**Исследование современных методов, способов и средств решения проблемы автоматизации информационных бизнес-процессов**

2.1 **Основные способы и технико-технологические решения автоматизации информационных бизнес-процессов**

На современном уровне развития автоматизация процессов представляет собой один из подходов к управлению процессами на основе применения информационных технологий. Этот подход позволяет осуществлять управление операциями, данными, информацией и ресурсами за счет использования компьютеров и программного обеспечения, которые сокращают степень участия человека в процессе, либо полностью его исключают. Основной целью автоматизации является повышение качества исполнения процесса. Автоматизированный процесс обладает более стабильными характеристиками, чем процесс, выполняемый в ручном режиме. Во многих случаях автоматизация процессов позволяет повысить производительность, сократить время выполнения процесса, снизить стоимость, увеличить точность и стабильность выполняемых операций.

На сегодняшний день автоматизация процессов охватила многие отрасли промышленности и сферы деятельности: от производственных процессов, до совершения покупок в магазинах. Вне зависимости от размера и сферы деятельности организации, практически в каждой компании существуют автоматизированные процессы. Соответственно, имеет место быть большое разнообразие способов, методов и технико-технологических решений по автоматизации тех или иных бизнес-процессов.

В данном дипломном проекте способом автоматизации информационных бизнес-процессов является применение автоматизированной информационной системы.

2.2 **Обоснование выбора способа технической реализации (приобретение типового готового решения без доработок, кастомизация/локализация готового решения, разработка нового решения)**

Существует несколько способов технической реализации автоматизации бизнес-процессов:

* Приобретение типового готового решения без доработок;
* Приобретение типового готового решения с дальнейшей кастомизацией/локализацией;
* Разработка нового решения.

Первый вариант, связанный с приобретением готового решения без доработок, не представляется возможным. Изучив рынок готовых решений, не нашлось того готового программного обеспечения, которое бы соответствовало всем требованиям заказчика.

Есть так же возможность видоизменить существующие решения под нужды заказчика. Для достижения этой цели существует 2 метода модернизации готового программного обеспечения под поставленные задачи, а именно модернизация и кастомизация.

Локализация программного обеспечения - процесс адаптации программного обеспечения к культуре какой-либо страны. Как частность — перевод пользовательского интерфейса, документации и сопутствующих файлов программного обеспечения с одного языка на другой. Это многоуровневая операция, которая требует взаимодействие программистов, дизайнеров и переводчиков. В ней присутствуют такие шаги, как:

* Обеспечить поддержку языка и национальных стандартов - необходимый минимум, чтобы программа могла выполнять свои функции в другой стране;
  + Корректная работа в локализированной операционной системе;
  + Соответствие товарным законам целевой страны;
  + Вывод на экран символов языка, подготовка локализированных шрифтов;
* Перевод текстов в интерфейсе программы на целевой язык;
  + Корректная расположение элементов интерфейса, с учетом того, что разные языки имеют существенно разные размеры;
  + Перевод терминологий;
  + Работа с изображениями, если они содержат текст;
* Тонкая настройка под целевую страну
  + Работа со словоформами;
  + Работы с дополнительными стандартами, не связанными с основной функциональностью программы, такими как дата и время, особенности типографики;
  + Корректировка клипарт-библиотек. Например, добавление картинок местных праздников;
  + Перерисовка графики под реалии другой страны.

Кастомизация – это индивидуализация продукции под заказы конкретных потребителей путем внесения конструктивных или дизайнерских решений. Её основная задача – создать у потребителя ощущение, что работа делается именно для него и нацелена на удовлетворение его конкретных потребностей. Основным инструментом, используемым для кастомизации и улучшения взаимодействия между производителем и потребителем, являются CRM системы (системы управления взаимоотношения с клиентом).

Исходя из вышеопределенных способов видоизменения программного продукта под нужды заказчика, можно прийти к выводу, что второй вариант с применением локализации и кастомизации готового решения был так же не является лучшим выбором.

Проанализировав большое количество готовых решений, был выявлен ряд минусов и несоответствий с потребностями заказчика, а именно:

* Функционал выбранного решения должен строго отвечать поставленным заказчиком требованиям, а все готовые решения, которые хотя бы отдаленно подходили под поставленные задачи, обладают крайне большим функционалом, выходящим за рамки требований и нужд заказчика, и который излишен для задач, поставленных преподавателем студентам в ходе образовательных процессов по дисциплинам «Программирование микроконтроллеров» и «Проектирование и разработка систем на микроконтроллерах». Так же это поднимает порог вхождения для использования выбранного решения студентами, что негативно скажется на их успеваемости;
* Не было найдено такое решение, которое бы даже после кастомизации, удовлетворяло бы всем требованиям. На рынке имеется ряд готовых, качественных решений по работе с микроконтроллерной подсистемой и большое количество решений по работе с базами данных, но не было найдено то, которое бы объединяло эти две сферы.
* Высокие финансовые затраты на переработку выбранного программного решения;
* Сложность в видоизменении программного продукта под поставленные нужды;
* Большие временные затраты, которые не укладываются в выделенное на дипломный проект время;

Для данного диплома было выбрано решение разработки нового программного решения. В пользу данного решения определен ряд следующих причин:

* Функционал данного программного обеспечения четко соответствует требованиям, предписанным заказчиком;
* Количество функций сведено к минимуму и отвечает всем задачам, которые стоят перед студентами в рамках соответствующих дисциплин;
* Минимальный порог вхождения для использования полученного программного обеспечения;
* Бесплатность данного решения;
* Универсальность полученного программного обеспечения;
* Полученное программное обеспечения имеет открытый исходный код, что дает возможность дальнейшего развития данного проекта студентами, оптимизацию ими существующих функций и добавление новых.

