

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕРАКТИВНОЙ ИНТЕЛЛЕКТУАЛЬНОЙ ВИКТОРИНЫ

Выпускная квалификационная работа бакалавра
по направлению 44.03.04 Профессиональное обучение (по отраслям)
профиля «Информатика и вычислительная техника»
специализация «Компьютерные технологии»

Идентификационный номер ВКР: 326

Екатеринбург 2016

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Российский государственный профессионально-педагогический университет»
Институт инженерно-педагогического образования
Кафедра информационных систем и технологий

К ЗАЩИТЕ ДОПУСКАЮ

Заведующая кафедрой ИС

_____ Н. С. Толстова

« ____ » _____ 2016 г.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНТЕРАКТИВНОЙ ИНТЕЛЛЕКТУАЛЬНОЙ ВИКТОРИНЫ

Выпускная квалификационная работа бакалавра
по направлению 44.03.04 Профессиональное обучение (по отраслям)
профиля «Информатика и вычислительная техника»
специализация «Компьютерные технологии»

Идентификационный номер ВКР: 326

Исполнитель:

студент группы ПУ-411оКТ

Д.А. Палехов

Руководитель:

канд. пед.наук, доцент

Н. С. Толстова

Нормоконтролер:

Б. А. Редькина

Екатеринбург 2016

РЕФЕРАТ

Пояснительная записка к выпускной квалификационной работе выполнена на 50 страницах, содержит 17 рисунков, 2 таблицы, 34 источников литературы.

Ключевые слова: ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ИНТЕРАКТИВНАЯ ВИКТОРИНА, СЕРВЕР, КЛИЕНТ, ИНТЕЛЛЕКТУАЛЬНАЯ ВИКТОРИНА, «СОБОЛЁНОК».

Объектом выпускной квалификационной работы является интерактивная викторина «Соболенок», проводимая среди учеников средних и старших классов городских школ г. Первоуральска.

Предметом выпускной квалификационной работы является программное обеспечение интерактивной викторины «Соболенок», проводимой среди учеников средних и старших классов городских школ г. Первоуральска.

Цель выпускной квалификационной работы — разработать программное обеспечение интерактивной викторины «Соболенок».

В соответствии с поставленной целью в работе решены следующие задачи:

1. Проведён анализ нормативно-справочной документации, материала, а также интернет источников по теме исследования.
2. Определена архитектура и средства реализации программного приложения.
3. Спроектирована и реализована серверная часть приложения.
4. Спроектирована и реализована клиентская часть приложения.

СОДЕРЖАНИЕ

| | |
|----------------------------------------------------------------------------------------------------------------|----|
| Введение..... | 5 |
| 1 Анализ литературы по теме «Интерактивная викторина» | 7 |
| 1.1 Анализ литературы | 7 |
| 1.2 Анализ интернет-источников | 9 |
| 1.3 Стандартизация педагогических программных средств и понятие стандарта | 9 |
| 1.4 Перечень стандартов программных продуктов | 15 |
| 1.5 Описание стандартов программных продуктов..... | 15 |
| 1.6 Общие требования к программному продукту для интерактивной викторины | 16 |
| 1.7 Анализ и выбор средств разработки программных продуктов под управлением операционной системы Windows | 17 |
| 1.8 Рекомендации и выбор сборки сервера | 19 |
| 2 Описание программного обеспечения интерактивной викторины «Соболёнок» | 21 |
| 2.1 Педагогический адрес..... | 21 |
| 2.2 Правила интерактивной викторины «Соболенок» | 21 |
| 2.3 Структура программного продукта «Соболёнок» | 22 |
| 2.4 Интерфейс и навигация в основном Windows-совместимом приложении..... | 23 |
| 2.5 Серверное приложение..... | 24 |
| 2.6 Клиентское приложение..... | 26 |
| 2.7 Архитектура приложения..... | 30 |
| 2.8 Описание программных модулей, используемых в проекте | 32 |
| 2.8.1 Windows Forms | 32 |
| 2.8.2 System.Net | 34 |
| 2.8.3 ADO.Net | 36 |

| | | |
|------|-----------------------------------------------------------------------|----------------------------------------|
| 2.9 | Аппаратно-программные требования..... | 37 |
| 2.10 | Инструкции по работе с программным продуктом для администратора 38 | |
| 2.11 | Инструкции по работе с программным продуктом для пользователя | 39 |
| | Заключение | 41 |
| | Список использованных источников | 43 |
| | Приложение 1 | Ошибка! Закладка не определена. |
| | Приложение 2 | 50 |

ВВЕДЕНИЕ

Учебно-воспитательная работа преподавателя специальных дисциплин должна носить профессионально-направленный характер. С целью повышения интереса к глубокому изучению профильных дисциплин, творческого развития личности, привития элементов исследовательской культуры преподаватели предметно-цикловой комиссии наряду с традиционными формами активно применяют дополнительные, нетрадиционные формы обучения и воспитания студентов.

Одной из эффективных форм профессионально-направленной деятельности преподавателя можно считать организацию тематических викторин.

Викторину можно провести как заключительное занятие-повторение в конце семестра или учебного года или как внеклассное мероприятие. В последнем случае ее можно провести как внутри группы, так и с привлечением студентов из нескольких групп.

Проведение тематической викторины является одной из нетрадиционных форм обучения.

Особенности занятия-викторины заключаются в том, что эта форма работы:

- является игровой формой контроля знаний умений;
- пробуждает и активизирует интерес учащихся к изучаемым дисциплинам;
- развивает творческую мыслительную деятельность;
- обеспечивает активность учащихся;
- учит сравнивать, анализировать, выдвигать гипотезы, объяснять, что является первыми шагами профессионально-направленной деятельности будущего специалиста;

- способствует выработке способности к психологической совместности в группе (команде), толерантности, творческому контакту с членами команды;
- способствует развитию таких качеств личности как конкурентоспособность, желание победить;
- и другие.

Объектом выпускной квалификационной работы является интерактивная викторина «Соболенок», проводимая среди учеников средних и старших классов городских школ г. Первоуральска.

Предметом выпускной квалификационной работы является программное обеспечение интерактивной викторины «Соболенок», проводимой среди учеников средних и старших классов городских школ г. Первоуральска.

Цель выпускной квалификационной работы — разработать программное обеспечение интерактивной викторины «Соболенок».

Задачи выпускной квалификационной работы:

1. Провести анализ нормативно-справочной документации, материала, а также интернет источников по теме исследования.
2. Определить архитектуру и средства реализации программного приложения.
3. Спроектировать и реализовать серверную часть приложения.
4. Спроектировать и реализовать клиентскую часть приложения.

1 АНАЛИЗ ЛИТЕРАТУРЫ ПО ТЕМЕ «ИНТЕРАКТИВНАЯ ВИКТОРИНА»

1.1 Анализ литературы

Книга автора Шарипова Ф.В. «Педагогика и психология высшей школы» [29] содержит множество полезной информации об игровых формах обучения, формах контроля знаний учащихся, которые в свою очередь помогают в формировании тестов, что даёт общее представление о некоторых возможностях, которые можно применить в составлении игровых интерактивных интеллектуальных викторин.

Книга авторов Киселёва Г.М., Бочковой Р.В. «Информационные технологии в педагогическом образовании» [12] содержит подробную информацию о типах игровых технологий, их классификацию, формы. Каждый тип или форма содержит примеры и пояснения, что делает информацию в книге легко воспринимаемой и не оставляет за собой вопросов. Также в книге содержатся правила оформления викторин, правила группировки тестовых заданий и информация об инструкции к тесту.

Книга за авторством Джона Скита. «C# для профессионалов: тонкости программирования» [10] несмотря на понятное изложение материала и год издания, данная книга не подходит даже для начинающих программистов, так как в ней представлены техники программирования профессионального уровня.

Книга автора Кристиана Нейгла C# 5.0 и платформа .NET 4.5 для профессионалов [14] материал, представленный в этой книге, включает в себя практически всё необходимое для знакомства как со средой .NET Framework, так и языком C#, имеется множество примеров кода, техник программирования и заданий. Материал хорошо иллюстрирован и легко воспринимается как новичком, так и только начинающим программистом. Также книга является

универсальным справочником, по основным понятиям и техникам программирования.

Книга: Язык программирования C#. Классика Computers Science. 4-е изд. Под авторством [30] А. Хейлсберг, М. Торгерсен, С. Вилтамут, П. Голд

Это четвертое издание главной книги по языку C#, написанной легендарным программированием — Андерсом Хейлсбергом, архитектором C#, Delphi и Turbo Pascal, совместно с другими специалистами, входившими в группу разработчиков C# компании Microsoft. Издание является наиболее полным описанием языка и самым авторитетным источником информации по этой теме, построенным в формате сборника спецификаций, включающих в себя описание синтаксиса, сопутствующие материалы и примеры, а также образцы кода. Эта книга — своего рода «библия» разработчика, которая с легкостью может заменить как MSDN, так и остальные книги по C#. Четвертое издание содержит описание новых особенностей C# 4.0, включая динамическое связывание, именованные и необязательные параметры, а также ковариантные и контрвариантные обобщенные типы. Цель этих новшеств — расширение возможностей C# для взаимодействия с объектами, не относящимися к платформе .NET. Отличительная особенность нового издания также состоит в том, что каждая глава книги содержит обширные комментарии, написанные известными «гуру» программирования, такими как Джон Скит, Джозеф Альбахари, Билл Вагнер, Кристиан Нейгел, Эрик Липперт и др.

Книга: Изучаем C#. 3-е изд. Под Авторством Э. Стиллмен и Дж. Грин [35]. В отличие от большинства книг по программированию, построенных на основе скучного изложения спецификаций и примеров, с этой книгой читатель сможет сразу приступить к написанию собственного кода на языке программирования C# с самого начала. Читатель может освоить минимальный набор инструментов, и даже принять участие в интересных программных проектах: от разработки карточной игры до создания серьезного бизнес-приложения. Второе издание книги включает последние версии C# 5.0, Visual Studio 2012 и .NET 4.5 Framework, и будет интересно всем, изуча-

ющим язык программирования C#. Особенностью данного издания является уникальный способ подачи материала, выделяющий серию «Head First» издательства O'Reilly в ряду множества скучных книг, посвященных программированию.

1.2 Анализ интернет-источников

Электронный ресурс «Современное программирование на языке C#» [23] является справочником, содержит большой объем справочной информации по основным понятиям C#, таким как типы данных, математические операции, базовые структуры и другое. Имеет целый раздел с заданиями на различные тематики, что поможет, как освежить память, так и закрепить знания или проверить себя.

Электронный ресурс «Современный учебник JavaScript» [24] информационный портал о языке клиентских сценариев JavaScript будет полезен не только при использовании базовых операторов и функций языка, но и при использовании библиотеки JQuery и веб-технологии Ajax.

Электронный ресурс «Мастера C#» [15] содержит большое количество книг на тему программирования, имеется форум для получения советов от программистов, чат для общения, статьи и новости с техниками программирования и логическими решениями. Подходит как для общения, так и для обращения за помощью к профессионалам.

1.3 Стандартизация педагогических программных средств и понятие стандарта

Стандартизация – это деятельность по установлению норм, правил и характеристик (требований) в целях обеспечения[22]:

- безопасности продукции, работ и услуг для окружающей среды, жизни, здоровья и имущества;

- технической и информационной совместимости, а также взаимозаменяемости продукции;
- качества продукции, работ и услуг в соответствии с уровнем развития науки, техники и технологии;
- единства измерений;
- экономии всех видов ресурсов;
- безопасности хозяйственных объектов с учетом риска возникновения природных и техногенных катастроф и других чрезвычайных ситуаций.

К нормативным документам по стандартизации, действующим на территории Российской Федерации, относятся государственные стандарты Российской Федерации (ГОСТ Р), межгосударственные стандарты стран СНГ (ГОСТ), международные (региональные) стандарты, правила, нормы и рекомендации по стандартизации, общероссийские классификаторы технико-экономической информации, стандарты отраслей, стандарты предприятий, стандарты научно-технических, инженерных обществ и других общественных объединений [22].

Основополагающим государственным стандартом Российской Федерации является ГОСТ Р 1.0-92 (Государственная система стандартизации Российской Федерации. (Основные положения) [1].

