|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Приднестровский государственный университет им. Т.Г. Шевченко | | | | | | | | | | | | | | | | |
| Инженерно-технический институт | | | | | | | | | | | | | | | | |
| Инженерно-технический факультет | | | | | | | | | | | | | | | | |
| Кафедра программного обеспечения вычислительной техники  и автоматизированных систем | | | | | | | | | | | | | | | | |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  |  |  | |  | | | | | | | |
|  |  |  |  | |  | | | | | | | |
|  |  |  |  | |  | | | | | | | |
|  |  |  |  | |  | |  | | |
|  |  | | |  | |  | | | | | | | |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  | | | | | | | | | | | | | | | | |
| **ОТЧЁТ** | | | | | | | | | | | | | | | | |
|  | | | | | | | | | | | | | | | | |
| **по преддипломной практике** | | | | | | | | | | | | | | | | |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  | Студент  группы ИТ15ДР62ПИ1 | | | | |
|  |  | | | | |  | |  |  | |  | Кодос | | | | |
|  |  | | | | | | |  |  | |  | Николай Николаевич | | | | |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | | | | | |  |  | |  | Руководитель, | | | | |
|  |  | | | | | | |  |  | |  | к. п. н., доцент | | | | |
|  |  | | |  | |  | |  |  | |  | Кирсанова | | | | |
|  |  | | |  | |  | |  |  | |  | Алла Владимировна | | | | |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |
|  |  | | |  | |  | |  |  | |  |  | | |  |  |

Тирасполь, 2019

**СОДЕРЖАНИЕ**

|  |  |  |
| --- | --- | --- |
| ВВЕДЕНИЕ | | 3 |
|  | Описание реализации | 4 |
|  | Описание пользовательского интерфейса | 7 |
|  | Тестирование и оценка надежности программного продукта | 13 |
|  | Расчет себестоимости от внедрения результатов | 14 |
|  | Охрана труда | 20 |
| ЗАКЛЮЧЕНИЕ | | 27 |
| СПИСОК ИСТОЧНИКОВ | | 28 |

**введение**

На сегодняшний день совершенно ясно, что современный мир становится все более зависимым от информационных технологий. Абсолютно не вызывает удивления повсеместное использование компьютерных технологий: в офисах крупных и мелких компаний, в высших и средних учебных заведениях, дома, даже на улице. Рождение компьютерных сетей было вызвано практической потребностью иметь возможность для совместного использования данных.

В современное время задача внедрения науки, образования и инновационной деятельности имеет важное место. Это является одним из решающих факторов развития экономики и общества. Потребность в высококвалифицированных и инициативных работниках возрастает в новых условиях, ведет к естественной интеграции вуза и основных работодателей, потребителей их услуг.

Одним из важных инструментом взаимодействия ВУЗа и предприятий является анкетирование. Процесс проведения анкетирования условно можно разделить на четыре этапа: подготовка анкет, проведение анкетирования, обработка собранных данных и формирование выводов о результатах анкетирования. Результаты анкетирования обрабатываются различными методами, результатами которых помогают создать условия для оптимизации педагогического процесса и принять эффективные управленческие решения для повышения качества предоставляемых образовательных услуг.

Так как известно, что применение современных информационных технологий позволит значительно повысить эффективность процесса анкетирования, было принято решение разработать программный продукт с использованием сетевого взаимодействия для упрощения процесса анкетирования предприятий высшим учебным заведением.

Описание реализации

Основной задачей клиент-серверного приложения, является обработка результатов выбранных анкет.

Для того, чтобы реализовать данный алгоритм, необходимо реализовать алгоритмы запрашивающие необходимую информацию у сервера. Такие алгоритмы реализовывались по аналогии с алгоритмом (листинг 1), запрашивающим информацию о предприятии.

q1.Type = "SELECT";

q1.Table = "Businesses";

q1.Query = "SELECT \* FROM questioning.businesses";

answer = QueryToDB.SendQuery(q1);

ListBusinesses LB = (ListBusinesses)QueryToDB.ProcessResponse(answer, q1);

answer = QueryToDB.SendQuery(q1);

ListBusinesses LB = (ListBusinesses)QueryToDB.ProcessResponse(answer, q1);

Листинг 1 – Фрагмент кода, реализующий запрос серверу

В этом алгоритме сначала задаются параметры запроса, после чего вызывается метод, отправляющий запрос и сразу ждущий ответ, который записывается в переменную *answer*. После чего ответ преобразуется в конкретный объект нужного класса, для дальнейшей работы с ним.

Обработка результатов анкетирования реализована листингом 2.

ListAnswersBlock LAB = (ListAnswersBlock)QueryToDB.ProcessResponse(answer, q1);

Table2 t2 = new Table2();

t2.ConversationInputData(LB, LAB);

t2.CalculateAll();

List<RowForResultTable> listRowsForRT = new List<RowForResultTable>();

foreach(RowForTable2 row in t2.rows)

{

RowForResultTable rowForRT = new RowForResultTable(row.nameQB, row.masData[0], row.masData[3], row.masData[4], row.masData[6]);

listRowsForRT.Add(rowForRT);

}

WindForResult window = new WindForResult();

window.listRows = listRowsForRT;

window.Owner = this;

window.Show();

Листинг 2 – Фрагмент кода, реализующий обработку результатов

В этом примере кода сначала принимаются необходимые результаты выбранных анкет. После чего вызываются методы классов обработки данных, которые сначала преобразуют данные в необходимый вид. Потом эти данные обрабатываются, и результаты обработки передаются другой форме, которая в итоге открывается и выводит их в виде таблиц и гистограмм.

Для принятия запроса сервером от клиента использовался алгоритм, приведенный в листинге 3.

async private Task BackgroundThreadMethodAsync()

{

await Task.Run(() => BackgroundThreadMethod());

}

private void BackgroundThreadMethod()

{

try

{

\_tcpListner = new TcpListener(new IPEndPoint(IPAddress.Parse("127.0.0.1"), 12000));

\_tcpListner.Start();

while (true)

{

\_tcpClient = \_tcpListner.AcceptTcpClient();

StreamReader sr = new StreamReader(\_tcpClient.GetStream());

jsonData = sr.ReadLine();

Invoke(new Action(() =>

{

txtLog.Text += DateTime.Now.ToString("hh:mm:ss") + " Получен строка в json: " + jsonData + Environment.NewLine;

txtLog.SelectionStart = txtLog.Text.Length;

txtLog.ScrollToCaret();

}));

q1 = JsonConvert.DeserializeObject<QueryToDB>(jsonData);

string answer = q1.RunQuery();

StreamWriter sw = new StreamWriter(\_tcpClient.GetStream());

sw.AutoFlush = true;

sw.WriteLine(answer);

Invoke(new Action(() =>

{

txtLog.Text += DateTime.Now.ToString("hh:mm:ss") + " Запрос клиента выполнен и был передан ответ" + Environment.NewLine;

txtLog.SelectionStart = txtLog.Text.Length;

txtLog.ScrollToCaret();

