


ДЕПАРТАМЕНТ ВНУТРЕННЕЙ И КАДРОВОЙ ПОЛИТИКИ БЕЛГОРОДСКОЙ ОБЛАСТИ
ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«АЛЕКСЕЕВСКИЙ КОЛЛЕДЖ»

УТВЕРЖДАЮ

Зам. директора


И.А. Злобина

20.08.2019

**КОМПЛЕКТ
КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ**

по УП.02 Учебная практика

09.02.07 Информационные системы и программирование

Алексеевка, 2019

Комплект контрольно – оценочных средств учебной практики разработан на основе Федерального государственного образовательного стандарта по специальности среднего профессионального образования 09.02.07 Информационные системы и программирование и с учетом профессионального стандарта «Специалист по информационным системам», утвержденного приказом Министерства труда и социальной защиты Российской Федерации от «18» ноября 2014 г. №896н

Рассмотрено на заседании предметно-цикловой комиссии общепрофессиональных дисциплин и профессиональных модулей специальностей 09.02.04 Информационные системы (по отраслям) и 09.02.07 Информационные системы и программирование

Протокол № 1 от «30» 09 2019 г.

Председатель  И.В. Косинова

Разработчик:

Е.И. Капустина, преподаватель ОГАПОУ «Алексеевский колледж»

СОДЕРЖАНИЕ

1. Область применения комплекта контрольно-оценочных средств	4
2. Паспорт комплекта контрольно-оценочных средств	5
3. Комплект контрольно-оценочных средств	6
4. Условия выполнения контрольно-оценочных средств	23
5. Информационное обеспечение	23
6. Критерии оценивания ответов обучающихся	24

1. Область применения комплекта контрольно-оценочных средств

Контрольно-оценочные средства (КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу УП.02 Учебная практика. КОС включают контрольные материалы для проведения промежуточной аттестации в форме дифференцированного зачета. КОС разработаны в соответствии с программой УП.02 Учебная практика по специальности СПО 09.02.07 Информационные системы и программирование.

Количество часов на освоение программы учебной практики:

Количество часов по программе, из них:	72
- теоретических	-
- практических	72
- лабораторных	-
- самостоятельная работа	-
Семестры изучения	4
Промежуточная аттестация	Дифференцированный зачет

2. Паспорт комплекта контрольно-оценочных средств

УП.02 Учебная практика

В результате аттестации по учебной практике осуществляется комплексная проверка следующих умений и знаний:

Результаты обучения (освоенные умения, усвоенные знания)	Формы и методы контроля и оценки результатов обучения
<p><u>освоенные умения:</u> использовать выбранную систему контроля версий; использовать методы для получения кода с заданной функциональностью и степенью качества.</p> <p><u>усвоенные знания:</u> модели процесса разработки программного обеспечения; основные принципы процесса разработки программного обеспечения; основные подходы к интегрированию программных модулей; основы верификации и аттестации программного обеспечения.</p>	<ol style="list-style-type: none">1. Интерпретация результатов наблюдений за деятельностью студентов в процессе освоения образовательной программы.2. Текущий контроль в форме: - защиты практических занятий;3. Промежуточная аттестация в форме дифференцированного зачета.

3. Комплект контрольно-оценочных средств

КОС предназначен для контроля и оценки результатов освоения УП.02 Учебная практика по специальности СПО 09.02.07 Информационные системы и программирование.

Вопросы к дифференцированному зачету

- 1 Понятия требований, классификация, уровни требований.
- 2 Методологии и стандарты, регламентирующие работу с требованиями
- 3 Современные принципы и методы разработки программных приложений.
- 4 Методы организации работы в команде разработчиков.
- 5 Системы контроля версий.
- 6 Основные подходы к интегрированию программных модулей.
- 7 Стандарты кодирования.
- 8 Описание требований: унифицированный язык моделирования - краткий словарь.
- 9 Диаграммы UML.
- 10 Описание и оформление требований (спецификация).
- 11 Анализ требований и стратегии выбора решения.
- 12 Цели и задачи и виды тестирования.
- 13 Стандарты качества программной документации.
- 14 Меры и метрики.
- 15 Опишите метрики Чидамбера и Кемерера и методику их применения.
- 16 Тестовое покрытие. Тестовый сценарий, тестовый пакет
- 17 Анализ спецификаций. Верификация и аттестация программного обеспечения.
- 18 Дайте определение понятия проект. Охарактеризуйте состав и структуру коллектива разработчиков, их функции.
- 19 Охарактеризуйте структурный подход к проектированию ИС. CASE - средства разработки ПО.
- 20 Опишите как осуществляется моделирование потоков данных (процессов). Внешние сущности. Системы и подсистемы. Процессы. Накопители данных. Потоки данных. Построение иерархии диаграмм потоков данных.
- 21 Охарактеризуйте метод моделирования IDEF3.
- 22 Охарактеризуйте, что представляет собой методология DFD как инструмент моделирования потоков данных. Опишите инструменты функционального моделирования бизнес-процессов и использованием стандарта IDEF0.
- 23 Сформулируйте понятие и принципы работы с инструментальными средствами разработки ПО
- 24 Опишите методы организации коллективной разработки ПО.
- 25 Охарактеризуйте процесс разработки сетевой модели.

- 26 Опишите элементы Microsoft Office Project 2007.
- 27 Опишите элементы графической нотации DFD.
- 28 Опишите элементы методологии IDEF0.
- 29 Охарактеризуйте процесс имитационного моделирования.
- 30 Опишите Case-метод Баркера.
- 31 Объясните как осуществляется генерация кода клиентской части с помощью ERwin.
- 32 Опишите нотацию ARIS eEPC.
- 33 Охарактеризуйте модель AS-IS.
- 34 Охарактеризуйте модель TO-BE.

Практические задания

Вариант 1.

1. Диаграмма классов.

2. Цифровой диктофон.

Программное обеспечение цифрового диктофона

Требуется разработать средствами UML модель программного обеспечения, управляющего работой цифрового диктофона.

