Министерство образования и науки РФ
Иркутский национальный исследовательский технический университет

Анализ предметной области

Методические указания
по выполнению лабораторных работ

Издательство
Иркутского национального исследовательского технического университета
2018

УДК *****

Рецензент

Доктор техн. наук, зав.каф. кафедры Информационные системы и защита информации ФГБОУ ВО «ИрГУПС» Л.В.Аршинский.

Анализ предметной области: Метод. указания по выполнению лабораторных работ / сост.: З.А.Бахвалова – Иркутск: Изд-во ИРНИТУ, 2018 -45с.

Соответствуют требованиям ФГОС ВО 09.03.01- Информатика и вычислительная техника и 09.03.02 - Информационные системы и технологии.

Данные методические материалы содержат теоретические материалы и задания для выполнения лабораторной работы, позволяющий освоить процесс разработки программных систем через поэтапное выполнение основных этапов жизненного цикла разработки ПО. Приведенные материалы используются при выполнении работ по курсам «Технологии разработки программных комплексов» и «Технология разработки программных комплексов».

Учебное издание

Анализ предметной области

Методические указания
по выполнению лабораторных работ

Составители:
Бахвалова Зинаида Андреевна

Оглавление

Введение

При построении эффективной информационной системы (ИС) первым этапом является исследование информационных потоков информации деятельности предприятия. Проектирование ИС начинается с анализа задач и, соответственно, функций, обеспечивающих реализацию информационных потребностей предприятия или пользователя. Основное внимание сосредотачивается на выделении существенных объектов — предметов и связей, информация о которых может быть использована при решении конкретных задач. Таким образом, определяется состав и структура предметной области, выделяется система объектов и их взаимосвязей с учетом функционально-информационных требований к их последующему представлению в виде системы взаимосвязанных данных.

Чтобы ПО было действительно полезным, , необходимо сначала выяснить, какие же задачи она должна решать для этих людей и какими свойствами обладать, важно, чтобы оно удовлетворяло реальные потребности людей и организаций, которые часто отличаются от непосредственно выражаемых пользователями желаний. Для выявления этих потребностей, а также для выяснения смысла высказанных требований приходится проводить достаточно большую дополнительную работу, которая называется анализом предметной области или **бизнес-моделированием**, если речь идет о потребностях коммерческой организации. В результате этой деятельности разработчики должны научиться понимать язык, на котором говорят пользователи и заказчики, выявить цели их деятельности. Так же стоит:

- выяснить какие задачи нужно решать для достижения этих целей,
- выяснить свойства результатов, которые хотелось бы получить,
- определить набор сущностей, с которыми приходится иметь дело при решении этих задач.

Кроме того, анализ предметной области позволяет выявить места возможных улучшений и оценить последствия принимаемых решений о реализации тех или иных функций.

После этого можно определять область ответственности будущей программной системы — какие именно из выявленных задач будут ею решаться, при решении каких задач она может оказать существенную помощь и чем именно. Определив эти задачи в рамках общей системы задач и деятельностей пользователей, можно уже более точно сформулировать требования к ПО.

Анализом предметной области занимаются системные аналитики или бизнес-аналитики, которые передают полученные ими знания другим членам проектной команды, сформулировав их на более понятном разработчикам языке. Для передачи этих знаний обычно служит некоторый набор моделей, в виде графических схем и текстовых документов.

ЛАБОРАТОРНАЯ РАБОТА «Анализ предметной области»

Цель работы: формирование у студентов теоретического мышления, позволяющего видеть процесс проектирования программного обеспечения системным и актуальным.

Задачи: В результате изучения курса студенты должны получить

- базовые представления о принципах объектно-ориентированного программирования;
- овладеть понятийным аппаратом и практическими навыками моделирования программно-аппаратных комплексов;
- научиться использовать унифицированный язык моделирования (UML – Unified Modeling Language) для решения широкого круга задач;
- получить новые возможности развивать потенциальные способности специалиста в области разработки программного обеспечения.

Требования к уровню освоения.
В результате выполнения работы студенты должны :

- продемонстрировать знание содержания курса, определенного рамками учебной программы;
- навыки объектно-ориентрованного проектирования программных продуктов;
- умения разрабатывать модели к задачам из различных предметных областей на базе языка UML.

Основные процессы разработки ПО

Жизненным циклом программного обеспечения называют период от момента появления идеи создания некоторого программного обеспечения до момента завершения его поддержки фирмой-разработчиком или фирмой, выполнявшей сопровождение.

Состав процессов жизненного цикла регламентируется международным стандартом ISO/IEC 12207: 1995 «Information Technologe - Software Life Cycle Processes» («Информационные технологии - Процессы жизненного цикла программного обеспечения»). ISO - International Organization for Standardization - Международная организация по стандартизации. IEC - International Electrotechnical Commission - Международная комиссия по электротехнике.

Этот стандарт описывает структуру жизненного цикла программного обеспечения и его процессы. *Процесс* жизненного цикла определяется как совокупность взаимосвязанных действий, преобразующих некоторые входные данные в выходные. На рис. 1 представлены процессы жизненного цикла по указанному стандарту. Каждый процесс характеризуется определенными

задачами и методами их решения, а также исходными данными и результатами.

Рисунок 1- Структура процессов жизненного цикла программного обеспечения

Процесс разработки в соответствии со стандартом предусматривает действия и задачи, выполняемые разработчиком, и охватывает работы по созданию программного обеспечения и его компонентов в соответствии с заданными требованиями, включая оформление проектной и эксплуатационной документации, а также подготовку материалов, необходимых для проверки работоспособности и соответствия качества программных продуктов, материалов, необходимых для обучения персонала, и т. д. По стандарту процесс разработки включает следующие действия:

− *подготовительную работу* - выбор модели жизненного цикл, стандартов, методов и средств разработки, а также составление плана работ;

− *анализ требований к системе* - определение ее функциональных возможностей, пользовательских требований, требований к надежности и безопасности, требований к внешним интерфейсам и т. д.;

− *проектирование архитектуры системы* - определение состава необходимого оборудования, программного обеспечения и операций, выполняемых обслуживающим персоналом;

− *анализ требований к программному обеспечению* - определение функциональных возможностей, включая характеристики производительности, среды функционирования компонентов, внешних интерфейсов, спецификаций надежности и безопасности, эргономических требований, требова-

ний к используемым данным, установке, приемке, пользовательской документации, эксплуатации и сопровождению;

− *проектирование архитектуры программного обеспечения* — определение структуры программного обеспечения, документирование интерфейсов его компонентов, разработку предварительной версии пользовательской документации, а также требований к тестам и плана интеграции;

− *детальное проектирование программного обеспечения* - подробное описание компонентов программного обеспечения и интерфейсов между ни ми, обновление пользовательской документации, разработка и документирование требований к тестам и плана тестирования компонентов программного обеспечения, обновление плана интеграции компонентов;

− *кодирование и тестирование программного обеспечения* - разработку и документирование каждого компонента, а также совокупности тестовых процедур и данных для их тестирования, тестирование компонентов, обновление пользовательской документации, обновление плана интеграции программного обеспечения;

− *интеграцию программного обеспечения* - сборку программных компонентов в соответствии с планом интеграции и тестирование программного обеспечения на соответствие квалификационным требованиям, представляющих собой набор критериев или условий, которые необходимо выполнить, чтобы квалифицировать программный продукт, как соответствующий своим спецификациям и готовый к использованию в заданных условиях эксплуатации;

− *квалификационное тестирование программного обеспечения* - тестирование программного обеспечения в присутствии заказчика для демонстрации его соответствия требованиям и готовности к эксплуатации; при этом проверяется также готовность и полнота технической и пользовательской документации

− *интеграцию системы* - сборку всех компонентов системы, включая программное обеспечение и оборудование;

− *квалификационное тестирование системы* - тестирование системы на соответствие требованиям к ней и проверка оформления и полноты документации;

− *установку программного обеспечения* - установку программного обеспечения на оборудовании заказчика и проверку его работоспособности;

− *приемку программного обеспечения* - оценку результатов квалификационного тестирования программного обеспечения и системы в целом и документирование результатов оценки совместно с заказчиком, окончательную передачу программного обеспечения заказчику.

Указанные действия можно сгруппировать, условно выделив следующие основные этапы разработки программного обеспечения [10] (в скобках указаны соответствующие *стадии разработки* по ГОСТ 19.102-77 «Стадии разработки»):

- постановка задачи (стадия «Техническое задание»);
- анализ требований и разработка спецификаций (стадия «Эскизный проект»);
- проектирование (стадия «Технический проект»);
- реализация (стадия «Рабочий проект»).

Порядок выполнения работы

Постановка задачи

В процессе *постановки задачи* четко формулируют назначение программного обеспечения и определяют основные требования к нему. Каждое требование представляет собой описание необходимого или желаемого свойства программного обеспечения. Различают *функциональные требования,* определяющие функции, которые должно выполнять разрабатываемое программное обеспечение, и *эксплуатационные требования* определяющие особенности его функционирования.

Для формулирования требований к программному обеспечению, не имеющему аналогов, иногда необходимо провести специальные исследования, называемые предпроектными. В процессе таких исследований определяют разрешимость задачи, устанавливают наиболее существенные характеристики разрабатываемого программного обеспечения. Этап постановки задачи заканчивается разработкой *технического задания,* фиксирующего принципиальные требования, и принятием основных проектных решений

Цели этапа:

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

Исходными данными для этапа анализа являются:

- регламенты работы отделов и должностные инструкции сотрудников этих отделов;
- анкеты опроса заинтересованных лиц;
- записи интервью с заинтересованными лицами;
- другие документы, имеющие отношение к исследуемому объекту.