2.3 **Сравнительный анализ технико-технологических решений автоматизации информационных бизнес-процессов**

На текущий момент существует гигантское количество средств, с помощью которых можно достигнуть автоматизации для любого бизнес процесса.

Поскольку программное обеспечение данного дипломного проекта является нативным, то можно провести сравнительный анализ самых популярных языков разработки для десктопных приложений, используемых на данный момент.

2.3.1 **Сравнение языков разработки**

Существует большое количество языков программирования. Все они являются инструментом разработчика для решения тех или иных задач. На сегодняшний день любой язык программирования является практически универсальным инструментом благодаря различного рода дополнениям, плагинам, расширениям и библиотекам, но каждый из них особенно силен в определенных задачах, так как изначально был разработан именно для них. На основании того, что в данной работе речь идет о нативных приложениях под операционную систему Windows, рассмотрим 3 самых популярных языка программирования на данный момент. Это «C++», «C#», «Python».

2.3.1.1 **C++**

C++ - компилируемый, статически типизированный язык программирования общего назначения. Поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков. Он широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр. Синтаксис C++ унаследован от языка C.

В C++ доступны следующие встроенные типы:

* Символьные: char, wchar\_t;
* Целочисленные знаковые: signed char, short int, int, long int;
* Целочисленные беззнаковые: unsigned char, unsigned short int, unsigned int, unsigned long int;
* С плавающей точкой: float, double, long double;
* Логический: bool, имеющий значения true или false.

К особенностям С++ можно отнести следующие элементы:

* совместимость С++ с языком программирования С достаточно велика;
* вычислительная производительность;
* поддержка стилей программирования;
* вызов деструкторов автоматизирован;
* перегрузка операторов;
* управление константностью объектов;
* шаблоны;
* встраивание предметно-ориентированных языков программирования;
* доступность.

C ++ широко используется в разработке программного обеспечения для встраиваемых систем. Он также популярен в коммуникациях и играх. Одна из причин, по которой программисты выбирают C ++, заключается в том, что он хорошо взаимодействует с другими языками. Еще один плюс в том, что это высокая производительность.

C ++ регулярно используется в бэкэнде, когда организации достаточно велики, чтобы оправдать инвестиции как в горизонтальное, так и в вертикальное масштабирование. Разработчики в средних компаниях часто недооценивают вертикальное и горизонтальное масштабирование, но оба имеют свою полезность. Горизонтальное масштабирование относится к процессу простого добавления большего количества оборудования для обработки пользовательской базы, где вертикальное масштабирование относится к процессу использования более качественного оборудования или более эффективного использования существующего оборудования для обеспечения роста пользователей. Для проектов малого и среднего размера и пользовательских баз акцент делается на проверке возможностей и минимизации затрат на разработку.

2.3.1.2 **Python**

Python - высокоуровневый язык программирования общего назначения с динамической строгой типизацией и автоматическим управлением памятью, ориентированный на повышение производительности разработчика, читаемости кода и его качества, а также на обеспечение переносимости написанных на нём программ. Язык является полностью объектно-ориентированным — всё является объектами.

Python является мультипарадигмальным языком программирования, поддерживающим императивное, процедурное, структурное, объектно-ориентированное программирование и функциональное программирование. Задачи обобщённого программирования решаются за счёт динамической типизации.

Стандартная библиотека включает большой набор полезных переносимых функций, начиная от функционала для работы с текстом и заканчивая средствами для написания сетевых приложений. Дополнительные возможности, такие как математическое моделирование, работа с оборудованием, написание веб-приложений или разработка игр, могут реализовываться посредством обширного количества сторонних библиотек, а также интеграцией библиотек, написанных на Си или C++, при этом и сам интерпретатор Python может интегрироваться в проекты, написанные на этих языках.

Наличие библиотек также гарантирует то, что нет необходимости писать весь код самостоятельно, и можно импортировать его из тех алгоритмов, которые уже существуют в библиотеках.

Python стал одним из самых популярных языков, он используется в анализе данных, машинном обучении, DevOps и веб-разработке, а также в других сферах, включая разработку игр. За счёт читабельности, простого синтаксиса и отсутствия необходимости в компиляции язык хорошо подходит для обучения программированию, позволяя концентрироваться на изучении алгоритмов, концептов и парадигм.

К особенностям Python можно отнести следующие элементы:

* имеет открытый исходный код;
* Высоко динамичный язык;
* Поддержка GUI;
* Объектно-ориентированный подход;
* Обширный набор библиотек;
* Порто тируемый язык;
* Поддержка других языков программирования.

2.3.1.3 **C# и .NET Framework**

Представляет из себя объектно-ориентированный язык программирования, который был разработан в 1998 году группой разработчиков, под руководством Андерса Хейлсберга, специалиста в компании Microsoft. С# позиционируется как язык разработки, применимый для разработки приложений на платформе Microsoft.NET. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java.

C# разрабатывался как язык программирования прикладного уровня для исполняющей среды для байт-кода (CLR) и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает стандартная библиотека классов платформы «.NET Framework» (BCL). Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др.

Язык успешно реализует возможность работы со следующими элементами разработки:

* + статическая типизация;
  + полиморфизм;
  + перегрузка операторов;
  + делегаты;
  + атрибуты;
  + события;
  + свойства;
  + итераторы;
  + анонимные функции;
  + исключения.