Цифровой диктофон – это бытовое электронное устройство, предназначенное для записи и воспроизведения речи. Звуковые сообщения записываются через встроенный микрофон и сохраняются в памяти устройства. Сообщения воспроизводятся через встроенный громкоговоритель. Работа устройства осуществляется под управлением центрального процессора.

Примерный внешний вид устройства изображен на рисунке.

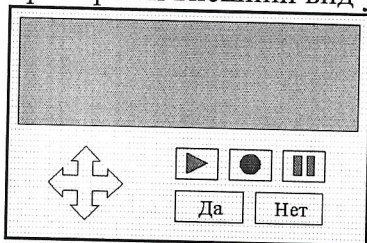


Рис. 4.1. Внешний вид диктофона

Диктофон хранит до 10 звуковых сообщений. Длина каждого сообщения ограничена размером свободной памяти. Диктофон осуществляет прямой (по номеру сообщения) доступ к любому сообщению из памяти. Пользователь имеет возможность воспроизводить сообщения, хранящиеся в памяти диктофона, стирать их, записывать новые. Исполнителем должна быть разработана схема базы данных для хранения сообщений в памяти диктофона.

Интерфейс с пользователем осуществляется при помощи экранного меню и управляющих кнопок на корпусе диктофона. При помощи кнопок-стрелок осуществляется навигация по пунктам меню. Кнопки «Да», «Нет» служат для подтверждения или отмены пользователем выбора той или иной опции меню (структуру меню исполнитель должен разработать самостоятельно). Имеются также кнопки «Воспроизведение», «Пауза» и «Запись» для работы со звуковыми сообщениями.

Во время записи сообщения на экране отображается время, в течение которого ведется запись, при воспроизведении – длительность воспроизведенной части сообщения. Если диктофон не используется, через 30 секунд он автоматически переходит в режим сбережения энергии. В этом режиме никакие операции над звуковыми сообщениями не возможны. Энергия расходуется только на сохранение памяти диктофона в неизменном состоянии. Переход из режима сбережения энергии в обычный режим осуществляется при нажатии пользователем любой кнопки.

В диктофоне имеется датчик уровня заряда батарей. При падении уровня заряда ниже установленного предела диктофон автоматически переходит в режим сбережения энергии (независимо от того используется он в данный момент или нет). Переход в обычный режим становится возможным только после восстановления нормального уровня заряда батарей.

Вариант 2.

1. Диаграмма вариантов использования.

2. Торговый автомат

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

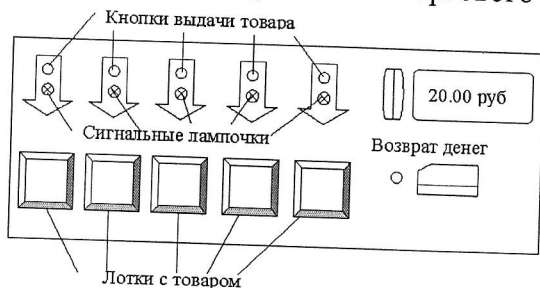


Рис. 4.2. Лицевая панель торгового автомата

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

Вариант 3.

1. Диаграмма последовательностей.

2. Табло на станции метро

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

Табло расположены на каждой станции метро. Они работают под управлением единого пункта управления (ПУ) информационной службы метро. Табло отображает текущее время (часы, минуты, секунды) и время, прошедшее с момента отправления последнего поезда (минуты, секунды). Момент прибытия и отправления поезда определяется при помощи датчиков, устанавливаемых на путях. Все табло метро синхронизованы, текущее время отсчитывается и устанавливается из центральной службы времени, находящейся на ПУ.

На табло высвечивается конечная станция назначения прибывающего поезда. Эти данные

содержатся в расписании движения поездов, которое хранится в памяти табло и периодически обновляется с ПУ.

В «бегущей строке» табло отображается рекламная информация. Память табло хранит до 10 рекламных сообщений. Сообщения отображаются друг за другом с небольшими паузами, циклически. Содержание рекламных сообщений поступает с ПУ. Дополнительная функция табло – по запросу с ПУ оно пересылает данные о нарушениях расписания (преждевременных отправлениях поездов или опозданиях). В ходе выполнения задания должна быть создана схема базы данных для хранения рекламных сообщений, расписания и сведений о нарушении расписаний. *Пояснение: в задании требуется разработать модель ПО только для табло, но не для пункта управления информационной службы.*

Вариант 4.

1. Диаграмма кооперации

2. Система автоматизации для пункта проката видеокассет

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

При необходимости служащий может запросить у системы следующие данные:

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

Постоянным клиентам предоставляются скидки, а также от них принимаются заявки на пополнение ассортимента кассет. Заявки регистрируются в системе. По ним готовится итоговый отчет, руководствуясь которым, служащие пункта проката обновляют ассортимент кассет.

Необходимо разработать схему базы данных для хранения каталога, учетных записей о прокате кассет и заявок на пополнение ассортимента.

Вариант 5.

1. Диаграмма состояний

2. Мини-АТС

Требуется разработать средствами UML модель программного обеспечения встроенного микропроцессора учрежденческой мини-АТС (автоматической телефонной станции). Мини-АТС осуществляет связь между служащими учреждения. Каждый абонент подключен к ней линией связи. Мини-АТС соединяет линии абонентов (осуществляет коммутацию линий). Абоненты имеют номера, состоящие из трех цифр. Специальный номер «9» зарезервирован для внешней связи.