Выходными данными, или результатом, этого тапа системного анализа являются:

- перечень заинтересованных лиц;
- список потребностей заинтересованных лиц в разрабатываемом ПО;
- описание объектов автоматизации;
- модель объектов автоматизации или предметной области.

Этап постановка задачи необходим для общего представления о деятельности и целях организаций, в которых будет работать будущая программная система, о ее предметной области и ее проблемах.

На этом этапе следует определить более четко, какие именно задачи система будет решать. Кроме того, важно понимать, какие из задач стоят наиболее остро и обязательно должны быть поддержаны уже в первой версии, а какие могут быть отложены до следующих версий или вообще вынесены за рамки области ответственности системы. Эта информация выявляется при анализе потребностей возможных пользователей и заказчиков.

Часто бывает, что невозможно изложить что-то, не опираясь на какие-то базовые и однозначно понимаемые всеми термины (понятия). В таких случаях сначала нужно определить всю необходимую терминологию (предоставить описание предметной области), а только затем приступать к изложению. Для этого создают словарь терминов.

Словарь терминов представляет собой краткое описание основных понятий, используемых при составлении спецификаций, содержащих:

- определение основных понятий предметной области;
- описание структур элементов данных, их типов и форматов (на этапе анализа требований и определение спецификаций**);**
- описание сокращений и условных обозначений.

Он предназначен для повышения степени понимания предметной области и исключения риска возникновения разногласий при обсуждении моделей между заказчиками и разработчиками.

Обычно описание термина в словаре выполняют по следующей схеме:

- термин;
- категория (понятие предметной области, элемент данных, условное обозначение и т. д.);
- краткое описание.

Цель проекта

На следующем этапе разработки проекта очень важно сформулировать **цель и задачи проекта.** Формулирование цели и задач проекта является одной из важнейших стадий проекта разработки программного обеспечения. Важность её заключается в том, что на этой стадии заказчик программного продукта знакомит исполнителя со своим замыслом, задаёт направление для развития будущего проекта и определяет предельно допустимые рамки, за которые проект не должен выйти. Если заказчик не расскажет исполнителю о своей идее в доступной для исполнителя форме, то исполнитель не сможет выполнить работу. Если заказчик неверно или неточно определит цель будущего проекта, то исполнитель не будет иметь ориентира для своей деятельности и вряд ли сможет осуществить замысел заказчика.

Понятие «**цель**» означает результат, на достижение которого направлен определенный процесс.

Цель проекта задаёт направление развития проекта. Она должна коротко и ясно указать эффект, который окажет создаваемая или модернизируемая система на бизнес-процессы заказчика.

Цели представляют собой общие намерения применительно к проекту. Проекты могут иметь более одной цели, и много задач применительно к одной цели.

Цели и задачи должны быть четкими заявлениями о намерениях. Каждая цель должна иметь собственное стремление, которое влияет на конечный результат проекта. Цели и задачи должны быть измеряемы.

Для достижения желаемого результата исследователь осуществляет определенные действия, которые называются задачами. Другими словами, **задачи** – это способы достижения поставленной цели и этапы в продвижении к ней.

От того насколько четко сформулирована задача, зависит качество исследования. Задачи ставятся в виде утверждений, перечисляются по порядку в зависимости от сложности. Количество задач зависит от глубины исследования и определяется исследователем: в среднем от трех до семи задач.

Для формулирования задач используются такие слова, как:
«изучить…»;
«проанализировать…»;
«исследовать…»;
«рассчитать…»,
«рассмотреть…».

Главное, что необходимо понимать – цель это **конечный** результат. Решение задач – промежуточные этапы. Если в работе неправильно обозначена цель, то результат может отличаться.

Потребности заказчика

Потребности определяются на основе наиболее актуальных проблем и задач, которые пользователи и заказчики видят перед собой. При этом требуется аккуратное выявление значимых проблем, определение того, насколько хорошо они решаются при текущем положении дел, и расстановка приоритетов при рассмотрении недостаточно хорошо решаемых, поскольку чаще всего решить сразу все проблемы невозможно.

Формулировку потребностей следует разбить на следующие этапы:

1 Выделить одну-две-три основных проблемы.

2 Определить причины возникновения проблем, оценить степень их влияния и выделить наиболее существенные из проблем, влекущие появление остальных.

3 Определить ограничения на возможные решения.

Формулировка потребностей не должна накладывать лишних ограничений на возможные решения, удовлетворяющие им. Нужно попытаться сформулировать, что именно является проблемой, а не предлагать сразу возможные решения.

Например, формулировки "система должна использовать СУБД Oracle для хранения данных", "нужно, чтобы при вводе неверных данных раздавался звуковой сигнал" не очень хорошо описывают потребности. Исключением в первом случае может быть особая ситуация, например, если СУБД Oracle уже используется для хранения других данных, которые должны быть интегрированы с рассматриваемыми: при этом ее использование становится внешним ограничением. Соответствующие потребности лучше описать так: "нужно организовать надежное и удобное для интеграции с другими системами хранение данных", "необходимо предотвращать попадание некорректных данных в хранилище".

После выделения основных потребностей нужно решить вопрос о разграничении области ответственности будущей системы, т.е. определить, какие из потребностей надо пытаться удовлетворить в ее рамках, а какие — нет.

При этом все заинтересованные лица делятся на пользователей, которые будут непосредственно использовать создаваемую систему для решения своих задач, и вторичных заинтересованных лиц, которые не решают своих задач с ее помощью, но чьи интересы так или иначе затрагиваются ею. Потребности пользователей нужно удовлетворить в первую очередь и на это нужно выделить больше усилий, а интересы вторичных заинтересованных лиц должны быть только адекватно учтены в итоговой системе.

Функции разрабатываемого ПО

На основе выделенных потребностей пользователей, отнесенных к области ответственности системы, формулируются возможные **функции** будущей системы, которые представляют собой услуги, предоставляемые системой и удовлетворяющие потребности одной или нескольких групп пользователей (или других заинтересованных лиц). Идеи для определения таких функций можно брать из имеющегося опыта разработчиков (наиболее часто используемый источник) или из результатов мозговых штурмов и других форм выработки идей.

Формулировка функций должна быть достаточно короткой, ясной для пользователей, без лишних деталей. Например:

- Все данные о сделках и клиентах будут сохраняться в базе данных.
- Статус выполнения заказа клиент сможет узнать через Интернет.
- Система будет поддерживать до 10000 одновременно работающих пользователей.

- Расписание проведения ремонтных работ будет строиться автоматически.

Предлагая те или иные функции, нужно уметь аккуратно оценивать их влияние на структуру и деятельность организаций, в рамках которых будет использоваться ПО. Это можно сделать, имея полученные при анализе предметной области модели их текущей деятельности.

Имея набор функций, достаточно хорошо поддерживающий решение наиболее существенных задач, с которыми придется работать разрабатываемой системе, можно составлять требования к ней, представляющие собой детализацию работы этих функций. Соотношение между проблемами, потребностями, функциями и требованиями показано на рис.2

Рисунок 2- Соотношение между проблемами, потребностями, функциями и требованиями

Требования к разрабатываемому ПО

Требования к ПО определяют, какие свойства и характеристики оно должно иметь для удовлетворения потребностей пользователей и других заинтересованных лиц. Однако сформулировать требования к сложной системе не так легко. В большинстве случаев будущие пользователи могут перечислить набор свойств, который они хотели бы видеть, но никто не даст гарантий, что это — исчерпывающий список. Кроме того, часто сама формулировка этих свойств будет непонятна большинству программистов: могут прозвучать фразы типа "должно использоваться и частотное, и временное уплотнение каналов", "передача клиента должна быть мягкой", "для обычных швов отмечайте бригаду, а для доверительных — конкретных сварщиков", и это еще не самые тяжелые для понимания примеры.

Каждое требование раскрывает детали поведения системы при выполнении ею некоторой функции в некоторых обстоятельствах. При этом часть

требований исходит из потребностей и пожеланий заинтересованных лиц и решений, удовлетворяющих эти потребности и пожелания, а часть — из внешних ограничений, накладываемых на систему, например, основными законами той предметной области, в рамках которой системе придется работать, государственным законодательством, корпоративной политикой и пр.

Требования к программному обеспечению–это описание того, что, как и для чего должна сделать программная система.

Постановка задачи пишется в повествовательной форме *в будущем времени*, в ней должны быть обязательно взаимоувязаны виды автоматизируемой деятельности (с привязкой к объекту(ам) автоматизации) со всеми ограничениями, накладываемыми на них, учтены особенности разрабатываемого информационного обеспечения и перечислены функции, которые должна выполнять система (с привязкой к процессам и информационному обеспечению).

Стандарт интерфейса пользователя должен определять:

- правила оформления экранов (шрифты и цветовую палитру), состав и расположение окон и элементов управления;
- правила пользования клавиатурой и мышью;
- правила оформления текстов помощи;
- перечень стандартных сообщений;
- правила обработки реакции пользователя.

Все описанные выше проектные решения существенно влияют на трудоемкость и сложность разработки. Только после их принятия следует переходить к анализу требований и разработке спецификаций проектируемого программного обеспечения.

Разработка технического задания

Техническое задание представляет собой документ, в котором сформулированы основные цели разработки, требования к программному продукту, определены сроки и этапы разработки и регламентирован процесс приемно-сдаточных испытаний. В разработке технического задания участвуют как представители заказчика, так и представители исполнителя. В основе этого документа лежат исходные требования заказчика, анализ передовых достижений техники, результаты выполнения научно-исследовательских работ, предпроектных исследований, научного прогнозирования и т. п.

Основные факторы, определяющие характеристики разрабатываемого программного обеспечения, являются:

- исходные данные и требуемые результаты, которые определяют функции программы или системы;

- среда функционирования (программная и аппаратная) -может быть задана, а может выбираться для обеспечения параметров, указанных в техническом задании;
- возможное взаимодействие с другим программным обеспечением и/или специальными техническими средствами -также может быть определено, а может выбираться исходя из набора выполняемых функций.

