Частное учреждение образование

Колледж бизнеса и права

ОТЧЕТ

ПО ТЕХНОЛОГИЧЕСКОЙ ПРАКТИКЕ

## ОП Т.492001

Руководитель практики (О.А. Макарова)

от предприятия

М.П.

Руководитель практики

от колледжа (Л.В. Басалыга)

Учащийся (А.С. Беганская)

2018

Содержание

Изм.

Лист

№ докум.

Подпись

Дата

Лист

2

ОП Т.492001.401

Разраб.

А.С.Беганская

Провер.

Л.В. Басалыга.

Реценз.

Н. Контр.

Утверд.

Отчет

по технологической практике

Лит.

Листов

КБП

1. Структура производства, характеристики основных видов деятельности
2. Должностные обязанности оператора ЭВМ, техника-программиста, инженера-программиста
3. План мероприятий по разработке и внедрению программы
4. Техно-рабочий проект
   1. Постановка задачи
   2. Выбор языка программирования
   3. Определение структуры данных
   4. Алгоритм решения задачи
   5. Руководство программиста
   6. Инструкция по эксплуатации
5. Опытная эксплуатация.
   1. Ошибки, выявленные в процессе опытной эксплуатации

Выводы

Список используемых источников

Приложение А Текст программных модулей

Приложение Б Формы входных и выходных документов

1 Структура производства, характеристики основных видов деятельности

Место прохождения практики - закрытое акционерное общество (ЗАО) «Минский транзитный банк» («МТбанк»)

ЗАО «МТБанк» было создано 14 марта 1994 года и стало первым в Республике Беларусь банком с участием иностранного капитала.

В настоящее время обслуживание клиентов ведется в 127 отделениях банка, расположенных по всей стране.

ЗАО «МТБанк» – член Ассоциации белорусских банков, Сообщества всемирных межбанковских финансовых коммуникаций (S.W.I.F.T.), международных платежных систем VISA International и MasterCard. С начала своей деятельности Банк является универсальным и имеет на белорусском рынке репутацию надежного банка.

МТБанк активно сотрудничает с ОАО «Банк развития Республики Беларусь» и международными финансовыми институтами, как Европейский банк реконструкции и развития (ЕБРР), Международная финансовая корпорация (IFC), Нидерландская компания финансового развития (FMO), АКА Ausfuhrkredit-Gesellschaft mbH, международный финансовый институт NEFCO (Финляндия), Немецкий инвестиционный фонд IIV Mikrofinanzfonds и др.

Также с 2006 года МТБанк является первым банком-участником Глобального Договора ООН в Беларуси – общепризнанной инициативы ООН в области корпоративной социальной ответственности.

Структура правления банка представлена на рисунке 1.

Председатель правления

Заместитель Председателя правления

Директор по корпоративному бизнесу

Заместитель Председателя правления

Финансовый директор

Рисунок 1 - Структура предприятия

Примеры продукции ЗАО «МТБанк» представлены в таблице 1.1.

Таблица 1.1 - Неполный список продуктов ЗАО «МТБанк» и краткое их описание.

|  |  |
| --- | --- |
| Название  проекта | Краткое описание |
| «Халва» | Карта беспроцентной рассрочки для магазинов партнеров |
| «Халва +» | Бонусная карта. Данная карта обеспечивает возврат средств – кэшбэк, в магазинах-партнерах |
| «Икс Карта» | Бонусная карта. Совмещает в себя возможности всех вышедших продуктов. |
| «Автокарта» | Бонусная карта. Обеспечивает возврат средств на АЗС. |

2 Должностные обязанности оператора ЭВМ, техника-программиста, инженера-программиста

Для выполнения возложенных на специалиста функций он обязан:

- Учувствовать в функционировании второй линии поддержки Service Desk, а именно:

- своевременно реагировать на поступившие заявки и оперативно их выполнять;

- принимать участие в администрировании пользователей локальной вычислительной сети;

- проводить установку и сопровождение системного и прикладного программного обеспечения;

- вести журнал сбойных ситуаций, давать отчет на выполнение заявки;

- осуществлять администрирование рабочих мест, коммутацию, установку и перемещение вычислительной техники сотрудников банка;

- консультировать сотрудников банка по использованию программных средств общего назначения;

- своевременно осуществлять техническое обслуживание и возможный ремонт средств вычислительной техники;

- вести учет вычислительной техники в соответствующем программном обеспечении, по принятым в банке нормам учета;

- соблюдать требования секретности коммерческой информации, принимать меры по предотвращению ее неправомерного применения, не по назначению или утечки;

- неукоснительно применять средства антивирусной защиты информации;

- учувствовать в разрешении нештатных ситуаций в работе, систематизировать возникающие проблемы и принимать меры к их устранению и уничтожению условий их возникновения.