Существует несколько реализаций C#:

* Реализация C# в виде компилятора csc.exe включена в состав .NET Framework (включая .NET Micro Framework, .NET Compact Framework и его реализации под Silverlight и Windows Phone 7);
* В составе проекта Rotor (Shared Source Common Language Infrastructure) компании Microsoft;
* Проект Mono включает в себя реализацию C# с открытым исходным кодом.

Таблица 1 – Сравнительный анализ языков программирования

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | C# | Python | C++ |
| Кроссплатформенность | Да | Да | Да |
| Простота использования | Да | Да | Нет |
| Лицензионные условия приобретения и использования | Бесплатный | Бесплатный | Бесплатный |
| Наличие библиотек для работы с графикой и мультимедиа | Да | Да | Да |
| Порог вхождения для студентов | Низкий | Низкий | Высокий |
| Количество различных компонент, готовых к использованию | Большое | Большое | Большое |

Для данного дипломного проекта был выбран язык программирования C#. Основополагающей причиной был тот факт, что данный дипломный проект разрабатывался под операционную систему Windows, а язык C# является продуктом компании Windows, что означает постоянную поддержку со стороны Microsoft. Так же данный язык программирования, согласно исследованиям на IT рынке, входит пятерку самых популярных языков программирования, что предполагает большое количество разного рода расширений и плагинов для тех или иных задач. Так же это означает, что C# обладает большим сообществом единомышленников, что способствует появлению огромного количества информации на просторах интернета, которая может помочь с решением сложностей и проблем, возникающих на стадии разработки. Так же его популярность, как в прочем и популярность операционной системы Windows, с которой C# связан, положительно влияет на востребованность студентов на рынке труда.

2.3.2 **Сравнение сред разработки (IDE)**

Для языка программирования C# существует не малое количество сред разработки, от простых редакторов кода до полноценных сред разработки.

Рассмотрим самые популярные и востребованные из них.

2.3.2.1 **MS** **Visual Studio**

Интегрированная среда программирования MS Visual Studio – полнофункциональная и расширяемая интегрированная среда разработки для создания современных приложений Android, iOS и Windows, а также веб-приложений и облачных служб. В Visual Studio возможно программирование на C#, Visual Basic, F#, C++, HTML, JavaScript, TypeScript, Python, PHP. Являясь расширяемой, она может быть средой программирования для любого языка. Среда программирования Visual Studio даёт возможность для разработки веб-приложений с помощью ASP.NET, Node.js, Python и JavaScript. В распоряжении мощные веб-платформы, такие как AngularJS, jQuery, Bootstrap, Django и Backbone.js. MS Visual Studio предоставляет управление исходным кодом в репозиториях Git, размещенных любым поставщиком, например: GitHub.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода (как, например, Subversion и Visual SourceSafe), добавление новых наборов инструментов (например, для редактирования и визуального проектирования кода на предметно-ориентированных языках программирования) или инструментов для прочих аспектов процесса разработки программного обеспечения.

Плюсы MS Visual Studio:

* Официальная. Так как и язык, и среда разработки созданы в Microsoft, логично предположить, что ничего более функционального не найти во всем Интернете. В некоторых случаях без Visual Studio не обойтись — например, при использовании технологий UWP и WPF;
* Бесплатная. Версии «Community edition» для рядового пользователя будет достаточно;
* Функциональная. В Visual Studio множество качественных плагинов. С их помощью можно расширить функциональность приложения и подключить другие языки;
* Поддерживает платформы .NET. Visual Studio имеет широкие возможности по разработке приложений под Windows, в том числе в .NET-сегменте;
* Облачные хранилища. Зарегистрировавшись в сообществе Visual Studio можно получить доступ к облачному хранилищу, где можно располагать файлы проектов.
* Корпоративность. Технология бэклога позволяет членам команды взаимодействовать при гибкой методологии разработки.

Минусы MS Visual Studio:

* Баги при переходах с триал-версии. При переходе на платную версию могут теряться настройки и нарушаться работа корпоративного сервера.
* Сложность. Самостоятельно освоить Visual Studio новичку будет непросто — слишком много доступных функций, спрятанных в подразделах меню.

2.3.2.2 **Eclipse**

Eclipse IDE - интегрированная среда разработки работающая на виртуальной Java-машине JVM. Включает в себя несколько IDE для разработки на языках C / C ++ IDE, JavaScript / TypeScript IDE, PHP IDE и многое другое. Eclipse это одна из самых богатых функционалом IDE с открытым исходным кодом.

Изначально она главным образом использовалась для разработки на Java, но сейчас поддерживает большее разнообразие языков. Эта IDE поставляется с отличным графическим пользовательским интерфейсом и функционалом drag-and-drop. Eclipse IDE доступна для Windows, Linux и MacOS. Эта среда предоставляет много продвинутых особенностей, таких как автоматический анализ кода, интеграция git, статический анализ кода и т. д.

Eclipse IDE открытая платформа для профессиональных разработчиков. Имеет бесплатный и открытый исходный код, выпущенный в соответствии с Eclipse Public License 2.0. Можно легко объединить поддержку нескольких языков и другие функции в любой из пакетов по умолчанию, а Eclipse Marketplace обеспечивает практически неограниченную настройку и расширение. Все больше и больше Eclipse IDE поддерживается отдельными участниками(спонсорами) по всему миру.

Плюсы:

* Множество плагинов. У Eclipse едва ли не самое большое число надстроек — «на все случаи жизни».
* Активное сообщество. Помогает быстрее освоить среду разработки, выпускает новые плагины.
* Отличные компилятор и отладчик. Первый работает на порядок быстрее, чем у конкурентов, второй — показывает потоки, пересечения, позволяет гибко управлять ходом отладки.
* Кастомизация. Благодаря плагинам и настройкам можно полностью персонализировать Eclipse.
* Бесплатность. Это open-source проект, абсолютно бесплатный.
* Высокая функциональность. Благодаря разработчикам-официалам и членам сообщества с помощью Eclipse можно провести любой C#-продукт по полному циклу разработки.