Разработка технического задания выполняется в следующей последовательности:

−устанавливается набор выполняемых функций;

−устанавливается перечень и характеристики исходных данных;

−определяется перечень результатов проектирования;

−определяется характеристики и способы представления результатов;

−уточняются среда функционирования программного обеспечения:

а) конкретную комплектацию и параметры технических средств;

б) версию используемой операционной системы;

в) версии и параметры установленного программного обеспечения;

−регламентируются действия программы в случае сбоев оборудования и энергоснабжения.

На техническое задание существует стандарт ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению». В соответствии с этим стандартом техническое задание должно содержать следующие разделы:

- введение;
- основания для разработки;
- назначение разработки;
- требования к программе или программному изделию;
- требования к программной документации;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки.

При необходимости допускается в техническое задание включать приложения.

Введение должно включать наименование и краткую характеристику области применения программы или программного продукта, а также объекта (например, системы), в котором предполагается их использовать. Основное назначение введения -продемонстрировать актуальность данной разработки и показать, какое место эта разработка занимает в ряду подобных.

Раздел «Основание для разработки» должен содержать наименование документа, на основании которого ведется разработка, организации, утвердившей данный документ, и наименование или условное обозначение темы разработки. Таким документом может служить план, приказ, договор и т.п.

Раздел «Назначение» должен содержать описание функционального и эксплуатационного назначения программного продукта с указанием категорий пользователей.

Раздел «Требования к программе или программному изделию» должен включать следующие подразделы:

- требования к функциональным характеристикам;
- требования к надежности;
- условия эксплуатации;
- требования к составу и параметрам технических средств;
- требования к информационной и программной совместимости;
- требования к маркировке и упаковке;
- требования к транспортированию и хранению;
- специальные требования.

Наиболее важным из перечисленных выше является подраздел «**Требования к функциональным характеристикам».**

В этом разделе должны быть перечислены выполняемые функции и описаны состав, характеристики и формы представления исходных данных и результатов. В этом же разделе при необходимости указывают критерии эффективности: максимально допустимое время ответа системы, максимальный объем используемой оперативной и/или внешней памяти и др.

В подразделе «Требования к надежности» указывают уровень надежности, который должен быть обеспечен разрабатываемой системой и время восстановления системы после сбоя. Для систем с обычными требованиями к надежности в этом разделе иногда регламентируют действия разрабатываемого продукта по увеличению надежности результатов (контроль входной и выходной информации, создание резервных копий промежуточных результатов и т. п.).

В подразделе «Условия эксплуатации» указывают особые требования к условиям эксплуатации: температуре окружающей среды, относительной влажности воздуха и т. п. Как правило, подобные требования формулируют, если разрабатываемая система будет эксплуатироваться в нестандартных условиях или использует специальные внешние устройства, например для хранения информации. Здесь же указывают вид обслуживания, необходимое количество и квалификация персонала. В противном случае допускается указывать, что требования не предъявляются.

В подразделе «Требования к составу и параметрам технических средств» указывают необходимый состав технических средств с указанием их основных технических характеристик: тип микропроцессора, объем памяти, наличие внешних устройств и т. п. При этом часто указывают два варианта конфигурации: минимальный и рекомендуемый.

В подразделе «Требования к информационной и программной совместимости» при необходимости можно задать методы решения, определить язык или среду программирования для разработки, а также используемую операционную систему и другие системные и пользовательские программные

средства, с которыми должно взаимодействовать разрабатываемое программное обеспечение. В этом же разделе при необходимости указывают, какую степень защиты информации необходимо предусмотреть.

В разделе «Требования к программной документации» указывают необходимость наличия руководства программиста, руководства пользователя, руководства системного программиста, пояснительной записки и т. п. На все эти типы документов также существуют ГОСТы.

В разделе «Технико-экономические показатели» рекомендуется указывать ориентировочную экономическую эффективность, предполагаемую годовую потребность и экономические преимущества по сравнению с существующими аналогами.

В разделе «Стадии и этапы разработки» указывают стадии разработки, этапы и содержание работ с указанием сроков разработки и исполнителей.

В разделе «Порядок контроля и приемки» указывают виды испытаний и общие требования к приемке работы.

В приложениях при необходимости приводят: перечень научно-исследовательских работ, обосновывающих разработку; схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые следует использовать при разработке. В зависимости от особенностей разрабатываемого продукта разрешается уточнять содержание разделов, т.е. использовать подразделы, вводить новые разделы или объединять их. В случаях если какие-либо требования, предусмотренные техническим заданием, заказчик не предъявляет, следует в соответствующем месте указать «Требования не предъявляются».

Анализ требований и определение спецификаций

Спецификациями называют точное формализованное описание функций и ограничений разрабатываемого программного обеспечения. Соответственно различают *функциональные* и *эксплуатационные* спецификации. Совокупность спецификаций представляет собой *общую* логическую модель проектируемого программного обеспечения.

Для получения спецификаций выполняют

- анализ требований технического задания,
- формулируют содержательную постановку задачи,
- выбирают математический аппарат формализации,
- строят модель предметной области,
- определяют подзадачи и выбирают или разрабатывают методы их решения.

Спецификации программного обеспечения при структурном подходе

Часть спецификаций может быть определена в процессе предроектных исследований и, соответственно, зафиксирована в техническом задании.

Функциональные спецификации описывают функции разрабатываемого программного обеспечения.

Эксплуатационные спецификации определяют требования к техническим средствам, надежности, информационной безопасности и т. д.

Точные спецификации можно определить, только разработав некоторую формальную модель разрабатываемого программного обеспечения.

Формальные модели, используемые на этапе определения спецификаций можно разделить на две группы:

- модели, зависящие от подхода к разработке (структурного или объектно-ориентированного);
- модели, не зависящие от него.

В рамках структурного подхода на этапе анализа и определения спецификаций используют три типа моделей:

- ориентированные на функции;
- ориентированные на данные;
- ориентированные на потоки данных.

Каждую модель целесообразно использовать для своего специфического класса программных разработок.

Диаграммы переходов состояний определяют основные аспекты поведения программного обеспечения во времени.

Диаграммы потоков данных -направление и структуру потоков данных.

Концептуальные диаграммы классов -отношение между основными понятиями предметной области.

Поскольку разные модели описывают проектируемое программное обеспечение с разных сторон, рекомендуется использовать сразу несколько моделей и сопровождать их текстами: словарями, описаниями и т. п., которые поясняют соответствующие диаграммы.

Методологии структурного анализа и проектирования, основанные на моделировании потоков данных, обычно используют комплексное представление проектируемого программного обеспечения в виде совокупности моделей:

- диаграмм потоков данных (DFD-Data Flow Diagrams), описывающих взаимодействие источников и потребителей информации через процессы, которые должны быть реализованы в системе;
- диаграмм «сущность-связь» (ERD-Entity-Relationship Diagrams), описывающих базы данных разрабатываемой системы;

- диаграмм переходов состояний (STD-State Transition Diagrams), характеризующих поведение системы во времени;
- спецификаций процессов
- словаря терминов.

Спецификации процессов представляют в виде краткого текстового описания, диаграмм, схем алгоритмов, псевдокодов, Flow-форм или диаграмм Насси-Шнейдермана.

Методологии структурного анализа

Функциональные диаграммы

Функциональными называют диаграммы, в первую очередь отражающие взаимосвязи функций разрабатываемого программного обеспечения. Отображение взаимосвязи функций активностной модели осуществляется посредством построения иерархии функциональных диаграмм, схематически представляющих взаимосвязи нескольких функций. Каждый блок такой диаграммы соответствует некоторой функции, для которой должны быть определены: исходные данные, результаты, управляющая информация и механизмы ее осуществления - человек или технические средства.

Все перечисленные выше связи функции представляются дугами, причем тип связи и ее направление строго регламентированы. Дуги, изображающие каждый тип связей, должны подходить к блоку с определенной стороны, а направление связи должно указываться стрелкой в конце дуги.

Физически дуги исходных данных, результатов и управления представляют собой наборы данных, передаваемые между функциями. Дуги, определяющие механизм выполнения функции -исполнителей-людей или соответствующие технические средства, в основном используют при описании спецификаций сложных информационных систем, которые включают как автоматизированные, так и ручные операции. Блоки и дуги маркируются текстами на естественном языке.

Блоки на диаграмме размещают по «ступенчатой» схеме в соответствии с последовательностью их работы или доминированием, которое понимается как влияние, оказываемое одним блоком на другие. В функциональных диаграммах SADT различают пять типов влияний блоков друг на друга:

- вход -выход блока подается на вход блока с меньшим доминированием, т. е. следующего;
- управление -выход блока используется как управление для блока с меньшим доминированием (следующего);
- обратная связь по входу – выход блока подается на вход блока с большим доминированием (предыдущего);

- обратная связь по управлению - выход блока используется как управляющая информация для блока с большим доминированием (предыдущего);
- выход-исполнитель - выход блока используется как механизм для другого блока.

Дуги могут разветвляться и соединяться вместе различными способами. Разветвление означает, что часть или вся информация может использоваться в каждом ответвлении дуги. Дуга всегда помечается до ветвления, чтобы идентифицировать передаваемый набор данных. Если ветвь дуги после ветвления не помечена, то непомеченная ветвь содержит весь набор данных. Каждая метка ветви уточняет, что именно содержит данная ветвь. Построение иерархии функциональных диаграмм ведется поэтапно с увеличением уровня детализации: диаграммы каждого следующего уровня уточняют структуру родительского блока. Построение модели начинают с единственного блока, для которого определяют исходные данные, результаты, управление и механизмы реализации. Затем он последовательно детализируется с использованием метода пошаговой детализации. При этом рекомендуется каждую функцию представлять не более чем 3-7 блоками. Во всех случаях каждая подфункция может использовать или продуцировать только те элементы данных, которые использованы или продуцируются родительской функцией, причем никакие элементы не могут быть опущены, что обеспечивает непротиворечивость построенной модели.

