Частное учреждение образование

«Колледж бизнеса и права»

ОТЧЕТ по ПРАКТИКЕ

по РАЗРАБОТКЕ и сопровождению программного обеспечения

ОП Т.993012

Выполнил А.П.

Руководители практики Н.В. Ржеутская

Е.В.Багласова

2016

Содержание

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

ОП Т.093016.401

Разраб.

Провер.

Т. Контр.

Н. Контр.

Утверд.

Отчет по практике

по разработке и сопровождению

Лит.

Листов

КБП

1 Программа практики 3

2 Объектно-ориентированный анализ и проектирование системы 4

2.1 Сущность задачи 4

2.2 Проектирование модели 5

3 Вычислительная система

3.1 Требования к аппаратным и операционным ресурсам

3.2 Инструменты разработки

4 Проектирование задачи

4.1 Требования к приложению

4.2 Концептуальный прототип

4.3 Организация данных

4.4 Функции и элементы управления

4.5 Проектирование справочной системы приложения

5 Описание программного средства

5.1 Общие сведения

5.2 Функциональное назначение

5.3 Входные и выходные данные

6 Программирование на языке С# в среде VisualStudio

6.1 Технические требования

6.2 Функциональное тестирование

7 Применение

7.1 Назначение программы

7.2 Условия применения

7.3 Справочная система

Заключение

Список используемых источников

## 1 Программа практики

Целями практики по разработке и сопровождению программного обеспечения являются:

- закрепление знаний, связанных с технологией обработки информации;

- приобретение навыков разработки программ с использованием современных средств обработки экономической и деловой информации;

- выработка умений применять средства стандартных библиотек в практических задачах.

Задачами практики по разработке и сопровождению ПО являются:

- углубленное изучение принципов организации программного обеспечения и технологии его проектирования;

- владение методами надежного программирования;

- умение разрабатывать программы в соответствии с промышленными требованиями, обеспечивая высокий уровень качества программного обеспечения и экономической эффективности;

- изучение способов определения экономической себестоимости и эффективности разработки программного обеспечения;

- оформление комплекта документации на созданное программное обеспечение.

##### Таблица 1 - Календарный график работы

|  |  |
| --- | --- |
| Период | Наименование и содержание работ |
| 09.02.18 | Изучение требований к рабочей дисциплине и технике безопасности.  Изучение нормативных документов. |
| 11.02.18-  17.02.18 | Ознакомление с различным программным обеспечением, используемым для разработки программ |
| 18.02.18-  24.02.18 | Получение индивидуальных заданий по созданию программ для обработки экономической и деловой информации. Обсуждение тем индивидуальных заданий. |
| 25.02.18-  02.03.18 | Изучение общих требований к программному обеспечению обработки экономической и деловой информации. Требования к интерфейсу пользователя, выходным документам (печатные, экранные формы, форматы результирующих файлов с результатами расчетов). |
| 03.03.18-  23.03.18 | Анализ предметной области индивидуального задания. Выбор и обоснование языка и среды программирования. Описание функциональных возможностей задачи. |
| 24.03.18-  06.04.18 | Технология создания программного обеспечения. Правила построения диаграмм UML. Проектирование ПО при помощи CASE-средства Rational Rose |
| 07.04.18-  20.04.18 | Конструирование алгоритма и его реализация в выбранной визуальной среде программирования. Разработка интерфейса .Проектирование классов, определение свойств и методов, определение связей между классами. |
| 21.04.18-  04.05.18 | Создание информационной базы задачи. Тестирование и отладка программы с использованием подготовленных тестовых примеров.. Проектирование модулей, позволяющих осуществить защиту программного продукта от несанкционированного доступа.. |
| 05.05.18-  25.05.18 | Критерии оценки качества программного продукта. Изучение нормативных документов, регламентирующих оформление технического проекта. Создание документации по эксплуатации и сопровождению разработанного ПС. Проведение испытания ПС в нормальных и критических условиях.. |
| 26.05.18-  01.06.18 | Анализ возможности модернизации проекта с учетом альтернативных решений. Оформление аннотации на программное средство. |
| 02.06.18-  15.06.18 | Оформление отчета по практике. Подготовка листингов программ с дополнением комментариев. Распечатка экранных форм. |
| 16.06.18-  21.06.18 | Демонстрация программ по индивидуальным заданиям.  Защита отчета по практике. |

2 Объектно-ориентированный анализ и проектирование системы

2.1 Сущность задачи

Ремонт квартир, сложный технологический процесс и выполнять его нужно поэтапно. Первый этап - это составление всех документов, сметы на производство ремонтно - отделочных работ и стройматериалов, оформление договора. Смета это - основной документ на основании которого проводится ремонт квартир. Поэтому она должна быть составлена с учетом всех видов работ и стройматериалов. При обращении, специалист выезжает к на объект, собирает всю необходимую информацию, делает все замеры, а далее составляется смета быстро, квалифицированно и бесплатно. Именно для этих целей и создаётся данное программное средство. Когда будут соблюдены все формальности по оформлению документов на производство ремонтно - отделочных работ, заключен договор, начинается второй этап ремонта. Ремонт квартир на втором этапе работ заключает в себе следующие виды работ; демонтажные работы, строительные работы (возведение стен, перегородок). Если необходимо изменить размеры комнат, совместить или наоборот разделить санузел, перенести и расширить дверные проемы, то для этого необходимо будет оформление перепланировки.