Государственные стандарты разрабатываются на продукцию, работы и услуги, имеющие межотраслевое значение. Они содержат:

- требования к продукции, работам и услугам по их безопасности для окружающей среды, жизни, здоровья и имущества, требования пожарной безопасности, требования техники безопасности и производственной санитарии;
- требования по технической и информационной совместимости, а также взаимозаменяемости продукции;
- основные потребительские (эксплуатационные) характеристики продукции, методы их контроля, требования к упаковке, маркировке, транспортированию, хранению, применению и утилизации продукции;

- правила и нормы, обеспечивающие техническое и информационное единство при разработке, производстве, использовании (эксплуатации) продукции, выполнении работ и оказании услуг, в том числе правила оформления технической документации, допуски и посадки, общие правила обеспечения качества продукции, работ и услуг, сохранения и рационального использования всех видов ресурсов, термины и их определения, условные обозначения, метрологические и другие общетехнические и организационно-технические правила и нормы [18].

Для обеспечения государственной защиты интересов Российской Федерации и конкурентоспособности отечественной продукции (услуг) в государственных стандартах в обоснованных случаях устанавливаются предварительные требования на перспективу, опережающие возможности традиционных технологий.

Соответствие продукции и услуг требованиям государственных стандартов может подтверждаться путем маркирования продукции и услуг знаком соответствия государственным стандартам. Предприятия и организации, которым выдана лицензия на маркирование продукции и услуг знаком соответствия государственным стандартам, а также сами продукция и услуги, маркированные этим знаком, вносятся в Государственный реестр продукции и услуг, маркированных знаком соответствия государственным стандартам.

Государство гарантирует экономическую поддержку и стимулирование субъектов хозяйственной деятельности, которые производят продукцию (оказывают услуги), маркированную знаком соответствия государственным стандартам, в том числе государственным стандартам с предварительными требованиями на перспективу, опережающими возможностями традиционных технологий [9].

Требования, устанавливаемые государственными стандартами для обеспечения безопасности продукции, работ и услуг по отношению к окружающей среде, жизни, здоровью и имуществу, для обеспечения техниче-

ской и информационной совместимости, взаимозаменяемости продукции, единства методов их контроля и единства маркировки, являются обязательными для соблюдения государственными органами управления, а также субъектами хозяйственной деятельности.

Иные требования государственных стандартов к продукции, работам и услугам подлежат обязательному соблюдению субъектами хозяйственной деятельности в силу договора либо в том случае, если об этом указывается в технической документации изготовителя (поставщика) продукции, исполнителя работ или услуг. При этом соответствие продукции и услуг этим требованиям государственных стандартов может определяться в порядке добровольной сертификации продукции и услуг.

Стандарты предприятий могут разрабатываться и утверждаться предприятиями самостоятельно, исходя из необходимости их применения в целях обеспечения требований Закона о стандартизации, а также в целях совершенствования организации и управления производством [22].

Требования стандартов предприятий подлежат обязательному соблюдению другими субъектами хозяйственной деятельности, если в договоре на разработку, производство и поставку продукции, на выполнение работ и оказание услуг сделана ссылка на эти стандарты.

Нормативные документы по стандартизации должны применяться государственными органами управления, субъектами хозяйственной деятельности на стадиях разработки, подготовки продукции к производству, ее изготовления, реализации (поставки, продажи), использования (эксплуатации), хранения, транспортирования и утилизации, при выполнении работ и оказании услуг, при разработке технической документации (конструкторской, технологической, проектной), в том числе технических условий, каталожных листов на поставляемую продукцию (оказываемые услуги).

Заказчик и исполнитель обязаны включать в договор условие о соответствии продукции, выполняемых работ и оказываемых услуг обязательным требованиям государственных стандартов.

Государственный контроль и надзор за соблюдением субъектами хозяйственной деятельности обязательных требований государственных стандартов осуществляется на стадиях разработки, подготовки продукции к производству, ее изготовления, реализации (поставки, продажи), использования (эксплуатации), хранения, транспортирования и утилизации, а также при выполнении работ и оказании услуг [26].

Государственные инспекторы при осуществлении возложенных на них обязанностей должны защищать интересы потребителей, субъектов хозяйственной деятельности и государства, руководствуясь законодательством. Государственный инспектор имеет право:

- свободного доступа в служебные и производственные помещения субъекта хозяйственной деятельности;
- получать от субъектов хозяйственной деятельности документы и сведения, необходимые для проведения государственного контроля и надзора;
- использовать технические средства и привлекать специалистов субъекта хозяйственной деятельности при проведении государственного контроля и надзора;
- проводить в соответствии с действующими нормативными документами по стандартизации отбор проб и образцов продукции, а также услуг для контроля их соответствия обязательным требованиям государственных стандартов с отнесением стоимости израсходованных образцов и затрат на проведение испытаний (анализов, измерений) на издержки производства проверяемых субъектов хозяйственной деятельности;
- выдавать предписания об устранении выявленных нарушений обязательных требований государственных стандартов на стадиях разработки, подготовки продукции к производству, ее изготовления, реализации (поставки, продажи), использования (эксплуатации), хранения, транспортирования и утилизации, а также при выполнении работ и оказании услуг;

- выдавать предписания о запрете или приостановке реализации (поставки, продажи), использования (эксплуатации) проверенной продукции, а также выполнения работ и оказания услуг в случаях несоответствия продукции, работ и услуг обязательным требованиям государственных стандартов;
- запрещать реализацию продукции, выполнение работ и оказание услуг в случае уклонения субъекта хозяйственной деятельности от предъявления продукции, работ и услуг для проверки.

В компетенцию главного государственного инспектора, помимо перечисленного, входит:

- принимать постановления о применении к субъектам хозяйственной деятельности штрафов за нарушения обязательных требований государственных стандартов;
- запрещать реализацию импортной продукции и оказание импортных услуг, не соответствующих обязательным требованиям государственных стандартов и не прошедших государственную регистрацию в соответствии с законодательством Российской Федерации [22].

За нарушение предписаний государственных инспекторов о запрете или приостановке реализации (поставки, продажи) продукции, выполнения работ и оказания услуг субъекты хозяйственной деятельности уплачивают штраф в размере стоимости реализованной продукции, выполненных работ и оказанных услуг или в установленном размере в случае, если стоимость реализованной продукции, выполненных работ и оказанных услуг определить невозможно или необходимая отчетная документация не представлена.

Государственные инспекторы в случае невыполнения выданных ими предписаний и постановлений субъектами хозяйственной деятельности направляют необходимые материалы в арбитражный суд, органы прокуратуры или суд.

1.4 Перечень стандартов на разработку программных продуктов

Российские стандарты на разработку программных продуктов [14]:

- ГОСТ 34.601 90. Информационная технология. Автоматизированные системы. Стадии создания [3].
- ГОСТ 34.602 89. Информационная технология. Техническое задание на создание автоматизированной системы [4].
- ГОСТ 34.201 89. Информационная технология. Виды, комплектность и обозначение документов при создании автоматизированных систем [5].
- РД 50 34.698 90. Автоматизированные системы. Требования к содержанию документов [6].
- ГОСТ 28195 89. Оценка качества программных средств. Общие положения [7].
- ГОСТ 34.603 92. Информационная технология. Виды испытаний автоматизированных систем [8].
- ГОСТ 28806 90. Качество программных средств. Термины и определения [9].

1.5 Описание стандартов на разработку программных продуктов

ГОСТ Р ИСО – 9241-3-2003. Требование к визуальному отображению информации [1].

Настоящий стандарт устанавливает требования к качеству изображения на стадии разработки и оценки одноцветных и многоцветных ВДТ. Применяют при эргономическом проектировании электронных дисплеев, используемых для решения офисных задач

ГОСТ Р ИСО – 9241-11-2010. Руководство по обеспечению пригодности использования [2].

Настоящий стандарт устанавливает понятие пригодности использования видеодисплейных терминалов и определяет информацию, необходимую для установления требований или оценки пригодности использования терминалов на основе критериев производительности работы и удовлетворенности пользователей. Стандарт предназначен для применения при проектировании, разработке, оценке, приобретении продукции и обмене информацией о пригодности ее использования.

1.6 Общие требования к программному продукту для интерактивной викторины

Для разрабатываемого продукта следует отметить следующие функциональные требования:

- простота сбора и проверки результатов ответов;
- возможность подгрузки настроек интернет подключения для клиентского приложения;
- создание базы данных полученных результатов;
- возможность вывода отчета.

Организация интерфейса – очень важное потребительское свойство продукта учебного назначения. Так как с программой работает ученик (неквалифицированный пользователь), а число сеансов работы обычно относительно невелико, так что особое значение приобретает быстрота и легкость освоения управлением. Поэтому необходимо обеспечить возможность сконцентрироваться на ходе викторины и как можно меньше думать о способах общения с компьютером. Интерфейс разрабатываемого продукта должен удовлетворять следующим требованиям:

- приемлемое цветовое решение интерфейса;
- единообразии в оформлении;
- для исключения возможности списывания с соседнего компьютера вывод вопросов случайным образом;

- отображение всех вариантов ответов на экране одновременно;
- автоматическая установка русской раскладки клавиатуры для ввода фамилии и имени.

При разработке дизайна необходимо постоянно ставить себя на место пользователя и все время стараться удовлетворить его потребности. В связи с этим, хотелось бы сказать несколько слов о таком понятии, как usability (к сожалению, из-за неустоявшейся терминологии в русском языке отсутствует общеупотребимый термин для этого широкого понятия; часто применяемые «используемость» или «практичность» не в полной мере отражают все аспекты этого понятия) [18]. Эксперт по usability Якоб Нильсен предлагает такое определение «практичности» для электронного приложения [36]:

- легкость освоения;
- запоминаемость;
- эффективность использования;
- надежность использования;
- удовлетворение пользователя.

1.7 Анализ и выбор средств разработки программных продуктов под управлением операционной системы Windows

На сегодняшний день существует множество языков программирования и сред, с помощью которых можно было бы реализовать все компоненты программного продукта, однако каждая среда и язык имеют ряд недостатков и преимуществ. Отталкиваясь от популярности языков, можно выделить следующие языки программирования и их среды:

- Visual Studio C# [41];
- Visual Studio C++ [19];
- Embarcadero Delphi [37];
- Embarcadero C++ Builder [19].

Достоинствами выбранных сред разработки являются:

- известность и популярность;
- множество вариантов расширений;
- возможность получить бесплатную ознакомительную лицензию среды разработки.

К недостаткам выбранных сред разработки можно отнести:

- стоимость полноценной лицензии;

Представим результаты сравнительного анализа данных сред по разработанным критериям (таблица 1).

Таблица 1 – Сравнительный анализ сред по основным параметрам

| Критерий сравнения | Embarcadero C++ Builder | Embarcadero Delphi XE2 | Visual Studio C++ | Visual Studio C# |
|----------------------------------------------------------|-------------------------|------------------------|-------------------|------------------|
| 1. Бесплатная ознакомительная лицензия | Есть | Есть | Есть | Есть |
| 2. Сложность освоения | Высокая | Средняя | Высокая | Средняя |
| 3. Уровень защиты исполняемого файла без протектора | Высокий | Выше среднего | Высокий | Низкий |
| 4. Размер выходного исполняемого файла | Средний | Большой | Средний | Низкий |
| 5. Нативное подключение к базе данных | Нет | Нет | Нет | Есть (.NET) |
| 6. Скорость разработки программных продуктов | Средняя | Высокая | Низкая | Высокая |
| 7. Уровень удобства разработки программных продуктов | Средний | Средний | Низкий | Средний |
| 8. Наличие стандартных средств разработки Web приложений | IntraWeb | IntraWeb | ASP.NET | ASP.NET |

Исходя из сравнительного анализа, можно четко увидеть сильные и слабые стороны языков программирования и их сред. Выбор, сделанный в пользу – Visual Studio C#, основан на личном опыте, скорости разработки, гибкости языка, что позволит достичь больших результатов за короткое время.

Язык программирования C++ является чрезвычайно мощным, содержащим средства создания эффективных программ практически любого

назначения, от низкоуровневых утилит и драйверов до сложных программных комплексов самого различного назначения. Использование этого языка в данном случае серьёзно усложнит процесс, сократит функционал финального результата и займет больше времени [19].