Стрелки, приходящие с родительской диаграммы или уходящие на нее, нумеруют, используя символы и числа. Символ обозначает тип связи: I - входная, C -управляющая, M -механизм, R-результат. Число -номер связи по соответствующей стороне родительского блока, считая сверху вниз и слева направо.

Все диаграммы связывают друг с другом иерархической нумерацией блоков: первый уровень –A0, второй -Al, A2 и т. п., третий —A11, A12, A13 и т. п., где первые цифры -номер родительского блока, а последняя -номер конкретного субблока родительского блока.

Детализацию завершают после получения функций, назначение которых хорошо понятно как заказчику, так и разработчику. Эти функции описывают, используя естественный язык или псевдокоды.

В процессе построения иерархии диаграмм фиксируют всю уточняющую информацию и строят словарь данных, в котором определяют структуры и элементы данных, показанных на диаграммах.Таким образом, в результате получают спецификацию, которая состоит из иерархии функциональных диаграмм, спецификаций функций нижнего уровня и словаря, имеющих ссылки друг на друга.

Диаграммы переходов состояний

Диаграмма переходов состояний является графической формой предоставления конечного автомата - математической абстракции, используемой для моделирования детерминированного поведения технических объектов или объектов реального мира.

На этапе анализа требований и определения спецификаций диаграмма переходов состояний демонстрирует поведение разрабатываемой программной системы при получении управляющих воздействий. Для построения диаграммы переходов состояний необходимо в соответствии с теорией конечных автоматов определить:

−основные состояния, управляющие воздействия (или условия перехода);

−выполняемые действия;

−возможные варианты переходов из одного состояния в другое

Диаграммы потоков данных

Диаграммы потоков данных позволяют специфицировать как функции разрабатываемого программного обеспечения, так и обрабатываемые им данные. При использовании этой модели систему представляют в виде иерархии диаграмм потоков данных, описывающих асинхронный процесс преобразования информации с момента ввода в систему до выдачи пользователю. На каждом следующем уровне иерархии происходит уточнение процессов, пока очередной процесс не будет признан элементарным.

В основе модели лежат понятия внешней сущности, процесса, хранилища (накопителя) данных и потока данных.

Внешняя сущность - материальный объект или физическое лицо, выступающие в качестве источников или приемников информации, например заказчики, персонал, поставщики, клиенты, банк и т. п.

Процесс - преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. Каждый процесс в системе имеет свой номер и связан с исполнителем, который осуществляет данное преобразование. Как в случае функциональных диаграмм, физически преобразование может осуществляться компьютерами, вручную или специальными устройствами. На верхних уровнях иерархии, когда процессы еще не определены, вместо понятия «процесс» используют понятия «система» и «подсистема», которые обозначают соответственно систему в целом или ее функционально законченную часть.

Хранилище данных - абстрактное устройство для хранения информации. Тип устройства и способы помещения, извлечения и хранения для такого устройства не детализируют. Физически это может быть база данных, файл, таблица в оперативной памяти, картотека на бумаге и т. п.

Поток данных - процесс передачи некоторой информации от источника к приемнику. Физически процесс передачи информации может происходить по кабелям под управлением программы или программной системы или вручную при участии устройств или людей вне проектируемой системы.

Таким образом, диаграмма иллюстрирует как потоки данных, порожденные некоторыми внешними сущностями, трансформируются соответствующими процессами (или подсистемами), сохраняются накопителями данных и передаются другим внешним сущностям - приемникам информации. В результате получают сетевую модель хранения/обработки информации.

Построение иерархии диаграмм потоков данных начинают с диаграммы особого вида - контекстной диаграммы, которая определяет наиболее общий вид системы. На такой диаграмме показывают, как разрабатываемая система будет взаимодействовать с приемниками и источниками информации без указания исполнителей, т. е. описывают интерфейс между системой и внешним миром. Обычно начальная контекстная диаграмма имеет форму звезды.

Если проектируемая система содержит большое количество внешних сущностей (более 10), имеет распределенную природу или включает уже существующие подсистемы, то строят иерархии контекстных диаграмм.

При разработке контекстных диаграмм происходит детализация функциональной структуры будущей системы, что особенно важно, если разработка ведется несколькими коллективами разработчиков.

Полученную таким образом модель системы проверяют на полноту исходных данных об объектах системы и изолированность объектов (отсутствие информационных связей с другими объектами).

На следующем этапе каждую подсистему контекстной диаграммы детализируют при помощи диаграмм потоков данных. В процессе детализации соблюдают правило балансировки - при детализации подсистемы можно использовать компоненты только тех подсистем, с которыми у разрабатываемой подсистемы существует информационная связь (т. е. с которыми она связана потоками данных).

Решение о завершении детализации процесса принимают в следующих случаях:

• процесс взаимодействует с 2-3 потоками данных;
• возможно описание процесса последовательным алгоритмом;
• процесс выполняет единственную логическую функцию преобразования входной информации в выходную.

На недетализируемые процессы составляют спецификации, которые должны содержать описание логики (функций) данного процесса. Описание может выполняться: на естественном языке, с применением структурирован-

ного естественного языка (псевдокодов), с применением таблиц и деревьев решений, в виде схем алгоритмов.

Декомпозицию потоков данных необходимо осуществлять параллельно с декомпозицией процессов.

Окончательно разработку модели выполняют в два этапа.

1 этап - построение контекстной диаграммы - включает выполнение следующих действий:

- классификацию множества требований и организацию их в основные функциональные группы - процессы;
- идентификацию внешних объектов - внешних сущностей, с которыми система должна быть связана;
- идентификацию основных видов информации - потоков данных, циркулирующей между системой и внешними объектами;
- предварительную разработку контекстной диаграммы;
- изучение предварительной контекстной диаграммы и внесение в нее изменений по результатам ответов на возникающие при изучении вопросы по всем ее частям;
- построение контекстной диаграммы путем объединения всех процессов предварительной диаграммы в один процесс, а также группирования потоков.

2 этап - формирование иерархии диаграмм потоков данных - включает для каждого уровня:

- проверку и изучение основных требований по диаграмме соответствующего уровня (для первого уровня - по контекстной диаграмме);
- декомпозицию каждого процесса текущей диаграммы потоков данных с помощью детализирующей диаграммы или - если некоторую функцию сложно или невозможно выразить комбинацией процессов, построение спецификации процесса;
- добавление определений новых потоков в словарь данных при каждом появлении их на диаграмме;
- проведение ревизии с целью проверки корректности и улучшения наглядности модели после построения двух-трех уровней.

Полная спецификация процессов включает также описание структур данных, используемых как при передаче информации в потоке, так и при хранении в накопителе. Описываемые структуры данных могут содержать альтернативы, условные вхождения и итерации. Условное вхождение означает, что соответствующие элементы данных в структуре могут отсутствовать.

Альтернатива означает, что в структуру может входить один из перечисленных элементов. Итерация означает, что элемент может повторяться некоторое количество раз.

Кроме того, для данных должен быть указан тип: непрерывное или дискретное значение. Для непрерывных данных могут определяться единицы измерений, диапазон значений, точность представления и форма физического кодирования. Для дискретных - может указываться таблица допустимых значений.

Полученную законченную модель необходимо проверить на полноту и согласованность. Под согласованностью модели в данном случае понимают выполнение для всех потоков данных правила сохранения информации: все поступающие куда-либо данные должны быть считаны и записаны.

Спецификации программного обеспечения при объектном подходе

На этапе анализа при объектном подходе ставятся две задачи:

- уточнить требуемое поведение разрабатываемого программного обеспечения;
- разработать концептуальную модель его предметной области с точки зрения поставленных задач.

В основе объектного подхода к разработке программного обеспечения лежит объектная декомпозиция, т. е. представление разрабатываемого программного обеспечения в виде совокупности объектов, в процессе взаимодействия которых через передачу сообщений и происходит выполнение требуемых функций.

При объектном подходе так же, как при структурном подходе, выполняют декомпозицию программного обеспечения.

UML (Unified Modeling Language унифицированный язык моделирования) -средство описания проектов, создаваемых с использованием объектно-ориентированного подхода.

Спецификация разрабатываемого программного обеспечения при использовании UML объединяет несколько моделей: использования, логическую, реализации, процессов, развертывания.

Всего UML предлагает девять дополняющих друг друга диаграмм, входящих в различные модели:

- диаграммы вариантов использования;
- диаграммы классов;
- диаграммы пакетов;
- диаграммы последовательностей действий;
- диаграммы кооперации;
- диаграммы деятельностей;
- диаграммы состояний объектов;
- диаграммы компонентов;
- диаграммы размещения.

Диаграммы используют единую графическую нотацию.

Помимо указанных диаграмм, как и при структурном подходе, спецификация включает словарь терминов, описания и текстовые спецификации.

Конкретный набор документации определяется разработчиком.

Определение «вариантов использования»

Разработку спецификаций программного обеспечения начинают с анализа требований к функциональности, указанных в техническом задании. В процессе анализа выявляют внешних пользователей разрабатываемого программного обеспечения и перечень отдельных аспектов поведения в процессе взаимодействия с конкретными пользователями. Аспекты поведения программного обеспечения были названы «вариантами использования» или «прецедентами» (use cases).

Вариант использования представляет собой характерную процедуру применения разрабатываемой системы конкретным действующим лицом, в качестве которого могут выступать не только люди, но и другие системы или устройства.