3 План мероприятий по разработке и внедрению программы

Разработка программного приложения происходит в несколько этапов. Обычно этих этапов пять. На каждом этапе выполняются определенные работы. В таблице 2 описаны все этапы разработки по порядку.

Таблица 2. Этапы разработки

|  |  |
| --- | --- |
| Описание этапа разработки | Срок выполнения |
| Первый этап по разработке приложения −постановка и утверждение технического задания. Целью разработки технического задания является обоснование необходимости разработки данного программного средства (ПС). На этом этапе выполняются работы по обследованию информационных потоков в подразделении, в котором осуществляется автоматизация управления, постановка задачи, сбор исходных материалов и определение структуры входных и выходных данных. | с 06.02.2018 по 07.02.2018 |
| Второй этап разработки — это разработка и утверждение эскизного проекта. На этом этапе выполняются работы по предварительной разработке структуры входных и выходных данных, уточнение методов решения задачи, разработка общего описания методов решения задачи, а также согласование и утверждение эскизного проекта. | с 08.02 по 13.02. |
| Третьим этапом является разработка и утверждение технического проекта. На этом этапе происходит уточнение структуры входных и выходных данных, разработка структуры ПС в контексте среды разработки, определение формы представления входных и выходных данных, разработка плана по разработке и внедрению программы. | с 14.02 по 13.03 |
| Четвертым этапом является разработка рабочего проекта. В рабочий проект входит программные модули и программная документация, то есть рабочий проект непосредственно сама разрабатываемая программа со всей вспомогательной и программной документацией. На этой стадии ведутся работы по программированию и отладке модулей, разработке программных документов в соответствии с требованиями единой системы программной документации (ЕСПД) и техническим заданием, а также корректировка программы и программной документации по результатам испытаний. | с 14.03 по 27.03 |
| Пятым этапом является внедрение программы на предприятие, то есть подготовка к передаче программы на предприятие. Здесь ведутся работы по подготовке и передаче программы и программной документации для сопровождения, оформление и утверждение акта о передаче программы на сопровождение, работа в службе сопровождения (по эксплуатации разработанной программы). Внесение изменений в соответствии с выявленными в процессе опытной эксплуатации ошибками и недоработками. | с 28.03 по 01.04. |

4 Техно-рабочий проекта

4.1 Исследование предметной области

Предметной областью решаемой задачи является закрытое акционерное общество (ЗАО) «Минский Транзитный Банк» (МТБанк). ЗАО «МТБанк» это уполномоченный банк Государственного таможенного комитета на право гарантировать уплату таможенных платежей, а также это уполномоченный депозитарий, выполняющий функции по учету прав на государственные ценные бумаги и ценные бумаги Национального банка Республики Беларусь, депозитарий, выполняющий функции по учету прав на корпоративные ценные бумаги, первичный инвестор на рынке государственных облигаций Республики Беларусь и облигаций Национального Банка Республики Беларусь. Каждый банк делает запросы в офис управления по контролю за иностранными активами (OFAC) на наличие клиента банка в санкционных списка данного управления. Идентификация клиента в банке — это комплекс процедур, направленный на борьбу с финансирование терроризма и отмывания преступных доходов. В комплекс этих мероприятий входят:

* проверка на наличие отрицательных кредитных историй;
* наличие связей с террористическими группировками;
* наличие судимостей по статьям мошенничества, отмывания денег.