1.8 Рекомендации и выбор сборки сервера

На начальных стадиях разработки приложения задумывалось серверную часть поместить на «веб хостинг», а в свою очередь клиентская часть могла быть реализована через браузер. Но так, как приложение было реализовано при помощи Windows Forms данная идея была отвергнута. В качестве альтернативы можно предложить некоторые рекомендации для разработчика.

Сервер – один из основных и важных компонентов работы системы. Сервер не только является хранилищем данных, но и «веб хостингом» и обработчиком PHP модулей.

Для разрабатываемого продукта следует отметить следующие общие требования к серверу:

- простота запуска;
- возможность подключения сторонних модулей;
- встроенный MySQL;
- встроенный Apache версии не ниже 2.1;
- встроенный PHP не ниже 5.2;
- скорость запуска.

При поиске в Интернет были выделены следующие готовые сборки серверов:

- Denwer [24];
- OpenServer Basic [33];
- USBWebServer [20].

Выбор данных сборок серверов основан на том, что они позволяют использовать основные возможности, необходимые для организации работы сервера.

Так же основным критерием выбора является доступность и свободное распространение.

Представим результаты проведенного сравнительного анализа данных сборок серверов (таблица 2).

В результате проведенного анализа, было принято решение взять за основу USBWebServer, т.к. данная сборка, является универсальной, поддерживает подключение любых модулей, что делает её идеальной для изучения. Ознакомившись с тем, как устроена связь между Apache, PHP и MySQL, сборка сервера USBWebServer была модифицирована и значительно улучшена:

- меньший размер сервера (131 Мбайт вместе с базой данных, 13 Мбайт Apache+PHP);
- чистые конфигурационные файлы;
- единый каталог хранения всех конфигурационных файлов;
- полный контроль модулей и запуск по отдельности;
- отсутствие неиспользуемых каталогов и модулей.

Таблица 2 – Сравнительный анализ сборок серверов по общим требованиям

| Критерий сравнения | Denwer | OpenServer Basic | USBWebServer |
|----------------------------------|-------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------|
| 1. Простота запуска | bat файл | при помощи специального приложения, запуск не требует специальных навыков | при помощи специального приложения, запуск не требует специальных навыков |
| 2. Подключение сторонних модулей | вручную | при помощи специального приложения | вручную |
| 3. Встроенный MySQL | 5.1 и выше | 5.1 и выше | 5.1 и выше |
| 4. Встроенный Apache | 2.1 и выше | 2.1 и выше | 2.1 и выше |
| 5. Встроенный PHP | 5.2 или 5.3 | 5.2 и выше | 5.2 и выше |
| 6. Скорость запуска | Средняя | Быстрая | Быстрая |

2 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИНТЕРАКТИВНОЙ ВИКТОРИНЫ «СОБОЛЁНОК»

2.1 Педагогический адрес

Электронное программное обеспечение интерактивной викторины «Соболенок» разработано для учеников среднего и старшего звена общеобразовательных школ г. Первоуральска.

2.2 Правила интерактивной викторины «Соболенок»

Ход проведения викторины проводится следующим образом:

- игроки делятся на команды произвольного количества человек;
- каждая команда занимает персональный компьютер с установленной на него программой считывания результатов;
- параллельно на экране, посредством проектора, который доступен всем командам, отображается ход викторины;
- викторина состоит из определённого количества вопросов, разделённых на категории цветом, каждый цвет соответствует одной тематике вопросов (рисунок 1).



Рисунок 1 — Интерфейс организаторской части

Непосредственно правила викторины выглядят следующим образом:

1. Очерёдность ответа команд определяется жребием.
2. Команда, чья очерёдность пришла, выбирает вопрос, озвучивая его.
3. Ведущий открывает выбранный командой вопрос, после чего всем командам дается время (5 сек.) на ввод его в программу сбора результатов.
4. Ввод в программу осуществляется следующим образом:
 - команда отмечает текущий номер вопроса;
 - затем вводит ответ в специальное поле;
 - для отправки результатов команда подтверждает выбранный ответ викторины.
5. После отправки результатов ведущий зачитывает правильный ответ на текущий вопрос.
6. После того, как вопросы истекли, подводятся итоги викторины.
7. По количеству данных правильных ответов выбирается команда победитель.

2.3 Структура программного продукта «Соболёнок»

Электронное программное обеспечение интерактивной викторины «Соболенок» состоит из двух компонентов, т.е. клиентская часть приложения, предназначенная для ввода ответов викторины, и серверная часть, предназначенная для сбора и подсчета результатов, что является логичным и удобным способом проведения и регистрации результатов викторины. Более современные системы уже имеют сервер и даже веб-интерфейс, что позволяет проходить тестирование удалённо и без затрат времени на размещение программы тестирования на все компьютеры. Чтобы программа получилась уникальной, следует выделить следующие требования к структуре программы:

- реализация Windows-совместимого приложения в архитектуре клиент-сервер;

- реализация серверной части и собственного веб-приложения;
- организация БД для хранения данных программы.

На рисунке 2 можно наглядно увидеть, как структуру, так и принцип работы будущей системы тестирования.

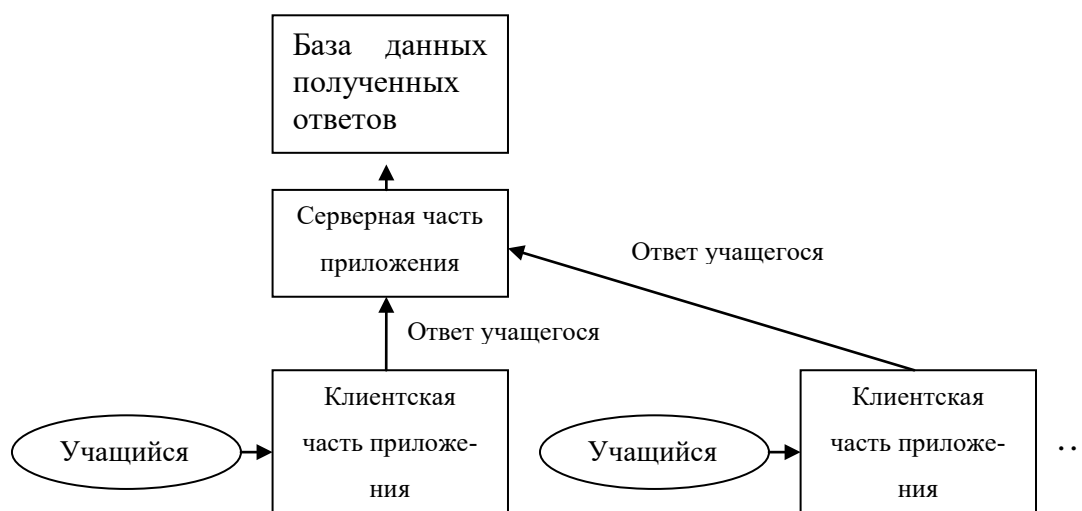


Рисунок 2 — Структура и принцип работы программного приложения

Схема выглядит несколько тривиально, однако и сам принцип работы не трудно понять. Windows-совместимое приложение управляет веб-сервером, сервером баз данных. Учащийся управляет ЭВМ, за которой установлена клиентская часть приложения. Параллельно по средствам мультимедиа на экран, или интерактивную доску выводится ход проведения викторины. Учащийся отвечают на вопросы викторины, далее данные об ответах пересылаются на сервер, где записываются в базу данных.

2.4 Интерфейс и навигация в основном Windows-совместимом приложении

Разработанное серверное приложение включает в себя следующие основные компоненты:

- контроллер сервера базы данных;
- монитор активности пользователей;
- монитор количества подключенных пользователей;

- компоненты правки и настроек объектов;
- инструмент формирования отчётов.

2.5 Серверное приложение

На главной форме серверного приложения сервисные окна и пункты меню формирования отчетов. В процессе загрузки приложения выполняется проверка на подключенных пользователей, который влияет на первую страницу, которую увидит пользователь (рисунок 3).

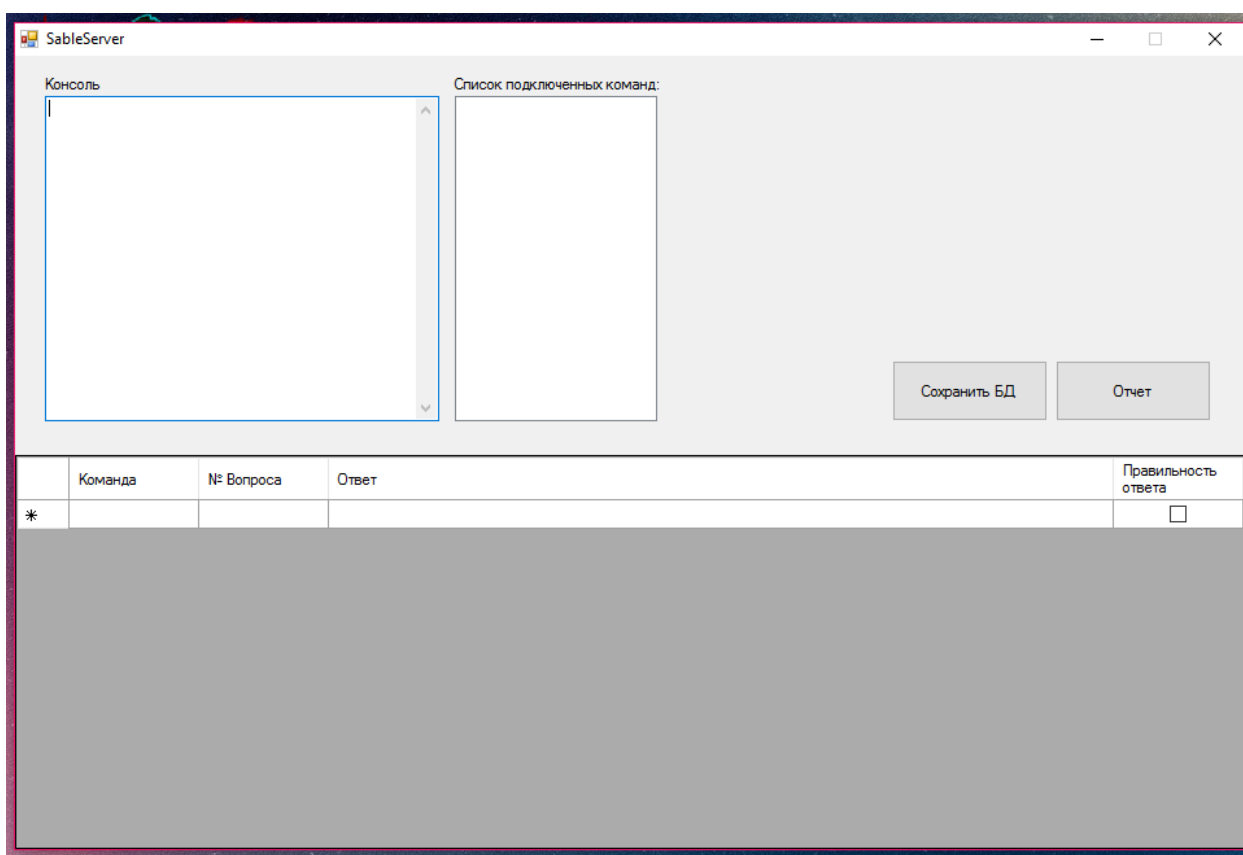


Рисунок 3 — Интерфейс серверного приложения

Здесь выполняются операции над сервером приложения – остановка, запуск, перезагрузка и конфигурация портов. Запуск сервера занимает от 5 до 30 секунд и зависит от скорости чтения и записи накопителя, с которого запущено приложения.