Телефонное соединение абонентов производится следующим образом. Абонент поднимает трубку телефона, и мини-АТС получает сигнал «Трубка». В ответ мини-АТС посылает сигнал «Тон». Приняв этот сигнал, абонент набирает телефонный номер (посылает три сигнала «Цифра»). Мини-АТС проверяет готовность вызываемого абонента. Если абонент не готов (его линия занята), мини-АТС посылает вызываемому абоненту сигнал «Занято». Если абонент готов, мини-АТС посылает обоим абонентам

сигнал «**Вызов**». При этом телефон вызываемого абонента начинает звонить, а вызывающий абонент слышит в трубке длинные гудки. Вызываемый абонент снимает трубку, и мини-АТС получает от него сигнал «**Трубка**», после чего осуществляет коммутацию линии. Абоненты обмениваются сигналами «**Данные**», которые мини-АТС должна передавать от одного абонента к другому. Когда один из абонентов опускает трубку, мини-АТС получает сигнал «**Конец**» и посылает другому абоненту сигнал «**Тон**». В любой момент абонент может положить трубку, при этом мини-АТС получает сигнал «**Конец**». После получения этого сигнала сеанс обслуживания абонента завершается. Если абонент желает соединиться с абонентом за пределами учреждения, то он набирает номер «9». Мини-АТС посылает по линии, соединяющей с внешней (городской) АТС, сигнал «**Трубка**» и в дальнейшем служит посредником между телефоном абонента и внешней АТС. Она принимает и передает сигналы и данные между ними, не внося никаких изменений. Единственное исключение касается завершения сеанса. Получив от городской АТС сигнал «**Конец**», мини-АТС посылает абоненту сигнал «**Тон**» и ждет сигнала «**Конец**» для завершения обслуживания абонента. Если 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.

Вариант 6.

1. Диаграмма деятельности

2. Телефон

Требуется разработать средствами UML модель программного обеспечения встроенного микропроцессора для аппарата учрежденческой телефонной сети. Аппарат подключен к линии связи, ведущей к мини-АТС. В его задачу входит прием и передача сигналов (в том числе и голосовых данных) мини-АТС. Аппарат имеет кнопочную панель управления, экран для отображения набираемых номеров, звонок и трубку, в которую встроены микрофон и громкоговоритель. В начальном состоянии трубка телефона повешена, телефон не реагирует на нажатия кнопок. Телефон реагирует только на сигнал «**Вызов**» от мини-АТС, при этом включается звонок.

При снятии трубки на АТС подается сигнал «**Трубка**». При получении ответного сигнала «**Тон**» от АТС телефон воспроизводит звуковой тон «**Готов**» (длинный непрерывающийся гудок) в трубку. При получении сигнала «**Занято**», в трубке воспроизводится тон «**Занято**» (частые короткие гудки).

Пользователь, слыша в трубке тон «**Готов**», набирает трехзначный номер. Номер может быть набран при помощи кнопок с цифрами или нажатием на специальную кнопку «**#**». При нажатии на кнопку с цифрой соответствующий ей сигнал «**Цифра**» передается АТС. Нажатия на кнопки с цифрами после третьего игнорируются. Во время набора номера введенные цифры отображаются на экране. Последний полностью набранный номер запоминается в памяти аппарата для того, чтобы можно было его воспроизвести при нажатии на кнопку «**#**». При нажатии на эту кнопку номер из памяти аппарата высвечивается на экране, и АТС передается последовательность из трех сигналов «**Цифра**». В ответ на набранный номер от АТС приходит либо сигнал «**Занято**», либо сигнал «**Вызов**». При получении сигнала «**Вызов**» телефон воспроизводит в трубку длинные гудки до того момента, когда АТС осуществит коммутацию и передаст сигнал «**Данные**».

Телефон воспроизводит данные, передаваемые с сигналом, в трубку. Ответ пользователя воспринимается микрофоном трубки, преобразуется в сигнал «**Данные**» и передается АТС. Обмен данными прерывается, если повешена трубка одного из телефонов, участвующих в обмене. О том, что трубку повесил вызываемый абонент, сообщает сигнал

«Занято», посылаемый АТС. После того, как трубка аппарата была повешена, телефон посылает АТС сигнал «Конец», и телефон переходит в начальное состояние.

Вариант 7.

1. Диаграмма объектов

2. Стиральная машина

Требуется разработать средствами UML модель программного обеспечения встроенного микропроцессора стиральной машины.

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

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

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

Вариант 8.

1. Диаграмма компонентов

2. Таксофон

Требуется разработать средствами UML модель встроенной системы управления работой таксофона городской телефонной сети.

Таксофон предназначен для оказания платных услуг телефонной связи. Он подключен к линии связи. В нем имеется кнопочная панель, дисплей, трубка со встроенным микрофоном и громкоговорителем, приемник карт – устройство для считывания телефонных карт, используемых для оплаты разговора.

В начальном состоянии трубка таксофона повешена, дисплей потушен, таксофон не реагирует на нажатия кнопок и какие-либо сигналы из линии. При снятии трубки таксофон выдает на дисплей сообщение «Вставьте карту» и ожидает, когда пользователь вставит карту в приемник. Дальнейшее функционирование таксофона осуществляется только при вставленной карте. Если карту вынимают, таксофон возвращается к началу и выдает сообщение о необходимости вставить карту. При попадании карты в приемник производится считывание информации с карты. Если кредит исчерпан или карта не пригодна (не удастся узнать кредит), то таксофон выдает соответствующее сообщение на дисплей таксофона. Если карта может быть использована для оплаты, то на дисплей выдается количество «единиц» на карте, и на телефонную станцию (АТС) подается сигнал «Трубка». При получении ответного сигнала «Тон» из линии таксофон воспроизводит звуковой тон «Готов» (длинный непрерывающийся гудок) в трубку. При получении

сигнала «Занято», в трубке воспроизводится тон «Занято» (короткие гудки). После получения от АТС сигнала «Тон» от пользователя принимаются семизначный номер вызываемого абонента, остальные нажатия на кнопки игнорируются. Когда пользователь нажимает на кнопку с цифрой соответствующий ей сигнал «Цифра» передается АТС. Во время набора номера введенные цифры отображаются на дисплее. В ответ на набранный номер от АТС приходит либо сигнал «Занято», либо сигнал «Вызов». При получении сигнала «Вызов» таксофон воспроизводит в трубку длинные гудки до того момента, когда АТС осуществит коммутацию и передаст сигнал «Данные». Таксофон воспроизводит данные, передаваемые с сигналом, в трубку. При получении данных из трубки, аппарат преобразует их в сигнал «Данные» и передает их АТС. Во время разговора на дисплее ведется отсчет времени и уменьшается кредит на телефонной карте - каждые 15 секунд вычитается четверть «единицы». Обмен данными прерывается, в следующих случаях:

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

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

Вариант 9.