В зависимости от цели выполнения конкретной процедуры различают следующие варианты использования:

- основные обеспечивают требуемую функциональность разрабатываемого программного обеспечения;
- вспомогательные - обеспечивают выполнение необходимых настроек: системы и ее обслуживание (например, архивирование информации и т. п.);
- дополнительные обеспечивают дополнительные удобства для пользователя (как правило, реализуются в том случае, если не требуют серьезных затрат каких-либо ресурсов ни при разработке, ни при эксплуатации).

Вариант использования можно описать кратко или подробно. Краткая форма описания содержит: название варианта использования, его цель, действующих лиц, тип варианта использования (основная, второстепенная или дополнительная) и его краткое описание.

Основные варианты использования обычно описывают подробно, стараясь отразить особенности предметной области разрабатываемого программного обеспечения. Подробная форма, кроме указанной выше информации, включает описание типичного хода событий и возможных альтернатив. Типичный ход событий представляют в виде диалога между пользователями и системой, последовательно нумеруя события. Если пользователь может выбирать варианты, то их описывают в отдельных таблицах. Также отдельно приводят альтернативы, связанные с нарушением типичного хода событий.

Спецификация (описание) варианта использования может содержать следующие основные разделы:

- название варианта использования (может быть указано в заголовке);
- цель;
- краткое описание;
- начальное состояние;
- основной поток событий;
- альтернативные потоки событий;
- специальные требования;
- предусловия;
- постусловия;
- дополнительные замечания**.**

В зависимости от особенностей разработки и требуемого уровня детализации моделей данный список может быть либо расширен (например, за счет добавления раздела «тип варианта» - основной или вспомогательный), либо сокращен (например, за счет разделов «предусловия», «постусловия», «дополнительные замечания» и т.д.).

Ниже приведены примеры описания вариантов использования *«Описание технического состояния детали»* разработанного при создании программной системы для прогнозирования технического состояния деталей машин в нефтехимии и отправки электронного письма, процесса известного всем и используемого в повседневной жизни очень часто.

Спецификация варианта использования «Описание технического состояния детали»

Цель: описать техническое состояние детали

Активные субъекты: аналитик, проектировщик, инженер.

Краткое описание: описание технического состояния детали путем указания наблюдаемых процессов, параметров этих процессов и их значений.

Основной поток событий:

1. Активный субъект инициирует процесс описания технического состояния.

2. Система публикует форму ввода параметров технического состояния. На экран выводятся проектные значения параметров технического состояния, полученные из базы данных.

3. Пользователь вводит фактические значения параметров технического состояния.

4. Система публикует форму ввода наблюдаемого процесса деградации. Ввод процессов предлагается сделать из справочника (табличная форма диалога).

5. Пользователь вводит наблюдаемый процесс, выбирая его из справочника (если нет, то активизируется один из альтернативных потоков 1,2,3

по выбору пользователя) и переходит к вводу наблюдаемых параметров указанных процессов.

6. Система публикует форму ввода наблюдаемых параметров процесса деградации и их значений. Ввод параметров предлагается сделать из справочника.

7. Пользователь вводит наблюдаемые параметры процесса деградации, выбирая их из справочника, указывает, по возможности, их значения (если нет, то активизируется один из альтернативных потоков 1, 2, 4 по выбору пользователя).

8. Система завершает работу варианта использования «Описание технического состояния детали».

Альтернативные потоки событий:

1. Активизация варианта использования «Планирование работ по уточнению причин изменения технического состояния».

2. Активизация варианта использования «Расширение справочника».

3. Пользователь возвращается к вводу фактических значений параметров технического состояния.

4. Пользователь возвращается к вводу наблюдаемого процесса деградации.

Специальные требования: нет.

Предусловия: Перед активизацией варианта использования должен быть выполнен вариант использования «Выбор детали».

Постусловия: После активизацией варианта использования должен быть активизирован вариант использования «Идентификация технического состояния» или «Регистрация технического состояния».

Дополнительные замечания: нет.

Спецификация варианта использования «Отправить электронное письмо»

Название: Отправить электронное письмо.

Цель: Отправить созданное письмо с проверкой корректности его атрибутов.

Начальное состояние: Выполнен ВИ «Создать новое письмо» или ВИ «Создать ответ на письмо».

Основной сценарий:

1. Пользователь выполняет команду отправки письма.

2. Программа проверяет, правильно ли заполнено поле «Адрес».

3. Если нет, программа сообщает об ошибке и отменяет отправку. Конец.

4. Если да, то программа проверяет, заполнено ли поле «Тема».

5. Если нет, программа выдает предупреждение, но не отменяет отправку.

6. Программа помещает письмо в папку «Исходящие» и отсылает его.

7. После отправки программа перемещает письмо в папку «Отправленные».

Сценарий обработки ошибок:

Предусловие: на шаге 6 основного сценария происходит ошибка отправки письма (сбой сети и т. п.)

7. Программа сообщает об ошибке и предлагает сохранить текст письма в файл.

8. Если пользователь согласен сохранить текст, выполняется ВИ «Сохранить черновик в файл».

Диаграммы вариантов использования

Диаграммы вариантов использования позволяют наглядно представить ожидаемое поведение системы. Основными понятиями диаграмм вариантов использования являются: действующее лицо, вариант использования, связь.

Действующее лицо внешняя по отношению к разрабатываемому программному обеспечению сущность, которая взаимодействует - с ним с целью получения или предоставления какой-либо информации. Как уже упоминалось выше, действующими лицами могут быть пользователи, другое программное обеспечение или какие-либо технические средства, взаимодействующие с разрабатываемым программным обеспечением.

Вариант использования некоторая очевидная для действующего лица процедура, решающая его конкретную задачу. Все варианты использования, так или иначе, связаны с требованиями к функциональности разрабатываемой системы и могут сильно отличаться по объему выполняемой работы.

Связь взаимодействие действующих лиц и соответствующих вариантов использования.

Варианты использования также могут быть связаны между собой.

Использование подразумевает, что существует некоторый фрагмент поведения разрабатываемого программного обеспечения, который повторяется в нескольких вариантах использования. Этот фрагмент оформляют как отдельный вариант использования и указывают связь с ним типа «использование».

Расширение применяют, если имеется два подобных варианта использования, различающиеся наличием в одном из них некоторых дополнительных действий. В этом случае дополнительные действия определяют как отдельный вариант использования, который связан с основным вариантом связью типа «расширение».

Диаграмма последовательностей системы

Диаграмма последовательностей системы - графическая модель, которая для определенного сценария варианта использования показывает генери-

руемые действующими лицами события и их порядок. При этом система рассматривается как единое целое.

Для построения диаграммы последовательностей системы необходимо:

- представить систему как «черный ящик» и изобразить для нее линию жизни - вертикальную пунктирную линию, подходящую к блоку снизу;
- идентифицировать каждое действующее лицо и изобразить для него линию жизни (много действующих лиц бывает в вариантах совместного использования программного обеспечения);
- из описания варианта использования определить множество системных событий и их последовательность;
- изобразить системные события в виде линий со стрелкой на конце между линиями жизни действующих лиц и системы, а также указать имена событий и списки передаваемых значений.

В отличие от внутренних, события, которые генерируются для системы действующими лицами, называют системными. Системные события инициируют выполнение соответствующего множества операций, также называемых системными. Каждую системную операцию называют по имени соответствующего сообщения.

Диаграммы деятельностей

В зависимости от степени детализации диаграммы деятельностей так же, как диаграммы классов, используют на разных этапах разработки. На этапе анализа требований и уточнения спецификаций диаграммы деятельностей позволяют конкретизировать основные функции разрабатываемого программного обеспечения.

Под деятельностью в данном случае понимают задачу (операцию), которую необходимо выполнить вручную или с помощью средств автоматизации. Каждому варианту использования соответствует своя последовательность задач. В теоретическом плане диаграммы деятельности являются обобщенным представлением алгоритма, реализующего анализируемый вариант использования. На диаграмме деятельность обозначается прямоугольником с закругленными углами.

Диаграммы деятельностей позволяют описывать альтернативные и параллельные процессы. Для обозначения альтернативных процессов используют ромб, условие указывают над ним слева или справа, а альтернативы «да», «нет» - рядом с соответствующими выходами. С помощью этого же блока можно построить циклический процесс. Множественность активации деятельности обозначают символом «*», помещенным рядом со стрелкой активации деятельности, и при необходимости уточняют надписью вида «для каждой строки».

Для обозначения параллельных процессов используют линейки синхронизации, причем условие синхронизации можно уточнить, указав его на диаграмме.

На этапе определения спецификаций имеет смысл уточнять только варианты использования, краткое описание которых недостаточно для понимания сущности решаемых проблем. Диаграммы деятельностей, таким образом, можно использовать вместо описания вариантов использования или как дополнение к ним.

Построение концептуальной модели предметной области

Диаграммы классов

Диаграммы классов — центральное звено объектно-ориентированных методов разработки программного обеспечения. UML предлагает использовать три уровня диаграмм классов в зависимости от степени их детализации:

- концептуальный уровень, на котором диаграммы классов, называемые в этом случае контекстными, демонстрируют связи между основными понятиями предметной области;
- уровень спецификаций, на котором диаграммы классов отображают интерфейсы классов предметной области, т. е. связи объектов этих классов;
- уровень реализации, на котором диаграммы классов непосредственно показывают поля и операции конкретных классов.

Каждую из перечисленных моделей используют на конкретном этапе разработки программного обеспечения:

- концептуальную модель - на этапе анализа;
- диаграммы классов уровня спецификации - на этапе проектирования;
- диаграммы классов уровня реализации - на этапе реализации.

Концептуальные модели в соответствии с определением оперируют понятиями предметной области, атрибутами этих понятий и отношениями между ними. Понятию в предметной области разрабатываемого программного обеспечения могут соответствовать как материальные предметы, так и абстракции, которые применяют специалисты предметной области.