Идентификация клиента классифицируется на полную и упрощенную. Полная идентификация представляет собой получение от клиента набора сведений до того, как банк (организация) начнет обслуживание. Полная идентификация проводится в следующих случаях:

* - по добровольному заявлению Клиента (в том числе, при желании Клиента использовать персонифицированный Кошелек и/или подключить дополнительную услугу, требующую Идентификации Клиента);
* - по факту подачи не идентифицированным Клиентом заявления.
* По требованию OFAC, в частности, в следующих случаях:
* - когда у OFAC есть основания предполагать, что Клиент нарушает условия

Соглашения, законодательство Республики Беларусь, права и законные интересы OFAC и/или иных третьих лиц;

* - когда идентификация обязательна в соответствии с требованиями законодательства о противодействии легализации (отмыванию) доходов, полученных преступным путем, и финансированию терроризма.
* Сведения несколько отличаются в зависимости от статуса клиента:
* - у физических лиц — граждан Республика Беларусь уточняют фамилию, имя, отчество, гражданство, дату рождения, данные документа, удостоверяющего личность;
* - у белорусских юридических лиц требуют представить информацию — название, правовая форма, ИНН, ОГРН, юридический адрес.

Чтобы появилась возможность провести упрощенную процедуру, должны соблюдаться такие критерии:

* - выполняемая клиентом транзакция не подлежит особому контролю;
* - клиент не вызывает подозрений у работников банка (или приравненной организации);
* - транзакция не является необычной, с сомнительным экономическим смыслом и не дает оснований предположить, что ее целью является избежание клиентом процедур полной проверки.

Для проведения упрощенной идентификации клиент обязан заполнить анкету, содержащую:

* - фамилию, имя (если иное не вытекает из закона или национального обычая) и отчество, должность лица, заполнившего анкету (досье) клиента в электронном виде;
* - подпись лица, заполнившего анкету (досье) клиента на бумажном носителе;
* - иные сведения по усмотрению банка.

Исходя из анализа предметной области можно выделить следующие задачи, подлежащие автоматизации:

* – организовать ведение базы данных, содержащей информацию о клиентах, о кредитах, о банках, о судимостях;
* - автоматизировать комплекс следующих процедуры, необходимых для идентификации клиента: автоматическая проверка на наличие отрицательных кредитных историй (наличие задолженностей по выплатам, наличие непогашенных кредитов), проверка на наличие связей с террористическими группировками, наличие судимостей по определенным статьям;
* - обеспечить возможность просмотра и поиска необходимой пользователю информации;
* - предоставить возможность сортировки данных по различным критериям;
* - создать отчёты по интересующей пользователя информации, в том числе и статистической с возможностью выбора аналитики и оцениваемых параметров; организовать возможность экспорта в файлы Microsoft Office интересующей пользователя информации (отчет по результатам идентификации). Создать справочную систему приложения;
* - обеспечить организацию интерфейса приложения средствами создания меню, кнопочных форм, панелей инструментов.

4.2 Структура входных и выходных данных

Входными данными являются данные, добавляемые в базу данных, вносимые при редактировании, а также данные, используемые при выборке информации по различным критериям.

Входными данными для добавления нового ингредиента в таблицу «Банк» будут являться:

- название;

Входными данными для добавления нового ингредиента в таблицу «Кредиты» будут являться:

- дата взятия;

- сумма кредита;

- название банка;

- комментарий.

Входными данными для добавления нового ингредиента в таблицу «Клиент\_Кредит» будут являться:

- номер кредита;

- номер клиента.

Входными данными для добавления нового ингредиента в таблицу «Клиент» будут являться:

- фамилия;

- имя;

- отчество;

- дата рождения;

- идентификационный номер;

- серия паспорта;

- номер паспорта.

- срок действия;

- орган выдачи.

Входными данными для добавления нового ингредиента в таблицу «Судимость» будут являться:

- номер статьи;

- номер клиента;

- Дата судимости.

Входными данными для добавления нового ингредиента в таблицу «Статьи» будут являться:

- номер статьи;

- описание статьи.

Все эти данные могут быть выбраны пользователем из подключённого к программе файла базы данных, имеющего по умолчанию расширение «\*.mdf».

4.3 Проектирование модели

Для того, чтобы разработчик имел возможность спроектировать информационную систему и любой другой программный продукт, необходимо осуществить моделирование данных.

