | *99 с.* | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | | *Графические документы* |  | | | | | |
|  | | | | |  |  | | | | | |
| *ГУИР 000000.001 ПЛ* | | | | | *Обзор системы управления* |  | | | | | |
|  | | | | | *содержимым CQ5* | *Формат А2* | | | | | |
|  | | | | |  |  | | | | | |
| *ГУИР 000000.003 ПД* | | | | | *Диаграмма классов сервисов* | *Формат А2* | | | | | |
|  | | | | |  |  | | | | | |
| *ГУИР 000000.002 ПЛ* | | | | | *Типичная конфигурация* |  | | | | | |
|  | | | | | *CQ-окружения* | *Формат А1* | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  | | | | |  |  | | | | | |
|  |  |  |  |  | *БГУИР ДП 1-53 01 02 06 008 Д1* | | | | | | |
|  |  |  |  |  |
|  |  |  |  |  |
| *Изм.* | *Л.* | *№ докум.* | *Подп.* | *Дата* | *Система поддержки составления расписания занятий*  *Ведомость курсового проекта* | | *Лит* | | | *Лист* | *Листов* |
| *Разраб.* | | *Студент* |  |  |  | *Т* |  | *99* | *99* |
| *Провер.* | | *Боброва* |  |  | *Кафедра ИТАС*  *гр. 720601* | | | | |
|  | |  |  |  |
|  | |  |  |  |
|  | |  |  |  |
|  | |  |  |  |

**ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Листинг кода**

**Класс Order**

**Order.h**

**(содержимое файла)**

**Order.cpp**

**(содержимое файла)**

Ит.д.

# ВАРИАНТЫ ЗАДАНИЙ НА КУРСОВОЕ ПРОЕКТИРОВАНИЕ

“

В каждом из предложенных вариантов требуется при помощи CASE-средства Rational Rose построить модель программного обеспечения. Процесс создания модели состоит из нескольких этапов:

*Процесс создания модели должен проходить так, как это описано в методическом пособии* [*[Вендров 2004].*](file:///C:\Users\MARYJANE\AppData\Roaming\Microsoft\umlpracticum2004.pdf)

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.

Сложные потоки событий (с ветвлениями, циклами) должны быть смоделированы с помощью **диаграмм деятельности.**

**При проектировании системы** требуется:

* создать иерархию классов системы;
* для классов указать стереотипы;
* разместить классы по пакетам в Design model, как это описано в методичке (придерживаться принципа: количество связей внутри пакета больше количества связей, проходящих за границы пакета);
* связать объекты на диаграммах взаимодействия с классами, а сообщения – 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);
* построить диаграммы классов системы, отображающие связи между классами;
* для описания поведения экземпляров отдельных классов построить диаграммы состояний;
* построить диаграммы деятельности для моделирования сложных операций (с альтернативами, циклами);
* 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 средствами Rational Rose.

Ниже перечислены варианты заданий.

1. [Видеопрокат](#var1)
2. [Торговый автомат](#var2)
3. [Табло на станции метро](#var3)
4. [Онлайновая театральная касса](#var4)
5. [Мини-АТС](#var5)
6. [Управление контактами с клиентами](#var6)
7. [Банкомат](#var7)
8. [Интернет-магазин](#var8)
9. [Библиотечная система](#var9)
10. [Web-форум](#var10)
11. [Каталог ресурсов Web](#var11)
12. [Генеалогическое дерево](#var12)
13. [Информационный киоск метро](#var13)
14. [Турникет метро](#var14)
15. [Игра «Тетрис»](#var15)
16. [Web-сайт авиакомпании](#var16)
17. [Система складского учета](#var17)
18. [Система поддержки составления расписания занятий](#var18)
19. [Телефон](#var19)
20. [Игра «Пасьянс-косынка»](#var20)
21. [Система начисления зарплаты](#var21)
22. [Система управления лифтами](#var22)
23. [Система для ввода информации при приеме сотрудника на работу](#var23)
24. [Служба занятости в рамках вуза](#var24)
25. [Метод потенциалов](#var25)
26. [Венгерский метод](#var26)
27. [Рейтинговая система](#var27)
28. [Контроль знаний](#var28)
29. [Учет машиночасов](#var29)
30. [Выключатель компьютеров класса](#var30)

\* \* \*

**Вариант 1. Видеопрокат**  
*Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение видеопроката*

Пункт проката видео нуждается в компьютерной системе. Его ассортимент составляет около тысячи видеокассет и пятьсот видеодисков. В прокате имеются видеодиски разных форматов: DVD, MPEG4, Blu-Ray, HD-DVD. Фильмы закупаются у разных поставщиков. Обычно один заказ поставщику делается на несколько фильмов. База данных хранит обычную информацию о поставщиках: их адреса, телефонные номера и т. д. В каждом заказе поставщику указывается: перечень фильмов; их количество, форматы кассет/дисков; отпускная цена.

Каждый видеоноситель при поступлении от поставщика снабжается штрих-кодом (содержащим уникальный идентификационный номер) для того, чтобы сканер, интегрированный в систему, мог поддерживать операции выдачи и возврата видеофильмов.

Каждому клиенту при первом обращении в видеопрокат выдается клиентская карточка со штрих-кодом для автоматизации обработки его запросов. Данные о клиенте (ф. и. о., телефон, адрес) заносятся в базу данных.

При выдаче фильма в прокат устанавливается конкретный период проката (исчисляемый в днях). Плата за прокат вычисляется как произведение количества дней на цену одного дня проката. Цена зависит от видеоносителя: кассета или диск; формата диска. Плата за прокат взимается в момент выдачи. За кассеты и диски, возвращенные позже срока, взимается дополнительная плата за период, превышающий срок проката. Если кассета/диск задержаны более чем на два дня, клиента ежедневно уведомляют о задержке. После двух уведомлений о задержке одной и той же кассеты/диска, клиент заносится в список нарушителей. При следующем его обращении в видеопрокат работник проката решает: оставить клиента в списке нарушителей и отказать в обслуживании или удалить из списка нарушителей и обслужить. При порче видеоносителя клиентом с него взимается штраф.

Система должна обладать поисковым механизмом по базе видео. Работники проката должны иметь возможность быстро получить ответ, имеется ли фильм в наличии, в каком количестве и на каких носителях. Если все носители фильма выданы в прокат, то система должна сообщить ближайшую дату возврата.

Постоянные клиенты (к ним относятся те, кто более десяти раз в течении 12 месяцев воспользовался услугами проката) могут оставлять заявки на фильмы, которых нет в прокате и которые не заказаны у поставщика. Фильмы из таких заявок включаются в следующий заказ поставщику, и в момент поступления фильмов от поставщика клиенты уведомляются о выполнении их заявок. Данные о выполненных заявках хранятся в течение 12 месяцев, после чего удаляются.