Минусы:

* Сложность. Как и любой функциональный продукт, Eclipse может показаться новичку слишком сложным.
* Нет гарантий надежности. Так как плагины создаются сообществом, за их качество отвечает только разработчик. Кроме того, сами создатели Eclipse с каждой новой версией плодят баги, не успевая порой исправлять старые.

2.3.2.3 **Project Rider**

Rider — это кросс-платформенная IDE для .NET-разработчиков, основанная на платформе IntelliJ и ReSharper, разрабатываемая компанией JetBrains. Rider поддерживает .NET Framework, новую платформу .NET Core и проекты на основе Mono. IDE позволяет разрабатывать десктопные приложения, .NET-сервисы и библиотеки, игры на движке Unity, мобильные приложения Xamarin, веб-приложения ASP.NET и ASP.NET Core. Rider предоставляет более 2200 инспекций кода, сотни контекстных действий и рефакторингов, заимствованных из ReSharper, в сочетании с продвинутой функциональностью сред разработки на основе платформы IntelliJ. Несмотря на большой набор функций, Rider — быстрая и отзывчивая IDE. Помимо умения запускать и отлаживать разные приложения в разных операционных системах, Rider сам по себе тоже поддерживает кросс-платформенность и работает на Windows, macOS и Linux.

Rider обладает следующим функционалом:

* Анализ кода. Rider предоставляет более 2200 инспекций кода и автоматизированных исправлений для устранения обнаруженных проблем как в индивидуальном, так и массовом порядке. Механизм анализа ошибок по всему решению будет искать ошибки в кодовой базе и сообщать о них, даже если проблемный файл не открыт в редакторе.
* Редактирование кода. Умный редактор Rider предоставляет различные виды автодополнения и шаблонов, автоматически вставляет парные скобки и импортирует недостающие пространства имен. Подсказки и иконки на полях помогают легко перемещаться по иерархии наследования, контекстные действия делают разработку удобной и эффективной.
* Рефакторинги. Rider заимствует из ReSharper более 60 рефакторингов и предусматривает более 450 контекстных действий для самых разных целей. Рефакторинги позволяют с легкостью переименовывать и извлекать методы, интерфейсы и классы, перемещать и копировать типы, использовать альтернативный синтаксис и выполнять другие преобразования.
* Инструмент запуска юнит-тестов. Rider помогает запускать и отлаживать юнит-тесты NUnit, xUnit.net и MSTest. ReSharper помогает исследовать тесты, группировать их по сессиям, просматривать результаты тестов и переходить к исходному коду из трассировки стека.
* Отладчик и другие инструменты. Встроенный отладчик для приложений на .NET Framework, Mono и .NET Core поддерживает пошаговое исполнение, позволяет вычислять выражения на лету, запускать программу от текущей выполняемой строки до строки с курсором, отслеживать и менять значения переменных. Кроме того, Rider включает в себя браузер NuGet, позволяет просматривать трассировку стека, поддерживает различные системы контроля версий и базы данных.
* Работа с базами данных и SQL. Можно работать с SQL и базами данных прямо в IDE. Rider помогает подключаться к базам данных, редактировать схемы и таблицы, выполнять запросы и анализировать схемы с помощью UML-диаграмм.
* Расширения. Rider поддерживает большой набор плагинов, разработанных для ReSharper и платформы IntelliJ. Встроенные плагины обеспечивают поддержку F#, Unity и систем контроля версий. Вы можете дополнительно подключить расширения для работы с Markdown, файлами .gitignore и Python-скриптами.

Плюсы:

* ReSharper. Это плагин, изначально разработанный для повышения производительности Visual Studio. Теперь на его основе выпущена IDE.
* Поддержка полного цикла. Фирменная черта продуктов JetBrains, воплощенная и в Project Rider. С ним вы сможете организовать весь цикл создания ПО: от идеи до поддержки.
* Функциональность. Project Rider позволяет подключить MSBuild и XBuild, работать с CLI-проектами и организовать отладку приложений .NET and Mono. Множество опций для быстрого создания кода улучшает производительность.
* Multiple runtime. Поддержка нескольких запущенных программ.
* Кроссплатформенность. Project Rider работает с Windows, Linux и MacOS.
* Контроль версий. Встроенный инструмент позволяет напрямую организовать работу с Git, Mercurial и TFS.

Минусы:

* Молодость. Часть функциональности еще в разработке, не все стартовые баги исправлены.
* Стоимость. Самая дешевая версия Project Rider обойдется в 139 долларов за первый год использования. Но есть триал-версия и специальные предложения для студентов и непрофильных организаций.

Таблица 2 – Сравнительный анализ средств разработки

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Rider | Visual Studio | Eclipse |
| Кроссплатформенность | Да | Да | Да |
| Простота использования | +/- | Нет | +/- |
| Лицензионные условия приобретения и использования | Платный с пробным периодом | Платный с пробным периодом | Бесплатный |
| Наличие расширений | Да | Да | Да |
| Порог вхождения для студентов | Средний | Высокий | Средний |
| Количество различных компонент, готовых к использованию | Большое | Большое | Большое |
| Отладчик кода | Да | Да | Да |
| Анализ кода | Да | Да | Да |
| Рефакторинг | Да | Да | Да |
| Работа с базами данных | Да | Да | Да |
| Системные требования | Средние | Большие | Средние |
| Средства тестирования | Мощные | Средние | Мощные |

Для разработки программного обеспечения для данного дипломного проекта была выбрана IDE Rider. Основными причинами являются:

* средний порог вхождения для начала использования;
* очень мощные средства отладки кода и поиска и устранения ошибок;
* довольно интуитивный и понятный интерфейс;
* активная поддержка со стороны разработчика;
* возрастающая популярность у пользователей и актуальность на рынке труда;
* возможность найти бесплатную версию на просторах интернета;
* большое количество плагинов и расширений;
* не высокие требования к аппаратной части компьютера;
* встроенные функции по работе с базами данных.