}));

\_tcpClient.Close();

}

}

catch (SocketException ex) when (ex.ErrorCode == 10004)

{

return;

}

catch (Exception ex)

{

Invoke(new Action(() =>

{

txtLog.Text += DateTime.Now.ToString("hh:mm:ss") + " Ошибка: " + ex.Message + Environment.NewLine;

txtLog.SelectionStart = txtLog.Text.Length;

txtLog.ScrollToCaret();

this.StopBackThread();

StartBackThread();

}));

}

}

Листинг 3 – Фрагмент кода, реализующий принятие запросов

В данном алгоритме видно, что сервер всегда в режиме прослушивания, благодаря бесконечному цикле. При получении запроса, сервер пытается перевести запрос из *JSON* строки в экземпляр класса *QueryToDB*. При успешной конвертации, сервер запускает метод *RunQuery*, который обрабатывает запрос и получает ответ в виде *JSON* строки. После чего отправляет сформулированный ответ обратно клиенту. Как видно, в листинге 3.3, метод *BackgroundThreadMethod* был обернут в асинхронный метод *BackgroundThreadMethodAsynс*, для того, чтобы обработка запросов от клиента выполнялась асинхронно. Для реализации асинхронности был выбран встроенный класс *Task*, так как он более высокого уровня по сравнению с классом *Thread*, что дает нам более простую реализацию. В данном методе реализовано логирование, которое выводится в текстовое поле пользовательской формы.

**Описание пользовательского интерфейса**

После запуска пользователем приложения, перед ним появляется окно авторизации (рисунок 1).

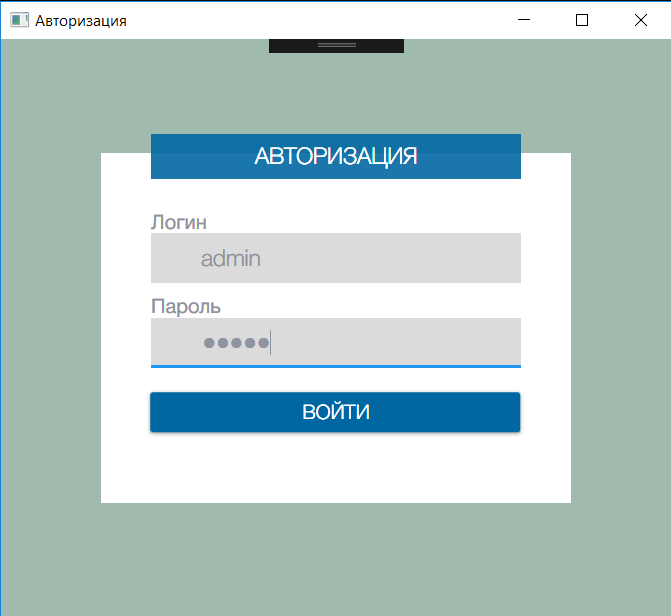


Рисунок 1 – Форма авторизации

Данная форма необходима для проверки прав доступа. После ввода правильного логина и пароля, пользователь попадает на главную форму программы (рисунок 2).

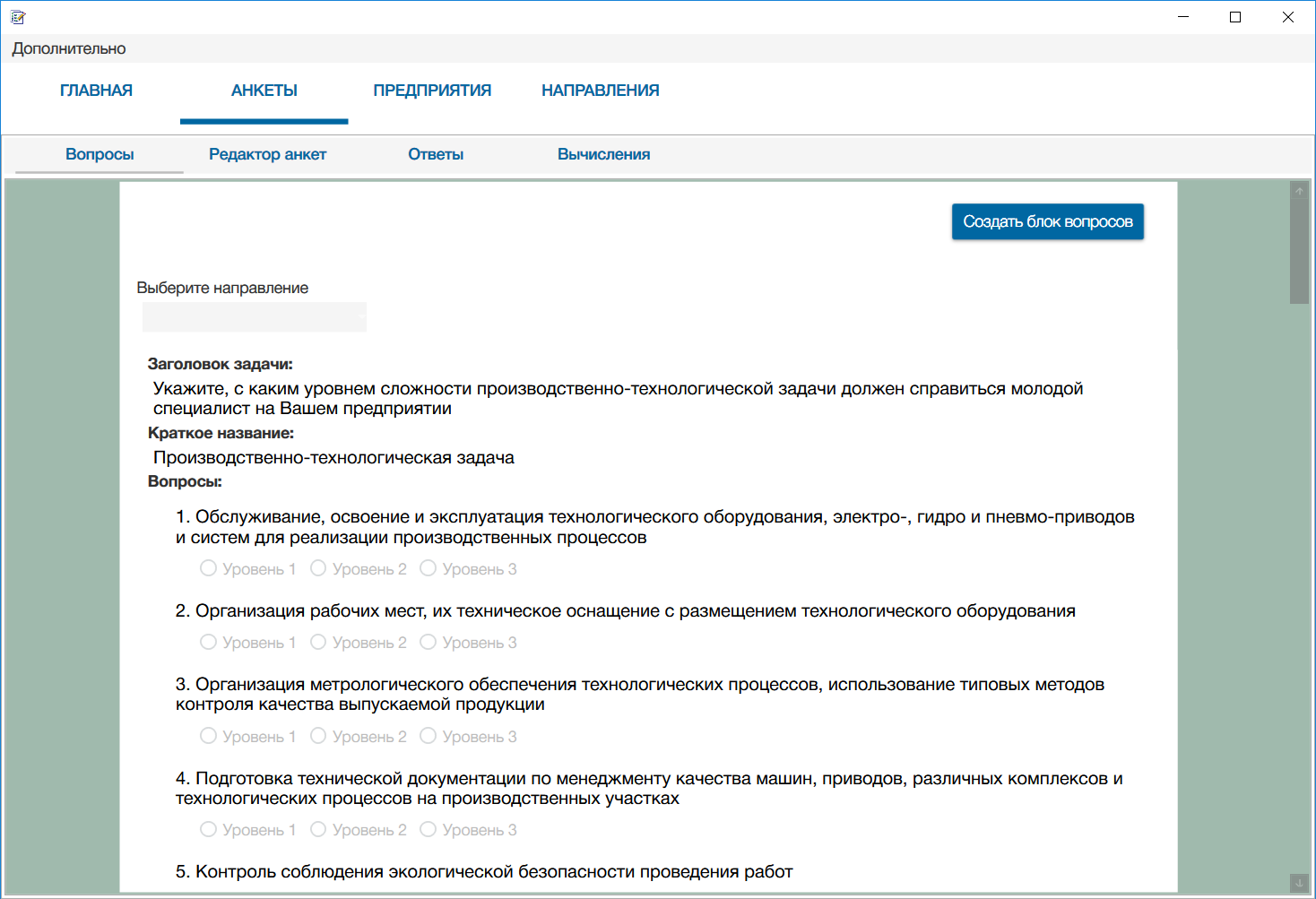


Рисунок 2 – Главное окно программы

При переключении между вкладками самого высокого уровня пользователь попадает в разделы таких сущностей, как анкеты, предприятия и направления. Переключив вкладку нижнего уровня, пользователь может реализовать над сущностями такие действия как: Просмотр, поиск, добавление, редактирование и удаление. На рисунке 3 предоставлен раздел редактирования предприятия.