При нажатии на кнопку «Отчет» над таблицей прохождения пользователю сервера будет предоставлено выбрать из нескольких видов отчетов, реализованных через sql-запросы к базе данных (рисунок 4). Данные отчеты можно разделить на группы:

- по учебным учреждениям;
- по номерам вопросов;
- по количеству правильных ответов;
- итоговый запрос с подсчетом победителей.

| | TeamName | QuestionNumber | Answer | check | Щелкните для добавления |
|----|----------|----------------|--------|-------------------------------------|-------------------------|
| 2 | Школа 1 | 1 | | <input checked="" type="checkbox"/> | |
| 3 | Школа 2 | 1 | | <input type="checkbox"/> | |
| 4 | Школа 3 | 1 | | <input checked="" type="checkbox"/> | |
| 5 | Школа 4 | 1 | | <input type="checkbox"/> | |
| 6 | Школа 5 | 1 | | <input checked="" type="checkbox"/> | |
| 7 | Школа 6 | 1 | | <input checked="" type="checkbox"/> | |
| 8 | Школа 7 | 1 | | <input type="checkbox"/> | |
| 9 | Школа 8 | 1 | | <input checked="" type="checkbox"/> | |
| 10 | Школа 9 | 1 | | <input checked="" type="checkbox"/> | |
| 11 | Школа 10 | 2 | | <input checked="" type="checkbox"/> | |
| 12 | Школа 1 | 2 | | <input type="checkbox"/> | |
| 13 | Школа 2 | 2 | | <input checked="" type="checkbox"/> | |
| 14 | Школа 3 | 2 | | <input type="checkbox"/> | |
| 15 | Школа 4 | 2 | | <input type="checkbox"/> | |
| 16 | Школа 5 | 2 | | <input type="checkbox"/> | |
| 17 | Школа 6 | 2 | | <input checked="" type="checkbox"/> | |
| 18 | Школа 7 | 2 | | <input checked="" type="checkbox"/> | |
| 19 | Школа 8 | 2 | | <input type="checkbox"/> | |
| 20 | Школа 9 | 3 | | <input checked="" type="checkbox"/> | |
| 21 | Школа 10 | 3 | | <input checked="" type="checkbox"/> | |
| 22 | Школа 1 | 3 | | <input type="checkbox"/> | |
| 23 | Школа 2 | 3 | | <input checked="" type="checkbox"/> | |
| 24 | Школа 3 | 3 | | <input checked="" type="checkbox"/> | |
| 25 | Школа 4 | 3 | | <input type="checkbox"/> | |
| 26 | Школа 5 | 3 | | <input checked="" type="checkbox"/> | |
| 27 | Школа 6 | 3 | | <input type="checkbox"/> | |
| 28 | Школа 7 | 3 | | <input checked="" type="checkbox"/> | |
| 29 | Школа 8 | 3 | | <input type="checkbox"/> | |
| 30 | Школа 9 | 4 | | <input checked="" type="checkbox"/> | |
| 31 | Школа 10 | 4 | | <input checked="" type="checkbox"/> | |
| 32 | Школа 1 | 4 | | <input checked="" type="checkbox"/> | |
| 33 | Школа 2 | 4 | | <input checked="" type="checkbox"/> | |
| 34 | Школа 3 | 4 | | <input type="checkbox"/> | |
| 35 | Школа 4 | 4 | | <input checked="" type="checkbox"/> | |
| 36 | Школа 5 | 4 | | <input type="checkbox"/> | |
| 37 | Школа 6 | 4 | | <input checked="" type="checkbox"/> | |
| 38 | Школа 7 | 4 | | <input type="checkbox"/> | |
| 39 | Школа 8 | 4 | | <input checked="" type="checkbox"/> | |
| 40 | Школа 9 | 4 | | <input type="checkbox"/> | |
| 41 | Школа 10 | 5 | | <input checked="" type="checkbox"/> | |

Рисунок 4 — Интерфейс базы данных

2.6 Клиентское приложение

Запуск клиентского приложения начинается с окна регистрации для ввода данных об учебном учреждении, которое представляет их команда (рисунок 5).

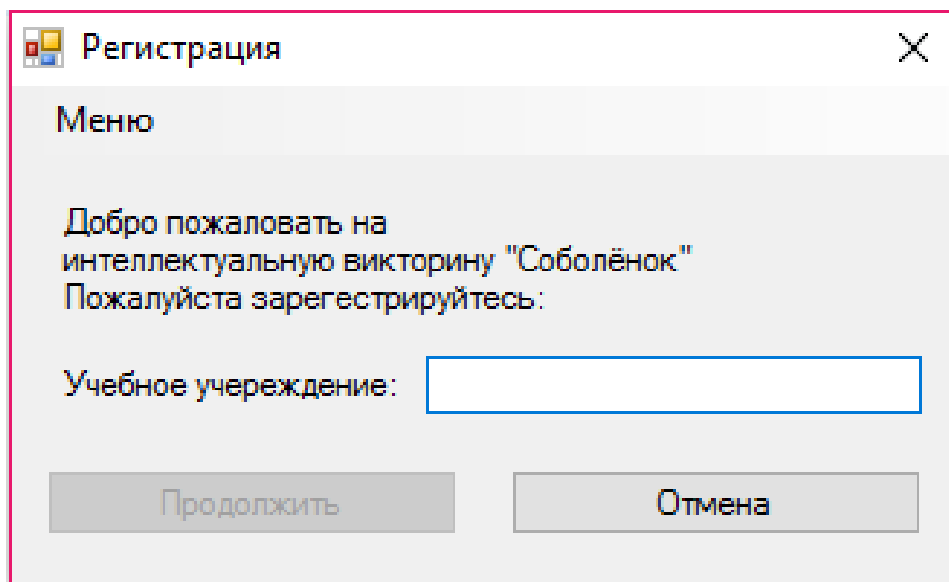


Рисунок 5 — Окно регистрации участника викторины

При первом запуске программы необходимо осуществить настройку параметров подключения. Для этого необходимо кликнуть на вкладку меню, и выбрать подпункт «Настройки подключения» (рисунок 6).

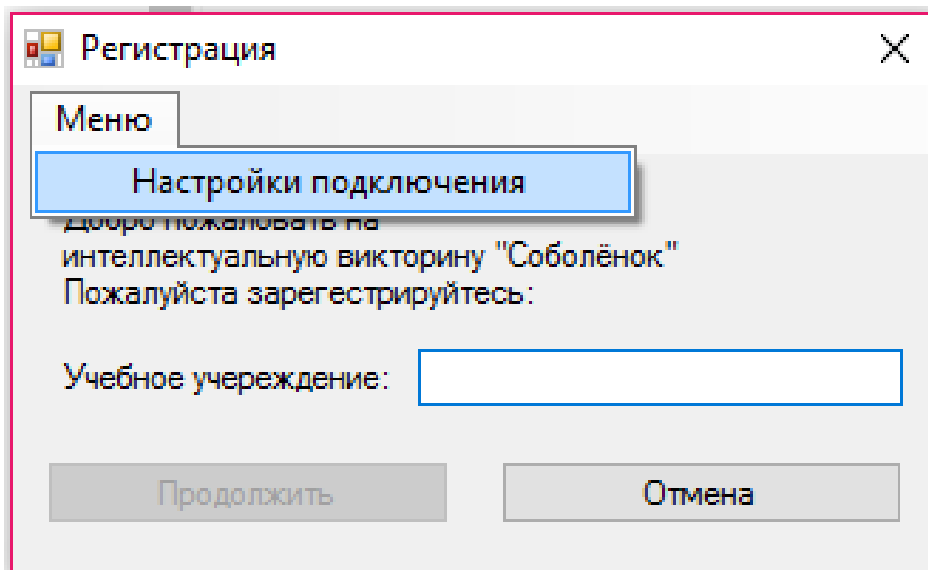
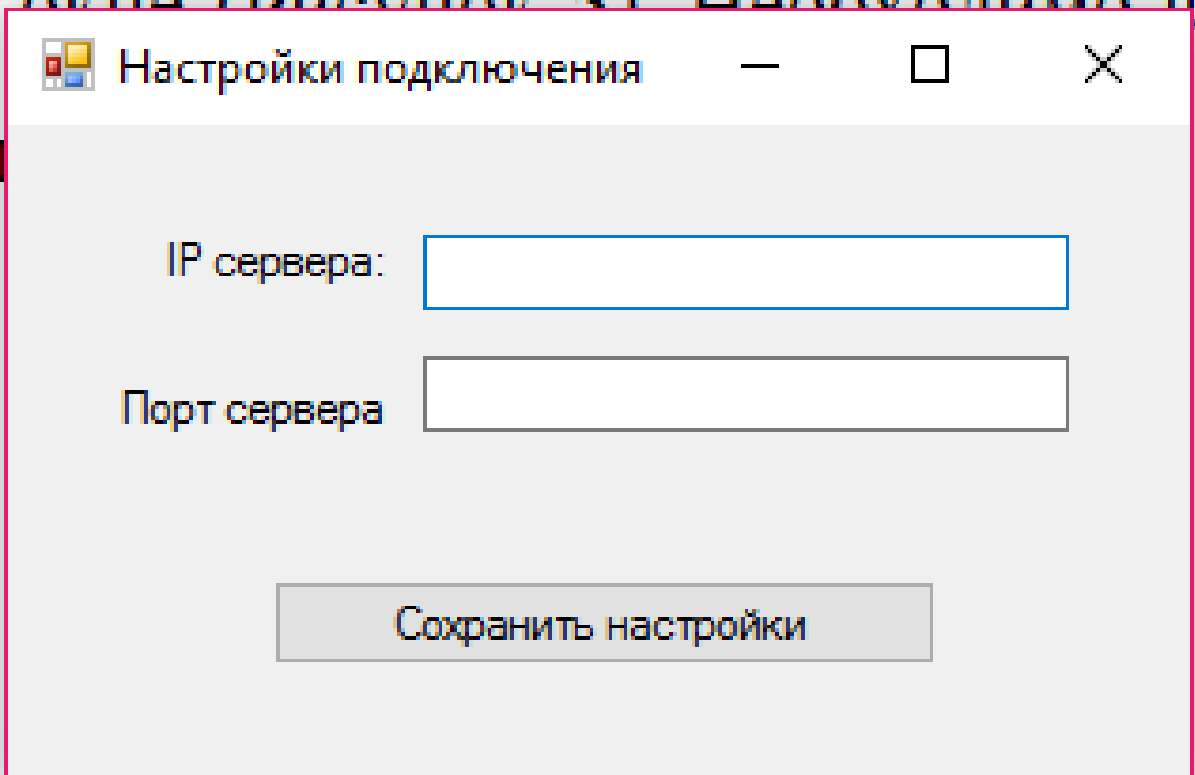


Рисунок 6 — Окно регистрации участника викторины с выпадающим меню

В всплывающем окне (рисунок 7). Необходимо ввести данные о сервере (IP адрес и порт сервера).



Настройки подключения

IP сервера:

Порт сервера

Сохранить настройки

Рисунок 7 — Окно настроек подключения к серверу

Если сервер не запущен или данные о подключении введены неверно клиентское приложение вернёт следующую ошибку (рисунок 8).

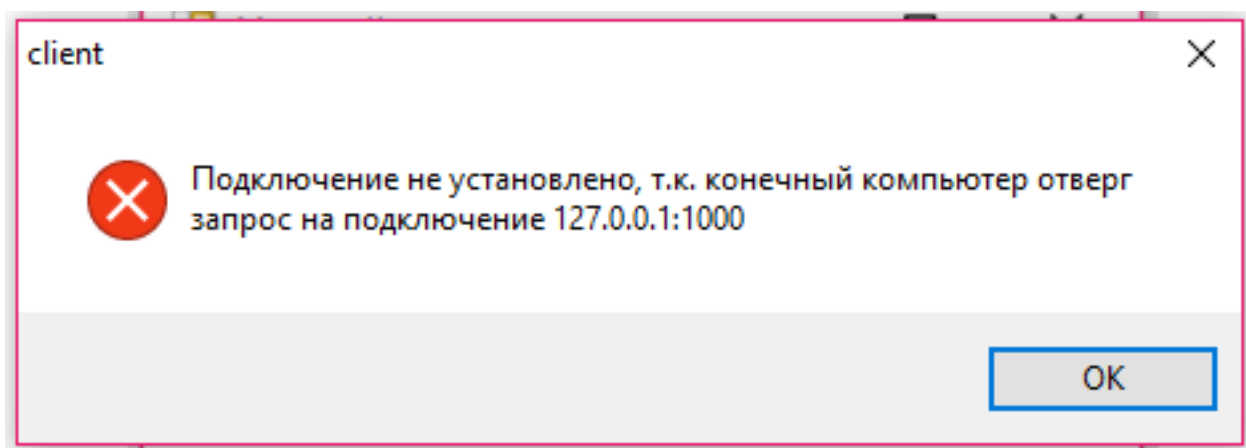


Рисунок 8 — Окно ошибки подключения

В случае удачного подключения учащемуся будет представлено окно выбора текущего вопроса викторины (рисунок 9).