2.3.3 **Сравнение баз данных**

Важнейшим элементом любой информационной системы является база данных. Именно благодаря базе данных пользователь может эксплуатировать информационную систему, обслуживать, модифицировать и развивать её при увеличении отделов или изменении информационных потоков организации. Перед выбором базы данных, стоит кратко описать существующие решения. На данный момент существует огромное количество решений для хранения информации, которые отличаются своими характеристиками и возможностями. Существует два вида – реляционный (SQL) и нереляционные (NoSQL) базы данных.

Реляционные - базы данных, основанные на реляционной модели данных. Проще говоря это связанная информация, хранимая в двухмерных массивах. В такой модели каждая база данных содержит таблицы. Каждая таблица представляется как совокупность строк и столбцов, где строки соответствуют экземпляру объекта, конкретному событию или явлению, а столбцы – атрибутам (характеристикам, свойствам) объекта.

Компании всех типов и размеров используют простую, но функциональную реляционную модель для обслуживания разнообразных информационных потребностей. Реляционные базы данных применяются для отслеживания товарных запасов, обработки торговых транзакций через Интернет, управления большими объемами критически важных данных заказчиков и т. д. Реляционные базы данных можно рекомендовать для обслуживания любых информационных потребностей, где элементы данных связаны между собой и необходимо обеспечивать безопасное и надежное управление ими на основе правил целостности. Реляционные базы данных появились в 1970-х годах. На сегодняшний день преимущества реляционного подхода сделали его самой распространенной моделью для баз данных в мире.

Нереляционная база данных — это база данных, в которой в отличие от большинства традиционных систем баз данных не используется табличная схема строк и столбцов. В этих базах данных применяется модель хранения, оптимизированная под конкретные требования типа хранимых данных. Кроме того, они, как правило, поддерживают определенные типы данных. Процесс запроса данных также специфический. Термин NoSQL применяется к хранилищам данных, которые не используют язык запросов SQL, а запрашивают данные с помощью других языков и конструкций. На практике NoSQL означает "нереляционная база данных", даже несмотря на то, что многие из этих баз данных под держивают запросы, совместимые с SQL. Однако базовая стратегия выполнения запросов SQL обычно значительно отличается от применяемой в системе управления реляционной базой данных (реляционная СУБД).

Таблица 3 – Сравнительный анализ баз данных

|  |  |  |  |
| --- | --- | --- | --- |
| Интегрированная среда разработки на языках SQL | Oracle Express/  SQL Developer | MySQL/  DBeaver | MongoDB/  Compass |
| Интерфейс | Интуитивный | Интуитивный | Не удобный |
| Тип представления данных | Табличный | Документный | Табличный |
| Простота использования | ДА | Нет | Да |
| Лицензионные условия приобретения и использования | Бесплатно | Платно | Бесплатно |
| Порог вхождения | Средний | Высокий | Низкий |
| Системные требования | Средние | Низкие | Средние |

Для данного дипломного проекта выбор базы данных являлся произвольным, так как база данных нужна лишь для демонстрации работоспособности разработанных DLL. Проведя анализ существующих решений была выбрана MySQL. Это решение бесплатно, не требовательно к аппаратным ресурсам компьютера и низким порогом вхождения.

2.4 **Технико-экономическое обоснование оптимального технико-технологического решения автоматизации информационных бизнес-процессов**

На основании того, что разработчиком данного дипломного проекта является студент КНУ, со стороны университета на разработку данного программного обеспечения финансовые затраты минимальны. Университет получил данную программу бесплатно и, следовательно, экономическая эффективность данного проекта составляет 100%.

Общее затраченное время на разработку и тестирование данного дипломного проекта составляет 3 месяца и 5 дней.

Если рассматривать ситуацию, в которой за разработку данного дипломного проекта взялся бы специалист, то анализ, разработка и тестирование подобной программы заняли бы у него примерно полтора – два месяца. Если проанализировать среднюю зарплату C# программиста на рынке Казахстана, то можно примерно подсчитать стоимость затрат на реализацию данного проекта. Она составила бы около одного миллиона тенге за 2 месяца при средней зарплате на рынке 500 000 тенге. Затраты на реализацию включают в себя фонд оплаты труда (ФОТ), расходы периода и накладные расходы. Расходы периода составляют 10% от фонда оплаты труда, накладные расходы в свою очередь 5%. В состав ФОТ входят затраты на разработку и тестирование.

2.5 **Потенциальные дивиденды от реализации проекта (экономические, политические, маркетинговые, учебно-методические, научные, инновационные, др). Преимущества, предоставляемые пользователям целевого предмета ДП**

Со стороны КНУ основным потенциальным дивидендом является получение в пользование прикладного программного обеспечения, обеспечивающего работу с микроконтроллерной подсистемой Arduino и базой данных, а также имеющей графический интерфейс для наглядной демонстрации.