1. Диаграмма развертывания

2. Банкомат

Требуется разработать средствами UML модель программного обеспечения 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. Банкомат подключен к линии связи для обмена данными 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. 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 (10, 50, 100, 200, 500, 1000 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.

В специальном отделении банкомата, закрываемом замком, есть "секретная кнопка", которая используется обслуживающим персоналом для загрузки денег. При нажатии на эту кнопку открывается доступ к хранилищу денег и конфискованным кредитным картам.

Вариант 10

1. Диаграмма классов.

2. Холодильник

Требуется разработать средствами UML модель программного обеспечения встроенного процессора холодильника. Холодильник состоит из нескольких холодильных камер для хранения продуктов. В каждой холодильной камере имеется регулятор температуры, мотор, термометр, индикатор, таймер, датчик открытия двери камеры и устройство для подачи звуковых сигналов.

При помощи терморегулятора устанавливается максимально допустимая температура в данной камере. Мотор предназначен для поддержания низкой температуры. Термометр постоянно измеряет температуру внутри камеры, а индикатор температуры, расположенный на дверце, постоянно высвечивает ее значение. При повышении температуры выше предела, определяемого текущим положением регулятора, включается мотор. При снижении температуры ниже некоторого другого значения, связанного с первым, мотор отключается.

Доступ в камеру осуществляется через дверцу. Если дверь холодильной камеры открыта в течение слишком долгого времени, подается звуковой сигнал. Звуковой сигнал также подается в любых нештатных ситуациях (например, при поломке мотора). Холодильник ведет электронный журнал, в котором отмечаются все происходящие события:

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

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

Вариант 11.

1. Диаграмма последовательностей.

2. Замок

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

Вариант 12.

1. Диаграмма вариантов использования.

2. Турникет метро

Требуется разработать средствами UML 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 перекрытия доступа, таймер, три оптических датчика для определения прохода пассажира, устройство подачи звуковых сигналов, индикаторы «Проход» и «Стоп».

В начальном состоянии турникета зажжен индикатор «Стоп», индикатор «Проход» потушен. Если один из датчиков посылает сигнал, то проход через турникет сразу же перекрывается, и подается предупредительный звуковой сигнал. Для прохода пассажир должен поместить карту в приемник карт. Турникет считывает с нее данные: срок годности карты и количество «единиц» на ней. Если данные не удается считать, или карта просрочена, или заблокирована, то карта возвращается пассажиру, и турникет остается в исходном состоянии. В другом случае с карты списывается одна «единица», карта возвращается из приемника, индикатор «Стоп» гаснет, зажигается индикатор «Проход», и пассажир может пройти через турникет. Получив от одного из датчиков сигнал, турникет ожидает время, отведенное на проход пассажира (5 секунд), после чего он возвращается в начальное состояние.

Наличие трех датчиков в турникете гарантирует, что при проходе пассажира хотя бы один из них подаст сигнал (датчики невозможно перешагнуть, перепрыгнуть и т. д.). Во время прохода пассажира возможна ситуация, когда все три датчика посылают сигналы. В этом случае принимается только первый сигнал и от момента его приема отсчитывается положенное время. Остальные сигналы игнорируются.

Турникет заносит в свою память время всех оплаченных проходов. В конце рабочего дня он передает всю информацию, накопленную за день, в АСУ метрополитена. В ходе выполнения этого варианта задания должна быть разработана схема базы данных о проходах через турникет.

Вариант 13.

1. Диаграмма кооперации

2. Система учета товаров

Требуется разработать средствами UML модель системы поддержки заказа и учета товаров в бакалейной лавке.

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

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

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

Вариант 14.

1. Диаграмма деятельности

2. Библиотека

Требуется разработать средствами UML модель системы автоматизирующей деятельность библиотеки.

Система поддержки управления библиотекой должна обеспечивать операции (добавление, удаление и изменение) над данными о читателях. В регистрационном списке читателей хранятся следующие сведения: фамилия, имя и отчество читателя; номер его

читательского билета и дата выдачи билета. Наряду с регистрационным списком системой должен поддерживаться каталог библиотеки, где хранится информация о книгах: название, список авторов, библиотечный шифр, год и место издания, название издательства, общее количество экземпляров книги в библиотеке и количество экземпляров, доступных в текущий момент. Система обеспечивает добавление, удаление и изменение данных каталога, а также поиск книг в каталоге на основании введенного шифра или названия книги. В системе осуществляется регистрация взятых и возвращенных читателем книг. Про каждую выданную книгу хранится запись о том, кому и когда была выдана книга, и когда она будет возвращена. При возврате книги в записи делается соответствующая пометка, а сама запись не удаляется из системы. Система должна выдавать следующую справочную информацию:

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

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

Вариант 15.

1. Диаграмма развертывания

2. Интернет-магазин

Требуется разработать средствами UML модель программного обеспечения Интернет-магазина.

Интернет-магазин позволяет делать покупки с доставкой на дом. Клиенты магазина при помощи программы-браузера имеют доступ к каталогу продаваемых товаров, поддержку которого осуществляет Интернет-магазин. В каталоге товары распределены по разделам. О каждом товаре доступна полная информация (название, вес, цена, изображение, дата изготовления и срок годности) Для удобства клиентов предусмотрена система поиска товаров в каталоге. Заполнение каталога информацией происходит автоматически в начале рабочего дня, информация берется из системы автоматизации торговли. При отборе клиентами товаров поддерживается виртуальная «торговая корзина». Любое наименование товара может быть добавлено в «корзину» или изъято в любой момент по желанию покупателя с последующим пересчетом общей стоимости покупки. Текущее содержимое «корзины» постоянно показывается клиенту.