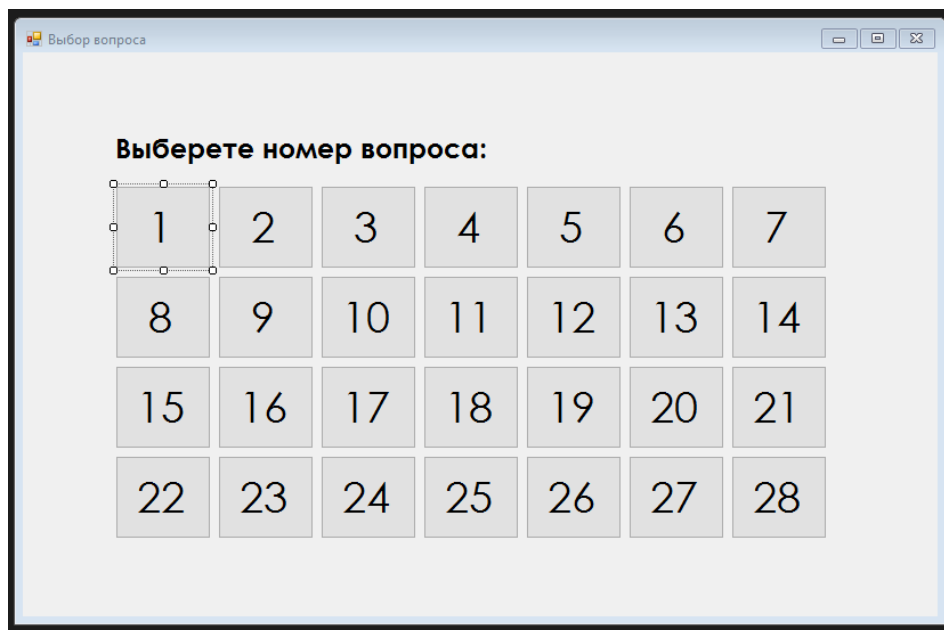


Рисунок 9 — Интерфейс окна выбора номера вопроса

После выбора номера вопроса пользователю будет предоставлено окно ввода ответа (рисунок 9).

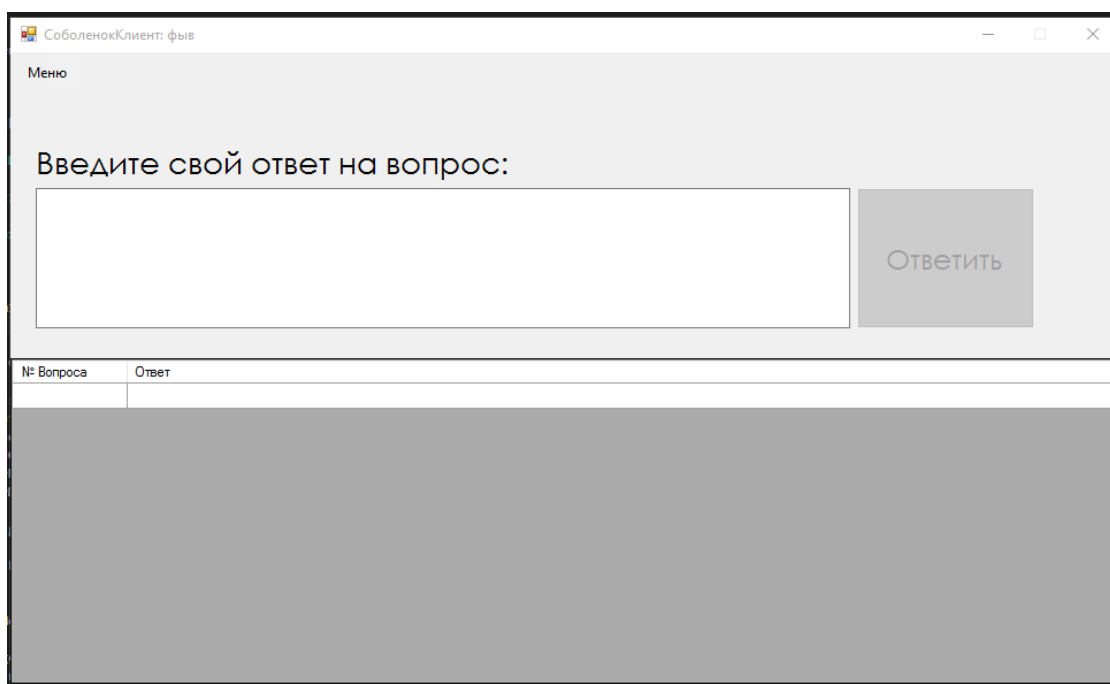


Рисунок 10 — Интерфейс окна приложения ввода ответа

В нижней части основного окна клиентского приложения ведётся протоколирование ответов команды учащегося. В текстовое поле «ответ» вво-

дится ответ на текущий вопрос викторины. Номер вопроса выбирается на счетчике слева от текстового поля для ввода ответов.

После завершения викторины, либо в случае преждевременного закрытия приложения пользователю будет представлено диалоговое окно (рисунок 11).

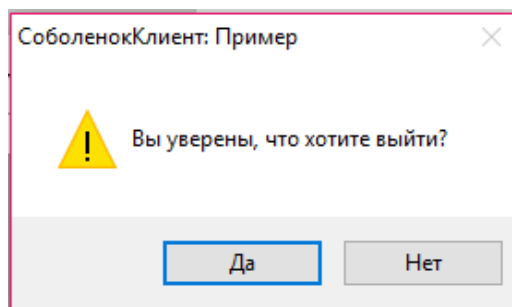


Рисунок 11 — Диалоговое окно подтверждения выхода из приложения

В случае разрыва подключения или других технических неполадок, при невозможности отправить сообщение на сервер пользователю будет представлено следующее сообщение об ошибке (рисунок 12).

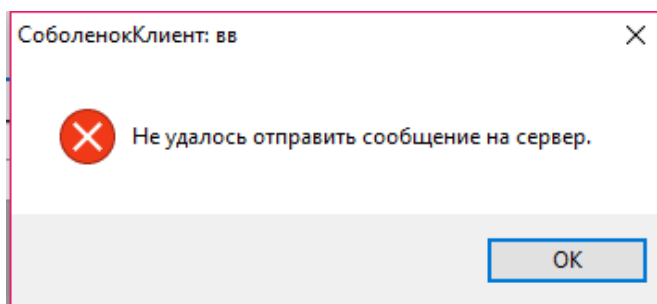


Рисунок 12 — Диалоговое окно ошибки отправки сообщения

При невозможности передать серверу команду об отключении клиента, и сохранении данных в отчет клиенту будет передано следующее сообщение (рисунок 13).

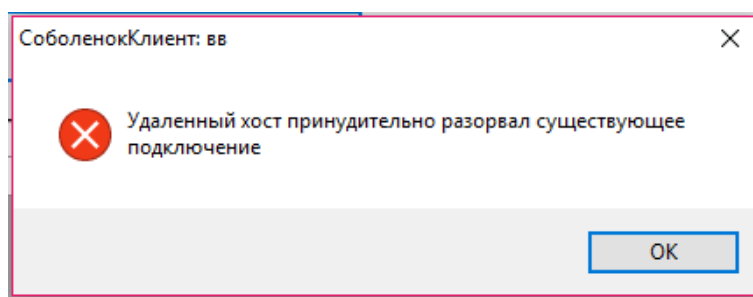


Рисунок 13 — Диалоговое окно ошибки преждевременного разрыва соединения с сервером

2.7 Архитектура приложения

Оба приложения, клиентская и серверная часть, работают через технологию TCP/IP протоколирования. Ниже приведены схемы архитектуры клиентской и серверной части приложения.

По сравнению с серверной частью программного продукта клиентская часть производит операции по передаче данных из текстового поля, кодируя сообщение из строковой переменной в базовую.

Архитектура клиентской части представлена на рисунке 14.

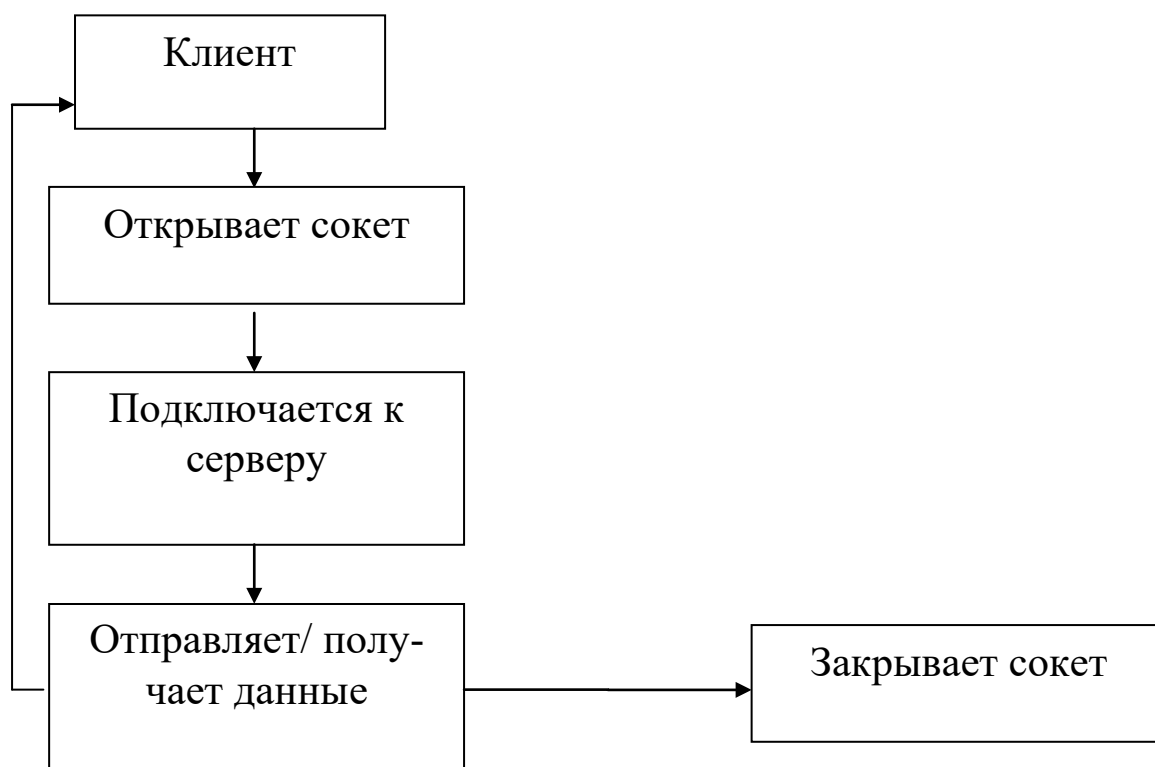


Рисунок 14 — Структура клиентской части программного продукта

Серверная часть приложения, получая данные от клиента декодирует их из базовой обратно в строковую, после чего добавляет их в базу данных. После этого сервер обновляет данные в таблице состояния прохождения викторины.

Архитектура серверной части представлено на рисунке 15.