Клиенту одновременно могут быть выданы несколько кассет или дисков, однако каждому взятому видеоносителю ставится в соответствие отдельная запись. Для каждого выдаваемого напрокат фильма фиксируются дата и время выдачи, стоимость проката, установленный и фактический срок возврата. При возврате запись о покате обновляется, чтобы отразить этот факт. Кроме того, запись хранит информацию о работнике, оформившем прокат. Записи хранятся в течение 12 месяцев, после чего удаляются.

[К списку вариантов](#list)

\* \* \*

**Вариант 2. Торговый автомат**  
*Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение торгового автомата*

В автомате имеется пять лотков для хранения и выдачи товаров.

Внешний вид автомата изображен на рисунке. 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 по линии связи и информирует покупателей (зажигается красная лампочка рядом с лотком данного товара).  
Автомат принимает к оплате бумажные купюры и монеты. После ввода денег клиент выбирает товар нажатием на кнопку нужного лотка и нажимает на кнопку выдачи товара. Выдача товара производится только в том случае, если товар имеется в наличии и если введенная сумма денег не меньше цены товара. Если сумма превышает цену, клиенту выдается сдача. 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.  
На автомате имеется информационное табло, на котором высвечивается текущая сумма денег, принятых автоматом к оплате, и сообщения для клиентов, такие как: "введите деньги", "выберите товар", "нажмите кнопку выдачи", "введенной суммы недостаточно", "товара нет в наличии", "заберите покупку", "заберите сдачу", "заберите деньги".  
В специальном отделении автомата, закрываемом замком, есть сервисная консоль, которая используется обслуживающим персоналом. С консоли производится управление доступом к ящику с деньгами для изъятия выручки, управление доступом к товарным лоткам для загрузки или замены товара, а также ввод данных о товарах на лотках в память автомата. 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.

[К списку вариантов](#list)

\* \* \*

**Вариант 3. Табло на станции метро**  
*Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение табло на станции метро*

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.*

[К списку вариантов](#list)

\* \* \*

**Вариант 4. Онлайновая театральная касса**  
*Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение онлайновой театральной кассы*

Онлайновая театральная касса "Билетов.Нет" представляет собой web-сайт службы бронирования и доставки билетов на спектакли и концерты.  
Перед тем как впервые воспользоваться услугами кассы клиент должен зарегистрироваться. В ходе регистрации он указывает данные о себе (ф. и. о., телефон, адрес электронной почты) и получает логин и пароль (логины и пароли разных клиентов не должны совпадать).  
Войдя в систему, клиент может ознакомиться с афишей, выбрать интересующее его мероприятие, указав название, дату и место проведения. Получив от системы сведения о билетах имеющихся в наличии, пользователь может забронировать нужное ему количество билетов. Билеты бывают разных типов: партер, балкон, ложа, бельэтаж, 1-2-3 ярус, vip-места и т. п. Цена билета зависит от его типа и расположения зрительского места. Билеты могут быть выкуплены в течение трех суток с момента бронирования, но не позднее пяти суток до начала спектакля. Клиент может самостоятельно выкупить забронированные билеты, приехав в офис, или заказать доставку билетов курьером, сделав пометку в заявке и указав адрес доставки. Стоимость доставки зависит от дальности: центр / спальный район / дальний пригород. Клиент может получить информацию обо всех своих заявках с web-страницы онлайновой кассы.  
Заявки клиентов хранятся в системе. В каждой указаны: сведения о клиенте, название спектакля, место и время проведения, количество и тип забронированных билетов, стоимость билетов, время создания заявки, время оплаты, вид доставки (самовывоз / курьер), адрес доставки, стоимость доставки, статус заявки (новая / рабочая / оплаченная / аннулированная). По истечении 12 месяцев с момента создания заявки данные автоматически удаляются из системы.

В обязанности работников онлайновой кассы входит внесение в систему сведений о мероприятиях и об имеющихся в продаже билетах. Данные о мероприятии – вид: концерт / шоу / спектакль; название; описание; место проведения; дата; – хранятся в системе. Один и тот же спектакль может идти в разные дни и в разных местах, но разные спектакли не могут пересекаться по времени и месту проведения. Запись о билете содержит название спектакля, дата, время, место проведения, тип билета, зрительский ряд, зрительское место, цену билета, статус билета (есть в наличии / забронирован / продан / передан для реализации). По истечении 12 месяцев с даты, указанной в билете, данные автоматически удаляются из системы.

Работник кассы, получив новую заявку клиента, связывается с ним для подтверждения и уточнения мест. Согласовав с клиентом зрительские места, работник делает пометку о бронировании билетов в системе (тем самым уменьшается количество билетов, имеющихся в наличии) и меняет статус заявки на "рабочая". После оплаты и/или доставки "рабочей" заявки билеты из заявки помечаются как проданные, а заявка – как оплаченная. За 5 суток до начала спектакля все не проданные билеты передаются для реализации в обычные кассы, в системе они автоматически помечаются как "передан для реализации", заявки на них аннулируются, клиенты, не успевшие оплатить заказанные билеты, информируются о снятии брони. Через 4 суток после создания "рабочие" заявки автоматически аннулируются, бронирование с билетов снимается, клиентам посылается соответствующее сообщение. Также должна быть возможность аннулирования заявок вручную работниками онлайновой кассы. При аннулировании заявки вручную работник должен уведомить клиента, изменить статус заявки, снять бронирование билетов (количество билетов в наличии возрастает).

[К списку вариантов](#list)

\* \* \*

**Вариант 5. Мини-АТС**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение мини-АТС

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 **"Занято"**. В любой момент разговора абонент может положить трубку, при этом мини-АТС получает сигнал **"Конец"**. После получения этого сигнала сеанс обслуживания абонента завершается.  
Если вызываемый абонент не подходит к телефону, то вызывающий абонент может, не дождавшись, повесить трубку. В этом случае мини-АТС получает сигнал **"Конец"** и завершает сеанс. Вызываемому абоненту посылается сигнал **"Сброс"** для отмены вызова.  
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. При завершении сеанса, получив от внешней АТС сигнал **"Занято"** (в случае если вызываемый абонент первым повесил трубку), мини-АТС посылает абоненту сигнал **"Занято"**, ждет сигнала **"Конец"** для завершения обслуживания абонента и передает его внешней АТС. Если вызывавший абонент первым вешает трубку, то мини-АТС получает сигнал **"Конец"** и 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. Если абонент **"000"** долго не отвечает на внешний вызов, от городской АТС может придти сигнал **"Сброс"**. Он передается абоненту **"000"**, и сеанс завершается.

