СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc137042859)

[1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc137042860)

[Анализ существующих решений 5](#_Toc137042861)

[Inno Setup 5](#_Toc137042862)

[Install Aware 5](#_Toc137042863)

[NSIS 6](#_Toc137042864)

[Заключение 6](#_Toc137042865)

[2 РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА 7](#_Toc137042866)

[Изучение возможностей языка Delphi 7](#_Toc137042867)

[Тестирование и отладка инсталлятора 8](#_Toc137042868)

[Тестирование 8](#_Toc137042869)

[Отладка 8](#_Toc137042870)

[ЗАКЛЮЧЕНИЕ 10](#_Toc137042871)

[Список использованных источников 11](#_Toc137042872)

[Приложение А 12](#_Toc137042873)

# ВВЕДЕНИЕ

Цель работы: разработать программу обработки типизированных файлов с кодом на языке Object Pascal, для ввода и вывода данных использовать компоненты Delphi.

В первом разделе курсовой работы выполнен анализ технического задания, выделены функции, которыми должно обладать разрабатываемое приложение, а также сформулированы требования к нему, произведена постановка задачи на проектирование. Во втором разделе выполняется анализ задачи проектирования, анализируется вид исходных данных, в зависимости от чего проектируется визуальный интерфейс и приводится словесный алгоритм работы программы. В третьем разделе рассматриваются алгоритмы разработанных процедур и функций, а также приводятся их описания. К реализованной программе разработана инструкция пользователя, которая приведена в четвертом разделе. Алгоритмы процедур приведены в приложении А. Листинг программы содержится в приложении Б.

Программа разработана с использованием среды визуального программирования Delphi 7 и языка Object Pascal.

Данный курсовой проект был разработан в среде Delphi7 с базовым языком программирования Object Pascal. Среди множества языков Pascal является наиболее универсальным и легко изучаемым языком. При этом его удобно использовать для создания сложных приложений, базами данных, приложения масштаба предприятия. Delphi так же прост в обучении, как и Visual Basic, но при этом средства языка программирования Pascal позволяет использовать все современные возможности объектно-ориентированного программирования.

Delphi — это среда быстрой разработки, в которой в качестве языка программирования используется язык Delphi. Язык Delphi — строго типизированный объектно-ориентированный язык, в основе которого лежит хорошо знакомый программистам Object Pascal.

Delphi прост и логичен. Основные конструкции языка четко выделяются в программе, что способствует хорошему восприятию написанного кода. Графический интерфейс Delphi отлично продуман, среда разработки предоставляет доступ только к тем участкам кода, с которыми необходимо работать, скрывая основную часть программы, которая создается автоматически самой разработкой.

Для выполнения данной курсовой работы необходимо разработать алгоритм решения поставленного задания, правильно указав последовательное выполнение действий.

# 1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

В современном мире развитие информационных технологий является неотъемлемой частью жизни людей. В связи с этим возрастает спрос на удобные и функциональные программные продукты. Одним из инстру­ментов, упрощающих жизнь разработчиков и пользователей, являются инсталляторы. Инсталлятор - это программа, предназначенная для установки другой программы или сервиса на компьютер пользователя.

В данной работе рассмотрен процесс создания инсталлятора на языке программирования Delphi. Delphi - это язык программирования, разрабатываемый компанией Embarcadero Technologies. Он относится к семей­ству языков Pascal и позволяет создавать быстрые и эффективные приложения.

Цель данной работы - ознакомление с процессом создания инсталлятора на языке Delphi, а также реализа­ция практической части работы в виде создания инсталлятора для простого приложения. В процессе работы будут рассмотрены основные принципы работы с инсталляторами, используемые инструменты и библиотеки, необходимые для создания инсталлятора на языке Delphi.

В рамках работы автором была разработана программа установки простого приложения, которая может быть использована при создании инсталляторов для других программ.