Третий этап ремонта квартиры - это электромонтажные работы, прокладка нового силового кабеля, телевизионного, телефонного и интернет кабеля, сантехнические работы, штукатурные работы. И последний этап - это отделочные работы: отделка потолка; отделка стен; настил полов.

В косметический ремонт квартиры входят следующие виды работ, представленные в таблице 2. Заказ включает в себя вызов мастера на место проведения работ для оценки стоимости работ. Вся расценка идет за 1 кв. м. работ, минимальный заказ 7 кв. м.

Таблица 2 – Расценки на некоторые виды косметических работ

|  |  |
| --- | --- |
| Наименование работ | Цена за 1 кв.м.,у.е |
| Расчистка потолков | 7 |
| Расчистка стен | 2 |
| Частичная шпаклевка стен | 6 |
| Частичная шпаклевка потолков | 3 |
| Покраска потолков | 6 |
| Наклеивание обоев | 15 |
| Настил покрытия (ламинат, линолеум, ковер) | 8 |

В капитальный ремонт квартиры входят следующие виды работ, представленные в таблице 1.2.

ЗДЕСЬ могут быть расчётные формулы, если они есть в теме задания.

Например, расчет зарплаты или вычисление процента по вкладу и т.п.

ИЛИ ДРУГОЙ ВАРИАНТ:

На данный момент существует несколько аналогов данного приложения от унитарного предприятия «ЛориТрансСервис», [http://loritransservis.by](http://loritransservis.by/) и ЧУП «Астра-лайн», <http://astra-lain.com/>. ». Однако все эти ПП не обеспечивают полноценного учета всей деятельности и всех его услуг, а лишь автоматизируют учет отдельных его частей. «Астра-лайн», не представляет подробного прайс листа с расценками для каждого вида работ, а «ЛориТрансСервис» не имеет возможности заполнения сметы на выполнение ремонтно-строительных работ. Для данного программного средства эти недостатки будут исправлены и доработаны.

2.2 Проектирование модели

Главной целью проектирования моделей является отображение функциональной структуры объекта, то есть производимые ими действия и связи между этими действиями. Наиболее распространенным средством моделирования данных являются диаграммы «сущность-связь» (ERD), которые 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. Основными понятиями данной нотации являются понятия сущности и связи. При этом под сущностью понимается произвольное множество реальных или абстрактных объектов, каждый из которых обладает одинаковыми свойствами и характеристиками. В этом случае каждый рассматриваемый объект может являться экземпляром одной и только одной сущности, должен иметь уникальное имя или идентификатор, а также отличаться от других экземпляров данной сущности. Связь определяется как отношение или некоторая ассоциация между отдельными сущностями. Примерами связей могут являться родственные отношения типа «отец-сын» или производственные отношения типа «начальник-подчиненный». Другой тип связей задается отношениями «иметь в собственности» или «обладать свойством».

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. Информационная модель базы представлена на диаграмме «Сущность-связь». Данная диаграмма представлена на рисунке 1.1.

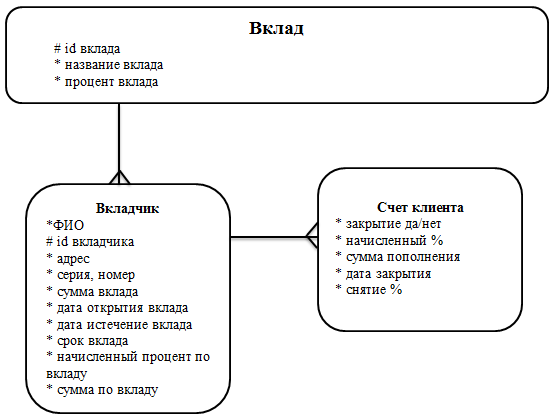


Рисунок 1.1 - Диаграмма «сущность-связь» в нотации Баркера

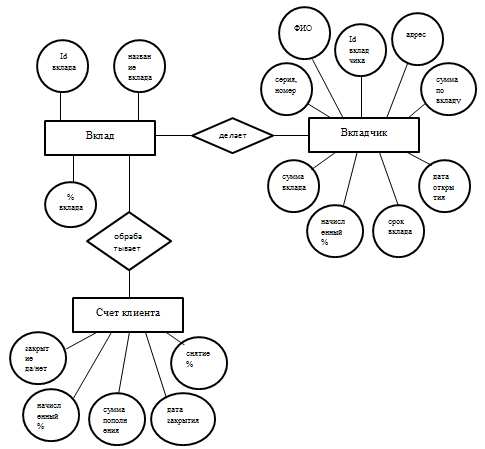


Рисунок 1.2 - Диаграмма ERD в нотации Ричарда Чена

Исходя из исследования предметной области, можно выделить следующие сущности разработки: вклад, вкладчик, счет клиента.