[К списку вариантов](#list)

\* \* \*

**Вариант 6. Управление контактами с клиентами**

Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение для управления контактами с клиентами

Компания, поставляющая оборудование, в рамках обеспечения своей коммерческой деятельности нуждается в системе управления контактами со своей клиентурой. Клиенты делятся на два вида: текущие – те, с которыми у компании заключены договора в текущий момент или ранее, и потенциальные.  
Система управления контактами находится в распоряжении всех работников компании. Система поддерживает функции "постоянного контакта" с наличной и потенциальной клиентской базой, так, чтобы откликаться на ее нужды, получать новые контракты, обеспечивать выполнение старых. Система позволяет сотрудникам планировать задания, которые необходимо провести в отношении контактных лиц. Некоторые сотрудники должны иметь доступ к планированию заданий только для себя, другие – и для других сотрудников, и для себя.  
Система хранит имена, номера телефонов и факсов, почтовые и электронные адреса и т. д. организаций и контактных лиц в этих организациях.  
Каждое задание связано с каким-либо контактным лицом. Примерами заданий являются телефонный звонок, визит, отправка факса, отправка электронного сообщения, проведение презентации и т. д. Некоторые задания связаны с выполнением контракта, например, отправка оборудования, поставка, установка, гарантийный и послегарантийный ремонт. В таких заданиях указывается необходимая информация: номер контракта, серийный номер ремонтируемого оборудования. Каждое задание имеет дату создания – время внесения ее в систему. Некоторые задания имеют срок исполнения – период времени от начальной даты до финальной, другие являются бессрочными. Дата создания задания не может изменяться, а срок исполнения – может. По исполнении задания дата и время его завершения фиксируются.  
Каждое задание имеет автора – сотрудника, который его создал. Исполнителем задания может быть сотрудник, не являющийся автором. Рядовые сотрудники не могут назначать задания кому-либо кроме себя. Менеджеры назначают задания себе или кому-либо из рядовых сотрудников. Менеджер в ходе выполнения созданного им задания может поменять исполнителя.  
Просматривать задание, автором которого является менеджер, может либо автор, либо исполнитель задания. Просматривать задание, автором которого является рядовой сотрудник, может автор и любой менеджер. Задания сотрудника отображаются на экране его компьютера в виде страницы календаря (один день на страницу). Приоритет каждого задания (низкий, средний, высокий) визуально выделяется на экране. Каждый менеджер может помимо своего календаря просматривать календари рядовых сотрудников. Помечать задание как выполненное и указывать дату завершения может либо автор, либо исполнитель задания. Вносить какие-либо другие изменения в задание может только автор. После завершения задания внесение в него изменений не допускается. По прошествии 12 месяцев после даты завершения задания сведения о нем удаляются из системы.  
Администратор системы управляет доступом сотрудников: выдает логины и пароли пользователям, формирует две группы пользователей: менеджеров и рядовых сотрудников. Он также имеет доступ к специальным функциям, например, может изменить автора задания или внести изменения в завершенное задание.  
Система имеет возможности для поиска в базе клиентов и контактных лиц по их атрибутам (названию, городу, имени контактного лица). Система генерирует отчет по исполнению заданий каким-либо сотрудником в течение периода времени, указываемого в параметре отчета. В отчете указывается: общее количество заданий для данного сотрудника в указанный период, сколько заданий завершено вовремя, сколько заданий завершено с нарушением срока исполнения, сколько заданий с истекшим сроком исполнения не завершено, и сколько не завершенных заданий, срок исполнения которых не истек.

[К списку вариантов](#list)

\* \* \*

**Вариант 7. Банкомат**

Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение банкомата

Банкомат – это автомат для выдачи наличных денег по кредитным пластиковым карточкам. В его состав входят следующие устройства: дисплей, панель управления с кнопками, приемник кредитных карт, хранилище денег и лоток для их выдачи, хранилище конфискованных кредитных карт, принтер для печати справок, сервисная консоль. Банкомат подключен к линии связи для обмена данных с банковским компьютером, хранящим сведения о счетах клиентов.  
Обслуживание клиента начинается с момента помещения пластиковой карточки в банкомат. После распознавания типа пластиковой карточки, банкомат выдает на дисплей приглашение ввести персональный код. Персональный код представляет собой четырехзначное число. Затем банкомат проверяет правильность введенного кода, сверяя с кодом, хранящимся на карте. Если код указан неверно, пользователю предоставляются еще две попытки для ввода правильного кода. В случае повторных неудач карта перемещается в хранилище карт, и сеанс обслуживания заканчивается. После ввода правильного кода банкомат предлагает пользователю выбрать операцию. Клиент может либо снять наличные со счета, либо узнать остаток на его счету.  
При снятии наличных со счета банкомат предлагает указать сумму (100, 200, 500, 1000, 5000, 10000 рублей). После выбора клиентом суммы банкомат запрашивает, нужно ли печатать справку по операции. Затем банкомат посылает запрос на снятие выбранной суммы центральному компьютеру банка. В случае получения разрешения на операцию, банкомат проверяет, имеется ли требуемая сумма в его хранилище денег. Если он может выдать деньги, то на дисплей выводится сообщение "Выньте карту". После удаления карточки из приемника, банкомат выдает указанную сумму в лоток выдачи. Банкомат печатает справку по произведенной операции, если она была затребована клиентом.  
Если клиент хочет узнать остаток на счету, то банкомат посылает запрос центральному компьютеру банка и выводит сумму на дисплей. По требованию клиента печатается и выдается соответствующая справка.  
Сервисная консоль, которая используется обслуживающим персоналом, находится в специальном отделении банкомата, закрываемом на замок. С консоли производится управление доступом к хранилищу денег для загрузки банкнот, управление доступом к хранилищу конфискованных карт, запуск самодиагностики банкомата, конфигурация сетевого соединения с банковским компьютером.

[К списку вариантов](#list)

\* \* \*

**Вариант 8. Интернет-магазин**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение Интернет-магазина

Магазин компьютеров предлагает возможность приобретения своих товаров через Интернет. Клиент может выбрать компьютер на web-странице магазина. Компьютеры подразделяются на серверы, настольные и портативные. Заказчик может выбрать стандартную конфигурацию из списка и детально ознакомиться с ней на отдельной web-странице. Если стандартная конфигурация ему не подходит, он может построить требуемую ему конфигурацию в диалоговом режиме. Компоненты конфигурации (такие, как оперативная память, процессор, жесткий диск и т. п.) представляются как список для выбора из доступных альтернатив. Для каждой новой конфигурации система может подсчитать цену.  
Чтобы оформить заказ, клиент должен заполнить электронную форму с адресами для доставки товара и отправки счета-фактуры, а также деталями, касающимися оплаты. Оплата компьютеров осуществляется наличными курьеру, осуществляющему доставку, или банковским переводом на счет Интернет-магазина. После ввода заказа система отправляет клиенту по электронной почте сообщение с подтверждением получения заказа вместе с относящимися к нему деталями (стоимость, номер счета, банковские реквизиты для безналичной оплаты и т. п.). Пока клиент ожидает прибытия компьютера, он может проверить состояние заказа (поставлен в очередь / собран / отправлен). Работник магазина проверяет, поступила ли оплата (в случае безналичного расчета) и делает соответствующую пометку при поступлении денег. Если деньги не поступают в течение 5 банковских дней, заказ аннулируется. После оплаты или в случае оплаты наличными работник печатает счет-фактуру и отправляет ее на склад вместе с требованием заказанной конфигурации. Заказ помечается как поставленный в очередь. Собранный компьютер вместе со счетом-фактурой и накладной передается со склада в отдел доставки, при этом заказ помечается как собранный. Компьютер поставляется клиенту (статус заказа – отправлен). Если заказ оплачивается наличными, курьер по возращении передает деньги в кассу, заказ помечается как оплаченный.  
По окончании работы с заказом, он помечается в системе как выполненный. Заказы хранятся в системе в течение 15 месяцев с момента создания для составления годовых и квартальных отчетов, после чего автоматически удаляются.