В свою очередь, разработанное программное обеспечение предоставит студентам двух образовательных программ «Программирование микроконтроллеров» и «Проектирование и разработка систем на микроконтроллерах» возможность выбора практически любого программного решения для реализации поставленных в рамках данных дисциплин задач. Имея готовое решение, студентам больше не нужно тратить свое время на поиски нужных компонент для тех или иных задач, и они смогут сфокусироваться на основных задачах этих дисциплин. Благодаря тому, что данное программное обеспечение имеет открытый исходный код, студенты могут самостоятельно его дорабатывать и улучшать. Это поможет студентам расширить свои теоретические и практические знания и умения. Исходя из того, что в данной программе использованы все актуальные на сегодняшний момент технологии, студенты станут более компетентными и востребованными на рынке труда.

Основываясь на том, что разработанное ППО имеет графический интерфейс, студентам будет легче вникнуть и понять основные принципы работы с микроконтроллерной подсистемой. Так же, исходя из того, то что данное ППО имеет открытый исходный код, студенты могут самостоятельно дорабатывать, развивать и улучшать данные программное решение, что положительно скажется на их навыках и умениях программирования. Они будут работать с актуальными технологиями, что в положительную сторону отразиться на их компетенции и востребованности на рынке труда.

Для преподавателей основным дивидендом является полученный инструмент, который можно использовать в учебных процессах соответствующих дисциплин. Так же благодаря графическому интерфейсу программы, преподавателю будет значительно легче и удобнее объяснять учебный материал и сделает процесс обучения более увлекательным и интересным.

При рассмотрении маркетинговых преимуществ и дивидендов можно выделить несколько самых важных.

Информирование общества о наличии и успешном функционировании, разработанного в ходе написания данного дипломного проекта, ППО в учебно-практической деятельности КНУ посредством вывода демонстрационных приложений на сайт КНУ, повысится рейтинг КНУ как среди потенциальных абитуриентов, так и среди потенциальных партнеров.

Внедрение данного ППО в учебные процессы КНУ так же повысит качество и скорость усвоения учебного материала, что ведет к положительному росту компетентности студентов на рынке труда, что в свою очередь поднимет репутацию КНУ среди работодателей.

2.6 **Ресурсные затраты по созданию и эксплуатации целевого предмета ДП.**

Со стороны студента основными издержками при реализации данного дипломного проекта являются затраты на разработку прикладного программного обеспечения. Для разработки данного ППО было использовано собственное аппаратное оборудование, в состав которого входит стационарный компьютер, монитор, мышь, клавиатура, роутер, интернет, электроэнергия, ноутбук.

Ниже приведены характеристики использованного аппаратного обеспечения.

Таблица 4 - Характеристики использованного стационарного компьютера

|  |  |
| --- | --- |
| Процессор | Intel core i9 9900k |
| Охлаждение процессора | Enermax ELC-LTTRTO360-TBP LiqTech |
| Кулеры | D-Cooling, DF-12025-ARGB в количестве 5 штук |
| Материнская плата | Asus ROG Maximus XI Code |
| Оперативная память | G.Skill 16 GB 3600 МГц |
| Жесткий диск | Kingston SH103s3120G  Samsung SSD 970 PRO 512 Gb  Western Digital ST1000dm010 |
| Корпус | ATX midi tower AeroCool, Cylon |
| Блок питания | Coolermaster v1200 |
| Видеокарта | Gigabate RTX 2070 Super |
| Операционная система | Windows 10 |

Таблица 5 – Характеристики использованного ноутбука

|  |  |
| --- | --- |
| Название | ASUS ROG Zephyrus G14 |
| Процессор | AMD Ryzen 7 4800HS |
| Объем оперативной памяти | 16 Gb |
| Твердотельный накопитель | SSD m2 512 Gb |
| Модель дискретной видеокарты | Geforce gtx 1660ti |
| Диагональ экрана, дюйм | 14 |
| Адаптер питания | 180 Вт |
| Операционная система | Windows 10 |
| Вес, кг | 1.73 |

Таблица 6 – Характеристики WIFI-адаптера

|  |  |
| --- | --- |
| Производитель | TP-Link |
| Модель | TL-WN8200ND |
| Сетевой стандарт | Wi-Fi EEE 802.11b  Wi-Fi EEE 802.11g  Wi-Fi EEE 802.11n |
| Интерфейс | USB |
| Безопасность | 64/128-битное шифрование WEP, WPA /WPA2, WPA-PSK/WPA2-PSK |
| Мощность передатчика | < 20 дБм или < 100 мВт |
| Частотный диапазон | 2.4 ГГц |
| Скорость передачи данных | 802.11b: до 11 Мбит/с,  802.11g: до 54 Мбит/с,  802.11n: до 300 Мбит/с |

Таблица 7 – Характеристики монитора

|  |  |
| --- | --- |
| Производитель | Xiaomi |
| Модель | Mi Curved Gaming Monitor 34 |
| Диагональ | 34 |
| Разрешение | 3440х1440 |
| Соотношение сторон | 21:9 |
| Частота обновления экрана | 144 Гц |
| Интерфейсы подключения | HDMI x2, DisplayPort x2 |

Со стороны КНУ основными финансовыми затратами являются амортизация на предоставленное аппаратное обеспечение, выданное студенту, а именно микроконтроллер Arduino Nano, датчик звука, датчик света, метео датчик, набор соединительных кабелей и беспаечная плата. Однако размер амортизационных издержек на столько мал, что им можно пренебречь. Так же был затрачен интеллектуальный ресурс преподавателей, которые передали необходимые знания для реализации данного дипломного проекта.

На сегодняшний день существует множество компаний, производящих микроконтроллеры, каждые из которых, имеют свои особенности. В качестве микроконтроллера университетом была выдана платформа Arduino Nano.