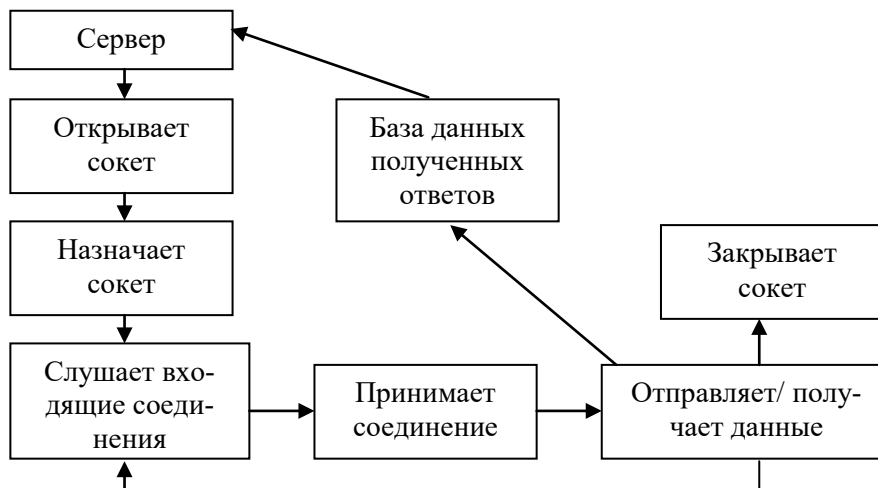


Рисунок 15 — Структура серверной части программного продукта

База данных состоит из 1 таблицы и 3 отчетов. Основная таблица содержит ключевую ячейку, название учебного учреждения, номер ответа команды, логическую ячейку для проверки правильности ответов на вопросы викторины.

Основой для отчетов являются SQL запросы к базе данных (рисунок 16) с группировкой

по:

- учебным учреждениям;
- номерам вопросов;
- количеству правильных ответов;
- итоговый запрос с подсчетом победителей.

| TeamName | QuestionNumber | Answer | check |
|-------------|----------------|--------|-------------------------------------|
| 2 Пример2 | 10 | | <input checked="" type="checkbox"/> |
| 3 Пример3 | 10 | | <input checked="" type="checkbox"/> |
| 4 Пример4 | 10 | | <input checked="" type="checkbox"/> |
| 5 Пример5 | 10 | | <input type="checkbox"/> |
| 6 Пример6 | 10 | | <input checked="" type="checkbox"/> |
| 7 Пример7 | 10 | | <input type="checkbox"/> |
| 8 Пример8 | 10 | | <input type="checkbox"/> |
| 9 Пример9 | 10 | | <input checked="" type="checkbox"/> |
| 10 Пример10 | 10 | | <input type="checkbox"/> |
| 11 Пример11 | 10 | | <input checked="" type="checkbox"/> |
| (№) | 0 | | <input type="checkbox"/> |

Рисунок 16 — База данных проведения викторины

2.8 Описание программных модулей, используемых в проекте

2.8.1 Windows Forms

Пространство имён (англ. *namespace*) — некоторое множество, под которым подразумевается модель, абстрактное хранилище или окружение, созданное для логической группировки уникальных идентификаторов (то есть имён). Идентификатор, определенный в пространстве имён, ассоциируется с этим пространством. Один и тот же идентификатор может быть независимо определён в нескольких пространствах. Таким образом, значение, связанное с идентификатором, определённым в одном пространстве имён, может иметь (или не иметь) такое же значение, как и такой же идентификатор, определённый в другом пространстве. Языки с поддержкой пространств имён определяют правила, указывающие, к какому пространству имён принадлежит идентификатор (то есть его определение) [10].

При написании программного продукта использовались следующие пространства имен.

- System.Windows.Forms;
- System.Net;
- System.Data.

Ниже будет приведено краткое описание вышеприведенных пространств имен.

Windows Forms — это технология интеллектуальных клиентов для .NET Framework. Она представляет собой набор управляемых библиотек, упрощающих выполнение стандартных задач, таких как чтение из файловой системы и запись в нее. С помощью среды разработки типа Visual Studio можно создавать интеллектуальные клиентские приложения Windows Forms, которые отображают информацию, запрашивают ввод от пользователей и обмениваются данными с удаленными компьютерами по сети [15].

В Windows Forms форма — это видимая поверхность, на которой выводится информация для пользователя. Обычно приложение Windows Forms строится путем помещения элементов управления на форму и написания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш. Элемент управления — это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.

При выполнении пользователем какого-либо действия с формой или одним из ее элементов управления создается событие. Приложение реагирует на эти события с помощью кода и обрабатывает события при их возникновении.

Windows Forms включает широкий набор элементов управления, которые можно добавлять на формы: текстовые поля, кнопки, раскрывающиеся списки, переключатели и даже веб-страницы. Список всех элементов управления, которые можно использовать в форме, представлены в разделе Элементы управления для использования в формах Windows Forms. Если существующий элемент управления не удовлетворяет потребностям, в Windows Forms можно создать пользовательские элементы управления с помощью класса `UserControl` [16].

В состав Windows Forms входят многофункциональные элементы пользовательского интерфейса, позволяющие воссоздавать возможности таких сложных приложений, как Microsoft Office. Используя элементы управления `ToolStrip` и `MenuStrip`, можно создавать панели инструментов и меню, содержащие текст и рисунки, подменю и другие элементы управления, такие как текстовые поля и поля со списками.

С помощью поддерживающего перетаскивание конструктора Windows Forms в Visual Studio можно легко создавать приложения Windows Forms. Достаточно выделить элемент управления курсором и поместить его в нужное место на форме. Для преодоления трудностей, связанных с выравниванием элементов управления, конструктор предоставляет такие средства, как линии сетки и линии привязки. И при использовании Visual Studio, и при ком-

пияции из командной строки вы можете использовать элементы управления FlowLayoutPanel, TableLayoutPanel и SplitContainer для создания сложных макетов форм за меньшее время.

В программном продукте Windows Forms использовалась в качестве каркаса приложения, через неё происходят вызовы форм.

2.8.2 System.Net

Пространство имен System.Net содержит сетевые классы для поиска IP-адресов, сетевой аутентификации, разрешений, отправки и получения данных. Рассмотрим эти классы, рассортировав их по группам [14].

Поиск имен

Чтобы получить IP-адрес из DNS-имени хоста или получить имя хоста из IP-адреса, можно использовать класс Dns. Класс DnsPermission представляет разрешение, необходимое для поиска имени. DnsPermissionAttribute — это класс атрибута, позволяющий отмечать сборки, классы и методы, нуждающиеся в этих полномочиях.

IP-адреса

IP-адреса обрабатываются в классе IPAddress. У одного хоста может быть несколько IP-адресов и алиасов. Вся эта информация содержится в классе IPHostEntry. Когда мы ищем имя, класс Dns возвращает объект типа IPHostEntry.

Аутентификация и авторизация

В классе AuthenticationManager есть статические методы, предназначенные для аутентификации клиентского пользователя. В этом классе-утилите используются модули, реализующие *интерфейс IAuthenticationModule*. Класс AuthenticationManager обращается к этим модулям, чтобы идентифицировать пользователя. Модули аутентификации получают информацию запроса и данные о личности пользователя с помощью

интерфейса *ICredentials* и возвращают объект *Authorization* для авторизованных пользователей, которым разрешается использовать ресурс [31].

Приложение-клиент может передать данные о пользователе на сервер экземпляром класса *NetworkCredential*.

Эти данные о пользователе могут быть занесены в кэш *BCredentialCache* (рисунок 17).

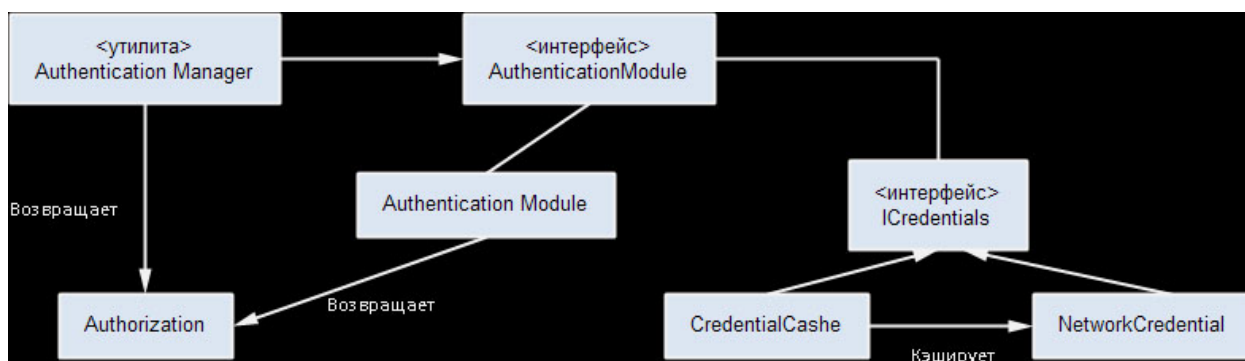


Рисунок 17 — Кэш данных

Управление соединениями

Классы *ServicePoint* и *ServicePointManager* играют важную роль для HTTP-соединений. Для ресурса экземпляр класса *ServicePoint* связан с URI и может обрабатывать несколько соединений. Класс-утилита *ServicePointManager* управляет объектами *ServicePoint*, создавая новые или отыскивая существующие объекты.

Пропускную способность приложения, которое одновременно запрашивает множество данных с сервера, можно повысить, увеличив число соединений на базе приложения. Чтобы ограничить выделение сетевых ресурсов, по умолчанию максимальное число соединений с одним и тем же сервером устанавливается равным двум.

Записи cookies

Cookies — это хранящиеся на стороне клиента наборы данных, которые используются сервером для запоминания некоторой информации между запросами. Когда для запросов данных с сервера используется такой Web-браузер, как Internet Explorer, он сам управляет приемом и сохранением записей cookies и посылает их обратно Web-серверу. При создании специального

приложения, запрашивающего данные с Web-сервера, посылающего записи cookies, можно их считывать в объект класса CookieCollection, возвращаемый свойством Cookies объекта HttpWebResponse.

С помощью класса CookieContainer передаются записи cookie серверу. Сама запись cookie представлена в классе Cookie.

Сокеты

Вместо Web-классов используются классы сокетов, при этом мы, приобретая дополнительные возможности и гибкость, сталкиваемся с определенной сложностью. Большинство классов, которые используются в программировании сокетов, можно найти в пространстве имен System.Net.Sockets.

Программирование сокетов не только позволяет осуществлять связь, ориентированную на соединения, как в случае с HTTP, но также и реализовывать связь без установления соединений, которая используется при групповой рассылке или широковещательной передаче с UDP. Программирование сокетов — чрезвычайно гибкий механизм, позволяющий пользоваться самыми разными протоколами: GGP, ICMP, IGMP, IPX, SPX [20].

Обзор наиболее важных классов из пространства имен System.Net закончен, и теперь более детально рассмотрим некоторые из этих классов.

В программном продукте «Соболёнок» System.Net использовался для связи клиентского приложения с серверным.

2.8.3 ADO.Net

ADO (ActiveX Data Objects) — это библиотека компонентов COM [33], получившая в последние несколько лет множество воплощений. ADO состоит, прежде всего, из объектов Connection, Command, Recordset и Field. С помощью ADO открывается соединение с базой данных, после чего некоторые данные извлекаются и помещаются в набор записей, состоящих из полей; эти данные затем претерпевают манипуляции и обновления на сервере, после че-

го соединение закрывается. Кроме того, ADO предлагает так называемый отключенный набор записей (disconnected record set), который используется, когда соединение с базой нежелательно удерживать открытым в течение длительного времени.

Существует несколько проблем, которые ADO не решает удовлетворительным образом. Наиболее заметная из них — громоздкость (в плане физического размера) отключенного набора записей. Потребность в этом средстве возрастает по мере развития веб-ориентированных вычислений, поэтому в данном случае понадобился свежий подход. Переход от ADO к ADO.NET не должен быть слишком трудным, поскольку между этими технологиями все же имеется некоторое сходство.

Более того, если вы используете SQL Server, существует замечательный набор управляемых классов, которые настроены на обеспечение максимальной производительности базы данных. Одного этого достаточно для перехода на ADO.NET [43].

ADO.NET поставляется с тремя пространствами имен клиента базы данных: одно для SQL Server, другое для источников данных Open Database Connectivity (ODBC) и третье для любой базы данных, доступной через OLE DB. Если выбрана база данных, отличная от SQL Server, отдавайте предпочтение OLE DB, если только не окажется, что нет другого выбора кроме ODBC.

В программном продукте «Соболёнок» ADO.Net использовалась в качестве связующего компонента между сервером и клиентским приложением.

2.9 Аппаратно-программные требования

Система проведения викторины предполагает наличие персонального компьютера с минимальными системными требованиями для работы с приложением:

- процессор Core 2 Duo 2.0 ГГц;

- объем оперативной памяти – 256 Мб;
- место на накопителе – 12 Мб;
- скорость чтения и записи накопителя – 10 Мбит;
- видеоадаптер – 128 Мб с поддержкой разрешения 1024x768;
- монитор;
- манипулятор – мышь;
- клавиатура.