[К списку вариантов](#list)

\* \* \*

**Вариант 9. Библиотечная система**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Система автоматизации для библиотеки

Система поддержки управления библиотекой должна обеспечивать операции над данными о читателях (добавление, удаление и изменение). В регистрационном списке читателей хранятся следующие сведения: фамилия, имя и отчество читателя; номер его читательского билета и дата выдачи билета, дата последней перерегистрации.  
Наряду с регистрационным списком системой должен поддерживаться каталог библиотеки, где хранится информация о книгах (наименованиях): 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 или фамилии автора. Читатели имеют доступ только к каталогу книг (они могут осуществлять в нем только поиск и просмотр).  
В системе поддерживается реестр экземпляров всех книг библиотеки. Каждый экземпляр имеет свой уникальный идентификационный номер, вообще говоря, не совпадающий с библиотечным шифром. В системе осуществляется регистрация взятых и возвращенных читателем книг. Про каждый выданный экземпляр в реестре хранится запись о том, кому и когда была выдана книга, и когда она должна быть возвращена. При возврате книги в записи делается пометка, о том, что данный экземпляр находится в наличии и указывается, какой читатель пользовался этой книгой последним. Если экземпляр приходит в негодность, запись реестра о нем удаляется. Если от поставщиков приходят новые книги, записи о них добавляются в реестр и каталог.  
При любом обращении читателя в библиотеку сначала осуществляется проверка, не является ли он нарушителем правил пользования. Нарушителем считается тот читатель, который не вернул по истечении срока какую-либо книгу. Нарушители библиотекой не обслуживаются, до тех пор не вернут книги и не заплатят штраф.  
Перерегистрация читателей проходит раз в два года. Она необходима для поддержания списка читателей в актуальном состоянии. Если какой-либо читатель пропускает перерегистрацию, то по истечении полугода с момента перерегистрации его читательский билет аннулируется, сведения о нем удаляются из системы.  
Система должна выдавать библиотекарям 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 данные о читателях, каталоге книг, реестре экземпляров.

[К списку вариантов](#list)

\* \* \*

**Вариант 10. Web-форум**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение Web-форума

Web-форум состоит из нескольких разделов. В каждом разделе содержатся темы, обсуждаемые его пользователями. Темы в разделе упорядочены по убыванию даты последнего ответа в тему. Каждая тема открывается заглавным сообщением и представляет собой древовидную структуру сообщений. Верхний уровень иерархии составляют сообщения, открывающие новые темы, а подуровни составляют сообщения, полученные в ответ на них и т. д.  
Сообщение состоит из текста и заголовка (который может не совпадать с заголовком темы). Каждое сообщение-ответ содержит ссылку на сообщение, ответом на которое оно является. Сообщения помечены именами их авторов и двумя датами (датой добавления сообщения и датой его последнего изменения).   
Начальной страницей конференции является список разделов, на которой находятся ссылки на первые страницы разделов. Количество тем в разделе может быть большим, поэтому на первой странице раздела отображается список из первых 20 сообщений темы, на второй – следующие 20 и т. д. В списке отображаются только заголовки тем, их авторы и даты последних ответов. Просматривая список, пользователь может перейти на страницу заглавного сообщения темы. Помимо текста заглавного сообщения темы на этой странице отображается список (иерархический) сообщений являющихся ответами на заглавное, ответами на ответы и т. д. С этой страницы пользователь может перейти на страницу сообщения-ответа, на которой также отображается текст сообщения и дерево ответов. На всех страницах сообщений содержатся ссылки на заглавную страницу форума, на страницу текущего форума и на страницу заглавного сообщения темы.  
Просматривать страницы форума могут любые пользователи Web. Зарегистрированные пользователи, осуществляют вход в форум, указывая имя и пароль. После входа пользователь может добавить ответ, заполнив форму на странице сообщения, также он может редактировать свои сообщения (в течение двух недель с момента их создания). Еще он имеет возможность начать новую тему, заполнив форму на странице раздела.  
Регистрирует новых пользователей администратор форума. При регистрации пользователь заполняет специальную форму, содержимое которой затем пересылается администратору и запоминается в базе пользователей. Администратор решает, регистрировать пользователя или нет, и отправляет свой ответ. Администратор может создавать, редактировать или удалять разделы.  
Администратор управляет правами пользователей, он может назначить кого-либо из них модератором (ведущим) какого-либо раздела. У одного раздела может быть несколько ведущих. Модератор имеет право удалять любые сообщения из раздела, редактировать их, переносить темы в другие разделы. Он также может наказывать пользователей, нарушающих правила поведения в форуме, лишая на некоторое время возможности добавлять и редактировать сообщения.  
Вариант задания включает в себя разработку схемы базы данных для хранения разделов, тем и сообщений форума, а также информации о зарегистрированных пользователях.

[К списку вариантов](#list)

\* \* \*

**Вариант 11. Каталог ресурсов Web**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение каталога ресурсов Web

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. Количество ресурсов в разделе может быть большим, поэтому пользователь может выбрать количество, отображаемое на одной странице, например 25, тогда на первой странице раздела отображается список из первых 25 ресурсов, на второй – следующие 25 и т. д. Ресурсы в списке могут быть упорядочены по дате обновления или по названиям (по алфавиту).  
Пользователям каталога должны быть предоставлены возможности по поиску ресурсов в каталоге. 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 результаты поиска 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 информации каталога.

[К списку вариантов](#list)

\* \* \*

**Вариант 12. Генеалогическое дерево**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программная система 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 "братья-сестры". Система обеспечивает возможность добавления данных о новых персонах и родственных связях, изменение введенных данных и удаление ненужных данных. Система следит за непротиворечивостью вводимых данных. 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, если таковые есть (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»).

[К списку вариантов](#list)

\* \* \*

**Вариант 13. Информационный киоск метро**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение информационного киоска метро.

Информационный киоск метро предоставляет пассажирам сведения о метрополитене, помогает им планировать свое передвижение под землей. В состав киоска входят следующие устройства: сенсорный дисплей, видеокамера, микрофон, кнопка тревоги, сервисная консоль.  
Киоск выдает информацию на нескольких языках (русском, английском, японском и др.), выбор языка осуществляется в главном меню киоска. Также в главном меню отображаются разделы информации, отображаемой киоском:

* интерактивная схема метро;
* правила пользования метрополитеном;
* исторические сведения о метрополитене в целом (с фотографиями);
* сведения о том, как доехать до вокзалов, крупных торговых центров, исторических памятников и т. п.