Исходя из исследования предметной области, можно выделить следующие сущности разработки: «Банк», «Кредиты», «Клиент\_Кредит», «Клиент», «Судимость», «Статья».

Для сущности «Банк» атрибутами будут являться:

* 1. наименование.

Для сущности «Кредиты» атрибутами будут являться:

* 1. ID\_Кредита;
  2. комментарий.

Для сущности «Клиент\_кредит» атрибутами будут являться:

* 1. ID\_Кредита;
  2. ID\_Клиента;
  3. Дата\_Взятия
  4. Сумма\_кредита

Для сущности «Клиент» атрибутами будут являться:

* 1. фамилия;
  2. имя;
  3. отчество;
  4. дата рождения;
  5. идентификационный номер;
  6. серия паспорта;
  7. номер паспорта;
  8. срок действия;
  9. орган выдачи.

Для сущности «Судимость» атрибутами будут являться:

* 1. номер статьи;
  2. дата судимости.

Для сущности «Статья» атрибутами будут являться:

* 1. содержание статьи;
  2. номер статьи.

Суть диаграммы вариантов использования состоит в том, что проектируемая система представляется в виде множества сущностей или актёров, 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.

Актёр представляет собой внешнюю по отношению к моделируемой системе сущность, которая взаимодействует с системой и использует её функциональные возможности для достижения определённых целей или решения частных задач. При этом актёры служат для обозначения согласованного множества ролей, которые могут играть пользователи в процессе взаимодействия с проектируемой системой. Каждый актёр может рассматриваться как некоторая отдельная роль относительно конкретного варианта использования.

Данный программный продукт имеет следующие основные (Include) функции:

1. проверка на наличие кредитных историй;
2. проверка на наличие судимостей.

К вспомогательным (Extend) функциям, расширяющим возможности системы, относятся следующие функции:

1. предоставление справочной информации;
2. предоставление информации о кредитах;
3. предоставление информации о банках;
4. предоставление информации о клиентах;
5. предоставление информации о судимостях интересуемого клиента;
6. предоставление информации о кредитной истории интересуемого клиента;
7. предоставление информации о статьях.

Диаграмма вариантов использования представлена в графической части на листе 1.

Диаграмма классов служит для представления статической структуры модели системы в терминологии классов объектно-ориентированного программирования. Диаграмма классов может отражать, в частности, различные взаимосвязи между отдельными сущностями предметной области, такими как объекты и подсистемы, а также описывает 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.

В данном курсовом проекте будут реализованы классы и их методы, представлены в таблице 1.

Таблица 1 - Классы

|  |  |  |
| --- | --- | --- |
| Класс | Поля | Методы |
| Клиент\_кредит,Клиент\_View,Кредит\_View, Bank, Stat | Button,Components, MenuStrip, Label, ListBox1, DatGridView, BindingSource, TableAdapterManadger, ComboBox, GroupBox, | Click,Item\_click,InitializeComponent, IndexChanged |

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.

Основная цель использования диаграмм деятельности - визуализация особенностей реализации операций классов, когда необходимо представить алгоритмы их выполнения.

Диаграмма деятельности для функции представлена в графической части на листе 3.

Для моделирования взаимодействия объектов в UML используются соответствующие диаграммы взаимодействия. Если рассматривать взаимодействия объектов во времени, тогда для представления временных особенностей передачи и приема сообщений между объектами используется диаграмма последовательности.

Временной аспект поведения имеет существенное значение при моделировании синхронных процессов, описывающих взаимодействия объектов. Именно для этой цели и используются диаграммы последовательности, в которых ключевым моментом является динамика взаимодействия объектов во времени. При этом диаграмма последовательности имеет как бы два измерения: одно - слева направо в виде вертикальных линий, каждая из которых изображает линию жизни отдельного объекта, участвующего во взаимодействии; второе - вертикальная временная ось, направленная сверху вниз, на которой начальному моменту времени соответствует самая верхняя часть диаграммы.

Диаграмма последовательности для проектируемой системы представлена в графической части на листе 4.

Диаграмма компонентов описывает объекты реального мира это компоненты программного обеспечения. Эта диаграмма позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами.

