

НАЗВАНИЕ УНИВЕРСИТЕТА
Название факультета
Название образовательной программы

СОГЛАСОВАНО

Должность согласовавшего

_____ ФИО
«_____» _____ 20__г.

УТВЕРЖДАЮ

Должность утвердителя

_____ ФИО
«_____» _____ 20__г.

СОЗДАНИЕ ИНСТАЛЛЯТОРА НА ЯЗЫКЕ DELPHI

Курсовая работа

RU.17701729.10.03-01 01-1-ЛУ

Исполнитель
Студент группы *номер*
образовательной программы
«Название программы»
_____ ФИО
«_____» _____ 20__г.

Город 2023

УТВЕРЖДЕН
RU.17701729.10.03-01 01-1-ЛУ

СОЗДАНИЕ ИНСТАЛЛЯТОРА НА ЯЗЫКЕ DELPHI

Курсовая работа

RU.17701729.10.03-01 01-1-ЛУ

Листов 12

Содержание

1	Введение	3
2	Анализ существующих решений	4
2.1	Inno Setup	4
2.2	InstallAware	4
2.3	NSIS	4
2.4	Conclusion	4
3	Изучение возможностей языка Delphi	5
4	Алгоритм создания инсталлятора	6
5	Проектирование интерфейса пользователя	7
6	Разработка инсталлятора на языке Delphi	8
7	Тестирование и отладка инсталлятора	9
7.1	Тестирование	9
7.2	Отладка	9
7.3	Оформление отчета	9
8	Заключение	10

1 Введение

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.

2 Анализ существующих решений

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

2.1 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.

2.2 InstallAware

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.

2.3 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.

2.4 Conclusion

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.

3 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.IO, 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.

4 Алгоритм создания инсталлятора

Алгоритм создания инсталлятора

1. Определение требований и функциональности

Первый шаг при создании инсталлятора — определение требований и функциональности. Необходимо определить, какие файлы и компоненты должны быть установлены на целевом компьютере, а также какие действия необходимо выполнить в процессе установки и удаления.

2. Проектирование пользовательского интерфейса

Следующим шагом является проектирование пользовательского интерфейса. Необходимо определить, какие страницы должны быть включены в инсталлятор, а также какие элементы управления должны быть использованы на каждой странице.

3. Создание проекта инсталлятора

После того, как требования и функциональность будут определены, необходимо создать проект инсталлятора. Для этого необходимо выбрать соответствующий шаблон инсталлятора в Delphi и настроить его параметры.

4. Программирование логики инсталлятора

Следующим шагом является программирование логики инсталлятора. Необходимо определить, какие команды и действия должны быть выполнены в процессе установки и удаления, а также какие проверки необходимо выполнить, чтобы убедиться в том, что все компоненты установлены корректно.

5. Создание проекта пакета

После того, как логика инсталлятора будет определена, необходимо создать проект пакета. Пакет является набором файлов и компонентов, которые необходимо установить на целевом компьютере. В проекте пакета также определяются зависимости между компонентами и файлами.

6. Настройка параметров пакета

После создания проекта пакета необходимо настроить его параметры. Необходимо задать соответствующие параметры для каждого компонента и файла, которые должны быть установлены.

7. Компиляция инсталлятора и пакета

Следующим шагом является компиляция инсталлятора и пакета. После компиляции инсталлятор и пакет будут готовы к выпуску.

8. Тестирование инсталлятора

Последним шагом является тестирование инсталлятора. Необходимо убедиться в том, что все файлы и компоненты устанавливаются корректно, а также что процесс установки проходит без ошибок.

5 Проектирование интерфейса пользователя

Для создания графического интерфейса пользователя (GUI) в программе-инсталляторе был выбран инструментальный компонент VCL (Visual Component Library) библиотеки компонентов Delphi.

Основным окном программы является главное окно, на котором расположены все элементы управления. В верхней части окна расположена панель меню, в которой содержатся основные команды управления программой: "Файл", "Настройки", "Помощь". Панель меню расширяется при нажатии на соответствующую кнопку в левом верхнем углу окна.

Основная часть окна занимает список действий, которые выполняются при установке программы на компьютер пользователя. В данном списке содержатся элементы управления для выбора опций установки, текстовые поля, кнопки для навигации в списке и выпадающие списки.

В нижней части окна расположены кнопки "Назад", "Далее" и "Отмена", которые используются для навигации по списку действий, а также для запуска и отмены процесса установки программы. Также здесь расположен прогресс-бар, отображающий ход установки.

Все элементы управления имеют подсказки при наведении курсора на них, чтобы облегчить понимание, как они работают. Также предусмотрена возможность изменения размеров окна, чтобы обеспечить удобство при работе на различных разрешениях экрана.

В процессе проектирования интерфейса пользовательского окна были учтены следующие требования:

- простота и интуитивность интерфейса;
- минимум необходимых элементов управления;
- удобство навигации в списке действий;
- возможность отменить процесс установки в любой момент;
- использование подсказок для пояснения работы элементов управления;
- регулировка размеров окна для поддержки различных разрешений экрана.

6 Разработка инсталлятора на языке Delphi

=====

Для разработки инсталлятора на языке Delphi были использованы следующие технологии и инструменты:

- Среда разработки Delphi
- Компоненты для создания интерфейса (VCL)
- Компоненты для установки файлов и настроек (InnoSetup)
- Компоненты для работы с реестром (Jedi Code Library)

В процессе разработки была создана форма с интерфейсом, в котором были добавлены необходимые элементы управления: кнопка "Установить", кнопка "Отмена", прогресс-бар, текстовое поле для вывода информации о процессе установки.

Для работы с файлами и настройками в процессе установки было использовано средство установки InnoSetup. Оно позволяет создавать установочные пакеты с необходимой конфигурацией и производить установку нужных приложений и файлов на компьютер.

Для работы с реестром была использована библиотека Jedi Code Library. Она предоставляет множество функций для работы с реестром, таких как запись и чтение значения, создание и удаление ключей и т.д.

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.

7 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.

7.1 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.

7.2 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 приступить к ее устранению. Необходимо исправить код и провести повторное тестирование, чтобы убедиться, что ошибка была устранена.

7.3 Оформление отчета

После проведения тестирования и отладки необходимо оформить отчет, который будет содержать информацию о проведенных тестах, выявленных ошибках и проведенной отладке. В отчете также могут быть приведены рекомендации по повышению качества инсталлятора.

Отчет должен быть оформлен в удобочитаемой форме и содержать четкие выводы и рекомендации. Он должен быть доступен для всех участников проекта, чтобы они могли ознакомиться с проведенными тестами и результатами.

8 Заключение

В заключении данной курсовой работы был рассмотрен процесс создания инсталлятора на языке Delphi. Было изучено, каким образом можно создать программу-установщик при помощи компонентов TInstall и TRegistry.

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>
- [3] Иванов В.И. Проектирование и разработка приложений в среде Delphi. М.: Издательский центр "Академия", 2012. 368 с.
- [4] Jones T. Delphi in a Nutshell. Sebastopol, CA: O'Reilly Media, 2000. 400 с.
- [5] Петров В.Д. Разработка многопользовательских приложений с помощью языка Delphi. СПб.: БХВ-Петербург, 2014. 448 с.