На интерактивной схеме отображаются все линии, их станции, переходы и станции пересадки. На схеме специально выделяется станция, на которой установлен данный киоск. Пользователь может запросить поиск маршрута по схеме, указав стартовую и финишную станции. Киоск высвечивает варианты движения на схеме (их может быть несколько), сопровождая их рассчитанным временем проезда. Пользователь может запросить подробную справку о какой-либо станции, и киоск выдаст в отдельном окне сведения об истории создания станции, фотографии, перечень туристических объектов, расположенных рядом.  
Правила пользования метрополитеном представляют собой перечень положений, который может не поместиться целиком на экране.  
Исторические сведения о метрополитене разбиты на страницы-экраны. Переход со страницы на страницу осуществляется нажатием на стрелки, высвеченные на экране.  
Четвертый раздел представляет собой список вокзалов, крупных магазинов и т. п. Маршрут проезда высвечивается в отдельном окне, если пользователь нажимает на название интересующего его объекта.  
В случае аварии, террористического акта и т. п. пассажир может оставить сообщение, нажав тревожную кнопку. Камера и микрофон записывают сообщение все время, пока кнопка нажата. Сообщение сразу передается по линии связи в отдел УВД при метрополитене.  
Сервисная консоль, которая используется обслуживающим персоналом, находится в специальном отделении киоска, закрываемом на замок. Персонал может провести самодиагностику киоска, нажав кнопку диагностики. Результат отображается на дисплее. Также есть возможность загрузить новую информацию. Для этого к консоли присоединяется носитель данных и нажимается кнопка загрузки. В ходе загрузки на дисплее отображается ход процесса. По окончании носитель данных отсоединяется.  
Разработанная модель должна содержать схему базы данных для хранения схемы метро и содержимого остальных информационных разделов киоска.

[К списку вариантов](#list)

\* \* \*

**Вариант 14. Турникет метро**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение встроенного процессора турникета для метрополитена

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"**, индикатор количества оставшихся поездок.  
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. Каждая карта имеет срок годности, по истечении которого она не может быть использована для прохода. Карты бывают двух типов: с фиксированным количеством поездок и с неограниченным количеством поездок. Турникет считывает с карты данные: срок годности карты, номер карты, тип карты и количество поездок. Если данные не удается считать, или карта просрочена, или количество поездок нулевое, то карта возвращается пассажиру, и турникет остается в исходном состоянии. Иначе с карты с фиксированным количеством поездок списывается одна поездка, карта возвращается из приемника, индикатор **"Стоп"** гаснет, зажигается индикатор **"Проход"**, индикатор количества оставшихся поездок высвечивает текущее значение 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. Если карта имеет неограниченное количество поездок, то ее номер запоминается, чтобы в течение пяти минут после прохода пассажира с этой картой блокировать попытки прохода с ней через все турникеты данной станции метро. При проходе с картой такого типа индикатор количества оставшихся поездок высвечивает символ ∞ (бесконечность).  
Наличие трех оптических 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 два или три датчика одновременно 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 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.

[К списку вариантов](#list)

\* \* \*

**Вариант 15. Игра «Тетрис»**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Игровая программа

Требуется разработать средствами Rational Rose модель игровой программы «Тетрис».  
В ходе игры программа отображает прямоугольное игровое поле размером 50 х 15 клеток – "стакан", в который падают одна за другой кости. Кости состоят из 4-х клеток и бывают следующих видов:

Рис. 15.1.Виды игровых костей.

Игрок во время падения кости может сдвигать ее влево или вправо, поворачивать ее относительно центра и "уронить" ее, не дожидаясь, пока она достигнет дна. Скорость падения кости зависит от количества набранных очков (увеличивается после каждой сотни). Очки начисляются игроку за полностью заполненные горизонтальные ряды (15 очков за ряд). Падение кости прекращается, когда непосредственно под любой из ее клеток оказывается клетка ранее упавшей кости или дно "стакана". По окончании падения анализируется ситуация на игровом поле. Все образовавшиеся заполненные ряды удаляются, при этом начисляются очки. Удаление рядов может привести к изменению положения клеток костей, под которыми удалена опора, и образованию новых заполненных рядов. Например (см. рис. 15.2), после падения квадратной кости удаляется второй снизу заполненный ряд, Т-образная кость теряет опору и смещается на дно, образуется еще один заполненный ряд, который также удаляется.

Рис. 15.2. Изменение игровой ситуации при удалении заполненных рядов.

По окончании анализа игровой ситуации случайным образом определяется тип и цвет очередной кости, которая появляется в середине верхней части "стакана" и начинает падать. Игра заканчивается, если после падения кости какая-либо ее клетка оказывается в верхнем ряду игрового поля. Если результат игрока оказывается одним из 10 лучших, программа запрашивает его имя и сохраняет запись о нем в таблице рекордов. Записи в таблице упорядочены по невозрастанию количества набранных очков. По окончании игры пользователь может начать новую или выйти из программы.  
В ходе игры пользователь выйти из игры, нажав кнопку выхода, при этом можно сохранить текущее состояние игрового поля, чтобы в следующий раз начать играть в сохраненном состоянии. Можно начать игру заново, нажав соответствующую кнопку.   
Для удобства игроков программу можно настроить, указав клавиши для управления падающей костью, клавишу выхода и др.  
В ходе выполнения этого варианта задания должна быть разработана схема базы данных о состоянии игрового поля и таблице рекордов.

[К списку вариантов](#list)

\* \* \*

**Вариант 16. Web-сайт авиакомпании**

Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение web-сайта авиакомпании

Коммерческий отдел авиакомпании предложил расширить свой web-сайт, добавив систему бронирования билетов, чтобы позволить пользователям:

* узнать о выполнении рейсов текущего дня;
* запросить информацию о расписании рейсов, стоимости билетов и наличии мест;
* забронировать билеты.

Постоянные клиенты, которые часто летают самолетами авиакомпании, смогут использовать также следующие функции:

* получить текущую информацию о состоянии своего личного счета (количество километров, проведенных в воздухе с начала года на данное число, премиальные очки за количество налетанных километров, которые могут быть использованы для получения поощрительного вознаграждения (бесплатного перелета) и т. д.;
* забронировать билеты более высокого класса по более низкой цене, если имеется достаточное количество премиальных очков;
* забронировать бесплатные билеты, если имеется достаточное количество премиальных очков.