По окончании выбора товаров производится оформление заказа и регистрация покупателя. Клиент указывает в регистрационной форме свою фамилию, имя и отчество, адрес доставки заказа и телефон, по которому с ним можно связаться для подтверждения сделанного заказа. Заказы передаются для обработки в систему автоматизации торговли. Проверка наличия товаров на складе и их резервирование Интернет-магазином не производятся. Дополнительно требуется разработать схему базы данных, хранящей заказы.

Следует определиться, по какому архитектурному шаблону будет строиться Web-приложение («тонкий клиент» или «толстый клиент»). В соответствии с выбранным шаблоном следует построить модели клиентской части магазина и серверной части, промоделировать связи между частями приложения. Для Web-приложений типичными являются следующие *классы*:

- клиентская Web-страница;
- серверная Web-страница (например, CGI-скрипт);
- HTML-форма;
- объект JavaScript.

Дополнительные *связи* между классами Web-приложений:

- link – ссылка с одной страницы на другую;

- build – связь между CGI-скриптом и клиентской страницей, генерируемой при его выполнении;
- submit – связь между формой и серверной Web-страницей, принимающей данные из формы.

Типичные компоненты:

- Web-страница (HTML-файл),
- ActiveServerPage (ASP),
- JavaServerPage (JSP),
- библиотека скриптов (например, подключаемый файл с Javascript-функциями).

Вариант 16.

1. Диаграмма компонентов

2. WWW-конференция

Требуется разработать средствами UML модель программного обеспечения WWW-конференции.

WWW-конференция представляет собой хранилище сообщений в сети Интернет, доступ к которому осуществляется при помощи браузера. Для каждого сообщения конференции хранятся значения следующих полей: номер сообщения, автор, тема, текст сообщения, дата добавления сообщения, ссылка на родительское сообщение. Начальной страницей конференции является иерархический список сообщений. Верхний уровень иерархии составляют сообщения, открывающие новые темы, а подуровни составляют сообщения, полученные в ответ на сообщения верхнего уровня. Сообщение-ответ всегда имеет ссылку на исходное сообщение. В списке отображаются только темы сообщений, их авторы и даты добавления. Просматривая список, пользователь выбирает сообщение и по гиперссылке открывает страницу с текстом сообщения. Помимо текста на этой странице отображается список (иерархический) сообщений являющихся ответами, ответами на ответы и т.д. Для удобства пользователей необходимо предусмотреть поиск сообщений по автору или по ключевым словам в теме или тексте сообщения. Сообщения добавляются в конференцию зарегистрированными пользователями, которые при отправке сообщения должны указать своё имя и пароль. Регистрирует новых пользователей модератор конференции - её ведущий. При регистрации пользователь заполняет специальную форму, содержимое которой затем пересылается модератору и запоминается в базе пользователей. Модератор решает, зарегистрировать пользователя или нет, и отправляет свой ответ.

При добавлении сообщений пользователь имеет возможность начать новую тему или ответить на ранее добавленные сообщения. После добавления сообщения оно доступно для чтения всем пользователям (даже незарегистрированным), и список сообщений обновляется.

Модератор имеет право по тем или иным причинам удалять сообщения любых авторов. Он также может наказывать пользователей, нарушающих правила поведения в конференции, лишая на некоторое время пользователя возможности добавлять и редактировать сообщения.

Вариант задания включает в себя разработку схемы базы данных для хранения сообщений конференции и информации об её участниках.

Выполняющим это задание полезно ознакомиться с заключительным замечанием к варианту «Интернет-магазин». Наиболее подходящей архитектурой для WWW-конференции является «тонкий клиент», поскольку клиентская часть практически не содержит «бизнес-логики». Единственным её элементом, который может выполняться на стороне клиента, является проверка правильного заполнения полей формы, перед отправкой её содержимого на сервер.

Вариант 17.

1. Диаграмма состояний

2. Каталог ресурсов Интернет

Требуется разработать средствами UML модель программного обеспечения каталога ресурсов сети Интернет.

В каталоге хранится следующая информация о ресурсах: название ресурса, уникальный локатор ресурса (URL), раздел каталога, в котором содержится ресурс, список ключевых слов, краткое описание, дата последнего обновления, контактная информация. Доступ пользователей к каталогу осуществляется при помощи браузера. Пользователи каталога могут добавлять новые ресурсы, информация о которых не была внесена ранее. Ресурсы в каталоге классифицируются по разделам. Полный список ресурсов каждого раздела должен быть доступен пользователям. Пользователям каталога должны быть предоставлены возможности по поиску ресурсов. Поиск осуществляется по ключевым словам. Если пользователь не доволен результатами поиска, он может уточнить запрос (осуществить поиск среди результатов предыдущего поиска). Должна быть возможность выдавать результаты поиска в разной форме (вывод всей информации о ресурсах или частичной). Пользователь может отсортировать список ресурсов по релевантности (соответствию ключевым словам из запроса) или по дате обновления. Поскольку содержание ресурсов Интернет со временем изменяется необходимо следить за датой последнего обновления, периодически опрашивая Web-сайты, URL которых хранятся в каталоге.

Вариант задания включает в себя разработку схемы базы данных для хранения сообщений конференции и информации об её участниках.

Выполняющим это задание полезно ознакомиться с заключительным замечанием к варианту подходящей архитектурой для каталога является «тонкий клиент», поскольку клиентская часть практически не включает в себя функций «бизнес-логики» кроме проверки содержимого форм перед пересылкой на сервер.

Вариант 18.

1. Диаграмма объектов

2. Будильник

Требуется разработать средствами UML модель программного обеспечения встроенного микропроцессора для будильника.

Будильник постоянно отображает текущее время (часы, минуты, например: 12 : 00).

Управление будильником осуществляется следующими кнопками:

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

На будильнике имеется переключатель режима работы со следующими положениями: «ВЫКЛ», «ВКЛ», «РАДИО» и «ТАЙМЕР».