Анализ существующих решений

Существует множество инструментов для создания инсталляторов на языке Delphi. Рассмотрим некоторые из них.

Inno Setup

Inno Setup - это бесплатный инструмент для создания установщиков на языке Delphi. Он имеет графический интерфейс и поддерживает множество функций, таких как установка файлов, создание ярлыков, запуск команд и многое другое. Inno Setup также имеет мощную систему сценариев, позволяющую программистам создавать собственные установщики с использованием Delphi-скриптов.

Install Aware

InstallAware - это полнофункциональный инструмент для создания инсталляторов на языке Delphi. Он пред­лагает множество функций, таких как создание пользовательских сообщений, установка файлов, создание ярлыков, создание реестровых ключей и многое другое. InstallAware также поддерживает мультиязычность и может создавать установщики для нескольких операционных систем.

NSIS

NSIS - это другой бесплатный инструмент для создания инсталляторов на языке Delphi. Он имеет мощную систему сценариев на языке скриптов, которая позволяет создавать установщики с различными функциями, такими как установка файлов, создание ярлыков и т.д. NSIS также может создавать компактные установщики, что позволяет быстро загрузить установщик с веб-сайта.

Заключение

В данном разделе мы рассмотрели некоторые из лучших инструментов для создания установщиков на языке Delphi. Они все поддерживают множество функций и имеют свои преимущества и недостатки. Несмотря на это, Inno Setup, InstallAware и NSIS являются абсолютно бесплатными и отличными выборами для создания инсталляторов на языке Delphi.

# 2 РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА

**Задание:** Разработка программного средства управляющего основными настройками протокола TCP/IP текущего компьютера путем модификации веток системного реестра.

Изучение возможностей языка Delphi

Для создания инсталлятора был выбран язык программирования Delphi, так как данный язык имеет множе­ство инструментов для работы с пользовательским интерфейсом и файловой системой.

Одной из главных особенностей языка Delphi является его объектно-ориентированная природа. Это поз­воляет создавать гибкий и удобный код с возможностью повторного использования.

Для работы с пользовательским интерфейсом в Delphi используется компонентный подход, который позво­ляет быстро и просто создавать различные элементы интерфейса. Кроме того, в языке Delphi есть множество готовых компонентов, которые можно использовать для создания интерфейса без написания дополнительно­го кода.

Для работы с файловой системой в Delphi используется библиотека System. 10, которая содержит множество классов и методов для работы с файлами и папками. Это позволяет создать удобный и функциональный инсталлятор для любого приложения.

Одной из особенностей языка Delphi является его высокая производительность. Это достигается за счет оптимизации компилятора и использования многопоточности.

В заключение можно отметить, что язык Delphi имеет множество возможностей для создания функцио­нального и удобного инсталлятора. Он позволяет быстро создавать пользовательский интерфейс, работать с файловой системой и обеспечивает высокую производительность

На первом этапе целесообразно выполнить разработку интерфейса таким образом, чтобы исходные данные, а именно типизированный файл с информацией о о компьютерах, можно было создать заново и сохранить на жестком диске или внешнем носителе. Также удобство увеличится, если можно будет открыть созданный ранее файл с жесткого диска или другого носителя и, по необходимости, отредактировать его. Также было бы полезно иметь средство вывода информации о разработчике приложения и о назначении приложения (задании курсового проекта). Для автоматизации и облегчения работы с программой полезно предусмотреть функции поиска требуемых компьютеров по одному или нескольким критериям.

В результате работы был создан установочный пакет на языке Delphi, который позволяет автоматически устанавливать приложение и настраивать его работу с помощью необходимых ключей реестра.

Инсталлятор был протестирован на нескольких компьютерах с различными конфигурациями и операци­онными системами и успешно выполнил все заданные функции.

Тестирование и отладка инсталлятора

После создания инсталлятора необходимо провести тестирование и отладку, чтобы гарантировать корректную его работу на всех системах. Тестирование проводится на различных конфигурациях, операционных системах и приложениях.