Чтобы гарантировать тайну частной информации и предотвратить несанкционированное использование данных о постоянных клиентах, при доступе к личным счетам необходимо потребовать, чтобы пользователь вошел в систему, введя логин и пароль. После входа пользователь должен увидеть начальную страницу с учетом его предпочтений и привычек, почерпнутых из базы данных, хранящей информацию о перелетах постоянных клиентов. Постоянные клиенты могут оперативно обновлять сведения о себе. Новым пользователям логины и пароли выдаются при регистрации на сайте.  
Чтобы сэкономить деньги, руководство компании приняло решение использовать ряд существующих систем:

* систему управления счетами, хранящую информацию о постоянных клиентах и балансе премиальных очков;
* маркетинговую базу данных, которая отслеживает данные о выполненных рейсах, тарифах, проданных билетах и др. Эти данные используются для формирования специальных уведомлений, которые включаются в ежемесячные выписки из лицевого счета постоянных клиентов;
* базу данных наличия билетов.

Система бронирования хранит заявки клиентов. Оформляя заявку, клиент указывает: тип билета (в одну сторону или "туда и обратно"); 1 или 2 номера рейсов и 1 или 2 даты вылета (в зависимости от типа билета); класс обслуживания (v.i.p. или эконом); количество мест; признак использования премиальных очков для бесплатного перелета или повышения класса обслуживания. Клиенту высвечивается тариф и общая стоимость заказанных билетов, запрашивается подтверждение брони. После окончания оформления заявки информация передается в базу наличия билетов и количество доступных билетов уменьшается. Клиент самостоятельно выкупает забронированные билеты в какой-либо из касс авиакомпании. Когда билеты выкуплены (информация об этом приходит из маркетинговой базы данных), заявка удаляется.  
Клиент может аннулировать заявку не позднее трех суток до вылета, при этом в БД наличия билетов делаются соответствующие изменения.  
В ходе выполнения этого варианта задания должна быть разработана схема базы данных клиентских заявок, а также логинов и паролей пользователей.

[К списку вариантов](#list)

\* \* \*

**Вариант 17. Система складского учета**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программная система складского учета

На продуктовом складе для каждого товара фиксируется место хранения (определенная полка), количество товара и его наименование. Разные товары имеют разные единицы измерения: штуки, килограммы, коробки, бутылки и др. Система складского учета должна обеспечивать добавление информации о новом товаре, изменение или удаление информации об имеющемся товаре, хранение (добавление, изменение и удаление) информации о поставщиках и покупателях, включающей в себя название фирмы, ее адрес и телефон. В системе учитывается приход товаров от поставщиков. В каждом приходе товаров могут содержаться несколько позиций, в каждой позиции указываются наименование товара и его количество. После оформления прихода товара в системе количество товара в инвентаризационной описи соответственно увеличивается.  
Товар со склада отпускается покупателям по расходным накладным. В каждой накладной могут содержаться несколько позиций, в каждой позиции указываются наименование товара и его количество. После оформления расхода товара в системе количество товара в инвентаризационной описи соответственно уменьшается.  
Система складского учета по требованию пользователя формирует и выдает на печать следующую справочную информацию:

* список всех товаров;
* инвентаризационную опись товаров, имеющихся в наличии;
* список товаров, количество которых необходимо пополнить;
* список товаров, поставляемых данным поставщиком;
* все позиции в каком-либо приходе товара;
* все позиции в какой-либо расходной накладной.

Система осуществляет поиск информации о клиенте или поставщике по части названия фирмы. Это необходимо, чтобы работники склада могли связаться с фирмой по какому-либо вопросу.  
В ходе выполнения этого варианта задания должна быть разработана схема базы данных, хранящей информацию о товарах, приходах и расходах товара, поставщиках и клиентах.

[К списку вариантов](#list)

\* \* \*

**Вариант 18. Система поддержки составления расписания занятий**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Система поддержки составления расписания занятий

Система обеспечивает составление расписания некоторого учебного заведения, внесение в него изменений, выдачу полного расписания и дополнительной информации (например, по итоговому расписанию составляется расписание указанной группы или указанного преподавателя 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. Некоторые занятия проводятся для нескольких групп: потока или всего курса.  
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 допускается одновременно два различных занятия и т. д.).  
Система по требованию пользователя должна осуществлять поиск свободных аудиторий на заданную дату и время.  
При работе над этим вариантом задания необходимо разработать схему базы о расписании, курсах, преподавателях.

[К списку вариантов](#list)

\* \* \*

**Вариант 19. Телефон**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение встроенного микропроцессора телефона

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) от мини-АТС. Аппарат имеет кнопочную панель управления, соединение с линией сети, 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. Пользователь телефона, желающий сделать звонок, снимает трубку. Телефон подает на АТС сигнал **"Трубка"**. При получении ответного сигнала **"Тон"** от АТС телефон воспроизводит звуковой тон **"Готов"** (длинный непрекращающийся гудок) в трубку. При получении сигнала **"Занято"** (посылается если АТС не может сейчас обслужить абонента) 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"** (если вызываемый абонент занят), либо сигнал **"Вызов"**. При получении сигнала **"Вызов"** телефон воспроизводит в трубку длинные гудки до того момента, когда АТС осуществит коммутацию и передаст сигнал **"Данные"**.  
Телефонный громкоговоритель воспроизводит данные, передаваемые с сигналом. 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.  
Получив сигнал **"Занято"** телефон ожидает, когда пользователь повесит трубку, игнорируя нажатия кнопок и т. п.  
После того, как трубка аппарата была повешена, телефон посылает АТС сигнал **"Конец"**, и переходит в начальное состояние. Сеанс завершается.  
В начальном состоянии телефон может получить от АТС сигнал **"Вызов"**, при этом его звонок начинает звонить. Пользователь, принимая вызов, поднимает трубку, при этом АТС получает сигнал **"Трубка"**, осуществляет коммутацию с вызывающим абонентом и начинает передачу голосовых данных. Завершение сеанса такое же, как ранее описанное.  
Если вызываемый абонент не подходит к телефону, то вызывающий абонент может, не дождавшись, повесить трубку. Его телефон в этом случае посылает АТС сигнал **"Конец"** и завершает сеанс. Телефон вызываемого абонента принимает сигнал **"Сброс"**, перестает звонить и переходит в начальное состояние.

[К списку вариантов](#list)

\* \* \*

**Вариант 20. Игра «Пасьянс-косынка»**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Игровая программа

Требуется разработать средствами Rational Rose модель игровой программы "Пасьянс-косынка".  
В игре участвует колода из 54 игральных карт. Устанавливается следующее старшинство (по возрастанию): туз, 2, 3, ..., 10, валет, дама, король. В колоде четыре масти: черные (пики, трефы) и красные (бубны, червы). В начале игры карты располагаются в колоде случайным образом, затем выкладываются на игровое поле. В поле располагаются семь стопок (в первой – одна, во второй – две, ..., в седьмой – семь карт), дом, состоящий из четырех столбцов, и не розданная часть колоды. Каждая стопка состоит из открытой и закрытой части. В начале игры в каждой стопке открыта лишь последняя карта. В не розданной колоде открыта верхняя карта.  
Цель игрока состоит в том, чтобы сложить в столбцах дома четыре масти карт по порядку старшинства (от туза до короля). Для этого он может делать следующие ходы с открытыми картами:

* поместить туза в пустой столбец дома, если туз является последним в стопке или верхним в открытой части колоды;
* поместить последнюю карту стопки или верхнюю карту открытой части колоды в дом, если она подходит по масти и старшинству (например, 3♠ на 2♠);
* поместить последнюю карту стопки или верхнюю карту открытой части колоды или верхнюю карту столбца дома в другую стопку так, чтобы соблюдался порядок – соседние карты имеют масти разного цвета и расположены подряд по убыванию старшинства (например, Д♠ переложить на К♡);
* сделать такой же ход с несколькими подряд идущими картами какой-либо стопки, включая последнюю(!) (например, 10♠, 9♡, 8♣, 7♢ переложить на В♢ из другой стопки);
* положить короля или несколько подряд идущих начинающихся с короля карт из стопки, включая последнюю(!), в начало пустой стопки.