Для установки текущего времени нужно нажать на кнопку режима установки и, при нажатой кнопке, нажимать на кнопки установки часов и минут. При каждом нажатии на кнопки, устанавливаемое значение увеличивается на одну единицу (один час или одну минуту соответственно). При достижении максимального значения производится сброс.

Для установки времени срабатывания будильника нужно нажать на кнопку режима установки времени срабатывания и, держа кнопку нажатой, нажимать на кнопки установки часов и минут. Когда переключатель режима работы находится в положении «ВКЛ», при достижении времени срабатывания происходит подача звукового сигнала в течение одной минуты. Сигнал можно прервать, нажав на кнопку «СБРОС». При этом сигнал должен быть возобновлен через пять минут. При установке переключателя в положение «ВЫКЛ» звуковой сигнал не подается.

Когда переключатель находится в положении «РАДИО» работает радиоприемник. При переводе переключателя в положение «ТАЙМЕР» включается радиоприемник на тридцать минут, а затем часы переходят в состояние будильника (аналогично положению

«ВКЛ»). При нажатии на кнопку режима установки времени, будильник должен отображать время срабатывания.

Вариант 19.

1. Диаграмма классов.

2. Генеалогическое дерево

Требуется разработать средствами UML модель системы для поддержки генеалогических деревьев.

Система хранит сведения о персонах (Ф.И.О., пол, дата рождения, дата смерти, биография) и о родственных связях между ними. Связи бывают только трех видов: «мужья-жены», «дети-родители» и «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;
- проследивать цепочку родственных связей от одной персоны до другой (например, если Петр является шурином Ивана, то на запрос о родственных связях между Петром и Иваном выдается такой результат: «Петр – брат Ольги, Ольга – жена Ивана»).

Вариант 20.

1. Диаграмма сущность-связь

2. Телевизор

Требуется разработать средствами UML модель встроенной системы управления работой телевизора.

В телевизоре имеются следующие устройства: приемник телевизионного сигнала, устройство отображения картинки, память каналов, память настроек, управляющие кнопки, пульт дистанционного управления (ДУ). Управление телевизором осуществляется при помощи кнопок на корпусе (их четыре: «ВКЛ / ВЫКЛ», « - », « + », кнопка начальной установки) и пульта ДУ. Кнопка «ВКЛ / ВЫКЛ» позволяет включать и выключать телевизор. После включения телевизора на экран отображается передача, идущая по каналу №1, при этом используются параметры изображения и значение громкости, сохраненные в памяти настроек.

Память каналов телевизора хранит до 60 каналов. Каналы нумеруются, начиная с нуля. Последовательное переключение каналов осуществляется при помощи кнопок « - » и « + ». Нажатие на « + » переключает телевизор на канал с номером, на единицу большим (с 59-го канала телевизор переключается на 0-ой). Нажатие на « - » переключает телевизор на канал с номером, на единицу меньшим (с 0-го канала телевизор переключается на 59-ый).

При нажатии на кнопку начальной установки очищается память каналов телевизора, после чего осуществляется поиск передач и сохранение их частот в памяти каналов. Поиск начинается с нижней границы рабочего диапазона телевизора. На экран телевизора выводится «синий экран». Рабочая частота постепенно увеличивается до тех пор, пока приемник не обнаружит телевизионный сигнал. Найденная передача выводится на экран в течение 10 секунд. Также отображается номер, под которым найденный канал будет сохранен в памяти (начиная с 1). Затем поиск продолжается до тех пор, пока не будет достигнута верхняя граница диапазона, или пока не заполнена вся память каналов.

Телевизор принимает управляющие сигналы с пульта ДУ. На пульте ДУ расположены следующие кнопки:

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

Для прямого переключения на нужный канал его номер набирается с помощью кнопок с цифрами. После нажатия первой цифры в течение 5 секунд ожидается нажатие второй. Если вторая цифра не была нажата, то номер канала считается состоящим из одной цифры и осуществляется переключение на него.

Кнопки «П-» и «П+» на пульте имеют те же функции, что и кнопки «-» и «+» на корпусе телевизора. Кнопки «Г-» и «Г+» позволяют увеличивать или уменьшать громкость. Каждое нажатие на «Г-» уменьшает громкость на одну единицу, пока она больше нуля, «Г+» увеличивает громкость на единицу, пока не достигнуто максимальное значение. Текущее значение громкости сохраняется в памяти настроек.

Кнопки «МЕНЮ», «<» и «>» позволяют устанавливать значения настроек, хранящихся в памяти телевизора. При нажатии на кнопку «МЕНЮ» внизу экрана возникает надпись «ЯРКОСТЬ» и полоса, отображающая текущее значение яркости. Кнопками «<» и «>» яркость можно уменьшить или увеличить. При работе с меню нажатия на все остальные кнопки игнорируются. После повторного нажатия на кнопку «МЕНЮ» значение яркости запоминается в памяти настроек, и осуществляется переход к настройке контрастности. Настройка контрастности и остальных параметров (четкости, цветовой гаммы) происходит аналогично. Нажатие на кнопку «МЕНЮ» по окончании настройки цветовой гаммы (последнего пункта меню) приводит к окончанию работы с меню. Выход из меню также осуществляется в том случае, если в течение 15 секунд не была нажата ни одна кнопка.

Вариант 21.

1. Диаграмма классов.

2. Система поддержки составления расписания занятий

Требуется разработать средствами UML модель системы поддержки составления расписания занятий.

Система обеспечивает составление расписания некоторого учебного заведения, внесение в расписание изменений, выдачу полного расписания и дополнительной информации (например, по итоговому расписанию составляется расписание указанной группы на заданный день или неделю).

В расписании фиксируются время и место проведения занятия, предмет и преподаватель, проводящий занятие, а также номер группы, для которой это занятие проводится. Расписание не должно содержать коллизий (например, разные занятия не должны пересекаться друг с другом по месту и времени их проведения, один преподаватель не может вести одновременно два разных занятия, в одно и то же время у одной и той же группы не может быть два различных занятия и т. д.).

При работе над этим вариантом задания необходимо разработать схему базы данных для хранения расписания.