Arduino Nano это полнофункциональное миниатюрное устройство на базе микроконтроллера ATmega328 (ArduinoNano 3.0) или ATmega168 (Arduino Nano 2.x), адаптированное для использования с макетными платами.

Arduino Nano – это компактная и дружественная к макету версия платы на базе процессора ATmega328. Это более или менее та же функциональность, что и Arduino UNO, но в другом пакете. Вместо того, чтобы использовать стандартный USB для подключения к компьютеру, он использует мини-USB, но без разъема питания для внешнего источника питания. Размерность Arduino Nano составляет всего 43 мм x 18 мм, он поставляется с 6 входами / выходами ШИМ из общего числа 14 цифровых входов / выходов, 8 аналоговых входов, тактовой частотой 16 МГц и флэш-памятью 32 КБ.

ArduinoNano может получать питание через подключение Mini-B USB, или от нерегулируемого 6-20В (вывод 30), или регулируемого 5В (вывод 27), внешнего источника питания. Автоматически выбирается источник с самым высоким напряжением.

Микросхема FTDI FT232RL получает питание, только если сама платформа запитана от USB. Таким образом при работе от внешнего источника (не USB), будет отсутствовать напряжение 3.3В, генерируемое микросхемой FTDI, при этом светодиоды RX и TX мигаю только при наличии сигнала высокого уровня на выводах 0 и 1.

Каждый из 14 цифровых выводов Nano, используя функции pinMode(), digitalWrite(), и digitalRead(), может настраиваться как вход или выход. Выводы работают при напряжении 5 В. Каждый вывод имеет нагрузочный резистор (стандартно отключен) 20-50 кОм и может пропускать до 40 мА. Некоторые выводы имеют особые функции:

* последовательная шина: 0 (RX) и 1 (TX). Выводы используются для получения (RX) и передачи (TX) данных TTL. Данные выводы подключены к соответствующим выводам микросхемы последовательной шины FTDI USB-to-TTL;
* внешнее прерывание: 2 и 3. Данные выводы могут быть сконфигурированы на вызов прерывания либо на младшем значении, либо на переднем или заднем фронте, или при изменении значения. Подробная информация находится в описании функции attachInterrupt();
* ШИМ: 3, 5, 6, 9, 10, и 11. Любой из выводов обеспечивает ШИМ с разрешением 8 бит при помощи функции analogWrite();
* SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Посредством данных выводов осуществляется связь SPI, которая, хотя и поддерживается аппаратной частью, не включена в язык Arduino;

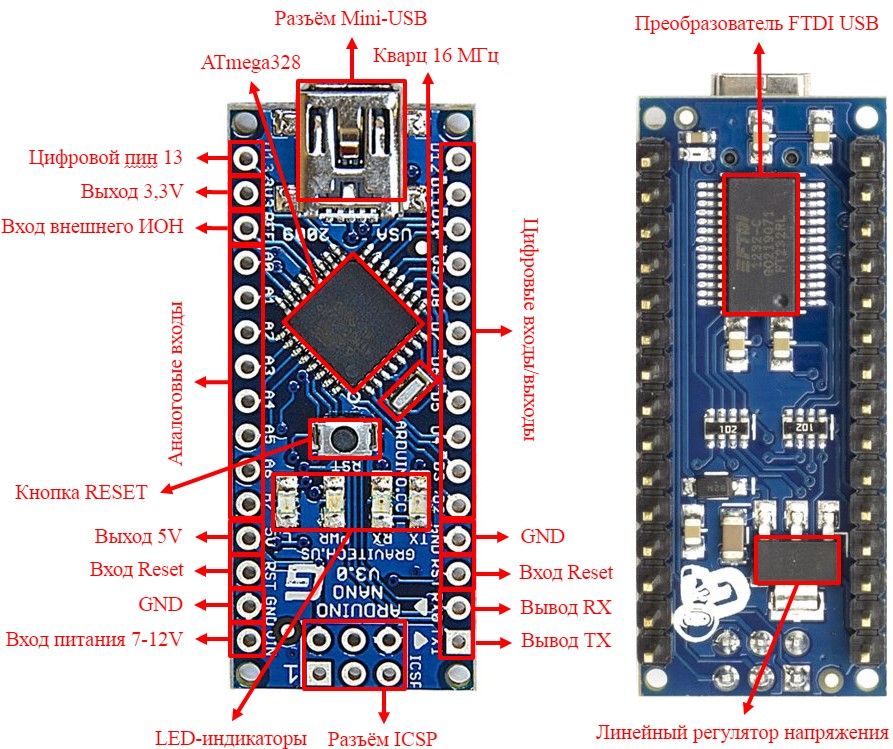


Рисунок 1 – Платформа Arduino Nano

Датчики освещенности (освещения), построенные на базе фоторезисторов, довольно часто используются в реальных Arduino проектах. Они относительно просты, не дороги, их легко найти и купить в любом интернет-магазине. Фоторезистор Arduino позволяет контролировать уровень. Датчик света имеет три контакта для подключения к Arduino Nano. Два контакта служат для питания датчика — 5V и GND, а третий контакт — выдает аналоговый (обозначен буквой S) или цифровой сигнал (обозначен D0) и подключается к соответствующим портам платы Arduino.