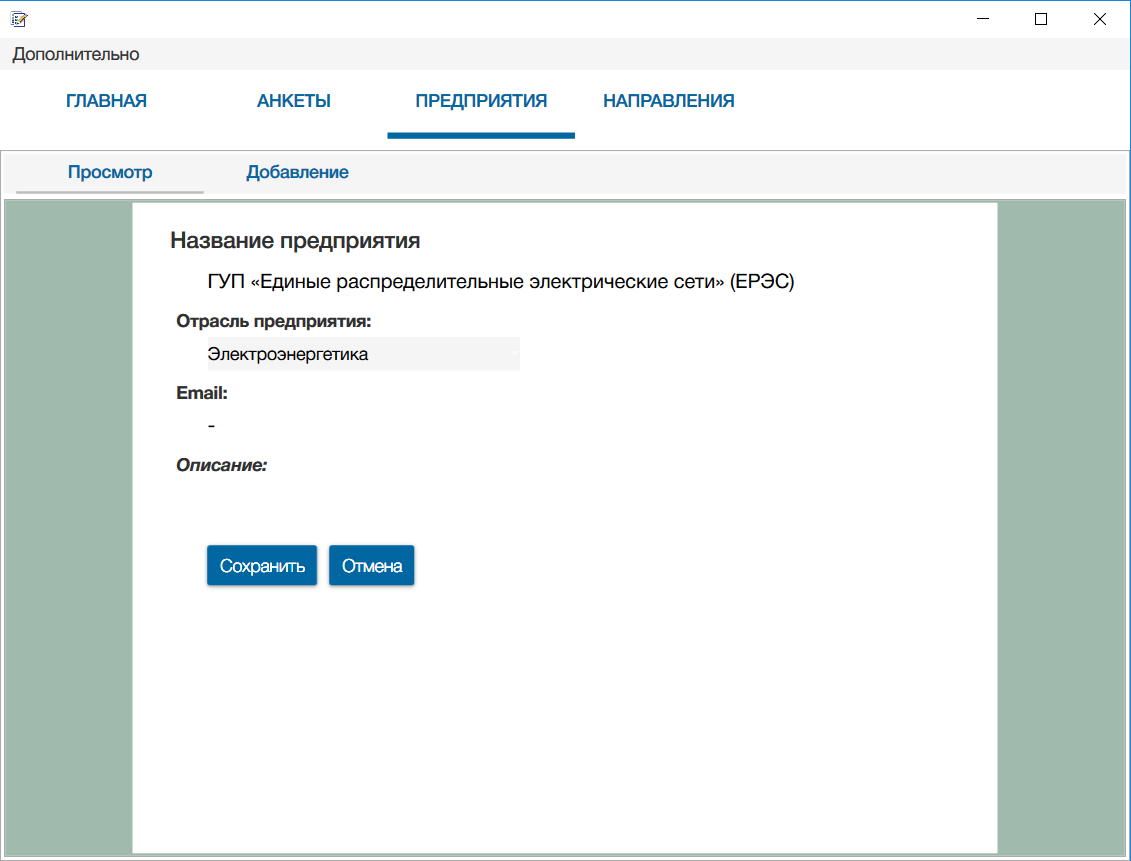


Рисунок 3 – Вкладка редактирования предприятия

На рисунке 4 изображено окно обработки результатов, на которое можно перейти, нажав на вкладку Анкеты и Вычисления.

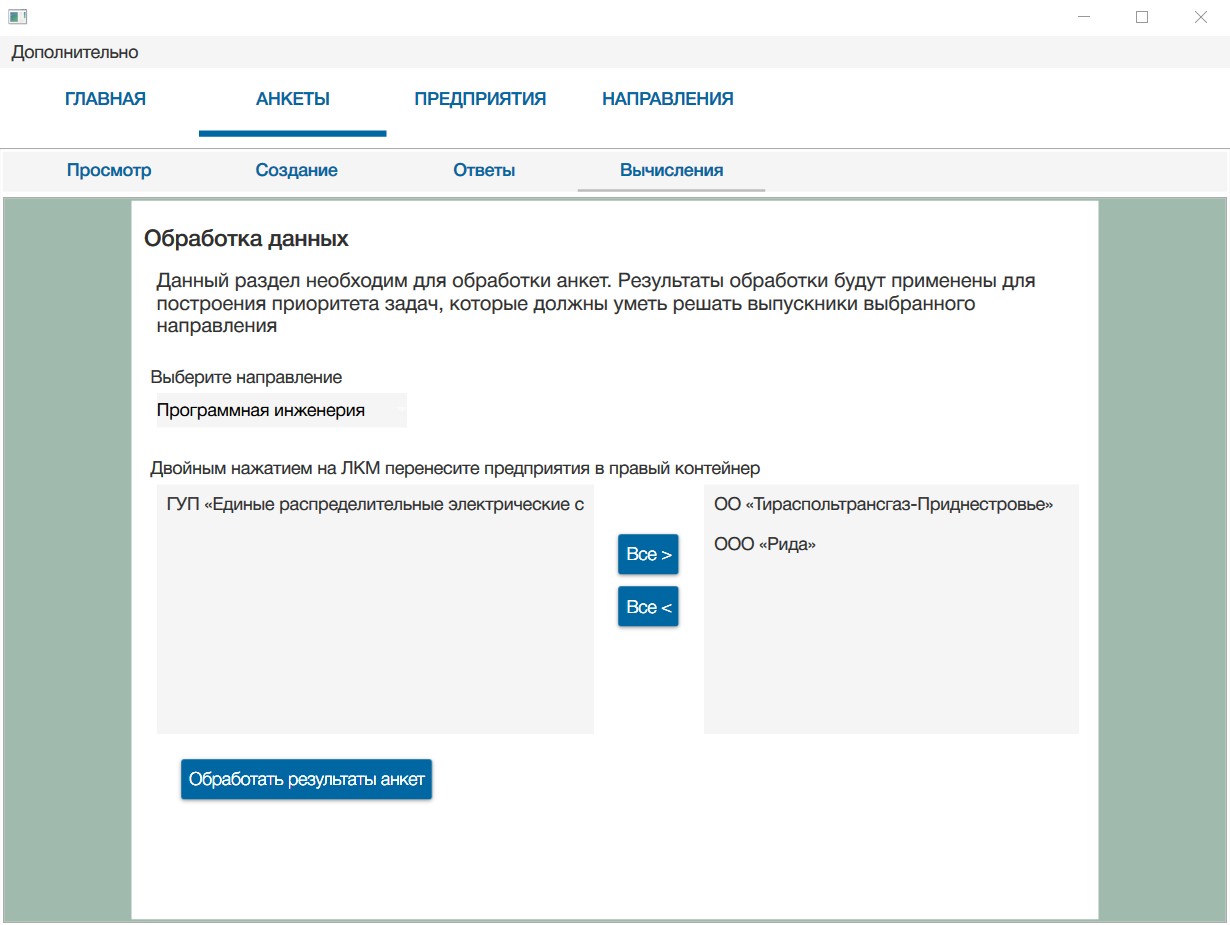


Рисунок 4 – Вкладка обработки результатов

Выбрав направление и, как минимум два предприятия, можно обработать результаты анкеты выбранного направления. Для этого нужно нажать на кнопку «Обработать результаты анкет». После нажатия появится окно обработки данных (Рисунок 5), на которой результаты показаны в таблице.