Вариант 22.

1. Диаграмма сущность-связь

2. Домофон

Требуется разработать средствами UML модель программного обеспечения встроенного микропроцессора домофона.

Домофон регулирует доступ в подъезд многоквартирного дома. В подъезде имеется дверь с замком. С наружной стороны двери установлена панель с кнопками на каждую квартиру, микрофон и динамик. В каждой квартире имеется кнопка «СВЯЗЬ», «БЛОКИРОВКА» и «ОТКРЫТЬ». Кроме того, в квартире имеется микрофон и динамик.

Жильцы могут открывать дверь ключом. Посетитель может нажать кнопку квартиры на внешней панели. При этом в квартире раздается звонок (если подача звонка в квартиру не заблокирована). Услышав звонок, жилец квартиры нажимает на кнопку «СВЯЗЬ» внутренней панели домофона, после чего домофон устанавливает звуковое сообщение между жильцом и посетителем. Звуки, произносимые посетителем в микрофон, установленный на внешней панели, воспроизводятся в динамике, установленном в квартире. Звуки из микрофона в квартире, передаются в динамик на внешней панели. После сеанса связи жилец может нажать на кнопку «ОТКРЫТЬ», чтобы замок на двери в подъезд открылся, и посетитель смог войти. По истечении минуты замок должен снова заблокировать вход в подъезд. Жилец, который желает, чтобы его не беспокоили, может отключить подачу звонка в свою квартиру, нажав на кнопку «БЛОКИРОВКА». Повторное нажатие на эту кнопку вновь включает подачу звонка.

Вариант 23.

1. Диаграмма последовательностей.

2. Мастерская по ремонту ТА (телефонных аппаратов)

Программа должна осуществлять учет заявок на ремонт телефонов.

Исходные данные к проекту:

- дата заявки;
- ФИО заказчика;
- адрес заказчика;
- вид ремонта;
- ФИО мастера, который принял заказ;
- дата выполнения заявки;
- сумма ремонта.

Так же необходимо организовать удобный интерфейс для пользователя. Программа должна выполнять поиск по заданным критериям, а также корректировку базы данных.

Вариант 24

1. Диаграмма вариантов использования.

2. Автоматизация рабочего места проектировщика

Необходимо обеспечить планирование инженерных задач (выбрать любую область самостоятельно). Программа должна выполнять следующие основные функции:

- Организовывать базу данных с объектами и работниками;
- Организовывать поиск и сортировку в базе данных по всем критериям;
- Иметь возможность ввода новых объектов и работников в базу данных;
- Должен быть удобный и приятный пользовательский интерфейс.

У пользователя должна быть возможность просмотра всей базы объектов и работников, внесения изменения в существующие записи, создания новых записей и удаления их.

Вариант 25

1. Диаграмма кооперации

2. АРМ по продаже программного обеспечения

Необходимо обеспечить пользователя удобным интерфейсом. База данных должна содержать: каталог товаров и учет покупок.

В программе необходимо предусмотреть возможность корректировки базы данных, то есть редактирования, добавления и удаления товаров. По запросу пользователя организовать поиск по:

- типу ПО;
- виду ПО;
- производителю.

Необходимо предусмотреть возможность сортировки по:

- коду;
- типу ПО;
- виду ПО;

- цене.

Также в программе необходимо вести подсчет итоговой суммы при продаже товаров, если отпускается не одна единица товара, то должна выводиться общая сумма покупки.

Вариант 26

1. Диаграмма состояний

2. Магазин по продаже дисков

Необходимо обеспечить пользователя удобным интерфейсом. База данных должна содержать:

- каталог товаров – в магазине должны продаваться диски таких типов как: Игры, Аудио и Видео.

- должен вестись учет покупок, т.е. должна быть создана таблица Клиентов.

В программе необходимо осуществить поиск по критериям, предусмотреть возможность корректировки базы данных и внесения новой информации в нее. Реализовать сортировку по типам и алфавиту.

Вариант 27

1. Диаграмма объектов

2. Автоматизированное рабочее место менеджера по продаже книг.

База данных должна содержать информацию о книгах (название, серийный номер, аннотация, стоимость, количество в магазине).

Разрабатываемая система должна решать следующие задачи:

1) Организация и ведение информации о книгах;

2) Поиск информации по критериям;

3) Составление отчёта по продажам, выдача чека.

4) Предусмотреть корректировку баз данных, а именно: добавление, удаление и редактирование записей.

Вариант 28

1. Диаграмма компонентов

2. Телефонный справочник

Необходимо создать базу данных, которая будет содержать следующую информацию:

- Фамилия, имя, отчество;

- Номер телефона;

- Домашний адрес (улица, номер дома, корпус и номер квартиры).

Если данный телефонный номер зарегистрирован на предприятие, то в колонке Фамилия необходимо указывать название предприятия.

В базе данных для пользователя должна существовать возможность:

1 Корректировки базы данных (добавление, удаление абонентов; изменение содержимого полей базы данных);

2 Осуществление поиска по критериям:

- по фамилии, имени, отчеству абонента;

- по адресу;

- по № телефона.

Вариант 29

1. Диаграмма деятельности

2. Автоматизация продажи авиабилетов

Разработать программу для автоматизированного контроля продажи авиабилетов.

Исходные данные:

- Фамилия И.О. пассажира (string);

Место назначения (string);

Серийный номер билета (integer)

Номер рейса (byte);

Категория (эконом –1, бизнес –2);

Количество билетов по каждой категории;

База содержит:
Номер рейса;
Общее количество мест по каждой категории;
Количество свободных мест по каждой категории;
Стоимость одного билета для каждой категории;
Дата и время отправления
Дата и время прибытия.
Постановка задачи:
Программа должна обеспечивать корректировку базы данных:
добавление новых рейсов;
снятие с рейса;
изменение даты и времени отправления или прибытия;
изменение стоимости билета.
Поиск билетов может осуществляться по следующим критериям: номеру рейса; месту назначения; количеству свободных мест по каждой категории.