Для сущности «Вклад» можно выделить следующие атрибуты:

- название вклада;

- процент по вкладу.

Для сущности «Вкладчик» можно выделить следующие атрибуты:

- фамилия имя и отчество вкладчика;

- серия и номер паспорта;

- адрес прописки;

- сумма по вкладу;

- дата открытия вклада;

- срок вклада.

Для сущности «Счет клиента» можно выделить следующие атрибуты:

- закрытие;

- начисленный процент;

- сумма пополнения;

- дата открытия;

- снимаемый процент.

В рамках языка UML все представления о модели сложной системы фиксируются в виде специальных графических конструкций – диаграмм. В терминах языка UML определены следующие виды диаграмм: диаграмма вариантов использования, диаграмма классов, диаграмма деятельности, диаграмма последовательности, диаграмма компонентов.

Перечень этих диаграмм представляет собой неотъемлемую часть графической нотации языка UML, сам процесс объектно-ориентированного программирования (ООП) неразрывно связан с процессом построения этих диаграмм. Совокупность построенных таким образом диаграмм содержит всю информацию, необходимую для реализации проекта сложной системы.

Диаграмма – граф специального вида, состоящий из вершин в форме геометрических фигур, которые связаны между собой ребрами или дугами.

Суть диаграммы вариантов использования состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью, так называемых вариантов использования.

Варианты использования описывают не только взаимодействия между пользователями и сущностью, но также реакции сущности на получение отдельных сообщений от пользователей и восприятие этих сообщений за пределами сущности. Варианты использования могут включать в себя описание особенностей способов реализации сервиса и различных исключительных ситуаций, таких как корректная обработка ошибок системы. Множество вариантов использования в целом должно определять все возможные стороны ожидаемого поведения системы.

В данной проектируемой системе в качестве актера выступает сотрудник отдела банка, который служит источником воздействия на моделируемую систему.

К основным функциям разрабатываемой программы относятся следующие функции:

* составить смету;
* осуществить работу с базой данных;
* фиксировать данные о клиенте;
* оформить договор заказа.

К вспомогательным функциям программы относятся:

* предоставить список ремонтно-строительных работ;
* предоставить справочной информации об авторе;
* предоставить справочной информации о программе;
* предоставить возможности выбора оплаты;
* предоставить информации о занятости бригады строителей.

Диаграмма вариантов использования представлена на листе 1 графической части.

Центральное место в ООП занимает разработка логической модели системы в виде диаграммы классов. Нотация классов в языке UML проста и интуитивно понятна. Схожая нотация применяется и для объектов — экземпляров класса, с тем различием, что к имени класса добавляется имя объекта, и вся надпись подчеркивается.

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма может отражать различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывать их внутреннюю структуру и типы отношений. На данной диаграмме не указывается информация о временных аспектах функционирования системы. С этой точки зрения диаграмма классов является дальнейшим развитием концептуальной модели проектируемой системы.

Диаграмма классов представляет собой некоторый граф, вершинами кото­рого являются элементы типа «классификатор» и которые связаны различными типами структурных отношений. Следует заметить, что диаграмма классов может также содержать интерфейсы, пакеты, отношения и даже отдельные экземпляры, такие как объекты и связи. Когда говорят о данной диаграмме, имеют в виду статическую структурную модель проектируемой системы. Поэтому диаграмму классов принято считать графическим представлением таких структурных взаимосвязей логической модели системы, которые не зависят от времени.

Класс в языке UML служит для обозначения множества объектов, которые обладают одинаковой структурой, поведением и отношениями с объектами из других классов. Графически класс изображается в виде прямоугольника, который дополнительно может быть разделен горизонтальными линями на разделы или секции. Здесь указывается: имя класса, атрибуты и операции (методы).

Обязательным элементов обозначения класса является его имя. На начальных этапах разработки диаграммы отдельный класс может обозначаться простым прямоугольником с указанием только имени соответствующего класса. По мере проработки отдельных компонентов диаграммы классов дополняются атрибутами и операциями. Предполагается, что окончательный вариант диаграммы содержит наиболее полное описание классов, которые состоят из трех разделов или секций. Состав некоторых основных классов проекта представлен в таблице 1.5. Диаграмма классов представлена на листе 2 графической части.