Вид диаграммы компонентов для данной проектируемой системы представлен в графической части на листе 5 и содержит следующие компоненты:

1. программные компоненты созданные в среде Visual studio 2017;
2. файл отчета экспортируемый в Microsoft Excel;
3. программные компоненты созданные в среде MS SQL server 2017;
4. файл справочной системы «help.сhm».

4.4 Концептуальный прототип

Концептуальный прототип состоит из описания внешнего пользовательского интерфейса, а именно, элементов управления.

При создании данного приложения важную роль играют формы, так как они являются основным диалоговым средством работы пользователя. Разрабатываемое приложение будет содержать одну основную форму. Такая структура интерфейса позволит классифицировать основные функции программы по группам.

В программе предполагается создание меню, состоящее из пункта «Файл», который содержит подпункты «Добавление кредита», «Добавление судимости», «Добавление статьи», «Добавление банка», «Клиент\_кредит»

При проектировании концептуального прототипа предполагается, что при загрузке программы первой будет загружаться форма «МТБанк». На ней будет отображаться соответствующая кнопочные формы для перехода на следующие формы управлениями остальными таблицами. Будут отдельные формы для табличного вида и ленточного, которые также можно редактировать.

4.5 Реализация функций

Рассмотрим основные функции программы.

При нажатии на кнопку «Сортировать» будет вызван метод «Button8\_Click», позволяющий отфильтровать таблицу по выбранному критерию из ListBox. Программный код реализации данного метода представлен ниже.

Private Sub Button8\_Click(sender As Object, e As EventArgs) Handles Button8.Click

Dim Col As System.Windows.Forms.DataGridViewColumn

Select Case ListBox1.SelectedIndex

Case 0

Col = DataGridViewTextBoxColumn2

Case 1

Col = DataGridViewTextBoxColumn3

Case 2

Col = DataGridViewTextBoxColumn4

Case 3

Col = DataGridViewTextBoxColumn5

Case 4

Col = DataGridViewTextBoxColumn6

Case 5

Col = DataGridViewTextBoxColumn7

Case 6

Col = DataGridViewTextBoxColumn8

Case 7

Col = DataGridViewTextBoxColumn9

Case 8

Col = DataGridViewTextBoxColumn10

End Select

If RadioButton1.Checked Then

КлиентDataGridView.Sort(Col, ComponentModel.ListSortDirection.Ascending)

Else

КлиентDataGridView.Sort(Col, ComponentModel.ListSortDirection.Descending)

End If

End Sub

При нажатии «Поиск» будет вызван метод «Button8\_Click» с помощью которого можно найти все столбцы в которых есть введенный в TextBox критерий.

Программный код реализации данного метода представлен ниже.

Private Sub Button6\_Click(sender As Object, e As EventArgs) Handles Button6.Click

For i = 0 To КлиентDataGridView.ColumnCount - 1

For j = 0 To КлиентDataGridView.RowCount - 1

КлиентDataGridView.Item(i, j).Style.BackColor = Color.White

КлиентDataGridView.Item(i, j).Style.ForeColor = Color.Black

Next j

Next i

For i = 0 To КлиентDataGridView.ColumnCount - 1

For j = 0 To КлиентDataGridView.RowCount - 1

If InStr(КлиентDataGridView.Item(i, j).Value, TextBox1.Text) Then

КлиентDataGridView.Item(i, j).Style.BackColor = Color.AliceBlue

КлиентDataGridView.Item(i, j).Style.ForeColor = Color.Blue

End If

Next j

Next i

End Sub

При нажатии на кнопку «Фильтровать» будет вызван метод «Button6\_Click», позволяющий отсортировать по убыванию или по возрастанию выбранный из ComboBox столбец. Программный код реализации данного метода представлен ниже.

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

КлиентBindingSource.Filter = "Фамилия= '" & ComboBox1.Text & "'"

End Sub

Private Sub Button5\_Click(sender As Object, e As EventArgs) Handles Button5.Click

Клиент

End Sub

При нажатии на кнопку «Добавить» будет вызван метод «Button1\_Click», после чего будет добавлена новая запись в таблице. Программный код реализации данного метода представлен ниже.

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

КлиентBindingSource.AddNew()

End Sub

При нажатии на кнопку «Удалить» будет вызван метод «Button2\_Click», после чего будет удалена выбранная запись в таблице. Программный код реализации данного метода представлен ниже.