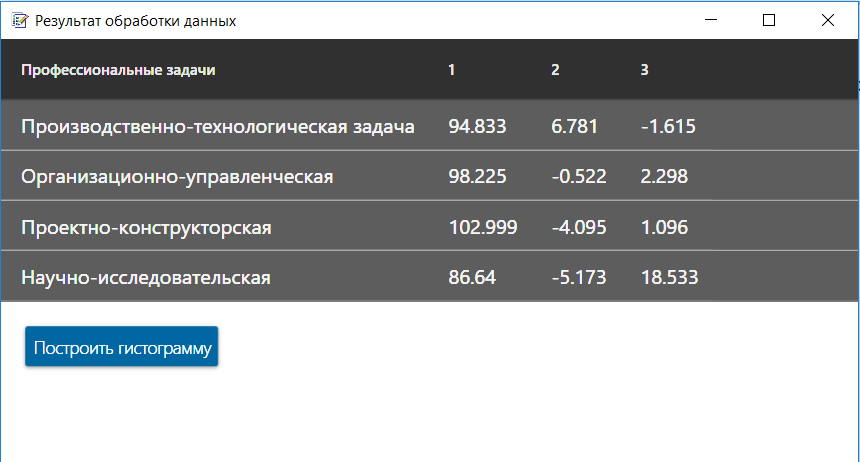


Рисунок 5 – Окно с результатами обработки данных

После нажатия на кнопку «Построить гистограмму», откроется новая форма с прорисованной гистограммой, как показано на рисунке 6.

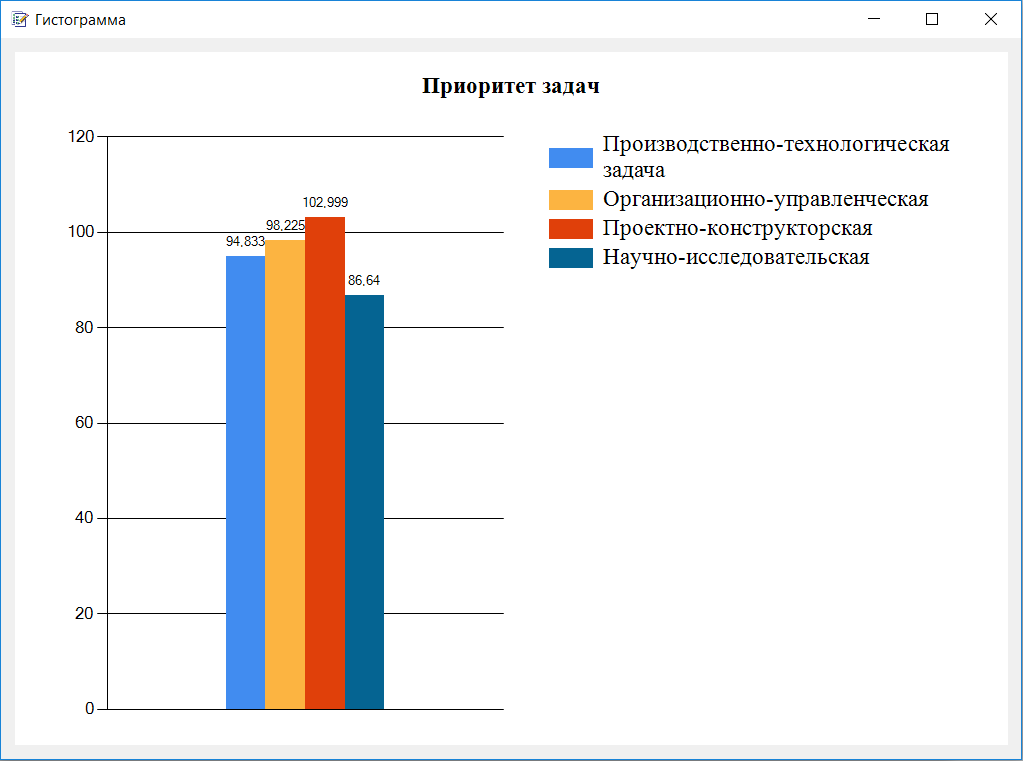


Рисунок 6 – Форма для отображения гистограммы

Серверное приложения состоит из одной главной формы (рисунок 7), на которой происходит запуск и остановка приложения, а также просмотр логирования.

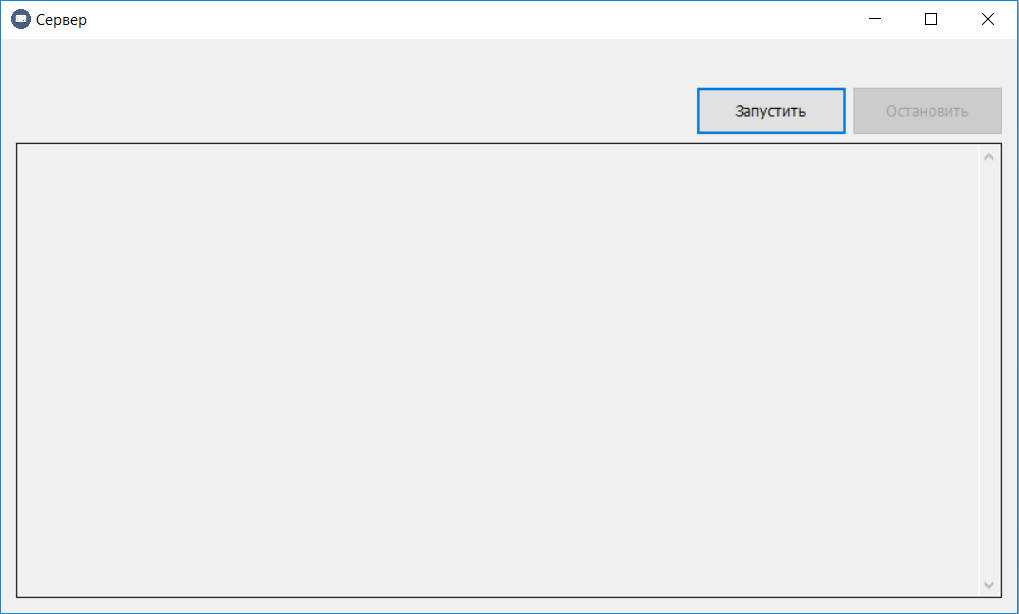


Рисунок 7 – Главная форма серверного приложения

Данная форма состоит всего из трех элементов: кнопки «Запустить» и «Остановить» и одно текстовое поле, занимающее большую часть формы.

Анкеты, созданные в *windows*-приложении, можно загрузить, пройдя в *web*-приложение (рисунок 8).

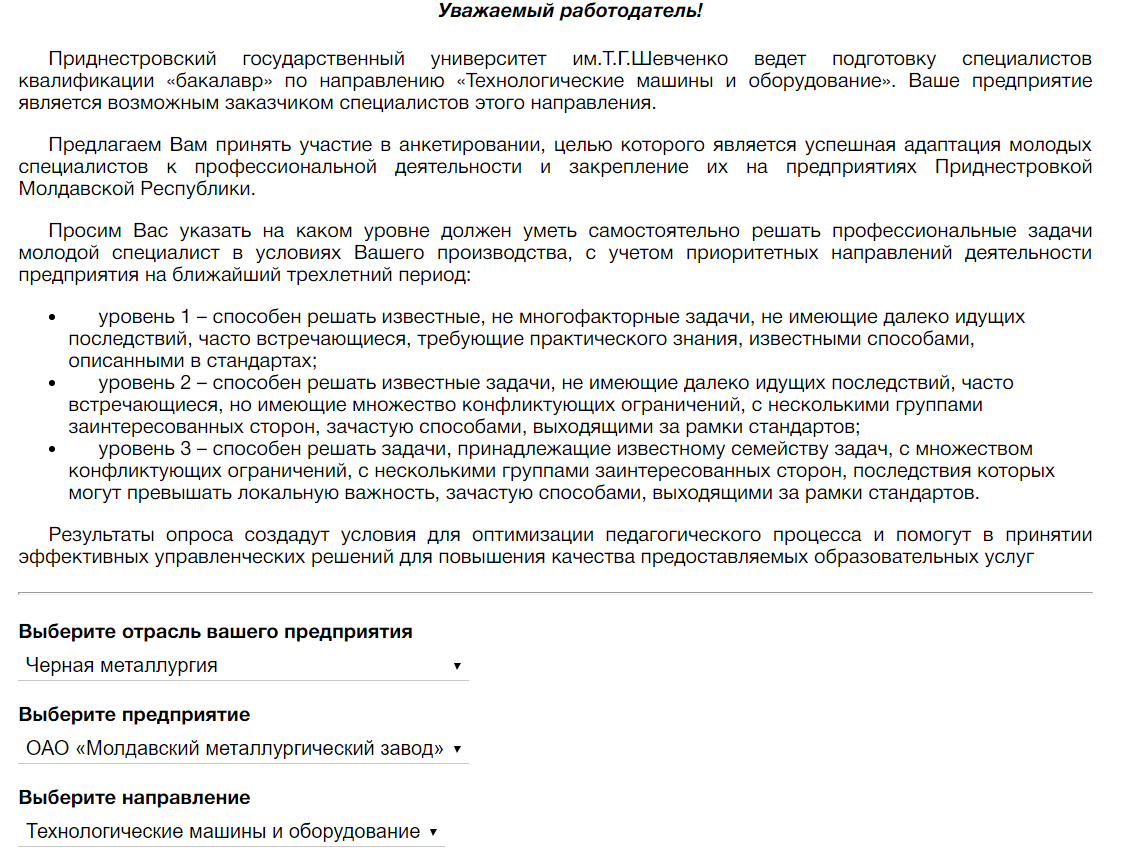


Рисунок 8 – Интерфейс *web*-приложения

На форме заполнения анкет сначала выведена краткая информация по данной анкете. Далее пользователю предлагается выбрать отрасль предприятия, предприятие и направление, по которому будет проводиться анкетирование. На рисунке 9 изображен фрагмент анкеты с *web*-приложения.

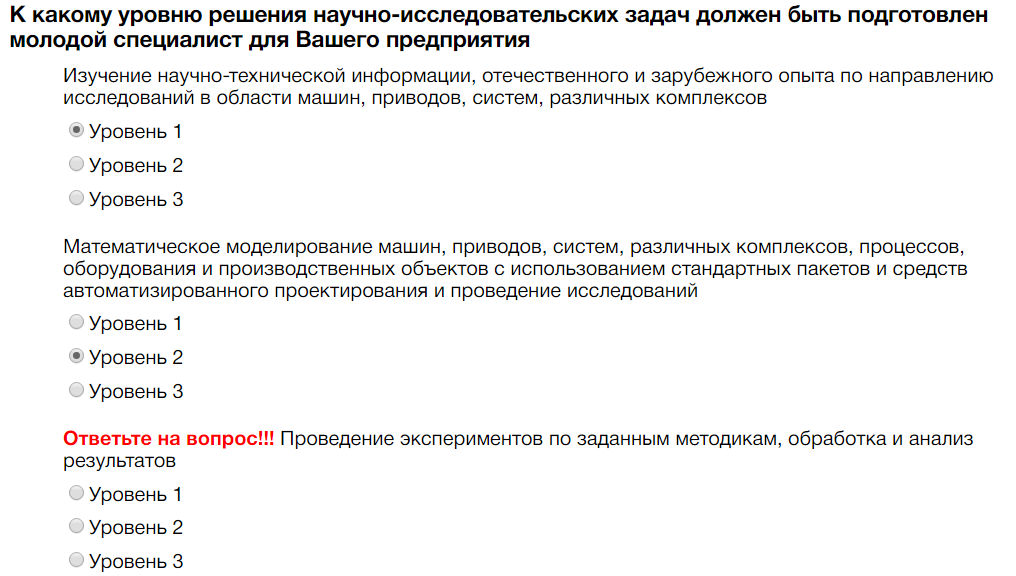


Рисунок 9 – Интерфейс *web*-приложения

Варианты ответов выполнены с помощью радиокнопок. После нажатие на кнопку «Отправить», результаты анкетирования не будут отправлены на сервер, пока пользователь не ответит на все вопросы. Вопросы будут помечены таким, образом, как последний вопрос на рисунке 9.

Тестирование и оценка надежности программного продукта

Первые этапы тестирования проходили непосредственно в процессе разработки программного продукта. Каждый разработанный объект программного продукта (направления, предприятия, анкеты, математические расчеты) тестировался отдельно сразу же после его разработки. В ходе тестирования были выявлены такие ошибки, как:

* некорректная передача данных серверу;
* повторение записей в базе данных;
* неполное изменение данных;
* сохранение связанных данных, после удаления основного объекта;
* и др.

При ликвидации ошибок в одном объекте в дальнейшей разработке остальных объектов, ошибок становилось все меньше.

После того, как программный продукт был полностью реализован, тестирование осуществлялось потенциальными пользователями. В ходе данного этапа тестирования был обнаружен серьезный недочет в проектировании программного продукта. Данный недочет заключался в том, что все созданные блоки вопросов были в открытом доступе для редактирования, удаления и дополнения. Этот недочет мог бы привести к тому, что в середине процесса анкетирования, какой-то из пользователей *windows*-приложения мог бы добавить, изменить или, что еще хуже, удалить вопросы и в дальнейшем это повлияло бы на результаты обработки данных.

Было принято решение изменить алгоритм создания анкет, разделив объект «Анкеты» на «Редактор анкет» и «Редактор вопросов». Таким образом, в «Редакторе вопросов» проводят все возможные операции над блоками вопросов. А в «Редакторе анкет» собираются нужные блоки вопросов в один объект «Анкета», в которой вопросы становятся независимыми от изменений.

Расчет себестоимости от внедрения результатов

Разработка программного продукта для автоматизации процесса анкетирования потребовала разработку программно-документальных ресурсов, требующих подготовку высокого уровня, и представляют собой объекты интеллектуальной собственности.

Разработанный программный продукт является объектом защиты также и в связи со сложностью и трудоемкостью восстановления его работоспособности, значимостью для работы информационной системы.

В структуре капитальных вложений, связанных с автоматизацией управления, выделяют капитальные вложения на разработку проекта автоматизации (предпроизводственные затраты) и капитальные вложения на реализацию проекта (затраты на внедрение):

*К = Кп + Кр,*  (3.1)

где *Кп* – капитальные вложения на проектирование;

*Кр* – капитальные вложения на реализацию проекта.

*Расчет капитальных вложений на проектирование*. Капитальные вложения на проектирование ПС определяются путем составления смет расходов и определяются по формуле:

*Кп=Км+ Кпр + Кмаш + Кс + Кн*, (3.2)

где *Км –* стоимость материалов;

*Кпр –* заработная плата основная и дополнительная с отчислениями в соцстрах инженерно-технического персонала, непосредственно занятого разработкой проекта;

*Кмаш –* затраты, связанные с использованием машинного времени на отладку программы;

*Кс –* оплата услуг сторонним организациям, если проектирование производится с привлечением сторонних организаций;

*Кн –* накладные расходы отдела проектирования.

Все расчеты будут производиться в условных единицах (у.е.), что соответствует стоимости одного доллара США в Приднестровском Республиканском Банке на момент разработки ПС.

*Затраты на материалы.* Определим смету затрат и рассчитаем стоимость материалов *Км*, пошедших на разработку ПС. Перечень материалов обусловлен темой дипломной работы. В их состав входит следующее: носители информации (бумага, магнитные диски) и быстроизнашивающиеся предметы труда (ручка, карандаш, резинка). Смета затрат на материалы представлена в таблице 3.1.

Таблица 3.1 – Смета затрат на материалы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Материал | Единица  измерения | | Цена за  единицу (у.е.) | Количество | Сумма (у.е.) |
| *CD-RW* диск | Шт. | | 1,50 | 1 | 1,5 |
| Бумага | Пач. | | 5 | 1 | 5 |
| Ручка | Шт. | | 0,30 | 2 | 0,6 |
| Файл А4 | Шт. | | 0,018 | 70 | 1,2 |
| Папка | Шт. | 0,2 | | 1 | 0,2 |
| ИТОГО |  | | | | 8,5 |
| Транспортно-заготовительные расходы (5 %) |  | | | | 1,115 |
| ВСЕГО |  | | | | 9,615 |

*Затраты на оплату труда*. Затраты на основную заработную плату проектировщика (*Кпр*) рассчитывается на основе данных о квалификационном составе разработчиков, их должностных окладах и общей занятости по теме. Дополнительная заработная плата начисляется в размере 10% от суммы основной заработной платы, а отчисления на социальные страхования – в размере 39% от фонда заработной платы. Смета затрат на оплату труда представлена в таблице 3.2.

Таблица 3.2 – Смета затрат на оплату труда

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Должность работника | Должностной оклад (у.е.) | Дневная ставка | Занятость  по теме | Сумма основной з/п (у.е.) |
| Программист | 130 | 5,2 | 80 | 416 |
| Руководитель  программного  продукта | 150 | 6,8 | 50 | 310 |
| ИТОГО |  | | | 726 у.е. |

Итого *Кпр =* 726 у.е

*Затраты на отладку программы*. Затраты, связанные с использованием машинного времени на отладку программы (*Кмаш*) учитываются для следующих этапов проектирования: разработка рабочего проекта; внедрение - проведение опытной эксплуатации задач и сдача их в промышленную эксплуатацию.

Затраты на отладку программы определяются по формуле:

, (3.3)

где *Смч* – стоимость одного часа машинного времени;

*tотлl* – время отладки программы (ч);

*Sпрl* – количество программистов.

Подставляя фактические данные, получаем величину затрат на отладку программы:

*Смч =* 0,2 у.е*.,*

*tотл =* 48 часов,

*Sпр =* 1 программист *Кмаш* = 0,2·48·1 = 9,6 у.е.

В связи с тем, что сторонние организации не привлекались к работе, то *Кс* = 0.

Накладные расходы на разработку дипломной работы берутся в размере 45% от основной заработной платы разработчиков для покрытия административно-хозяйственных и других непредусмотренных расходов:

*Кн = Кпр·*0,45. (3.4)

Так как затраты на основную заработную плату проектировщика (*Кпр*) равны 726 у.е., то накладные расходы составят: *Кн =* 726\*0,45 = 326,7 у.е.

Так как при реализации данной задачи не производилось специальных закупок техники и переустройства рабочих мест, капитальные вложения на реализацию задачи *Кр* равны нулю и общая величина капитальных вложений определяется затратами на предпроизводственные затраты. Общая величина капитальных вложений приведена в таблице 3.3.

Таблица 3.3 – Общая смета затрат на проектирование

|  |  |  |
| --- | --- | --- |
| Статьи | Затраты | |
| Сумма (у.е.) | Удельный вес статьи в общей стоимости (%) |
| Материалы и покупные полуфабрикаты | 9,615 | 0,72 |
| Основная заработная плата | 726 | 54,38 |
| Дополнительная заработная плата | 72,6 | 5,43 |
| Отчисления на единый социальный налог | 190,485 | 14,26 |
| Затраты на отладку программы | 9,6 | 0,71 |
| Накладные расходы | 326,7 | 24,47 |
| ИТОГО: | 1335 | 100 |

Итого общая величина капитальных вложений на реализацию проекта составляет 1335 у.е.

К затратам текущего характера относятся затраты, связанные с обеспечением нормального функционирования разработанного программного средства.

Это могут быть затраты на ведение информационной базы, эксплуатацию технических средств, реализацию технологического процесса обработки информации по задачам, эксплуатацию системы в целом.

Затраты, связанные с эксплуатированием задачи вычисляются по формуле:

*Сэз = Смч* ·*Тэ*, (3.4)

где *Смч* – стоимость одного часа работы технических средств;

*Тэ* –время эксплуатации задачи в течение года.

Подставляя реальные значения, полученные в ходе опытной эксплуатации задачи, получаем величину годовых эксплуатационных расходов с учетом оплаты за расход электроэнергии компьютера в год:

*Сэз* = 0,2·980 = 196 у.е.

*Определение экономической эффективности**от внедрения программы.* Экономический эффект, как реальная экономия, обусловлена следующими факторами: сокращением времени обработки информации; сокращением потерь рабочего времени.

Рассчитаем абсолютную годовую экономию на основе сокращения потерь рабочего времени, образующуюся в виде экономии на заработной плате за счет: снижение затрат на оплату простоев служащих; сокращение численности служащих; увеличение эффективности фонда времени одного служащего; сокращение сверхурочных работ.

Сокращения затрат при использовании программных средств для решения поставленной задачи обусловлено снижением трудоемкости работ по обработке информации и снижением затрат на оплату простоев сотрудников.

*Расчет экономии за счет снижения трудоемкости решения задачи.* Экономия за счет снижения трудоемкости решения определенного класса задач, рассчитывается по формуле:

*Этр = (А* · *В* · *Тр*· *Зчас* – *Кр*· *Тоб* · *Смч)* · *Ue*, (3.5)

где *А* – коэффициент, учитывающий дополнительную заработную плату;

*В* – коэффициент, учитывающий отчисления на соцстрах;

*Тр* – трудоемкость решения задачи вручную (ч);

*Зчас* – среднечасовая тарифная ставка работника (у.е.);

*Кр* – коэффициент использования технических средств;

*Тоб* – трудоемкость при автоматизированной обработке (ч);

*Смч* – стоимость одного машинного часа работы (у.е.);

*Ue* – периодичность решения задачи (раз/год).

Подставляя реальные данные, полученные в результате исследований при ручном (полуавтоматизированном) и автоматизированном способах планирования деятельности предприятия, получаем величину экономии за счет снижения трудоемкости решения задачи при условии, что

*А* = 1,1; *В* = 1,27; *Тр* = 2 ч; *Зчас* = 0,738 у.е. (при основной заработной плате 130 у.е., 8-мичасовом рабочем дне, 22 рабочих дня в месяц);

*Кр* = 1,13; *Тоб* = 0,2 ч; *Смч* = 0,1 у.е.; *Ue* = 700 раз в год.

*Этр* = (1,1·1,27·2·0,738 – 1,13·0,2·0,2) ·700 = 1475,04 у.е.

*Определение годового экономического эффекта*. Основной экономический показатель, определяющий экономическую целесообразность затрат на создание программного продукта – это годовой экономический эффект, который определяется по формуле:

*Эс=Этр – Ен*·*Кп – Сэз*, (3.6)

где *Этр* – годовая экономия от применения внедренной задачи;

*Ен* – нормативный коэффициент экономической эффективности капитальных вложений (*Ен*= 0,15);

*Кп* – единовременные затраты, связанные с внедрением задачи;

Подставляя в формулу (3.6) реальные данные, определяем величину годового экономического эффекта при *Кп =* 1335 у.е:

*Эс* = 1475,04 – 0,15·1335 – 211,2 = 1063,55 у.е.

*Расчет экономической эффективности.* Экономическая эффективность капитальных вложений, связанных с разработкой и внедрением программного продукта определяется по формуле:

*Ерс = Эс/Кп*. (3.7)

Подставляя в формулу фактические данные, определяем величину экономической эффективности: *Ерс* = 1063,55 / 1335 = 0,79.

Так как *Ерс* > *Ен*, то внедрение экономически эффективно. Определяем срок окупаемости внедренной задачи:

*Те = Кп/Эс* = 1335 / 1063,55 = 1,25 года.

Расчеты показали, что использование данного программного продукта является экономически оправданным и ведет к сокращению потерь рабочего времени за счет уменьшения времени решения «вручную», что в свою очередь приводит к значительной экономии человеческих ресурсов и финансовых средств.

**7 Охрана труда**

Для проектирования базы данных и архитектуры программного обеспечения, автоматизирующее процесс анкетирования требуется большое умственное напряжения. А для реализации конечного программного продукта необходимо затратить значительное количество времени, работая за компьютером.

Работа за компьютером обычно сопровождается нервно-эмоциональной нагрузкой операторов, высокой напряженностью зрительной работы и достаточно большой нагрузкой на мышцы рук при работе с клавиатурой ЭВМ. Поддержание наилучшей рабочей позы работника в основном зависит от рациональной конструкции и расположения элементов рабочего места.

Рабочее место программиста должно характеризоваться следующим образом: площадь не менее 6 м², высота помещения не менее 4 м, а объем – не менее 20 м3 на одного человека. После проведения анализа рабочего места программиста в помещении было выяснено, что площадь данного рабочего места составляет 4 м2, а объем 12 м3, что не соответствует приведенным требованиям. Также в результате анализа были выявлены нарушения в организации непосредственно самого рабочего места программиста. В связи с этим необходимо организовать рабочее место программиста, следующим образом. Высота над уровнем пола рабочей поверхности, за которой работает оператор, должна составлять 720 мм. Желательно, чтобы рабочий стол оператора при необходимости можно было регулировать по высоте в пределах 680-780 мм. Приемлемые размеры поверхности стола 1600х1000 мм2. Под столом должно иметься пространство для ног с размерами по глубине 450 мм. Рабочий стол оператора должен также иметь подставку для ног, расположенную под углом 15 к поверхности стола. Длина подставки 400 мм, ширина – 350 мм. Удаленность клавиатуры от края стола должна быть не более 300 мм, что обеспечит оператору удобную опору для предплечий. Расстояние между глазами оператора и экраном видеодисплея должно составлять 40-80 см.

Рабочий стул программиста должен быть снабжен подъемно-поворотным механизмом. Высота сиденья должна регулироваться в пределах 400-500 мм. Глубина сиденья должна составлять не менее 400 мм, а ширина – не менее 400 мм. Высота опорной поверхности спинки не менее 300 мм, ширина – не менее 380 мм. Угол наклона спинки стула к плоскости сиденья должен изменяться в пределах 90-110.

Основным источником электромагнитных полей является монитор. Основной мерой по борьбе с электромагнитными полями является максимальное разнесение оператора и монитора друг от друга. Электромагнитное поле, генерируемое монитором, вызывает электризацию пластмассовых деталей перед ним, поэтому не рекомендуется оснащать ВЦ мебелью из пластмасс. Все оборудование должно быть заземлено, а в особенности принтер, т.к. при трении ленты между бумагой и головкой весь этот узел сильно электризуется. Допустимый уровень напряженности электростатического поля не должен превышать 20 кВ/м. В качестве меры защиты от статического электричества проводится влажная уборка помещения. Также для защиты от электромагнитных излучений используется защита временем, расстоянием и экранирование.

*Электробезопасность.* В ходе разработки программного продукта текущей квалификационной работы было использовано всяческое электрическое оборудование, в том числе различную вычислительную технику.

Лаборатория относится к помещениям повышенной опасности. Основной мерой защиты от поражения электрическим током являются: применение для облицовки современных электроизоляционных материалов; выполнение электропроводки закрытого типа с возможностью быстрого отключения на легкодоступном щите; обязательное заземление.

*Расчет выносного заземления*. Рассчитаем выносное заземляющее устройство. Преимуществом такого типа заземляющего устройства является возможность выбора места размещения электродов с наименьшим сопротивлением грунта (сырое, глинистое).

Сопротивление группового заземлителя рассчитывается, если:

* мощность установки менее 2 кВА;
* вертикальный заземлитель – стальной прут диаметром 25 мм и длиной 3 м;
* горизонтальный заземлитель – стальная полоса шириной 25 мм, толщиной 5 мм;
* удельное сопротивление грунта (глина) 70 Ом⋅м.

Сопротивление одиночного вертикального заземлителя рассчитывается по формуле:

 (3.8)



где  − удельное сопротивление грунта (Ом⋅м);

*l* − длина вертикального заземлителя (м);

*d* − диаметр вертикального заземлителя (м);

*t* − глубина заложения.

 (3.9)

.

Расстояние между заземлителями (м):

 (3.10)

.

Ориентировочное количество вертикальных заземлителей (шт):

 (3.11)

где *Rзаз* – нормируемая величина сопротивления заземления (*Rзаз=*4*Ом*);



Количество вертикальных заземлителей определяется по формуле:

 (3.12)

где  – коэффициент использования вертикальных заземлителей (так как ориентировочное *n=*6 и *la=*3, поэтому ).



Длина горизонтального заземлителя (м):

 (3.13)

.

Сопротивление горизонтального заземлителя рассчитывается по формуле:

 (3.14)

где *b1* – ширина полосы (м)

.

Сопротивление группового заземлителя:

 (3.15)

где  – коэффициент использования горизонтальных заземлителей ()

, 

Рассчитанное заземление подходит для помещения, в котором проводилась реализация программного продукта, и обеспечит защиту персонала от поражения электрическим током в случае неисправности оборудования (при пробое на корпус).

*Пожарная безопасность.* Степень огнестойкости зданий принимается в зависимости от их назначения, категории по взрывопожарной и пожарной опасности, этажности, площади этажа в пределах пожарного отсека.

Здание, в котором находится помещение, по пожарной опасности строительных конструкций относится к категории *K1* (малопожароопасное), поскольку здесь присутствуют горючие вещества (книги, мебель, оргтехника и т.д.), которые при взаимодействии с огнем могут гореть без взрыва.

По конструктивным характеристикам здание можно отнести к зданиям с несущими и ограждающими конструкциями из естественных или искусственных каменных материалов, бетона или железобетона.

Следовательно, степень огнестойкости здания можно определить как третью (III).

Помещение по функциональной пожарной опасности относится к классу Ф1.3 − многоквартирные жилые дома.

Здание оборудовано пожарным водопроводом высокого давления с пожарными кранами.

Требования, предъявляемые к пожарной безопасности:

* установка пожарно-сигнальной аппаратуры с соответствующими тепловыми и дымовыми датчиками;
* выполнение скрытой электропроводки в стенах;
* устранение неисправных выключателей и розеток;
* запрет на использование оголенных шнуров и проводов для соединения;
* необходимо иметь в доступном видимом месте углекислотные огнетушители (по характеру помещения – минимум 1 шт.).

*Причины возникновения пожара*. Пожар в помещении может привести к очень неблагоприятным последствиям (потеря ценной информации, порча имущества, гибель людей и т.д.), поэтому необходимо: выявить и устранить все причины возникновения пожара; разработать план мер по ликвидации пожара в помещении; план эвакуации людей из помещения.

Причинами возникновения пожара могут быть:

неисправности электропроводки, розеток и выключателей которые могут привести к короткому замыканию или пробою изоляции;

использование поврежденных (неисправных) электроприборов;

использование в помещении электронагревательных приборов с открытыми нагревательными элементами;

возникновение пожара вследствие попадания молнии в здание;

возгорание здания вследствие внешних воздействий;

неаккуратное обращение с огнем и несоблюдение мер пожарной безопасности.

Профилактика пожара. Пожарная профилактика представляет собой комплекс организационных и технических мероприятий, направленных на обеспечение безопасности людей, на предотвращении пожара, ограничение его распространения, а также создание условий для успешного тушения пожара. Для профилактики пожара чрезвычайно важна правильная оценка пожароопасности здания, определение опасных факторов и обоснование способов и средств пожаропредупреждения и защиты.

В случае возникновения пожара необходимо отключить электропитание, вызвать по телефону пожарную команду, эвакуировать людей из помещения и приступить к ликвидации пожара. При наличии небольшого очага пламени можно воспользоваться подручными средствами с целью прекращения доступа воздуха к объекту возгорания.

ЗАКЛЮЧЕНИЕ

В результате выполнения задания преддипломной практики были систематизированы сведения предметная область. Были изучены существующие методы решения поставленной задачи. Были выявлены определенные требования к разрабатываемому приложению, которые охватывают все плюсы существующих методов.

С целью определения основных требований к программному продукту был разработан ряд диаграмм и схем, которые отвечают за определенный вид требований.

Было изучено клиент-серверное взаимодействие приложений, а также платформа разработки веб-приложений, в состав которой входит: веб-сервисы, программная инфраструктура, модель программирования.

Было реализовано клиент-серверное приложение и веб-приложение, позволяющее отправлять по сети запросы серверу и получать данные.

Так же была спроектирована база данных, хранящая всю необходимую информацию, и разработан сервер, позволяющий обрабатывать получаемые запросы и формировать *SQL*-запросы, а также получать данные из базы данных.

На основе разработок может быть построена библиотека, позволяющая обрабатывать данные методом расслоенного эксперимента, а так же применена не только для обработки результатов анкетирования.

При создании классов обработки данных была изучена литература и получен навык по обработке экспериментальных данных, представленных в виде специально оформленных таблиц.

Программа является законченным средством, но обладает широким потенциалом для развития своих возможностей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Бен Форта Освой самостоятельно SQL за 10 минут – СПб.: ООО "И.Д. Вильямс", 2014. – 288 с.
2. Нанс Б. Компьютерные сети. – М.: БИНОМ, 1996 год. – 35 с.
3. Полякова Л.Н. Основы SQL – М.: Интернет-университет информационных технологий, 2014. – 432 с.
4. Ржеуцкая С.Ю. Базы данных. Язык SQL – Вологда: ВоГТУ, 2010 – 59 с.
5. Трей Нэш C# 2010: ускоренный курс для профессионалов – М.: ООО "И.Д. Вильямс", 2010. – 588 с.
6. Эндрю Стиллмен, Дженнифер Грин Изучаем C#, 3-е издание – СПб.: ООО "И.Д. Вильямс", 2014. – 810 с.