Помимо указанных ходов пользователь может:

* открыть последнюю закрытую карту стопки, если ее открытая часть пуста;
* открыть закрытую верхнюю карту колоды и положить ее сверху открытой части колоды;
* повторно закрыть колоду, если она вся была открыта, при этом порядок карт в колоде сохраняется.

Любое последнее действие пользователя может быть им отменено. Неправильные ходы не допускаются. Корректное состояние игры должно восстанавливаться после неверных ходов.  
Игра заканчивается, если нельзя сделать ни одного из указанных ходов, либо если все карты оказались в доме. В ходе игры ведется подсчет очков:

* за перемещение карты в дом +10 очков;
* за возврат карты из дома в стопку -15 очков;
* за перемещение карты из колоды в стопку +5 очков;
* за открытие карты в стопке +5 очков;
* за каждый повторный просмотр колоды -100 очков;
* при отмене пользователем последнего действия начисленные очки (в плюс или в минус) снимаются.

По окончании игры пользователь может разложить следующий пасьянс, при этом набранные очки сохраняются. В ходе игры пользователь может нажать на кнопку выхода. Если результат игрока оказывается одним из 10 лучших, программа запрашивает его имя и сохраняет запись о нем в таблице рекордов. Записи в таблице упорядочены по невозрастанию количества набранных очков. Перед выходом можно сохранить состояние игрового поля, чтобы в следующий раз начать играть в сохраненном состоянии.  
В любой момент можно начать игру заново, нажав соответствующую кнопку, при этом все набранные очки сгорают.  
В ходе выполнения этого варианта задания должна быть разработана схема базы данных о состоянии игрового поля и таблице рекордов.

[К списку вариантов](#list)

\* \* \*

**Вариант 21. Система начисления зарплаты**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение системы начисления зарплаты

Перед информационной службой компании поставлена задача создания новой системы начисления зарплаты взамен морально устаревшей существующей системы. Новая система должна предоставлять служащим возможность записывать электронным способом информацию из карточки учета рабочего времени и автоматически формировать чеки на оплату, учитывающие количество отработанных часов и общий объем продаж (для служащих, получающих комиссионное вознаграждение).  
Новая система должна предоставлять служащим возможность вводить информацию об отработанном времени, вводить заказы на поставку, изменять свои параметры (такие, как способ оплаты за работу), и формировать различные отчеты. Система должна работать на персональных компьютерах служащих всей компании. В целях обеспечения безопасности и аудита служащие должны иметь возможность доступа и редактирования только своих собственных карточек учета рабочего времени и заказов на поставку.  
В системе должна храниться информация обо всех служащих компании в различных странах. Система должна обеспечивать правильную и своевременную оплату работы каждого служащего в соответствии с указанным им способом. Компания из соображений экономии расходов желает сохранить без изменений одну из существующих баз данных (БД управления проектами), которая содержит всю информацию относительно проектов и тарифов. БД управления проектами функционирует в среде DB2 на мейнфрейме IBM. Новая система может читать данные из БД управления проектами, но не может обновлять их.  
Некоторые служащие получают почасовую зарплату. Она начисляется на основе карточек учета рабочего времени, каждая из которых содержит дату и количество часов, отработанных в соответствии с конкретным тарифом. Если какой-либо служащий отработал в день более 8 часов, сверхурочное время оплачивается с коэффициентом 1.5. Служащие-почасовики получают зарплату каждую пятницу.  
Некоторые служащие получают фиксированный оклад, однако они тоже представляют свои карточки учета рабочего времени. Благодаря этому система может вести учет количества часов, отработанных в соответствии с конкретными тарифами. Такие служащие получают зарплату в последний рабочий день месяца.  
Некоторые из служащих с фиксированным окладом также получают комиссионное вознаграждение, учитывающее объем продаж. Они представляют заказы на поставку, отражающие дату и объем продаж. Процент комиссионного вознаграждения определяется индивидуально для каждого служащего и может составлять 10%, 15%, 25% или 35%.  
Одной из наиболее часто используемых возможностей новой системы является формирование различных отчетов. запросить количество отработанных часов, суммарную зарплату, оставшееся время отпуска и т.д.  
Служащие могут выбирать способ оплаты за работу. Они могут получать свои чеки на оплату по почте, на счет в банке или на руки в офисе.  
Администратор системы ведет информацию о служащих. В его обязанности входит ввод данных о новых служащих, удаление данных и изменение любой информации о служащем, такой, как имя, адрес и способ оплаты, а также формирование различных отчетов для руководства.  
Приложение начисления зарплаты запускается автоматически каждую пятницу и в последний рабочий день месяца, рассчитывая в эти дни зарплату соответствующих служащих. Начисление зарплаты должно производиться автоматически, без ручного вмешательства.

[К списку вариантов](#list)

\* \* \*

**Вариант 22. Система управления лифтами**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение системы управления лифтами

В 100-этажном здании для перемещения между этажами жильцов и посетителей используется система из 10-ти лифтов. Каждый лифт состоит из шахты, кабины и мотора, приводящего кабину в движение. Внутри шахт на каждом этаже находятся датчики прибытия кабины на этаж. В кабине находятся следующие устройства:

* кнопки назначения (по одной на этаж), с помощью которых пассажиры указывают нужные им этажи;
* кнопка "СТОП", препятствующая закрыванию дверей;
* индикатор текущего этажа, отображающего этаж, на котором остановилась или мимо которого проезжает кабина;
* индикатор направления, в котором движется или будет двигаться кабина;
* датчик перегрузки;
* индикатор перегрузки, загорающийся при переполнении кабины;
* автоматически открывающиеся и закрывающиеся двери.