Основным понятиям в модели ставятся в соответствие классы. Класс при этом традиционно понимают как совокупность общих признаков заданной группы объектов предметной области. В соответствии с этим определением на диаграмме классов каждому классу соответствует группа объектов, общие признаки которых и фиксирует класс. Так класс Студент объединяет общие признаки группы людей, обучающихся в высших учебных заведениях.

Экземпляр класса или объект (например, И.И. Иванов) обязательно обладает всей совокупностью признаков своего класса и может иметь собственные признаки, не фиксированные в классе. Так, например, помимо того,

что И.И. Иванов является студентом, он еще может быть спортсменом, музыкантом и т. д. Строго говоря, таким собственным признаком является и идентифицирующее студента имя.

На диаграммах класс изображается в виде прямоугольника, внутри которого указано имя класса. При необходимости допускается указывать характеристики класса, например атрибуты, используя специальные секции условного обозначения.

Тесты

На этом этапе также целесообразно сформировать тесты для поиска ошибок в проектируемом программном обеспечении, обязательно указав ожидаемые результаты.

Варианты заданий.

1. Реализовать информационную систему информирования посетителей книжного магазина. Печатное издание характеризуется атрибутами: Название, Краткое описание, Издательство, Тираж, Количество экземпляров на складе. Печатные издания подразделяется на Книги (атрибуты: Характеристика переплета, ISBN, Автор, Год издания, Число страниц) и Периодические издания(атрибуты: Редактор, Периодичность, Дата выхода в тираж, Количество номеров в год). Издательство имеет атрибуты Название и Местоположение. Реализовать хранение списка издательств и списка печатных изданий на складе. Необходимо обеспечить поиск изданий по названию, описанию, издательству с фильтрацией по доступности. Должна присуствовать возможность вывода на печать списка выбранных посетителем книг.

2. Реализовать информационную систему магазина офисной мебели. Предмет мебели характеризуется атрибутами: Модель, Изготовитель, Габариты, Цвет. Предмет мебели подразделяется на Шкаф(атрибуты: Количество дверей, Наличие замка), Полка (атрибут: Максимальная масса содержимого), Стул (атрибуты: Количество ножек, Высота спинки), Кресло (Материал, Наличие газопатрона). Изготовитель имеет атрибуты: Наименование, Страна, Адрес, Телефон. Реализовать хранение списка изготовителей и списка мебели. Обеспечить возможность поиска каждого из типов мебели по заданному набору характеристик. :Реализовать возможность печати перечня имеющихся в наличии предметов мебели одного производителя.

3. Система учёта данных о переговорах по офисной АТС компании. Телефонные переговоры (атрибуты: Дата переговоров, Продолжительность, Телефон вызывающего абонента) подразделяются на Международные переговоры (атрибут: Страна вызываемого абонента, Номер вызываемого абонента), Междугородние переговоры (атрибут: Город вызываемого абонента, Номер телефона вызываемого абонента), Местные переговоры (атрибуты: Является ли разговор тарифицируемым, Телефон вызываемого абонента). Телефон вызывающего абонента имеет атрибуты: ФИО владельца, Адрес владельца и Номер телефона. Реализовать хранение списка телефонов и списка переговоров. Вывести все разговоры за заднный период с конкретного номера, вывести список всех звонков на конкретный номер. Для заданного телефона абонента вывести список телефонных раговоров за месяц с их стоимостью и итоговой оплатой за месяц.

4. Система поиска авиарейса для веб-сайта. Авиарейс (Аэропорт (вылета), Аэропорт (назначения), Наличие мест, Тип самолёта) может быть Регулярным (Номер рейса, Дни недели вылета и время вылета, Перечень доступных классов бронирования с ценой билета) и Чартерным (Дата вылета, Время вылета). Аэропорт характеризуется атрибутами: Название по кодификатору, Название города на русском языке и Название города на английском языке. Реализовать хранение списка аэропортов и списка рейсов. Вывести

варианты перелёта между двумя пунктами назначения на заданную дату. Вывести для печати расписание прилётов/вылетов для заданного аэропорта на заданную дату.

5. Система табелирования сотрудников. Работник (атрибуты: Табельный номер, Фамилия, Имя, Отчество, Год рождения, Признак занятости [штатный или совместитель], Кабинет) может быть Управляющим (атрибуты: Название подразделения, Список подчиненных) или Служащим (атрибуты: Вид выполняемых работ, Непосредственный начальник (Управляющий)). Данные о выходе сотрудника на работу хранятся в Табеле(Табельный номер сотрудника, Дата, Время прихода, Время ухода, Причина неявки [в случае отсутствия на работе]) Организовать хранение списка работников и их табелирование. Вывести список сотрудников заданного подразделения, опоздавших на работу. Вывести табель за заданный месяц сотрудников подразделения.

6. Система автоматизации деятельности видеопроката. Видеоноситель (атрибуты: Название, Краткое описание, Режиссер, Актёры, Дата выпуска, Страна, Кинокомпания, Продолжительность, Место расположения в хранилище) может быть Лазерным диском (атрибут: Количество дисков, Формат записи, Обложка), DVD диском (атрибут: Код региона, Количество слоёв на диске) или VHS-кассета (атрибут: Формат записи, Обложка). Видеоноситель выдаётся Клиенту(атрибуты: ФИО, Адрес, Телефон, Дата регистрации). При это оформляется Карточка выдачи(атрибуты: Дата выдачи, Срок, Видеоноситель, Клиент). Выдать на экран список всех просрочивших сдачу видеоносителя клиентов. Подготовить печатную версию истории аренды видеоматериалов клиентом.

7. Система автоматизации деятельности ресторана. В ресторане подаются различные Блюда (атрибуты: Название, Состав, Цена, Вес, Тип [Первое, Второе, Закуска, Десерт]). Блюда подают Официанты (атрибуты: ФИО, Список столиков) Клиентам (атрибуты: ФИО, Дата и время прихода в ресторан, Номер столика) и регистрируются в Счёте (атрибуты: Клиент, Официант, Список блюд). Вывести на экран список обслуженных за день официантом клиентом. Сформировать печатную версию меню ресторана.

8. Система автоматизации прачечной. Клиент (атрибуты: ФИО, Адрес, Телефон) сдаёт в праченую Вещи (атрибуты: Наименование, Оценочная стоимость, Описание состояния вещи), которые составляют Заказ (атрибуты: Дата заказа, Список вещей, Срок выдачи, Стоимость заказа, Номер заказа). Для каждой вещи технологи формирует Технологическую карту (атрибуты: Вещь, Список операций с вещью), которая включает в себя перечень Операций (атрибуты: Наименование, Ответственный, Дата начала, Дата окончания), которые будут произведены с вещью. Вывести на экран перечень операций с указанием заказов, которые были произведены в течение определённого дня. Сформировать печатную версию листка заказа клиента.

9. Система автоматизации деятельности службы технической поддержки компании с большой региональной сетью. Сотрудники (атрибуты: ФИО, отдел, контактный телефон, эл. почта, здание и кабинет) формируют Заявки (атрибуты: Сотрудник, Тип, Описание неисправности, Приоритет, Дата и время заявки), которая назначается Исполнителю (атрибуты: ФИО, контактный телефон, адрес электронной почты). Исполнитель формирует описание неисправности и выполненных действий, которые вносятся в Акт выполненных работ (атрибуты: Дата окончания работ, Исполнитель, Заявка, Описание выполненных работ). Вывести на экран журнал работ за неделю с сортировкой по исполнителям/заказчикам/дате окончания. Печатная версия акта выполненных работ для подписи заказчиком.

10. **Учет товаров на складах и торговых точках**

Имеются товары, склады, где они хранятся, и торговые точки, в которых нужно размещать товары. Каждый склад имеет уникальный номер, адрес, а также ФИО кладовщика. Реквизиты товара – регистрационный номер, наименование, единица измерения. Торговая точка имеет наименование и адрес. Для каждого товара на складе и торговой точке хранится количество и стоимость единицы.

Выходные документы: список товаров на каждом складе, отсортированный по товарам, с подсчетом суммы стоимости товаров на каждом складе; для заданной торговой точки выдать список товаров, с указанием их общего количества.

11.Успеваемость студентов

Имеются группы студентов, студенты и изучаемые ими предметы.

Группа имеет шифр специальности, год начала обучения и номер. Данные о студенте состоят из ФИО, номера зачетки, даты рождения и пола. Для каждого предмета имеется наименование предмета, количество часов и вид аттестации (зачет или экзамен). Каждый студент получает оценку или признак зачета /незачета по каждому предмету.

Выходные документы: для указанной группы выдать список оценок студентов по всем предметам, с подсчётом среднего балла студента и упорядочить по фамилии и предмету; ведомости по зачетам и экзаменам по каждому предмету каждой группы.

12.Библиотека

Имеются разделы, книги и читатели книг.

Информация, хранимая о читателе - ФИО, домашний адрес, паспортные данные. Каждая книга имеет регистрационный номер, название, количество страниц, год издания и принадлежит определенному разделу (учебник, художественная, общественно-политическая и т.д.). Раздел характеризуется кодом и названием. Необходимо регистрировать дату, когда и какой читатель берет или возвращает книгу.

Выходные документы: список читателей, которые держат на руках книги, упорядоченный по датам выдачи, с указанием количества книг; для заданного читателя выдать список прочитанных им книг определенного раздела, сортируя по датам получения.

13.Магазины

Имеются производители товаров, товары и магазины. Компания-производитель характеризуется наименованием (маркой), адресом, ФИО директора и поставляет товары, которые имеют наименование и сорт. Будем считать, что один и тот же товар может иметь только одного производителя. Магазин имеет номер, наименование, адрес и площадь. Для каждого товара в магазине хранится единица измерения, цена за единицу и количество единиц. Выходные документы: для заданного интервала номеров магазинов выдать информацию о товарах с указанием производителя и с подсчетом стоимости товара и стоимости всех товаров в каждом магазине, упорядоченную по товарам; информация о магазинах, упорядоченная по их адресам, с подсчетом средней площади всех магазинов.