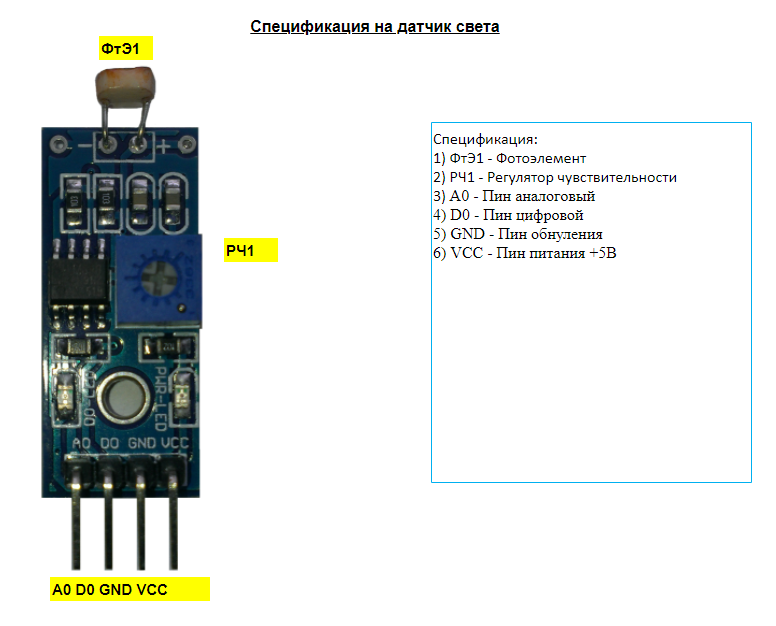


Рисунок 2 – Спецификация датчика света

Датчики DHT22 – очень популярен в среде Arduino и часто используются в проектах метеостанций и умного дома. Данные датчики не выделяются особенным быстродействием и точностью, но они просты в использовании. Датчик может питаться от 3.3-6 В, то есть спокойно может применяться со всеми типами Arduino. Для связи используется однопроводной протокол. Диапазоны измерения – температуры от -40 до 80°С, относительной влажности от 0 до 100%. Типичная погрешность – ±2% влажности и менее 0.5 °С температуры. Каждые 2 секунды датчик способен выдавать обновлённую информацию. Время передачи одного пакета данных – 5 мс. Размеры датчика – 21×15 мм, шаг контактов – 2 мм, то есть их можно вставить в беспаечную макетную плату. Датчик обеспечивает максимальную точность среди датчиков, подключающихся по однопроводному интерфейсу.

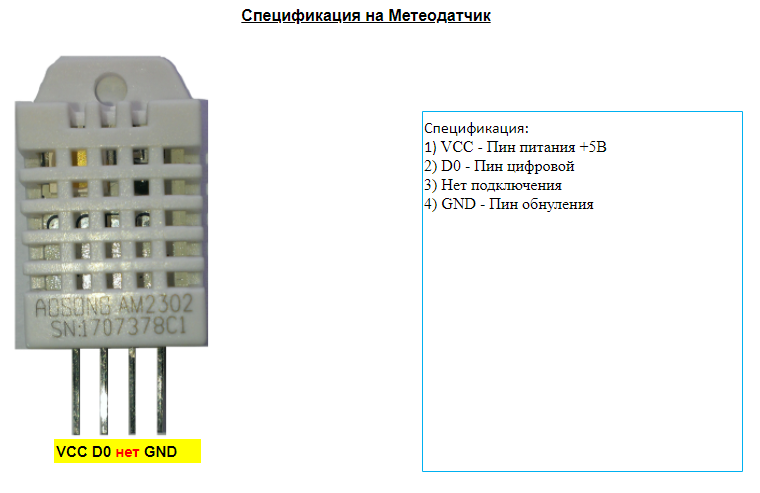


Рисунок 3 – Спецификации метео датчика

Arduino KY-037 - это звуковой Arduino модуль, который реагирует и срабатывает на любой акустический сигнал в зависимости от настроек чувствительности. Когда интенсивность звука достигает настроенного порога срабатывания, то на выходе модуля детектора звука устанавливается высокий логический уровень. Для управления силовыми нагрузками можно использовать модули реле, а также можно подключить детектор звука к контроллеру Arduino. Возможные сферы применения - акустический контроль, сигнализация, акустическое управление, детектор шума и прочее.

Технические характеристики:

* Напряжение питания: 3,3 — 5 В;
* Цифровой и аналоговый вывод;
* Максимальное расстояние обнаружения – 5 м;
* Размер модуля: 32×17×15 мм;
* Общий вес: 12.5 г.



Рисунок 4 – Спецификации датчика звука KY-037

2.7 **Основные методы, способы и средства проектирования, моделирования и разработки технических средств автоматизации информационных бизнес-процессов**

Моделирование — это деятельность, нацеленная на облегчение понимания, количественную оценку, визуализацию или моделирование изучаемой части мира ссылаясь на общепринятые знания. Существует большое множество моделей начиная от концептуальной до графической каждая из которых решает различные задачи. Для реализации графической модели данного дипломного проекта было выбрано решение Draw.io.

Draw.io - проект, в который входят бесплатные векторные редакторы деловой графики для работы в Windows, Linux, MacOS, web. В редакторе можно создавать блок-схемы, планы помещений, UML и Venn диаграммы, BPMN схемы, диаграммы связей, схемы компьютерных сетей, макеты интерфейсов программ, таблицы, инфографику и многое другое. Для вышеназванных тем предустановлены графические элементы, которые просто перетаскиваются мышкой на рабочий холст. Между элементами можно прокладывать соединительные линии по самым замысловатым траекториям. Линия-связь прочно прилипает к смежным элементами, так, что при перетаскивании элемента в другое место, конец связи будет следовать за ним. Результат работы экспортируется в разные форматы: свой собственный формат проекта, векторный формат SVG, растровые форматы PNG, JPG, WEBM, а также в HTML с удобным интерактивным просмотрщиком графики на странице браузера.