Таблица 1.5 – Состав некоторых основных классов проекта

|  |  |  |
| --- | --- | --- |
| Имя класса | Имя поля | Методы |
| UserForm | Label1,2;Edit1,2,3;  TADOQuery;  TADOConnection;  BitBtn1,2,3 | BitBtn1,2,3Click; FormCreate |
| VkladForm | DBGrid1; MainMenu1; MenuItem1..12; Labe1,2,3,4; Edit1,2,3,4; RadioGroup1,2,3; GroupBox1,2,3; Navigator; Button1,2,3. | MenuItem1..12Click; FormActivate; FormCreate; Button1,2,3Click; FormCreate |
| AddForm | Label1..8; BitBtn1,2; DBEdit1..6; DBComboBox1,2;. | BitBtn1,2Click; FormCreate |
| PopolForm | Label1..6; BitBtn1,2; DBEdit1..6. | FormCreate; FormCreate |

При моделировании поведения проектируемой или анализируемой системы возникает необходимость детализировать особенности алгоритмической и логической реализации выполняемых системой операций. Для моделирования процесса выполнения операций в языке UML используются так называемые диаграммы деятельности. Каждое состояние на диаграмме деятельности соответствует выполнению некоторой элементарной операции, переход в следующее состояние срабатывает только при завершении этой операции. Графически диаграмма деятельности представляется в форме графа, вершинами которого являются состояния действия, а дугами - переходы от одного состояния действия к другому.

Диаграммы деятельности - частный случай диаграмм состояний. Основная цель использования таких диаграмм - визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения. Диаграмма деятельности представлена на листе 3 графической части.

Для моделирования взаимодействия объектов в UML используется диаграмма последовательности. Диаграмма последовательности представлена на листе 4 графической части.

Все предыдущие диаграммы отражали концептуальные аспекты построения модели системы и относились к логическому уровню представления и оперировали понятиями, которые не имеют самостоятельного материального воплощения. Для описания реальных сущностей предназначен другой аспект модельного представления, а именно физическое представление модели. Для физического представления модели используются диаграмма компонентов.

Диаграмма компонентов описывает объекты реального мира – компоненты программного обеспечения. Эта диаграмма позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный, бинарный и исполняемый код.

Диаграмма компонентов представлена на листе 5 графической части и содержит следующие компоненты:

* программные компоненты, созданные в среде VStudio 2/0;
* файл базы данных «db.mdf»;
* файл справочной системы «help.chm»;
* текстовый документ «Zakaz.doc».

3 Вычислительная система

3.1 Требования к аппаратным и операционным ресурсам

3.2 Инструменты разработки

4 Проектирование задачи

4.1 Требования к приложению

Данное приложение не требует никаких специальных средств защиты, либо ограничений прав доступа к данным.

ИЛИ

В программном средстве будут предусмотрены доступ к данной программе с помощью пароля и логина пользователя, для защиты информации будет предусмотрено разграничение прав пользователей.

Разрабатываемое приложение должно иметь понятный и удобный в использовании интерфейс, чтобы взаимодействие между программой и пользователем было максимально упрощено. Для того чтобы интерфейс был понятен пользователю, необходимо оформить диалог приложения с пользователем, обладающий кнопочным меню, позволяющим перемещаться по записям. Программное приложение должно обеспечивать оперативный поиск необходимой информации по предоставленным критериям, для этого будут разработаны процедуры отбора информации. Кроме всего вышеперечисленного, разрабатываемое приложение позволит контролировать и вести статистику работ.

Все входные данные должны проверяться на ошибки. При совершении пользователем неверных действий или введении неверных данных будут выводиться сообщения с пояснением ошибок.

Функциональные требования представлены на диаграмме SADT, которая отображает вначале всю систему в виде простейшей компоненты – одного блока и дуг, изображающих интерфейсы с функциями вне системы. Единственный блок представляет всю систему как единое целое, имя, указанное в блоке, является общим. Это верно и для интерфейсных дуг – они также представляют полный набор внешних интерфейсов системы в целом. Диаграмма нулевого уровня представлена на рисунке 3.1.

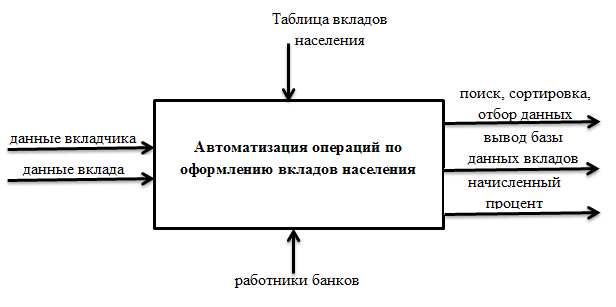


Рисунок 3.1 – Функциональная SADT диаграмма нулевого уровня

Далее блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами. Эти блоки представляют основные подфункции исходной функции. Данная декомпозиция выявляет полный набор подфункций, каждая из которых представлена как блок, границы которого определены интерфейсными дугами. Каждая из этих подфункций может быть декомпозирована подобным образом для более детального представления. Диаграмма первого уровня детализации представлена на рисунке 3.2.