14. Конструктор

Имеются изделия, детали и материалы. Изделие задается заводским номером и наименованием. Деталь имеет наименование и определенную массу в граммах. Каждое изделие состоит из набора деталей, входящих в него в определенном количестве. Каждая деталь изготовлена из какого - либо материала. Материал задается наименованием и ценой за грамм.

Выходные документы: выдать детали заданного изделия, отсортированные по наименованию деталей, и стоимость материалов, из которых изготовлены детали; по наименованию материала выдать список изделий, в производстве которых он используется, упорядоченный по массе материалов.

15. Расход товаров

Имеются товары, покупатели и накладные. Покупатель характеризуется наименованием и адресом. Товар имеет номер, наименование и вид упаковки. Покупатель приобретает товар по определенной цене в определенном количестве. Факт покупки сопровождается накладной, содержащей номер накладной, дату покупки товара покупателем. Покупатель может приобрести несколько товаров по одной накладной.

Выходные документы: напечатать все факты покупки товаров определенным покупателем, отсортировав их по дате; напечатать все накладные (содержащие номер, дата, покупатель, товар, цена, количество, общая стоимость).

16. Отдел кадров

Имеются сотрудники предприятия, их дети и отделы, в которых работают сотрудники. Информация о каждом сотруднике - табельный номер, ФИО, паспортные данные, адрес, возраст, пол, должность и количество де-

тей. Каждый работник обязательно принадлежит какому-либо отделу. Отдел имеет наименование, ФИО начальника. Необходимо регистрировать хронологию назначений - перемещений сотрудников (т.е. фиксировать дату начала работы в отделе и дату окончания). Информация о ребенке – номер свидетельства о рождении, имя, пол, дата рождения.

Выходные документы: сформировать список работников заданного отдела на заданную дату, упорядоченный по возрасту; выдать список сотрудников, имеющих более одного ребенка, сгруппированный по отделам с подсчетом максимального возраста сотрудников в отделах.

17.Учет выполнения курсовых работ

Имеются студенты, предметы и курсовые работы по предмету. Информация о студенте – шифр группы, ФИО и номер зачетной книжки. Предмет характеризуется названием, ФИО преподавателя и количеством часов. По предмету каждый студент должен выполнить курсовую работу. Курсовая работа по предмету характеризуется номером варианта и названием темы. Необходимо зарегистрировать дату получения студентом задания на курсовую работу, а при защите работы отметить дату сдачи и оценку.

Выходные документы: список тем курсовых работ по каждому предмету и указанием их количества, отсортировав по номеру варианта; выдать ведомость по определенному предмету определенной группы с подсчетом среднего балла. В ведомость включить только тех студентов, которые получили оценку.

18.Гараж

База данных гаража должна содержать сведения о машинах, их рейсах и ремонтах. Ремонт характеризуется видом и длительностью (в днях). Сведения о машине включают в себя марку, признак «легковая/грузовая», номер, цвет. Рейс характеризуется номером, начальным и конечным пунктами рейса и расстоянием в километрах между ними. Информация о рейсе машины - дата и время начала, дата и время окончания, и масса груза. Ремонт машины характеризуется датой и временем начала. Считать, что машина может подлежать конкретному ремонту только один раз (повторная неисправность приводит к ее списыванию).

Выходные документы: список масс перевезенного груза для каждой грузовой машины за определенный период, сортировка по номерам машин и массам; список ремонтов всех машин, отсортированный по длительности ремонта и по маркам машин.

19. Предприятие

Имеются цеха, участки, детали и операции. Цех и участок характеризуются номерами и ФИО начальника, а участок, кроме того, количеством изготавливаемых на нем деталей. Деталь изготавливается на участке и имеет

код и название. Деталь изготавливается посредством выполнения нескольких операций. Каждая операция имеет название и время выполнения.

Выходные документы: список участков заданного цеха, отсортированный по количеству изготавливаемых деталей; список цехов и входящих в них участков с указанием суммарного времени изготовления деталей на участке.

20. Учет материального положения студентов

Имеются факультеты, специальности, группы и студенты. Факультет характеризуется наименованием и ФИО декана, специальность – названием, шифром и квалификацией. Группа характеризуется шифром, годом поступления и номером. Студент характеризуется ФИО, номером зачетки, зарплатой отца, зарплатой матери и количеством членов семьи.

Выходные документы: список факультетов, специальностей, групп и студентов с подсчетом средней зарплаты студентов в группе; список студентов заданной группы, у которых средняя зарплата на одного члена семьи меньше заданного числа.

21. Изготовление деталей

Имеются цеха, детали, изготавливаемые в цехах, модификации деталей и станки, на которых изготавливаются детали. Деталь может иметь несколько модификаций. Цех характеризуется номером и ФИО начальника, деталь – заводским номером и названием, модификация детали – названием и трудоемкостью, станок – названием и временем обработки.

Выходные документы: список модификаций деталей цехов с минимальной трудоемкостью, упорядоченный по названиям деталей; список деталей указанного цеха, обрабатываемых определенном станке.

22. Потребность в лекарствах

Имеются лекарства, аптеки и больницы. Лекарство характеризуется названием и видом (таблетки, капсулы, ампулы и т.д.). Аптека характеризуется номером, адресом, названием; больница – номером, адресом, ФИО главного врача. В аптеках имеются лекарства по определенной цене за упаковку и в определенном количестве упаковок. Больницам также требуются лекарства в определенном количестве упаковок.

Выходные документы: список лекарств, имеющихся в определенной аптеке, упорядоченный по их названиям; список лекарств, купленных определенной больницей, упорядоченный по виду, с указанием общего количества упаковок каждого вида.

23. Подписка

Имеются почтовые отделения, подписчики и издания. Почтовое отделение имеет номер, адрес, ФИО заведующего. Каждый подписчик обслуживается по месту жительства в почтовом отделении и характеризуется фамилией и адресом. Издание характеризуется индексом, названием и месячной

стоимостью. Подписчик подписывается на каждое издание на определенное число месяцев.

Выходные документы: список подписчиков указанного почтового отделения и указанного издания, упорядоченный по фамилиям; список подписчиков и их изданий с подсчетом суммы подписки каждого подписчика, упорядоченный по фамилиям.

24 . Делопроизводство

Имеются организации, входящие и исходящие документы. Организация имеет название, почтовый адрес, адрес электронной почты. Входящие документы характеризуются номером, типом (письмо, счет, акт, проект, отчет и т.п.), названием, организацией-отправителем, датой отправления, организацией-получателем, датой получения. Исходящие документы имеют номер, тип, название, отправителя, дату отправления, получателя. Следует хранить справочную таблицу, содержащую типы документов.

Выходные документы: список входящих документов определенной организации, отсортированный по отправителю; список исходящих документов определенного типа всех организаций, сгруппированный по получателю, с подсчетом количества документов указанного типа.

25. Обучение на курсах

Организация (код, название, адрес, телефон, электронный адрес) проводит курсы. Каждый курс имеет код, название, тип (информационные технологии, менеджмент и т.п.), количество дней обучения, количество обучаемых, цену, цену с учетом 20% НДС. Цена со временем может меняться. Занятия проводят преподаватели (номер, ФИО, дата рождения, пол, образование, категория). Закрепление преподавателей за курсами осуществляется с помощью документа, в котором указано какой курс будет проводить данный преподаватель, дата начала обучения, дата окончания. В выходные дни – суббота, воскресение – занятия не проводятся. Организация принимает заявки на обучение (обучение проводится только по заявкам и только для организаций), в которой д.б. указаны: название организации, отправляющей сотрудников на обучение, адрес, телефон, электронная почта: количество человек, о каждом сообщается ФИО, должность; на какие курсы и в какой срок необходимо обучение.

Необходимо сформировать прайс-лист организации на заданную дату – перечень курсов, количество дней, цена, цена с учетом НДС, расписание заданного преподавателя за заданный период – название курса, дата начала, дата окончания, заданный курс, за заданный период .

26. Учет горюче-смазочных материалов на автобазе

На автобазе (код, название) ведется учет горюче-смазочных средств (ГСМ), заправляемых в автомобили, выполняющих рейсы по соответствующим путевым листам. При заправке автомобилей в гараже формируется раз-

даточная ведомость, в которой указаны: номер ведомости, число. В одной ведомости м.б. оформлены данные на нескольких водителей. В каждой позиции ведомости записывается: марка автомобиля, гос.номер автомобиля, номер путевого листа, фамилия, инициалы водителя, количество заправленного ГСМ, в литрах и килограммах. ГСМ – это бензин, дизтопливо, дизмасло, автол, солидол, нигрол и т.п.. Ведомость подписывает сотрудник, имеющий должность заправщика. Указываются его ФИО. В заголовке ведомости указывается автобаза, которой принадлежит автомобиль и гараж. У каждой автобазы м.б. несколько гаражей. Ведомости формируется отдельно для каждого гаража заправщиком гаража.

Необходимо подсчитать количество заданного вида ГСМ, заправленных в заданном гараже за заданный период времени, определить виды и количества ГСМ, полученных заданным водителем за заданный период, вывести перечень всех ГСМ, выданных за заданный период, их количество.