Минимальные требования к программному обеспечению:

- операционная система Windows XP или выше;
- установленный пакет библиотек Visual C++ 2008.

Программный комплекс был разработан с использованием объектно-ориентированной среды программирования Visual Studio и языка C#.

2.10 Инструкции по работе с программным продуктом для администратора

Для того чтобы начать работать с программным продуктом необходимо выполнить следующую последовательность действий:

- 1) открыть архив с названием Sabel.rar с компакт-диска;
- 2) распаковать архив в любую папку, не содержащую русских символов на компьютере;
- 3) открыть папку с распакованным дистрибутивом и установить файл dotNetFx35setup.exe;
- 4) проследовать указаниям установочного приложения;
- 5) после завершения установки библиотек .NET Framework запустить исполняемый файл SableServer.exe;
- 6) запуск будет сопровождаться загрузочным окном и если всё сделано верно, то запустится серверное приложение, в котором необходимо найти верхнюю панель с кнопками;
- 7) пользоваться системой;

8) для сохранения текущего состояния проведённой викторины необходимо нажать на кнопку «Сохранить базу данных»;

9) для вывода отчета необходимо кликнуть на раскрывающееся меню кнопки «отчеты», затем раскрыть щелчком мыши нужный отчет.

В том случае, если сервер не запускается, то обычно это сопровождается сообщениями, которые информируют пользователя о невозможности запуска конкретного модуля, чтобы избежать ошибок запуска, необходимо проделать следующее:

- 1) перезапустить серверное приложение;
- 2) отключить или настроить брандмауэр (firewall);
- 3) не пытаться запустить приложение в виртуальной среде;
- 4) попробовать отключить антивирус или добавить всю папку в список исключений;
- 5) проверить путь до каталога приложения, он не должен содержать русских символов;
- б) повторить запуск сервера.

2.11 Инструкции по работе с программным продуктом для пользователя

Для того, чтобы начать работать с системой регистрации ответов на вопросы викторины необходимо выполнить следующую последовательность действий:

- 1) узнать домен или адрес сервера системы тестирования получив их у администратора;
- 2) перейдя во вкладку «меню», кликнуть на пункт меню «настройки подключения»;
- 3) в предложенном окне ввести полученные от администратора данные о подключении в соответствующие поля «IP адрес сервера», «порт сервера»;

4) в предложенном окне нажать на кнопку текущего вопроса викторины;

5) в следующем предложенном окне ввести в текстовое поле ответ на данный вопрос викторины.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы был проведен анализ научной и учебной литературы, была изучена специфика программных продуктов для компьютерного сопровождения интеллектуальной викторины и осуществлена программная реализация системы проведения интеллектуальной викторины средствами объектно-ориентированной среды программирования Visual Studio на языке C#.

В результате были разработаны:

- серверное приложение;
- клиентское приложение;
- инструкции для пользователя;
- инструкции для администратора;
- аппаратно-программные требования.

В результате выполнения выпускной квалификационной работы создана система регистрации прохождения викторины «Sabel». Данная система разделена на два приложения: клиентское приложение «SabelClient», позволяющая клиенту выбирать текущий вопрос викторины, и давать на него ответ пользуясь текстовым полем и серверное приложение «SabelServer» позволяющее организаторам викторины отслеживать ход проведения викторины, ответы учащихся, и формировать отчеты на основе собранной в базе данных информации. Данный программный продукт предназначен для проведения интеллектуальной викторины «Соболёнок» среди учащихся или персонала и имеет значительные преимущества по сравнению с ранее используемыми системами. А именно наличием клиент-серверного взаимодействия, возможностью подключения баз данных, возможностью управления этими базами данных, а также обособленностью самого приложения, в сравнении с предыдущим методом проведения данной викторины.

Ранее использовавшийся текстовый процессор Google docs не соответствовал данным требованиям, что и являлось причиной написания данной выпускной квалификационной работы.

В результате выполнения работы решены следующие задачи:

1. Проведён анализ нормативно-справочной документации, материала, а также интернет источников по теме исследования.
2. Определена архитектура и средства реализации программного приложения.
3. Спроектирована и реализована серверная часть приложения.
4. Спроектирована и реализована клиентская часть приложения.
5. Составлена сопровождающая данный продукт документация

В рамках проектирования выпускной квалификационной работы цель исследования была достигнута, задачи решены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГСС РФ. Основные положения [Электронный ресурс]. – Режим доступа: http://ohranatruda.ru/ot_biblio/normativ/data_normativ/4/4787/ (дата обращения: 16.05.2016).

2. ГОСТ Р ИСО 9241-11-2010 - Федеральное агентство по техническому регулированию и метрологии [Электронный ресурс]. – Режим доступа: http://standard.gost.ru/wps/wcm/connect/76126680413f5d06a4e0fe7ab9890bef/GOST_R_ISO_9241-11-2010 (дата обращения: 16.05.2016).

3. ГОСТ 34.601-90 АС. Стадии создания [Электронный ресурс]. Режим доступа: http://dit.mos.ru/legislation/methods/tekhnicheskaya_politika/590462/ (дата обращения: 16.05.2016).

4. ГОСТ 34.602-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы [Электронный ресурс]. – Режим доступа: <http://webportalsrv.gost.ru/portal/katalog.nsf/3bf603832bf0c306c225712900445930/74d46fa6b649f9b5c32563fc0049f97c?OpenDocument> (дата обращения: 16.05.2016).

5. ГОСТ 34.201-89 Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем [Электронный ресурс]. – Режим доступа: <http://webportalsrv.gost.ru/portal/katalog.nsf/3bf603832bf0c306c225712900445930/218dd19fe354042ec32563fc0049f8e4?OpenDocument> (дата обращения: 16.05.2016).

6. РД 50-34.698-90 Автоматизированные системы. Требования к содержанию документов. [Электронный ресурс]. – Режим доступа: http://www.rugost.com/index.php?option=com_content&view=article&id=98:50-34698-90&catid=22&Itemid=53 (дата обращения: 16.05.2016).

7. ГОСТ 28195-89 Оценка качества программных средств. Общие положения [Электронный ресурс]. – Режим доступа: <http://webportalsrv.gost.ru/portal/katalog.nsf/3bf603832bf0c306c225712900445930/498f5d34b0f832c3c32563fc004fcf47?OpenDocument> (дата обращения: 16.05.2016).

8. ГОСТ 34.603-92 Информационная технология. Виды испытаний автоматизированных систем [Электронный ресурс]. – Режим доступа: <http://webportalsrv.gost.ru/portal/katalog.nsf/3bf603832bf0c306c225712900445930/6abb32a3e0087d6bc32563fc0049f98d?OpenDocument> (дата обращения: 16.05.2016).

9. ГОСТ 28806-90 Качество программных средств. Термины и определения [Электронный ресурс]. – Режим доступа: <http://webportalsrv.gost.ru/portal/katalog.nsf/3bf603832bf0c306c225712900445930/88834f268a091842c32563fc004ffd40?OpenDocument> (дата обращения: 16.05.2016).

10. Джон Скит. С# для профессионалов: тонкости программирования. 3-е издание, новый перевод = С# in Depth, 3rd ed. [Текст]: учебник/ – М.: «Вильямс», 2014. — 608 с.

11. Заочные электронные конференции [Электронный ресурс]. – Режим доступа: <http://econf.rae.ru/article/6870> (дата обращения: 16.05.2016).

12. Киселёв Г.М. Информационные технологии в педагогическом образовании [Текст]: учебник / Г.М. Киселёв, Р.В. Бочкова – М.: Дашков и К, 2013. – 308 с.

13. Культин Н.Б. Основы программирования в среде Delphi XE [Текст]: учебник / Н.Б. Культин М.: «БХВ-Петербург», 2011. – 416 с.

14. Кристиан Нейгел и др. С# 5.0 и платформа .NET 4.5 для профессионалов = Professional C# 5.0 and .NET 4.5. [Текст]: учебник/ – М.: «Диалектика», 2013. — 1440 с.
15. Мастера С# [Электронный ресурс]. – Режим доступа: <http://www.charpmaster.ru/> (дата обращения: 12.06.2016).
16. Мастера С# [Электронный ресурс]. – Режим доступа: <http://www.charpmaster.ru/articles/intraweb/> (дата обращения: 20.05.2016).
17. Осипов Д.Л. Delphi XE2 [Текст]: учебник / Д.Л. Осипов – СПб.: «БХВ-Петербург», 2012. – 912 с.
18. Портал «Знать» [Электронный ресурс]. – Режим доступа: <http://www.znannya.org/?view=elearning-tools-6-2-2> (дата обращения: 01.05.2016).
19. Программирование на С и С++ [Электронный ресурс]. – Режим доступа: <http://cpp.com.ru/> (дата обращения: 01.05.2016).
20. Рубанцев В.Д. Большой самоучитель Delphi XE3 [Текст]: учебник / В. Рубанцев – М.: «Я + R», 2012. – 1274 с.
21. Руководство по PHP [Электронный ресурс]. – Режим доступа: <https://secure.php.net/manual/ru/index.php> (дата обращения: 01.05.2016).
22. РОССТАНДАРТ Федеральное агентство по техническому регулированию и метрологии [Электронный ресурс]. – Режим доступа: <http://www.gost.ru/wps/portal/> (дата обращения: 20.06.2016).
23. Современное программирование на языке Delphi [Электронный ресурс]. – Режим доступа: <http://pascalabc.net/downloads/pabcnethelp/> (дата обращения: 01.05.2016).
24. Современный учебник JavaScript [Электронный ресурс]. – Режим доступа: <https://learn.javascript.ru/> (дата обращения: 06.04.2016).
25. Справочное руководство по MySQL [Электронный ресурс]. – Режим доступа: <http://www.mysql.ru/docs/man/Reference.html> (дата обращения: 10.01.2016).

26. Стандарты ГОСТ Р ИСО – базовая система стандартизации в РФ [Электронный ресурс]. – Режим доступа: <http://www.rospromtest.ru/content.php?id=256>

27. Федеральный государственный образовательный стандарт среднего общего образования [Электронный ресурс]. – Режим доступа: <http://минобрнауки.рф/документы/2365> (дата обращения: 03.06.2016).

28. Хабрахабр [Электронный ресурс]. – Режим доступа: <https://habrahabr.ru/post/144242/> (дата обращения: 03.06.2016).

29. Шарипов Ф. В. Педагогика и психология высшей школы [Текст]: учебник / Ф. В. Шарипов – М.: Логос, 2012. – 448 с.

30. Язык программирования C#. Классика Computers Science. 4-е издание. [Текст]: учебник / Хейлсберг А., Торгерсен М., Вилтамут С., Голд П. – СПб.: «Питер», 2011– 786 с.

31. AlphaControls [Электронный ресурс]. – Режим доступа: <http://www.alphaskins.com/> (дата обращения: 20.05.2016).

32. Apache HTTP Server [Электронный ресурс]. – Режим доступа: <https://httpd.apache.org/docs/2.4/> (дата обращения: 05.01.2016).

33. Axialis [Электронный ресурс]. – Режим доступа: <http://www.axialis.com/stock-icons/> (дата обращения: 01.02.2016).

34. CSS справочник [Электронный ресурс]. – Режим доступа: <http://css.manual.ru/> (дата обращения: 03.12.2015).

35. C# Expert [Электронный ресурс]. – Режим доступа: <http://www.csharpexpert.ru/> (дата обращение 15.06.2016).

36. Design Mania [Электронный ресурс]. – Режим доступа: <http://design-mania.ru/web-design/articles/yakob-nilsen/> (дата обращения: 03.06.2016).

37. HelpDelphi [Электронный ресурс]. – Режим доступа: <http://helpdelphi.ru/pagecontrol/> (дата обращения: 05.05.2016).

38. HTML справочник [Электронный ресурс]. – Режим доступа: <http://htmlbook.ru/> (дата обращения: 03.04.2016).

39. IconArchive [Электронный ресурс]. – Режим доступа: <http://www.iconarchive.com/> (дата обращения: 01.05.2016).

40. JQuery [Электронный ресурс]. – Режим доступа: <https://jquery.com/download/> (дата обращения: 22.05.2016).

41. MSDN Справочник по C# [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru-ru/library/618ayhy6.aspx> (дата обращения: 05.05.2016).

42. MSDN Руководство по программированию C# [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx> (дата обращения: 05.05.2016).

43. MSDN ADO.NET [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru-ru/library/e80y5yhx%28v=vs.110%29.aspx> (дата обращения: 05.05.2016).

44. Nevlabs разработка ПО [Электронный ресурс]. – Режим доступа: <http://nevlabs.ru/about/articles/web/php-vs-aspnet/> (дата обращения: 01.05.2016).

45. SQL.ru – всё про SQL [Электронный ресурс]. – Режим доступа: <http://www.sql.ru/forum/742162/intraweb-webbroker-websnap> (дата обращения: 05.05.2016).

46. Professor web – ADO.NET [Электронный ресурс]. – Режим доступа: http://professorweb.ru/my/ADO_NET/base/level1/ado_net_index.php (дата обращения: 05.05.2016).

47. W3Schools [Электронный ресурс]. – Режим доступа: <http://www.w3schools.com/> (дата обращения: 21.05.2016).

ПРИЛОЖЕНИЕ 1

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

«Российский государственный профессионально-педагогический университет»

Институт инженерно-педагогического образования
Кафедра информационных систем и технологий
направление 44.03.04 Профессиональное обучение (по отраслям)
профиль «Информатика и вычислительная техника»
профилизация «Компьютерные технологии»

УТВЕРЖДАЮ
Заведующий кафедрой
_____ Н. С. Толстова
« ____ » _____ 2016 г.

ЗАДАНИЕ
на выполнение выпускной квалификационной работы бакалавра

студента 4 курса, группы ПУ – 411 оКТ Палехова Данила Александровича

1. Тема Программное обеспечение интерактивной интеллектуальной викторины утверждена распоряжением по институту от 28.03.2016 г. № 57
2. Руководитель Н.С. Толстова, канд. пед.наук, доцент каф. ИС
3. Место преддипломной практики РГППУ «Российский государственный профессионально-педагогический университет», г.Екатеринбург
4. Исходные данные к ВКР
Джон Скит. С# для профессионалов: тонкости программирования. 3-е издание, новый перевод = C# in Depth, 3rd ed. [Текст]: учебник/ – М.: «Вильямс», 2014. — 608 с.
Заочные электронные конференции [Электронный ресурс]. – Режим доступа: <http://econf.rae.ru/article/6870> (дата обращения: 16.05.2016). Киселёв Г.М. Информационные технологии в педагогическом образовании [Текст]: учебник / Г.М. Киселёв, Р.В. Бочкова – М.: Дашков и К, 2013. – 308 с.
5. Содержание текстовой части ВКР (перечень подлежащих разработке вопросов)
 - Провести анализ нормативно-справочной документации, материала, а также интернет источников по теме исследования.
 - Определить архитектуру и средства реализации программного приложения.
 - Спроектировать и реализовать серверную часть приложения.
 - Спроектировать и реализовать клиентскую часть приложения.
6. Перечень демонстрационных материалов
Презентация выполненная в Microsoft PowerPoint

7. Календарный план выполнения выпускной квалификационной работы

| № п/п | Наименование этапа дипломной работы | Срок выполнения этапа | Процент выполнения ВКР | Отметка руководителя о выполнении |
|-------|------------------------------------------------------------------------------------------------------------------|-----------------------|------------------------|-----------------------------------|
| 1 | Сбор информации по выпускной работе и сдача зачета по преддипломной практике | 28.03.2016 | 15 % | |
| 2 | Выполнение работ по разрабатываемым вопросам их изложение в выпускной работе: | | 65 % | |
| | Провести анализ нормативно-справочной документации, материала, а также интернет источников по теме исследования. | 11.04.2016 | 10 % | |
| | Определить архитектуру и средства реализации программного приложения. | 15.04.2016 | 10 % | |
| | Спроектировать и реализовать серверную часть приложения. | 10.05.2016 | 20 % | |
| | Спроектировать и реализовать клиентскую часть приложения. | 18.05.2016 | 25 % | |
| 3 | Оформление текстовой части ВКР | 26.05.2016 | 5 % | |
| 4 | Выполнение демонстрационных материалов к ВКР | 04.06.2016 | 5 % | |
| 5 | Нормоконтроль | 08.06.2016 | 5 % | |
| 6 | Подготовка доклада к защите в ГЭК | 10.06.2016 | 5 % | |

8. Консультанты по разделам выпускной квалификационной работы

| Наименование раздела | Консультант | Задание выдал | | Задание принял | |
|----------------------|-------------|---------------|------|----------------|------|
| | | подпись | дата | подпись | дата |
| | | | | | |
| | | | | | |
| | | | | | |

Руководитель _____
подпись дата

Задание получил _____
подпись студента дата

9. Выпускная квалификационная работа и все материалы проанализированы. Считаю возможным допустить Палехова Д.А. к защите выпускной квалификационной работы в государственной экзаменационной комиссии.

Руководитель _____
подпись дата

10. Допустить Палехова Д.А. к защите выпускной квалификационной работы в государственной экзаменационной комиссии (протокол заседания кафедры от «08» июня 2016 г., № 15/10-02)

Заведующий кафедрой _____
подпись дата

ПРИЛОЖЕНИЕ 2

Модуль кода серверного приложения, отвечающий за регистрацию участников викторины:

```
private void OnAccept(IAsyncResult ar)
{
    try
    {
        Socket clientSocket = serverSocket.EndAccept(ar);

        //Start listening for more clients
        serverSocket.BeginAccept(new AsyncCallback(OnAccept), null);

        //Once the client connects then start receiving the commands from her
        clientSocket.BeginReceive(byteData, 0, byteData.Length, SocketFlags.None,
            new AsyncCallback(OnReceive), clientSocket);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "SableServer",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Модуль кода серверного приложения, отвечающий за принятие ответов участников викторины:

```
private void OnAccept(IAsyncResult ar)
{
    try
    {
        Socket clientSocket = serverSocket.EndAccept(ar);

        //Start listening for more clients
        serverSocket.BeginAccept(new AsyncCallback(OnAccept), null);

        //Once the client connects then start receiving the commands from her
        clientSocket.BeginReceive(byteData, 0, byteData.Length, SocketFlags.None,
            new AsyncCallback(OnReceive), clientSocket);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "SableServer",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void OnReceive(IAsyncResult ar)
{
    try
    {
        Socket clientSocket = (Socket)ar.AsyncState;
        clientSocket.EndReceive(ar);

        //Transform the array of bytes received from the user into an
```

```

//intelligent form of object Data
Data msgReceived = new Data(byteData);

//We will send this object in response the users request
Data msgToSend = new Data();

byte [] message;

//If the message is to login, logout, or simple text message
//then when send to others the type of the message remains the same
msgToSend.cmdCommand = msgReceived.cmdCommand;
msgToSend.strName = msgReceived.strName;

switch (msgReceived.cmdCommand)
{
    case Command.Login:

        //Добавление клиента в список команд

        ClientInfo clientInfo = new ClientInfo();
        clientInfo.socket = clientSocket;
        clientInfo.strName = msgReceived.strName;

        clientList.Add(clientInfo);

        //Set the text of the message that we will broadcast to all users
        msgToSend.strMessage = "Команда " + msgReceived.strName + " при-
соединилась к серверу";
        listOfTeams.Items.Add(msgReceived.strName);
        break;

    case Command.Logout:

        //When a user wants to log out of the server then we search for
        //in the list of clients and close the corresponding connection

        int nIndex = 0;
        foreach (ClientInfo client in clientList)
        {
            if (client.socket == clientSocket)
            {
                clientList.RemoveAt(nIndex);
                break;
            }
            ++nIndex;
        }

        clientSocket.Close();

        msgToSend.strMessage = "Команда " + msgReceived.strName + " отсо-
единилась от сервера";
        listOfTeams.Items.Remove(msgReceived.strName);
        break;

    case Command.Message:

        //Set the text of the message that we will broadcast to all users
        msgToSend.strMessage = msgReceived.strName + ": " + msgRe-
ceived.strMessage;

        listOfTeams.Items.AddRange(msgReceived.strMessage.Split('*'));
        listOfTeams.Items.Remove(listOfTeams.Items.Count - 1);
        string TeamName = msgReceived.strName;
        string AnswerText = msgReceived.strMessage;
        string[] qNumber = AnswerText.Split(':');
        string QuestionNumber = qNumber[0];

```

```

        string Answer = qNumber[1];
        dataGridView1.Rows.Add(TeamName, QuestionNumber, Answer);
        break;

    case Command.List:

        //Send the names of all users in the chat room to the new user
        msgToSend.cmdCommand = Command.List;
        msgToSend.strName = null;
        msgToSend.strMessage = null;

        //Collect the names of the user in the chat room
        foreach (ClientInfo client in clientList)
        {
            //To keep things simple we use asterisk as the marker to sep-
            arate the user names
            msgToSend.strMessage += client.strName + "*";
        }

        message = msgToSend.ToByte();

        //Send the name of the users in the chat room
        clientSocket.BeginSend(message, 0, message.Length, Socket-
Flags.None,
            new AsyncCallback(OnSend), clientSocket);
        break;
    }

    if (msgToSend.cmdCommand != Command.List) //List messages are not
    broadcasted
    {
        message = msgToSend.ToByte();

        foreach (ClientInfo clientInfo in clientList)
        {
            if (clientInfo.socket != clientSocket ||
                msgToSend.cmdCommand != Command.Login)
            {
                clientInfo.socket.BeginSend(message, 0, message.Length, Sock-
etFlags.None,
                    new AsyncCallback(OnSend), clientInfo.socket);
            }
        }

        txtLog.Text += msgToSend.strMessage + "\r\n";
    }

    //If the user is logging out then we need not listen from her
    if (msgReceived.cmdCommand != Command.Logout)
    {
        //Start listening to the message send by the user
        clientSocket.BeginReceive(byteData, 0, byteData.Length, Socket-
Flags.None, new AsyncCallback(OnReceive), clientSocket);
    }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "SableServer", MessageBoxButtons.OK, Message-
BoxIcon.Error);
}
}
}

```

Модуль кода программного продукта обеспечения интерактивной интеллектуальной викторины, в котором осуществляется общение между серверным приложением и клиентским приложением:

```
class Data
{
    //Default constructor
    public Data()
    {
        this.cmdCommand = Command.Null;
        this.strMessage = null;
        this.strName = null;
    }

    //Converts the bytes into an object of type Data
    public Data(byte[] data)
    {
        //The first four bytes are for the Command
        this.cmdCommand = (Command)BitConverter.ToInt32(data, 0);

        //The next four store the length of the name
        int nameLen = BitConverter.ToInt32(data, 4);

        //The next four store the length of the message
        int msgLen = BitConverter.ToInt32(data, 8);

        //This check makes sure that strName has been passed in the array of bytes
        if (nameLen > 0)
            this.strName = Encoding.UTF8.GetString(data, 12, nameLen);
        else
            this.strName = null;

        //This checks for a null message field
        if (msgLen > 0)
            this.strMessage = Encoding.UTF8.GetString(data, 12 + nameLen, msgLen);
        else
            this.strMessage = null;
    }

    //Converts the Data structure into an array of bytes
    public byte[] ToByte()
    {
        List<byte> result = new List<byte>();

        //First four are for the Command
        result.AddRange(BitConverter.GetBytes((int)cmdCommand));

        //Add the length of the name
        if (strName != null)
            result.AddRange(BitConverter.GetBytes(strName.Length));
        else
            result.AddRange(BitConverter.GetBytes(0));

        //Length of the message
        if (strMessage != null)
            result.AddRange(BitConverter.GetBytes(strMessage.Length));
        else
            result.AddRange(BitConverter.GetBytes(0));

        //Add the name
        if (strName != null)
            result.AddRange(Encoding.UTF8.GetBytes(strName));
    }
}
```

```
//And, lastly we add the message text to our array of bytes
if (strMessage != null)
    result.AddRange(Encoding.UTF8.GetBytes(strMessage));

return result.ToArray();
}
```