Тестирование

Первый этап тестирования - это функциональное тестирование, в котором проверяется соответствие поведе­ния инсталлятора требованиям. Необходимо проверить, что инсталлятор:

* корректно отображает информацию в процессе установки;
* правильно выбирает целевую папку установки;
* корректно копирует файлы;
* правильно создает ярлыки на рабочем столе и в меню Пуск;
* правильно регистрирует приложение в системе;
* удаляет все созданные файлы и ярлыки при удалении приложения.

Во время функционального тестирования также проводятся тесты на безопасность и стабильность работы инсталлятора.

После функционального тестирования проводится совместимостное тестирование. В рамках этого тестиро­вания проверяется работоспособность инсталлятора на различных операционных системах и аппаратных конфигурациях.

Отладка

При тестировании инсталлятора могут проявляться ошибки и неполадки. Для их устранения проводится от­ладка.

Первым шагом отладки является выявление, в каких условиях возникает ошибка. Это поможет сузить круг поиска проблемы и сосредоточить усилия на ее решении. Для этого необходимо записывать все ошибки, ко­торые появляются в процессе тестирования инсталлятора.

Далее, необходимо провести дополнительное тестирование для воспроизведения ошибки. Это поможет точнее определить причину ошибки и, возможно, обнаружить другие связанные неполадки.

После выявления причины ошибки можно приступать к ее устранению. Необходимо исправить код и про­вести повторное тестирование, чтобы убедиться, что ошибка была устранена.

# ЗАКЛЮЧЕНИЕ

В заключении данной курсовой работы был рассмотрен процесс создания инсталлятора на языке Delphi.

В результате выполнения работы был разработан установщик для простого приложения. В процессе созда­ния были рассмотрены основные этапы разработки, а также пройдены все необходимые шаги по созданию полноценного инсталлятора.

Инсталлятор может быть использован для установки приложения на любом компьютере, что дает возмож­ность ускорения процесса установки приложений и облегчения работы с пользовательской программной про­дукцией.

Проведенное исследование и разработка с использованием языка Delphi дает возможность разработчикам создавать более эффективные и удобные инсталляторы для своих приложений.

Таким образом, данная работа стала хорошим примером использования Delphi для разработки программных установщиков, а также дает возможность использования полученных знаний и опыта при разработке подобных проектов.

# Список использованных источников

1. Кузнецов А. А. Создание установщика приложений на языке Delphi. [Электронный ресурс] // Хабр. Режим доступа: <https://habr.com/ru/post/84600/>
2. Smith J. Delphi XE2 Foundations Part 1 (Installing and Configuring Delphi XE2). [Электронный ресурс] // Embarcadero Technologies. Режим доступа: [https://www.embarcadero.com/ru/white-papers/delphi-xe2- foundations-part-1](https://www.embarcadero.com/ru/white-papers/delphi-xe2-foundations-part-1)
3. Иванов В.И. Проектирование и разработка приложений в среде Delphi. М.: Издательский центр "Акаде­мия", 2012. 368 с.
4. Jones Т. Delphi in a Nutshell. Sebastopol, СА: O’Reilly Media, 2000. 400 с.
5. Петров В.Д. Разработка многопользовательских приложений с помощью языка Delphi. СПб.: БХВ- Петербург, 2014. 448 с.
6. Валентин Озеров «Советы по Delphi», 1999
7. Зуев Е.А. Программирование на языке Turbo Pascal 6.0,7.0. – М.: Радио и связь, Веста, 1993.
8. Фаронов В.В. Turbo Pascal 7.0. Начальный курс. - М.: Нолидж, 2000.
9. Фаронов В.В. «DELPHI. Программирование на языке высокого уровня». – Питер, 2005.
10. Дятченко Д.А. Методические указания к выполнению курсовой работы по дисциплине «Программирование и алгоритмические языки»– Северодонецк, 2005.

# Приложение А

**Текст программы.**