Private Sub Button2\_Click(sender As Object, e As EventArgs) Handles Button2.Click

КлиентBindingSource.RemoveCurrent()

End Sub

End SubПри нажатии на кнопку «Сохранить» будет вызван метод «Button3\_Click», после чего будут сохранены изменения в таблице. Программный код реализации данного метода представлен ниже.

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

Me.Validate()

Me.КлиентBindingSource.EndEdit()

Me.TableAdapterManager.UpdateAll(Me.IdentificationDataSet)

End Sub

Весь текст программы представлен в приложении А.

4.6 Функциональное тестирование

При функциональном тестировании осуществляется проверка каждого пункта меню, каждой операции, с целью проверки выполнения всех функций, определенных на этапе объектно-ориентированного анализа и проектирования. Функциональное тестирование должно гарантировать работу всех элементов управления в автономном режиме.

В качестве тестового примера использовались данные, структура которых соответствует структуре типизированного файла, и которые являются входной информацией в работе пользователя с данным программным продуктом. Тестовые данные представлены в таблице 5.1, а проверка выполнения функций – в таблице 5.2.

Таблица 5.1 – Тестовые данные

|  |  |  |  |
| --- | --- | --- | --- |
| Запись в типизированном файле БД | Поле записи | Название поля | Содержимое |
| Клиент | Фамилия | Фамилия | Банцевич |
| Имя | Имя | Светлана |
| Отчество | Отчество | Валерьевна |
| Дата рождения | День\_Рождения | 12.12.1995 |
| Идентификационный номер | Идентификационный номер | 1234567891112 |
| Серия паспорта | Серия\_паспорта | MP |
| Номер паспорта | Номер\_Паспорта | 1234567 |
| Срок действия | Срок\_действия | 12.12.2025 |
| Орган выдачи | Орган\_выдачи | Московское РУВД |

Таблица5.2 – Проверка выполнения функций программы

|  |  |
| --- | --- |
| Проверяемая функция | Результат выполнения |
| Добавление записи о клиенте | Приложение Б, рисунок Б.1 |

Продолжение таблицы 5.2

|  |  |
| --- | --- |
| Проверяемая функция | Результат выполнения |
| Удаление записи о клиенте | Приложение Б, рисунок Б.2 |
| Сортировка списка клиентов по названию в алфавитном порядке | Приложение Б, рисунок Б.3 |
| Сортировка списка клиентов по дате рождения в порядке убывания | Приложение Б, рисунок Б.4 |

5 Опытная эксплуатация

5.1 Ошибки, выявленные в процессе опытной эксплуатации

Ошибки и проблемы, которые возникли на стадии разработки средства, приведены в таблице 5.1.

Таблица 5.1 - Ошибки и их описание

|  |  |
| --- | --- |
| Классификация ошибок | Содержание |
| Ошибки работы с базой данных | В программе возникли ошибки неверно построенных SQL запросов. |
| Неверно работающая функциональность. | Функция «get user recommendations» работает не так, как предусмотрено спецификацией. |
| Ошибки конфигурации сервера | Неверно настроенное программное окружение. |
| Пропущенная информация. | Ошибка интерфейса: отсутствие признаков активности выполнения программы. |
| Ошибки отображения. | Данные отражаются формально правильно, но в каком-то неестественном виде, результат запросов выводится в форме, не понятной для пользователя. |

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, 2015
3. Гради Буч. Объектно - ориентированный анализ и проектирование / Гради Буч. – 2-е изд. – М.: Бином, 1998
4. Стивен Хольцнер. PHP в примерах / Стивен Хольцнер – М.: Бином, 2007
5. ГОСТ 2.105-95 ЕСКД. Общие требования к текстовым документам. – М.: Из-во стандартов, 1995
6. Web-технологии [Электронный ресурс]. – Web-технологии: HTML, DHTML, JavaScript, PHP, MySQL, XML+XLST, Ajax. – сор. 2008-2013 – Режим доступа: <http://htmlweb.ru/>

Приложение А

(обязательное)

Текст программных модулей

Приложение Б

(справочное)

Формы входных и выходных документов