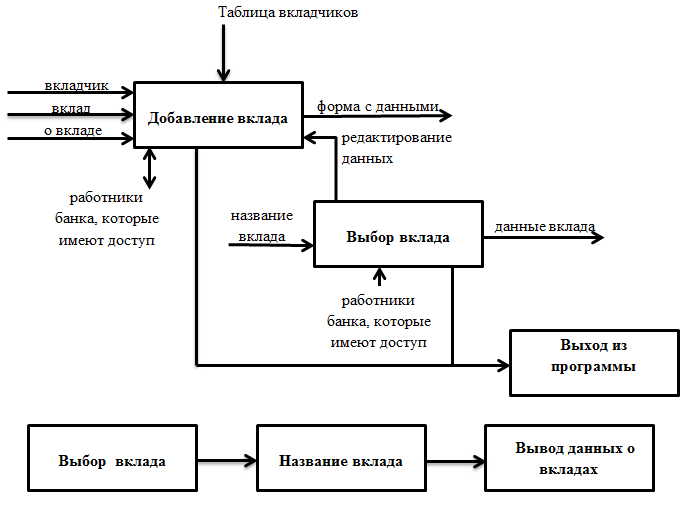


Рисунок 3.2 – Функциональная SADT диаграмма первого уровня детализации

Кроме этого, при разработке форм необходимо соблюдать определённые требования: формы в приложении должны быть выполнены в едином стиле, сдержанной цветовой гамме, иметь стандартные элементы управления данными на всех формах, не должны быть перегружены информацией, и в свое время понятны простому пользователю. При конструировании форм в необходимых случаях нужно предусмотреть возможность защиты данных от изменения, установить ограничения на корректировку записей, а также отсутствие данных при выполнении отбора записей в базе данных.

В создаваемом программном средстве должна быть справочная система, в которой будут предусмотрены все возможные способы работы системы. Справочная система должна содержать ответы на вопросы, которые могут возникнуть у пользователей.

4.2 Концептуальный прототип

Концептуальный прототип представляет собой описание внешнего пользовательского интерфейса – систему меню и форм.

Все действия можно будет выполнить с помощью главной формы, которая будет появляться после формы с приветствием. Главная форма будет иметь кнопочное меню программы позволяющее вызывать остальные формы программы, осуществлять различные манипуляции с данными, а также получать справочную информацию о работе с приложением. Используя меню, пользователь должен иметь возможность завершить работу всего приложения с сохранением сделанных изменений.

Главное меню программы будет представлено формой Main на которой находятся основные элементы управления программой. С помощью кнопок на главной форме можно получить доступ к другим формам оформления заказа, просмотра услуг, редактирования заказов, справке и возможность выйти из приложения. Структура меню представлена в таблице 3.1.

Таблица 3.1 – Структура меню

|  |  |  |
| --- | --- | --- |
| Файл |  | |
| 1.1 | | Открыть в Excel |
| 1.2 | | Информация о вкладах |
| 1.3 | | Пользователи |
| 1.4 | | Виды вкладов |
| 1.5 | | Выход |
| Операции с вкладом |  | |
| 2.1 | | Просмотр вклада |
| 2.2 | | Добавить вклад |
| 2.3 | | Редактировать вклад |
| 2.4 | | Пополнить вклад |
| 2.5 | | Удалить вклад |
| 2.6 | | Закрыть досрочно |
| Справка |  | |
| 3.1 | | О программе |
| 3.2 | | Справка |

Для добавления нового вкладчика будет разработана специальная форма ввода, представленная на рисунке 3.2.

ФИО клиента

Адрес:

Серия, номер

Вид вклада

Сумма вклада, руб.

Дата открытия вклада

Срок вклада

Дата истечение вклада

Фамилия клиента

Адрес прописки:

Серия и номер паспорта

Вид вклада

Сумма вклада, руб.

Дата открытия вклада

Срок вклада

Дата истечения срока

Сохранить

Отмена

Рисунок 3.2 – Проект формы для добавления нового вкладчика

Панель инструментов для организации поиска и сортировки информации представлена на рисунке 3.3.