На этажах кроме первого и последнего расположены две кнопки вызова (вверх и вниз, на первом – только вверх, на последнем – только вниз), а также возле каждой лифтовой шахты находятся индикаторы текущего этажа и направления движения кабины. На первом этаже находится переключатель аварийного режима, при переключении которого все кабины должны спуститься на первый этаж и открыть двери. При обратном переключении после спуска всех кабин лифты переходят в обычный режим работы.  
Все лифты управляются единой программной системой. Управление должно обеспечить доставку пассажиров на нужный им этаж. Например, поднимающаяся кабина не должна двигаться вниз, до тех пор, пока не остановится на всех назначенных пассажирами этажах. Аналогично, опускающаяся кабина не должна двигаться вверх, пока не доставит всех желающих спуститься. Движущиеся кабины (пустые или полные) должны останавливаться на промежуточных этажах, забирая попутных пассажиров. Кабина должна открыть двери, ждать 15 секунд для входа/выхода пассажиров, после чего закрыть двери. При переполнении кабины должен загораться индикатор перегрузки, перегруженная кабина не должна закрывать двери и приходить в движение. Нажатие на кнопку "СТОП", пока кабина неподвижна, должно вызывать открытие дверей, 15-тисекундное ожидание и закрытие дверей. В кабине, пришедшей в движение, нажатие на кнопку "СТОП" должно игнорироваться. Система должна принять нужные пассажирам этажи, реагируя на нажатия кнопок назначения. О принятии назначения сигнализирует подсветка кнопки. Когда назначение выполнено, подсветка выключается. Если вошедшие пассажиры назначают этажи в направлении, не совпадающем с текущим, такие назначения должны игнорироваться. Для экономии энергии остановившиеся кабины без пассажиров должны быть неподвижны, пока нет вызовов.

[К списку вариантов](#list)

\* \* \*

**Вариант 23. Система для ввода информации при приеме сотрудника на работу**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения. Программное обеспечение для приема сотрудника на работу

Подразделение/кафедра выбирается путем позиционирования на определенной (нужной) строке. Подразделения меняются сравнительно часто и поэтому их названия и коды хранятся в соответствующих справочниках.  
Должности также выбираются в соответствии со штатным расписанием. Если соответствующие должности уже заняты, то ввод не может быть осуществлен.  
Каждому сотруднику в соответствии с имеющимися ограничениями должен быть присвоен тарифный разряд.  
Наряду с другими сведениями, вводится информация о знании иностранных языков. Языки выбираются из списка. Выбор может быть множественным (т.е. сотрудник может владеть более чем одним языком). Все выбранные позиции должны быть видны; пользователь может корректировать свой выбор перед тем, как окончательно занести данные в БД. Кроме названия языка, фиксируется еще и степень владения языком.  
Запрещается принимать на работу лиц пенсионного возраста.  
При приеме на преподавательскую должность образование должно быть высшим.  
При приеме на должность выше ассистента научно-педагогический стаж должен быть больше 3 лет. Научно-педагогический стаж не может быть больше общего стажа.  
После занесения всей информации о сотруднике в БД каждому сотруднику автоматически присваивается табельный номер.

[К списку вариантов](#list)

\* \* \*

**Вариант 24. Служба занятости в рамках вуза**  
Тема: Объектно-ориентированный анализ и проектирование программного обеспечения системы службы занятости

Система предназначена для того, чтобы помочь студенту устроиться на работу уже в процессе его обучения в вузе. Подав заявление в систему, студент становится ее клиентом и начинает обслуживаться на протяжении всего обучения в вузе. Заявление представляет собой анкету. Система предлагает профессиональные (основанные на изучаемых предметах), психологические тестирования, проводимые регулярно (раз в семестр (полгода). Особое внимание уделяется обучению студента, по итогам успеваемости составляются экспертные оценки. На основе собранной информации составляется резюме, представляющее собой полную характеристику человека. Это резюме отсылается всем организациям, имеющим необходимые вакансии.  
Основным назначением системы является автоматизация ввода и хранения отчетных данных по студентам, составления характеристик и резюме, поиска вакансий в фирмах. Система позволяет изменять, дополнять, вести поиск и просмотр информации о студентах, накладывать ограничения доступа к системе, хранить списки студентов, закончивших обучение, в виде архива, контролировать выдачу студенту заданий на курсовые работы и проекты, связывать институт с фирмами, заинтересованными в поиске сотрудников.  
Данная система также может быть использована для составления отдельных списков групп, печати зачетных ведомостей, печати полной базы данных и для статистики.  
Система состоит из четырех подсистем:

* контроля успеваемости студентов;
* профессиональных и психологических тестов;
* обработки запросов, определения категорий полномочий пользователей;
* экспертных оценок.

Подсистема контроля успеваемости студентов отвечает за статистическую отчетность по успеваемости отдельного студента, группы или целого факультета, а также за хранение и правильность ее ввода.  
Входными данными подсистемы являются: оценки, даты сдачи экзаменов, имена студентов, номера групп, факультет. На выходе подсистема выдает обработанные данные: средний балл по студенту, группе или факультету, процентное соотношение оценок у студента в группе или на факультете, имена и количество стипендиатов в группе или на факультете. Подсистема "Контроль успеваемости студентов" может функционировать отдельно от всей системы, что дает возможность установить и использовать ее независимо, если это необходимо.

[К списку вариантов](#list)

**Вариант 25. Метод потенциалов**

Тема: Объектно-ориентированный анализ и проектирование программного обеспечения учебного процесса

Представить модуль интерактивного решения транспортных задач методов потенциалов с возможностью работы в демонстрационном и тестовом режимах

[К списку вариантов](#list)

**Вариант 26. Венгерский метод**

Тема: Объектно-ориентированный анализ и проектирование программного обеспечения учебного процесса

Представить модуль интерактивного решения задач о назначении венгерским методом с возможностью работы в демонстрационном и тестовом режимах

[К списку вариантов](#list)

**Вариант 27. Рейтинговая система**

Тема: Объектно-ориентированный анализ и проектирование программного обеспечения учебного процесса

Предложить модуль расширения рейтинговой системы факультета информационных технологий и управления БГУИР, исключающий этап ручного расчета и ввода промежуточных оценок по списку студентов, максимально используя данные рабочего учебного плана групп, текущие данные о фактах посещения занятий, выполнения и защиты лабораторных работ, а также полученных оценок

[К списку вариантов](#list)

**Вариант 28. Контроль знаний**

Тема: Объектно-ориентированный анализ и проектирование программного обеспечения учебного процесса

Выполнить анализ и обратное проектирование системы контроля знаний, представленной сайтом "http:\\infotest.by"

[К списку вариантов](#list)

**Вариант 29. Учет машиночасов**

Тема: Объектно-ориентированный анализ и проектирование программного обеспечения

Предложить систему автоматического контроля использования и степени загрузки персональных ЭВМ класса для диагностики непригодности конкретных ЭВМ к выполнению конкретных работ в классе

[К списку вариантов](#list)

**Вариант 30. Выключатель компьютеров класса**

Тема: Объектно-ориентированный анализ и проектирование программного обеспечения

Предложить систему автоматического выключения всех компьютеров класса после завершения лабораторных занятий по команде обслуживающего персонала или по расписанию. Компьютер должен завершать работу, используя стандартные интерфейсы операционной системы.

[К списку вариантов](#list)