Вариант 30

1. Диаграмма развертывания

2. Магазин по продаже сотовых телефонов

База данных должна содержать информацию о товарах (наименование, характеристика, стоимость, количество в магазине, гарантийный срок).

Разрабатываемая система должна решать следующие задачи:

- 1) Организация и ведение информации о покупках;
- 2) Формирование гарантийного талона;
- 3) Поиск информации по критериям;
- 4) Составление отчёта по продажам, выдача чека.
- 5) Предусмотреть корректировку баз данных, а именно: добавление, удаление и редактирование записей.

4. Условия выполнения контрольно-оценочных средств

Реализация программы учебной практики предполагает наличие лаборатории программного обеспечения и сопровождения компьютерных систем.

Оборудование лаборатории: доска; автоматизированные рабочие места на 14 обучающихся с наличием локальной и глобальной компьютерной сети: 14 столов, 14 стульев; автоматизированное рабочее место преподавателя (ПК, принтер), мультимедийный проектор, интерактивная доска, маркерная доска.

Основное оборудование: стенды «Техника безопасности», «Студенческий блог», «Современное программное обеспечение», «Технические средства информатизации», «Уголок здоровья», комплект учебно-методической документации.

Демонстрационные средства обучения:

программное обеспечение общего и профессионального назначения, мультимедийные презентации для проведения учебных занятий, необходимая для проведения практических занятий методическая и справочная литература (в т.ч. в электронном виде), мультимедийные презентации.

5. Информационное обеспечение

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительная литература.

Основные источники:

1. Федорова Г.Н. Разработка программных модулей программного обеспечения для компьютерных систем. (2-е изд., стер.) учебник/Федорова Г.Н. –М. ИЦ Академия, 2017- 336 с.

2. Рудаков А.В. Технология разработки программных продуктов (11-е изд., стер.). Учебник/ Рудаков А.В. –М. ИЦ Академия, 2017 г.-208 с.

3. Богомазова Г.Н. Модернизация программного обеспечения персональных компьютеров, серверов, периферийных устройств и оборудования: учебник.- М.: ИЦ Академия, 2016.-192 с.

4. Гвоздева В. А. Информатика, автоматизированные информационные технологии и системы. Учебник.- М.: ИД ФОРУМ, 2017.- 544 с.

5. Игошин В.И. Элементы математической логики: учебник.– М.: ИЦ Академия, 2017

Дополнительные источники:

1. Гагарина, Л. Г. Технология разработки программного обеспечения: учеб.пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул; Под ред. Л. Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2017.-400 с

2. Калайда В.Т., Романенко В.В. Технология разработки программного обеспечения: учебное пособие.-Томск: Томский межвузовский центр дистанционного образования, 2007.-257 с.

3. Черпаков И.В. Основы программирования: Учебник и практикум для СПО.- М.: Юрайт, 2017.-219 с

Интернет-ресурсы:

1. От модели объектов - к модели классов. Единое окно доступа к образовательным ресурсам. http://real.tepkom.ru/Real_OM-CM_A.asp
2. НОУ ИНТУИТ: <http://www.intuit.ru/department/se/devis/>

6. Критерии оценивания ответов обучающихся

Оценка «отлично» ставится если студент:

- ✓ полностью выполнил все требования индивидуального задания;
 - ✓ отвечал самостоятельно без наводящих вопросов преподавателя.
- Возможны одна - две неточности при освещении второстепенных вопросов или в выкладках, которые студент легко исправил по замечанию преподавателя.

Оценка «хорошо» ставится если ответ удовлетворяет основным требованиям, но при этом имеет один из недостатков:

- ✓ в выполнении допущены небольшие неточности, не исказившие решение задания;
- ✓ допущены один – два недочета при освещении основного содержания ответа, исправленные по замечанию преподавателя;
- ✓ допущены ошибка или более двух недочетов при освещении второстепенных вопросов или в выкладках, легко исправленные по замечанию преподавателя.

Оценка «удовлетворительно» ставится в следующих случаях:

- ✓ допущены неточности в выполнении индивидуального задания, но показано общее понимание вопроса;
- ✓ имелись затруднения или допущены ошибки в выполнении индивидуального задания, но осуществлены значительные исправления после нескольких наводящих вопросов преподавателя;
- ✓ не в полном объеме решена поставленная задача;
- ✓ обнаружены значительные отклонения в выполнении индивидуального задания;
- ✓ после нескольких замечаний преподавателя не исправлены неточности в выполнении индивидуального задания.

Текущий контроль успеваемости студентов осуществляется по всем видам аудиторных работ, предусмотренным учебной программой междисциплинарного курса. Текущий контроль осуществляется преподавателем, ведущим аудиторные занятия.

Текущий контроль успеваемости может проводиться в следующих формах:

- устная (устный опрос, защита письменной работы и т.д.)
- письменная (письменный опрос, выполнение расчетно-графического задания и т.д.);
- тестовая (устное, письменное, компьютерное тестирование).

Рекомендуются следующие виды текущего контроля:

- проверка исходного уровня подготовленности студента и его соответствия предъявляемым требованиям для изучения междисциплинарного курса;
- проверка усвоения студентами отдельных тем междисциплинарного курса;
- систематическая проверка выполнения индивидуальных домашних заданий, подготовки к занятиям, выполнения практических работи т.д.;
- единовременное подведение итогов текущей успеваемости (рубежи текущего контроля) в течение семестра.

Методы, используемые в процессе текущего контроля успеваемости, определяются с учетом специфики учебной дисциплины, ее содержания, трудоемкости (количества зачетных единиц), согласно утвержденной учебной программы междисциплинарного курса. Выбираемый метод должен обеспечить наиболее полный и объективный контроль (уровня освоения учебного материала с использованием фонда оценочных средств учебно-методического комплекса дисциплины. Запрещается использование антигуманных, а также опасных для жизни или здоровья студентов методов обучения и текущего контроля успеваемости.

Результаты текущего контроля успеваемости фиксируются в журнале занятий с соблюдением требований по его ведению.