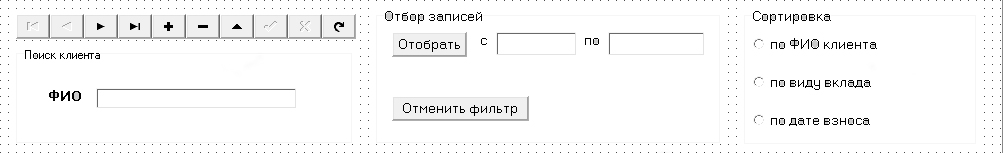


Рисунок 3.3 – Проект панели инструментов

Для отображения сформированных ведомостей о графике работы персонала, о оборудованиях и медицинских препаратов поликлиники будет выбран текстовый редактор Word с помощью, которого можно будет просмотреть и вывести документ на печать.

4.3 Организация данных

СУБД Access - широко распространенное офисное приложение Microsoft Office, которое служит удобным инструментом для ввода, анализа и представления данных и обеспечивает высокую скорость разработки приложений. В Access база данных обозначает файл, содержащий набор информации. Каждая база данных состоит из основных объектов: таблиц, запросов, форм, отчётов, страниц, макросов и модулей. MS Access может работать одновременно только с одной базой данных. Но одна БД Access может включать сотни таблиц, форм, запросов, отчётов, макросов и модулей, которые хранятся в одном файле с расширением mdb.

Таблица «Вкладчики» хранит информацию о вкладах, структура приведена в таблице 3.1

Таблица 3.1 — Структура таблицы «Вкладчики»

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | Тип данных | Размер,  байт | Описание |
| Nom | Счетчик | 255 | Идентификационный номер аттестации |
| Fio | Текстовый | 255 | ФИО вкладчика |
| Adres | Текстовый | 255 | Адрес прописки вкладчика |
| Pasport | Текстовый | 255 | Паспортные данные |
| Vid | Текстовый | 255 | Название вклада выбранного вкладчикам |
| Summa | Числовой | 255 | Сумма внесения денег по вкладу |
| D\_open | Дата/время | 255 | Дата заключение договора |
| Srok | Числовой | 255 | Срок на который заключается договор |
| D\_close | Дата/время | 255 | Дата истечения вклада по договору |
| Procent | Числовой | 255 | Процент по текущему вкладу |

Таблица «Вклад» хранит информацию о видах вкладов, которые предоставляются вкладчику для выбора вклада, структура приведена в таблице 3.2

И так далее по всем таблицам!!!!

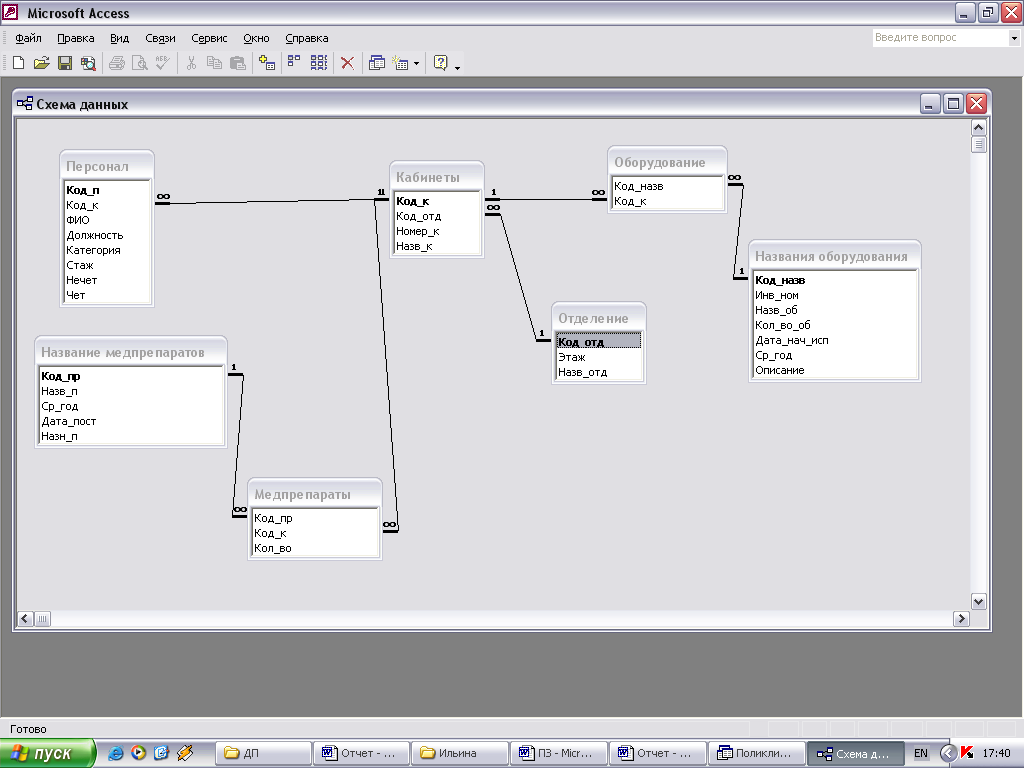


Рисунок 3.4- Схема данных

4.4 Функции и элементы управления

На основании диаграммы вариантов использования в программном приложении «Автоматизация управления работой поликлиники» были реализованы следующие функции:

* составление сметы ;
* осуществление работы с базой данных;
* регистрация клиента;
* оформление договора заказа.

Рассмотрим основные функции программы.

За добавление вклада отвечает метод BitBtn2Click, код которого представлен ниже

procedure TAddForm.BitBtn2Click(Sender: TObject);

begin

if DataModule1.VkladchikTable.Modified

then begin DataModule1.VkladchikTable.Post; end;

AddForm.Close;

end;

За поиск по вкладчику отвечает метод FilterFamEditChange, код которого представлен ниже

procedure TVkladForm.FilterFamEditChange(Sender: TObject);

begin

If Length(FilterFamEdit.Text)>0 then DataModule1.VkladchikTable.Filtered:=True

else DataModule1.VkladchikTable.Filtered:=False;

DataModule1.VkladchikTable.Filter:='ФИО>'''+FilterFamEdit.Text+'''';

end;

За удаление данных отвечает метод N6Click, код которого представлен ниже

procedure TVkladForm.N6Click(Sender: TObject);

begin

if messagedlg('Вы действительно хотите удалить : '+DataModule1.VkladchikTableDSDesigner.AsString+'?',mtWarning, [mbYes,mbNo],0)=mrYes then DataModule1.VkladchikTable.Delete;

end;

Весь программный код представлен в приложении А.

4.5 Проектирование справочной системы приложения

Для полной работы пользователя с приложением его необходимо обеспечить качественной справочной системой, в которой должны быть приведены методы и приёмы работы с приложением, данные о том, что произойдет после нажатия на определенную кнопку или при выборе пункта меню, сведения о том, какую информацию и в каком виде следует вводить в соответствующие поля, каким образом можно вносить изменения в базу данных, и как вывести данные на печать.

Пользователь должен иметь доступ к справочной информации на протяжении всей работы с программой. Для этого пользователь может воспользоваться пунктом меню «Справка». Система справочной информации данного проекта будет содержать два вида подсказок. Это всплывающая подсказка, возникающая при наведении на объект курсором мыши и наиболее полный вариант, представляющий собой отдельное приложение с расширением .chm.

Данная справочная информационная система будет разработана с помощью программы HTML Help Workshop. Будут созданы справочные Web-документы, с помощью текстового процессора Word и сохранены как Web-документы. Затем, с помощью вышеназванной программы, будет создана справочная система (программа преобразует все Web-документы и создаст справку с именем - Справка.chm).

Система будет выдавать справочную информацию по следующим разделам:

- введение (рассматривается общая информация о данном приложении);

- удаление записей (описывается, как удалить записи: списанное оборудование, списанные препараты, увольнение, уход персонала);

-добавление записей (описывается, как добавить запись: поступление оборудования, препаратов, принятие нового персонала на работу);

- редактирование записей (описывается, как внести изменения, поправки в записи: ошибка в названии оборудования, препаратов. Изменение графика работы врачей, название кабинетов и так далее);

- поиск информации (рассматривается поиск по критериям представленные пользователю: найти врача по фамилии, по кабинету и другие критерии);

- запросы пользователей (анализ по персоналу поликлиники, запрос на поступление и списание оборудования в количествах и так далее );

- документация (ведомости, выводимые на печать: ведомость поступивших медицинских препаратов).

5 Описание программного средства

5.1 Общие сведения

- обозначение и наименование приложения

- программное обеспечение, необходимое для функционирования программы

* **перечень модулей (библиотек), подключаемых файлов (связь с диаграммой компонентов)**
* **объем занимаемой памяти приложением, объем памяти для файлов и т.д.**

- инсталляция программного средства (если есть)

5.2 Функциональное назначение

- классы решаемых задач и назначение программы

- средства защиты и разграничения доступа к данным

- сетевая поддержка

5.3 Входные и выходные данные

- характер, организация и предварительная подготовка

- формат, описание и способ кодирования

6 Методика испытаний

6.1 Технические требования

Минимальные системные требования к приложению:

* 10 Мб свободного места на винчестере;
* 10 Мб свободной оперативной памяти;
* процессор Intel Pentium 300 MHz или совместимый аналог;
* видеокарта с объемом памяти не менее 16 Мб.
* операционная система семейства Microsoft Windows NT;
* клавиатура, мышь.

6.2 Функциональное тестирование

В процессе написания программного продукта необходимо производить тестирование на правильность работы приложения. Одной из основных задач тестирования является устранение ошибок, происходящих при вводе данных.

Функциональное тестирование – это тестирование функций приложения на соответствие требованиям. Оценка производится в соответствии с ожидаемыми и полученными результатами (на основании функциональной спецификации), при условии, что функции отрабатывали на различных значениях.

Тестирование программы будет производиться последовательно, переходя из одной части программы в другую. Во время теста будут проверяться все действия с программой, навигация пунктам меню, которые может произвести пользователь. После чего, все собранные и найденные ошибки будут исправлены.

6 Применение

6.1 Назначение программы

описать сведения о назначении программного средства, области применения, классе решаемых задач, ограничениях, накладываемых на область применения.

6.2 Условия применения

необходимо описать требования к необходимым для данного ПС техническим и программным ресурсам.

Для применения данного программного средства необходимы следующие технически требования:

* процессор Intel Core 2 Duo или выше;
* минимальный объем оперативной памяти — 512 Мбайт;
* операционная система Windows 7 и выше;
* Framework v4.0
* рекомендуется монитор типа VGA или с лучшей разрешающей способностью;
* устройство для чтения дисков;
* клавиатура;
* мышь.

6.3 Справочная система

необходимо описать:

* структуру справочной системы;
* методику использования справочной системы.

Справочная система программного средства представляет собой отдельный файл «help.chm» с полным описанием основных функций программы в формате \*.chm. В справочной системе даны ответы на типичные вопросы, возникающие при работе с приложением, что, несомненно, должно помочь при освоении программного средства.

Справка имеет следующие разделы:

* о программе;
* введение;
* учащийся;
* учителя;
* нормативы;
* мониторинг;
* меню.

Структура справочной системы представлена на рисунке 29.

Заключение

За время практики по созданию и сопровождению программного обеспечения был изучены процессы создания, внедрения и сопровождения программных приложений в различных средах программирования. Во время практики удалось накопить неоценимый опыт в разработке приложений в интегрированной среде визуального программирования Borland C++ Builder 6, развить умения самостоятельно находить оптимальный метод решения задачи. Кроме того, изучить принципы организации и технологии реализации программного обеспечения, развить умения разработки программ в соответствии с требованиями технического задания, обеспечивая высокий уровень качества программного обеспечения и экономической эффективности. Получить навыки оформления комплекта документации на созданное программное обеспечение.

При создании программных приложений входящих в обязательную программу практики были отработаны навыки владения методами надежного программирования и изучены способы определения экономической себестоимости и эффективности разработки программного обеспечения

Список используемых источников

1 Багласова Т.Г. Методические указания по выполнению дипломного проекта для учащихся по специальности 2-40 01 01 «Программное обеспечение технологий» / Т.Г.Багласова. – Минск: КБП, 2017

2 Багласова Т.Г. Методические указания по оформлению курсовых и дипломных проектов / Т.Г.Багласова, К.О.Якимович. – Минск: КБП, 2013

3 Вендров, А.М. Проектирование программного обеспечения экономических информационных систем: Учебник. / А.М. Вендров. – 2-е изд., перераб. и доп.– М. : Финансы и статистика, 2006. — 544 с.

4 Мацяшек. Анализ и проектирование информационных систем с помощью UML 2.0 / Мацяшек, А. М. Лешек: – М. – ООО “И.Д. Вильямс”, 2008. — 816 с.

5 Орлов С. А. Технологии разработки программного обеспечения: Учебник для вузов. 4-е изд. / С. А. Орлов, Б. Я. Цилькер – СПб. : Питер, 2012. — 608 с.

6 Просницкий А. Самоучитель «Управление проектами в Microsoft Project 2010» / А. Просницкий. – Київ, Казань, Красноярск, Рига, Санкт-Петербург, Львів, Чернівці, Івано-Франківськ, 2011.

7 Рудаков А. В., Технология разработки программных продуктов. Практикум: учеб. пособие для студ. учреждений сред. проф. образования / А.В. Рудаков, Г.Н. Федорова М. : Издательский центр «Академия»; 2014. — 192 с.

8 Тепляков С. Паттерны проектирования на платформе .NET. / С. Тепляков – СПб.: Питер, 2015. — 320 с.

9 ГОСТ 19701-90 Схемы алгоритмов, программ, данных. – М.: Из-во стандартов, 1990

10 ГОСТ 19301-2000 Программа и методика испытаний. – М.: Из-во стандартов, 2000

11 ГОСТ 19401-2000 Текст программы. – М.: Из-во стандартов, 2000

12 ГОСТ 19402-2000 Описание программы. – М.: Из-во стандартов, 2000

13 ГОСТ 2.105-95 Общие требования к тестовым документам. – М.: Из-во   
стандартов, 1995

14 Справочные материалы по информационным технологиям [Электронный ресурс] / Режим доступа : http://itteach.ru/bpwin/

15 Руководство по программированию [Электронный ресурс] / Режим доступа : msdn.microsoft.com/ru-ru

ДИАГРАММЫ

В рамках А4

Диаграмма вариантов использования

Диаграмма классов

Диаграмма деятельности

Диаграмма последовательности

Диаграмма компонентов