27. Банк данных туристических путевок сети турбюро

Предприятие – туристическое бюро. Офисы сети туристических бюро предлагают информацию о имеющихся турах, наличии путевок, их стоимости. Каждый тур характеризуется типом (отдых на море, шор-тур, горные лыжи и т.п.), продолжительностью. Каждому конкретному туру может соответствовать один или более различных населенных пунктов, принадлежащих разным странам. В информации о туре приводятся сведения о наличии гостиниц в населенных пунктах, название, количество звездочек. Цена на конкретный тур и на конкретную дату определяется прайс-листами, периодически выпускаемыми сетью турбюро. Для каждого тура также указывается вид транспорта, пункт отправления группы.

Необходимо сформировать полную информацию о заданном туре, список всех пятизвездочных отелей, через которые проходят туры (конкретный тур), прайс-лист на заданную дату, в котором д.б. указаны название организации владельца сети турбюро, её юридический адрес, телефон, ФИО контактного лица и перечень всех имеющихся туров.

28 Учет животных, птиц, рептилий в зоопарке

Предприятие – крупный зоопарк. Каждому новому питомцу зоопарка присваивается уникальный номер, имя. Необходимо также хранить дату рождения, пол. О птицах дополнительно необходимо хранить сведения о месте зимовки (если такое существует – код, название страны, дата улета, дата прилета), для рептилий необходимо хранить сведения о его нормальной температуре, сроки зимней спячки. Каждому питомцу назначается рацион кормления, который характеризуется номером, названием, типом (детский, диетический, усиленный и т.п. Каждый тип рациона может содержать несколько названий рационов. Рацион может со временем меняться.

Необходимо также учитывать зону обитания животного (название, характеристика). Каждое животное относится к одной зоне обитания. Также

необходимо хранить информацию о том, к какому смотрителю на текущий момент прикреплен питомец. За каждым животным закреплен обязательно один смотритель, а каждый смотритель одновременно может обслуживать нескольких. Также в зоопарке есть ветеринары, которые тоже закреплены за животными. Каждый сотрудник имеет свой личный номер, имя, дату рождения, также необходимо знать номер телефона и семейное положение сотрудника. Если кто-то из одной семьи работает вместе (супруги), необходимо об этом знать.

Необходимо вывести полную информацию по типу и имени питомца зоопарка, список сотрудников, работающих семейными парами – ФИО, дата рождения, телефон, перечень всех животных на текущую дату и номера их рационов.

29 Банк данных насаждений парков

Предприятие по благоустройству парков. Предприятие оказывает такие виды услуг, как: формирование ландшафтов, насаждение парков, озеленение улиц и скверов. Фирма имеет название, юридический адрес. Каждый обслуживаемый парк делится на зоны. Каждому высаживаемому растению присваивается уникальный номер в пределах зоны. Необходимо хранить дату высадки растения и возраст растения. Растение м.б. высажено в парке в многолетнем возрасте. Каждое растение относится к какому-либо одному виду. Режим полив каждого растения зависит от возраста растения и его вида. Каждый полив характеризуется днем (каждый, один раз в неделю и т.п.) временем полива, нормой воды в литрах. Насаждения поливаются максимум один раз в день. Также необходимо иметь информацию о служителях парка, которых ухаживают за насаждениями (ФИО, телефон, адрес). Каждый служитель закрепляется за насаждением графиком (дата). на каждую дату закреплен за насаждением только один служитель. Также есть декораторы парка, о них необходимо хранить информацию о ФИО, телефоне, адресе, образовании, названием законченного учебного заведения, категорией (высшая, средняя и т.п.).

Необходимо вывести полную информацию о насаждениях заданного вида, список сотрудников, работающих на заданную дату – ФИО, дата рождения, телефон, перечень всех растений заданного вида на текущую дату и режимы их полива.

30 Выставочные залы города»

Предприятие – областной союз художников.

Словесное описание предметной области: необходимо иметь информацию о выставочных залах города, выставках, проводимых в них, участниках выставок. Каждый выставочный зал характеризуется название, площадью, адресом, телефоном. Зал может принадлежать какому-либо владельцу – это м.б. городская организация. областная, общественная, частное лицо. Необхо-

димо иметь сведения о владельцах (название или имя, адрес, телефон). Также необходимо хранить информацию о видах выставок, проводимых в выставочных залах – это могут быть выставки изобразительного искусства, прикладного, скульптура и т.п., датах проведения выставок О художниках, которые принимают участие в выставках, необходимо хранить: имя, место и дату рождения, краткую биографическую справку, сведения об образовании. Каждый художник на выставке может представлять несколько работ, необходимо хранить название работы, её исполнение (краски, акварель, скульптура и т.п.), дату создания, размеры: высота, ширина, если это скульптура – объем.

Необходимо вывести список участников и их работы с заданной выставки – название выставки, дата проведения, список работ – название. исполнение, ФИО автора, возраст автора, дата создания, список всех выставочных залов города – дата, название, адрес, занимаемая площадь, владелец, перечень выставок, проходящих в городе с адресами залов на текущий момент – дата, список – название выставки, адрес зала.

Требования к выполнению работы

В данной лабораторной работе студент должен выполнить действия процесса разработки программного обеспечения «Постановка задачи», «Разработка технического задания» и «Анализ требований и определение спецификаций».

При разработке программного обеспечения студенты могут придерживаться следующей последовательности выполнения работы:

1. Выработка системных требований (постановка задачи).

Текстовое описание разрабатываемого программного продукта на русском языке. Формируется на основе задания к работе. В задание могут быть внесены дополнительные функции и ограничения (лучше не увлекаться). Также описываются условия эксплуатации программного продукта. Текстовое описание включает:

- одностраничное текстовое описание предметной области (1-3 стр),
- составление словарей: предметной области, по Абботу, объектно-
- ориентированного,
- формулировка цели и задач проекта ,
- описание функционального назначения системы,
- определение действующих лиц разрабатываемой системы,
- определение отношений между действующими лицами и функциями системы,
- формулировку требований функциональных и не функциональных на разрабатываемое ПО.

2. Разработка технического задания (объединение всех требований в единый документ).

Рекомендуется придерживаться при формировании структуры документа видение из RUP:

- Введение,
- Позиционирование,
- Описания совладельцев и пользователей,
- Краткий обзор продукта,
- Возможности продукта,
- Ограничения,
- Показатели качества,
- Старшинство и приоритеты.
- Анализ и управление рисками,
- Другие требования к изделию,
- Требования к документации.

Анализ и формальное описание требований к программному продукту может включать:

- описание функционального назначения системы, используя SADT диаграммы (детализация минимум до 3-го уровня),
- определение отношения между действующими лицами и вариантами использования для этого определить роли разрабатываемой системы и идентифицировать варианты использования системы,
- формализованный набор требований, разбитый по ролям пользователей программного продукта. Обязательно должен включать ограничения, накладываемые на программный продукт (не менее трех),
- составить полную диаграмму (или несколько диаграмм-по ролям) использования.
- реализовать выбранные варианты использования (один или два) в виде записи сценария на естественном языке и оставшиеся, из выбранных, вариантов использования реализовать диаграммами UML,
- описать нефункциональные и специальные требования, если они необходимы,
- описать поведение разрабатываемой системы;
- построить инфологическую модель разрабатываемой предметной области (нотация Чена).

Требование к отчету

Все отчёты предоставляются на проверку в печатном виде. Отчет по лабораторной работе должен содержать:

Титульный лист,

Задание к лабораторной работе,
Подробное описание выполненной работы,
Перечень используемой литературы.

Контрольные вопросы

1. Почему в процессе определения требований необходимо различать разработку пользовательских требований и разработку системных требований?
2. С чего начинается процесс разработки информационной системы?
3. Какие этапы предшествуют непосредственно реализации системы?
4. Чем отличается Пользователь от Роли? Приведите примеры.
5. Охарактеризуйте назначение концептуальной модели предметной области.
6. Из каких элементов состоит концептуальная модель предметной области?
7. Назовите виды отношений между сущностями в концептуальной модели предметной области.
8. Какая диаграмма UML используется в качестве нотации концептуальной модели предметной области?
9. Чем отличается диаграмма классов уровня анализа от диаграммы классов уровня проектирования?
10. Каковы основные этапы анализа предметной области?
11. В чем суть функционально-модульного и объектно-ориентированного подходов при декомпозиции предметной области?
12. Какие типы связей используются для описания взаимодействия объектов предметной области?
13. Какие модели UML используются для описания поведения системы на этапе анали.
14. Какая модель характеризует статические свойства разрабатываемого программного обеспечения?
15. Что такое Вариант использования?

.

Список использованной литературы

1. Орлов, Сергей Александрович . Технологии разработки программного обеспечения: Разраб. слож. програм. систем : учеб. пособие для вузов по направлению подгот. бакалавров и магистров "Информатика и вычисл. техника" / С. А. Орлов, 2003. - 473 с.

3. Иванова Г. С. Технология программирования : учебник для вузов по направлению "Информатика и вычислительная техника" / Г. С. Иванова, 2011.

4. Лаврищева Е. М. Программная инженерия и технологии программирования сложных систем [Электронный ресурс] : Учебник / Лаврищева Е.М., 2018. - 432 с.

Липаев В. В. Проектирование программных средств : учеб. пособия для вузов по специальности "Автоматизир. системы обраб. информ. и управл. " / В. В. Липаев, 1990. - 301 с.

3. Якобсон, Айвар . Унифицированный процесс разработки программного обеспечения : [Пер. с англ.] / А. Якобсон, Г. Буч, Дж. Рамбо, 2002. - 492 с.

5. Фаулер, Мартин . Архитектура корпоративных программных приложений : [Пер. с англ.] / Мартин Фаулер при участии Д. Райса и др, 2004. - 539 с.

7. Лаврищева Е. М. Программная инженерия. Парадигмы, технологии и CASE-средства : учебник для вузов / Е. М. Лаврищева, 2016. - 280 с.

8. Константайн Л., Локвуд Л. Разработка программного обеспечения. – СПб.:Питер, 2004. – 592 с.

9. ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы

10. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению