**Теоретическая часть**

1.1 Техническое задание

Постановка задачи

Общее описание вопроса. Описание моментов, которые подлежат проработке и автоматизации. В этом разделе описывается назначение программного средства (далее ПС), т.е. какую задачу оно решает и где может найти применение. Указывается полное наименование ПС с расшифровкой слов, которые представлены аббревиатурой. Указывается условное обозначение ПС, которое дали ему авторы разработки, например пакет прикладных программ (ППП) «OMEGA» или программный комплекс (ПК) «НАДЕЖДА» и др. Представляются авторы разработки: Фамилия, имя, отчество, основное место учебы (учеба и работа), а также та часть работы, которая выполнялась в данной курсовой работе.

1.2 Требования к программному средству

В этом разделе должна содержаться следующая информация:

- требования к функциональным характеристикам (состав выполняемых функций, организация входных и выходных данных и т.п.);

- требования к структуре ПС (возможность модернизации, увеличению функциональных возможностей);

- требования к надежности (надежное функционирование, контроль входной и выходной информации и т.п.);

- условия эксплуатации (температура, влажность, количество и

квалификация персонала);

- требования к составу и параметрам технических средств (состав

технических средств с указанием их технических характеристик);

- требования к информационной и программной совместимости

(требования к информационном структурам, языкам программирования);

- требования к транспортировке и хранению;

- специальные требования.

1.3 Требования к программной документации

Общие сведения о документации, сопровождающей программный продукт.

Указание видов документации, которые должны быть разработаны к проектируемому программному продукту.

1.4 Стадии и этапы разработки

Описание жизненного цикла программного продукта и его привязка к разрабатываемой программе. Указываются сроки разработки, т. е. временной период, в течение которого данная работа должна быть выполнена. А также, на какие этапы данная работа разбита.

Анализ требований и определение спецификаций к эскизному проекту по пред­метной области «Детский сад» сначала для структурного подхода, а затем для объектного подхода будет представлен ниже в виде набора диаграмм. Для реализации анализа требований при структурном подходе необходимо создать следующий набор диаграмм: диаграмма потоков данных, диаграм­ма «сущность-связь», диаграмма переходов состояний, функциональная

диаграмма, спецификация процессов (этот вариант схемы выполняется са­мостоятельно учащимся), словарь терминов (этот вариант схемы выполня­ется самостоятельно учащимся). При объектном подходе к программиро­ванию надо создать следующий набор диаграмм: диаграмма вариантов ис­пользования, диаграмма классов, диаграмма последовательности действий, диаграмма деятельности.

Диаграммы потоков данных (Data Flow Diagrams - DFD) представ­ляют собой иерархию функциональных процессов, связанных потоками данных. Цель такого представления - продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Для построения DFD традиционно используются две различные но­тации, соответствующие методам Йордона-ДеМарко и Гейна-Сэрсона. Эти нотации незначительно отличаются друг от друга графическим изображе­нием символов (далее в примерах используется нотация Гейна-Сэрсона).

Сначала строится контекстная диаграмма потоков данных, где опре­деляются внешние сущности и потоки данных между программой и внеш­ней сущностью. Контекстная диаграмма потоков данных к эскизному про­екту при структурном анализе для предметной области «Детский сад» по­казана на рисунке 2.

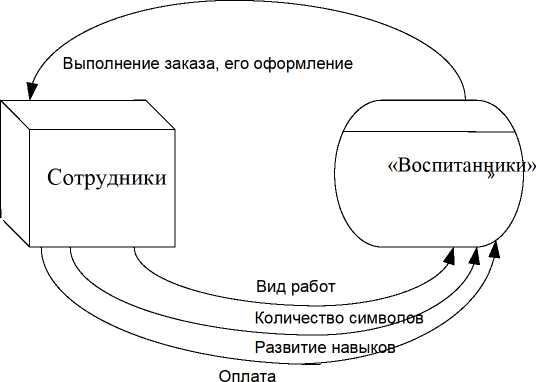


Рис. 2. Контекстная диаграмма потоков данных

После построения контекстной диаграммы, выполняется детализация проекта с последующим построением детализирующей диаграммы пото­ков данных, как показано на рисунке 3.

Просмотр данных



Рис. 3. Детализирующая диаграмма потоков данных

Диаграмма «сущность-связь» (ER - Entity-Relationship) предназначе­ны для графического представления моделей данных разрабатываемой программной системы и предлагают некоторый набор стандартных обо­значений для определения данных и отношений между ними. С помощью этого вида диаграмм можно описать отдельные компоненты концептуаль­ной модели данных и совокупность взаимосвязей между ними, имеющих важное значение для разрабатываемой системы.

Графическая модель данных строится таким образом, чтобы связи между отдельными сущностями отражали не только семантический харак­тер соответствующего отношения, но и дополнительные аспекты обяза­тельности связей, а также кратность участвующих в данных отношениях экземпляров сущностей.

Диаграмма «сущность-связь» для предметной области «Детский сад» представлена на рисунке 4.

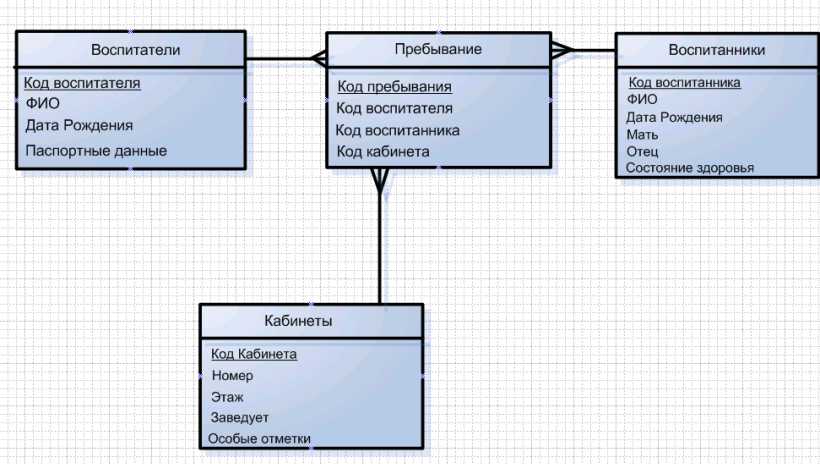


Рис. 4. Диаграмма «сущность-связь»

Диаграммы переходов состояний (STD - State Transition Diagrams)) предназначены для моделирования и документирования аспектов систем, зависящих от времени или реакции на событие. Они позволяют осуществ­лять декомпозицию управляющих процессов и описывают отношения между входными и выходными управляющими потоками для управляю­щего процесса-предка.

Диаграмма переходов состояния для программного продукта по предметной области «Детский сад» представлена на рисунке 5.



Рис 5. Диаграмма переходов состояния

Функциональными называют диаграммы, в первую очередь отража­ющие взаимосвязи функций разрабатываемого программного обеспечения.

Они создаются на ранних этапах проектирования систем, для того чтобы помочь проектировщику выявить основные функции и составные части проектируемой системы и, по возможности, обнаружить и устранить существенные ошибки.

Блоки на диаграмме размещаются по «ступенчатой» схеме в соответ­ствии с последовательностью их работы или доминированием, которое по­нимается как влиянием, которое понимается как влияние блоком на дру­гие.

Построение модели начинают с единственного блока, для которого определяют исходные данные, результаты, управление и механизмы реа­лизации.

Функциональная диаграмма для информационной системы «Детский сад» представлена на рисунке 6.

18



Рис. 6. Функциональная диаграмма по предметной области

Методология функционального моделирования IDEF0 - это техноло­гия описания системы в целом как множества взаимозависимых действий или функций.

При объектном подходе по предметной области «Детский сад» необ­ходимо создать следующие диаграммы: диаграмма вариантов использова­ния, диаграмма классов, диаграмма последовательности действий, диа­грамма деятельности, диаграмма состояний.

Диаграмма вариантов использования (use case diagram) - диаграмма, на которой изображаются отношения между актерами и вариантами ис­пользования [1].

Диаграмма вариантов использования - это исходное концептуальное представление или концептуальная модель системы в процессе ее проек­тирования и разработки.

По предметной области «Детский сад» информационные процессы на диаграмме можно представить согласно рисунку 7.

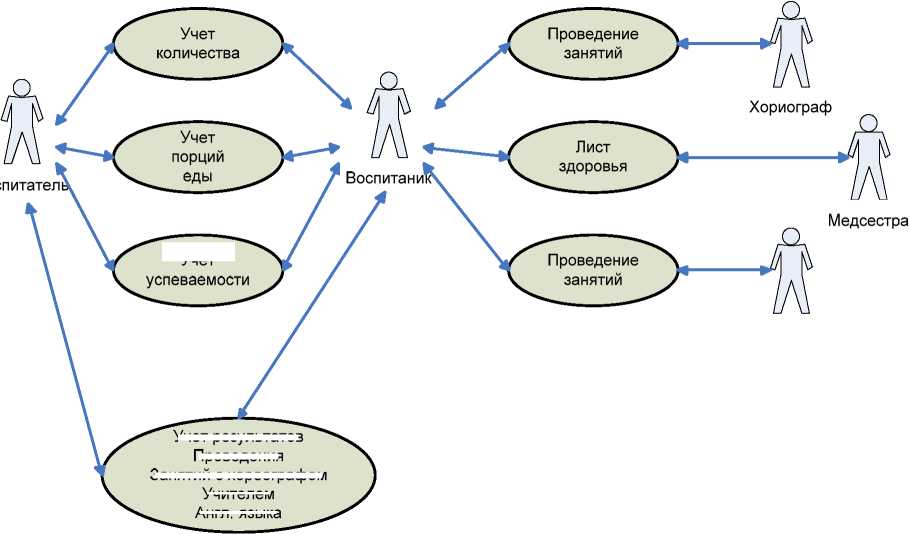


Рис. 7. Диаграмма вариантов использования

Диаграмма классов может отражать различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и под­системы, а также описывает их внутреннюю структуру и типы отношений. Диаграммы классов обычно содержат следующие сущности: классы; ин­терфейсы; кооперации; отношения зависимости, обобщения и ассоциации.

Диаграмма классов по предметной области «Детский сад» представ­лена на рисунке 8.

Диаграмма классов



Рис. 8. Функциональная диаграмма по предметной области

Диаграмма последовательностей системы (sequence diagram) - гра­фическая модель, которая для определённого сценария варианта использо­вания показывает динамику взаимодействия объектов во времени [1].

Диаграмма последовательности является одной из разновидности диаграмм взаимодействия и предназначена для моделирования взаимодей­ствия объектов Системы во времени, а также обмена сообщениями между ними. Диаграмма последовательности действий представлена на рисунке 9.



Требование

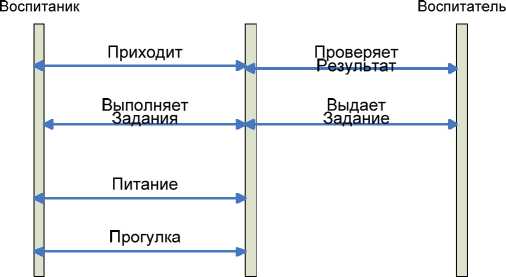


Рис. 9. Диаграмма вариантов использования

Для моделирования процесса выполнения операций в языке UML используются так называемые диаграммы деятельности. На этапе анализа требований и уточнения спецификаций диаграммы деятельностей позво­ляют конкретизировать основные функции разрабатываемого программно­го обеспечения. Диаграмма деятельности по предметной области «Детский сад» представлена на рисунке 10 [3, с. 148].





Рис. 10. Диаграмма вариантов использования

Главное предназначение диаграммы состояний - описать возможные последовательности состояний и переходов, которые в совокупности ха­рактеризуют поведение элемента модели в течение его жизненного цикла. По большей части под этим подразумевается моделирование поведения реактивных объектов. Реактивным называется объект, поведение которого лучше всего характеризуется его реакцией на события, произошедшие вне его собственного контекста. У реактивного объекта есть четко выражен­ный жизненный цикл, когда текущее поведение обусловлено прошлым. Если внешние действия, изменяющие состояния системы, инициируются в 21

произвольные случайные моменты времени, то говорят об асинхронном поведении модели [3, с.151]. Диаграмма состояний по предметной области «Детский сад» представлена на рисунке 11.

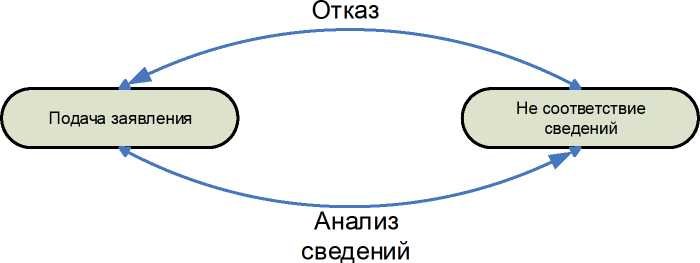


Рис. 11. Диаграмма вариантов использования

Диаграммы состояний используются для моделирования динамиче­ских аспектов системы.