

Самостоятельная работа № 5

Тема. Оценочное тестирование программного продукта.

Цель. Научиться создавать инсталляционные файлы; выполнять оценочное тестирование программного продукта.

Оборудование. ПК

Ход работы

1. Ознакомиться с теоретической частью.
2. Выполнить практическое задание.
3. Ответить на контрольные вопросы.
4. Оформить отчет.

Теоретическая часть

После завершения комплексного тестирования приступают к оценочному тестированию, целью которого является тестирование программы на соответствие основным требованиям. Эта стадия тестирования особенно важна для программных продуктов, предназначенных для продажи на рынке.

Оценочное тестирование, которое также называют «тестированием системы в целом», включает следующие виды:

- *тестирование удобства использования* - последовательная проверка соответствия программного продукта и документации на него основным положениям технического задания;
- *тестирование на предельных объемах* - проверка работоспособности программы на максимально больших объемах данных, например, объемах текстов, таблиц, большом количестве файлов и т. п.;
- *тестирование на предельных нагрузках* - проверка выполнения программы на возможность обработки большого объема данных, поступивших в течение короткого времени;
- *тестирование удобства эксплуатации* - анализ психологических факторов, возникающих при работе с программным обеспечением; это тестирование позволяет определить, удобен ли интерфейс, не раздражает ли цветовое или звуковое сопровождение и т. п.;
- *тестирование защиты* - проверка защиты, например, от несанкционированного доступа к информации;
- *тестирование производительности* - определение пропускной способности при заданной конфигурации и нагрузке;
- *тестирование требований к памяти* - определение реальных потребностей в оперативной и внешней памяти;
- *тестирование конфигурации оборудования* - проверка работоспособности программного обеспечения на разном оборудовании;
- *тестирование совместимости* - проверка преемственности версий: в тех случаях, если очередная версия системы меняет форматы данных, она должна предусматривать специальные конверторы, обеспечивающие возможность работы с файлами, созданными предыдущей версией системы;
- *тестирование удобства установки* - проверка удобства установки;
- *тестирование надежности* - проверка надежности с использованием соответствующих математических моделей;
- *тестирование восстановления* - проверка восстановления программного обеспечения, например системы, включающей базу данных, после сбоев оборудования и программы;
- *тестирование удобства обслуживания* - проверка средств обслуживания, включенных в программное обеспечение;
- *тестирование документации* - тщательная проверка документации, например, если документация содержит примеры, то их все необходимо попробовать;
- *тестирование процедуры* - проверка ручных процессов, предполагаемых в системе и др.

Естественно, целью всех этих проверок является поиск несоответствий техническому заданию. Считают, что только после выполнения всех видов тестирования программный продукт может быть представлен пользователю или к реализации. Однако на практике обычно выполняют не все виды оценочного тестирования, так как это очень дорого и трудоемко. Как правило, для каждого типа программного обеспечения выполняют те виды тестирования, которые являются для него наиболее

важными. Так базы данных обязательно тестируют на предельных объемах, а системы реального времени - на предельных нагрузках.

Системы для создания инсталляторов

Практика разработки коммерческого программного обеспечения показывает, что далеко не все пользователи умеют работать с архивами. Поэтому программы рекомендуется поставлять в виде исполняемых файлов, которые автоматически создают необходимые папки в файловой системе, копируют туда файлы программы, создают необходимые файлы настроек или ключи в реестре, а так же пункты меню запуска программы и ярлыки на рабочем столе. Для упрощения создания инсталляторов существует много специализированных программных продуктов.

Знакомство пользователя с программой чаще всего начинается с запуска инсталлятора. Внешний вид («упаковка») и функциональность продукта определяется разработчиком. Пользователю нужно иметь возможность проконтролировать процесс, выставив нужные параметры установки. Для разработчика же важно, чтобы, как минимум, его программа была установлена корректно, а инсталлятор был совместим с необходимыми платформами.

Решений для создания инсталляторов достаточно много. Чаще всего используется подсистема Windows Installer, которая уже входит в инструментарий операционной системы. Но существуют и альтернативные решения – как платные, так и бесплатные, различной функциональности. Зачастую с их помощью можно создавать пакеты с инсталлятором, не зависящим от Windows Installer.

Основные критерии выбора системы создания инсталлятора следующие:

- среда разработки, интерфейс, поддержка сценариев;
- работа с проектом, типы создаваемых пакетов, возможности импорта проектов из других сред разработки;
- пользовательские опции инсталлятора: поддержка языков, профилей и другие опции;
- поддержка расширений.

Свободные программы для создания инсталляторов:

– NSIS (Nullsoft Scriptable Install System) – один из самых популярных инсталляторов. Обладает богатыми возможностями, которые присутствуют в большинстве коммерческих продуктов. Позволяет устанавливать различные параметры сжатия при создании дистрибутива;

– IzPack – java инсталлятор. Это универсальный инсталлятор, способен создавать дистрибутивы для Unix, Linux, FreeBSD, Mac OS X и Windows 2000, XP. Позволяет создавать как обычные пакеты инсталляции, так и Web инсталляторы, которые подгружают необходимые файлы по мере необходимости. Данная возможность позволяет свести к минимуму количество загружаемых файлов в зависимости от требуемой конфигурации установки;

– Inno Setup – довольно популярный простой инсталлятор. Содержит встроенный скриптовый язык;

– WiX (Windows Installer XML) – специализированный продукт от Microsoft для создания MSI и MSM инсталляционных пакетов.

Коммерческие программы для создания инсталляторов:

– InstallShield – один из самых известных продуктов в ряду инсталляторов;

– WISE – простой в освоении с богатыми возможностями генератор инсталляторов;

– VISE - профессиональный инсталлятор для Windows, MacOS X и Macintosh;

– CreateInstall это универсальный, гибкий и мощный инсталлятор как для профессиональных разработчиков, так и для начинающих. С помощью этой программы Вы можете создать полнофункциональные инсталляционные программы для Ваших приложений, а также самораспаковывающиеся архивы с высокой степенью сжатия и многое другое;

– Advanced Installer – позволяет создавать инсталляторы для java приложений. Создает дополнительный исполняемый файл.

CreateInstall

Домашняя страница: <http://www.createinstall.ru/>

CreateInstall – инструментарий для создания установщиков. В его основу заложено две особенности – контроль над процессом установки и неограниченная расширяемость. Обе возможности реализованы благодаря языку программирования Gentee, применяемому для написания сценариев.

Интерфейс CreateInstall разбит на 3 вкладки – «Проект», «Скрипт установки» и «Скрипт деинсталляции». Первый раздел позволяет задать общие настройки инсталлятора: информация о

продукте, поддерживаемые языки, пути, внешний вид. Дополнительно, инсталлятор можно защитить цифровой подписью и установить пароль.

«Проект» – не равноценная замена двух последующих разделов, т. е. для создания дистрибутива нужно тщательно настроить скрипты установки и деинсталляции. Соответствующие параметры отображаются в виде групп, можно отобразить их единым списком.

Дополнением для CreateInstall служит утилита Quick CreateInstall (рисунок 1). Она значительно упрощает создание инсталлятора, предоставляя только базовые настройки проекта. Из Quick CreateInstall в дальнейшем проект можно импортировать в CreateInstall.

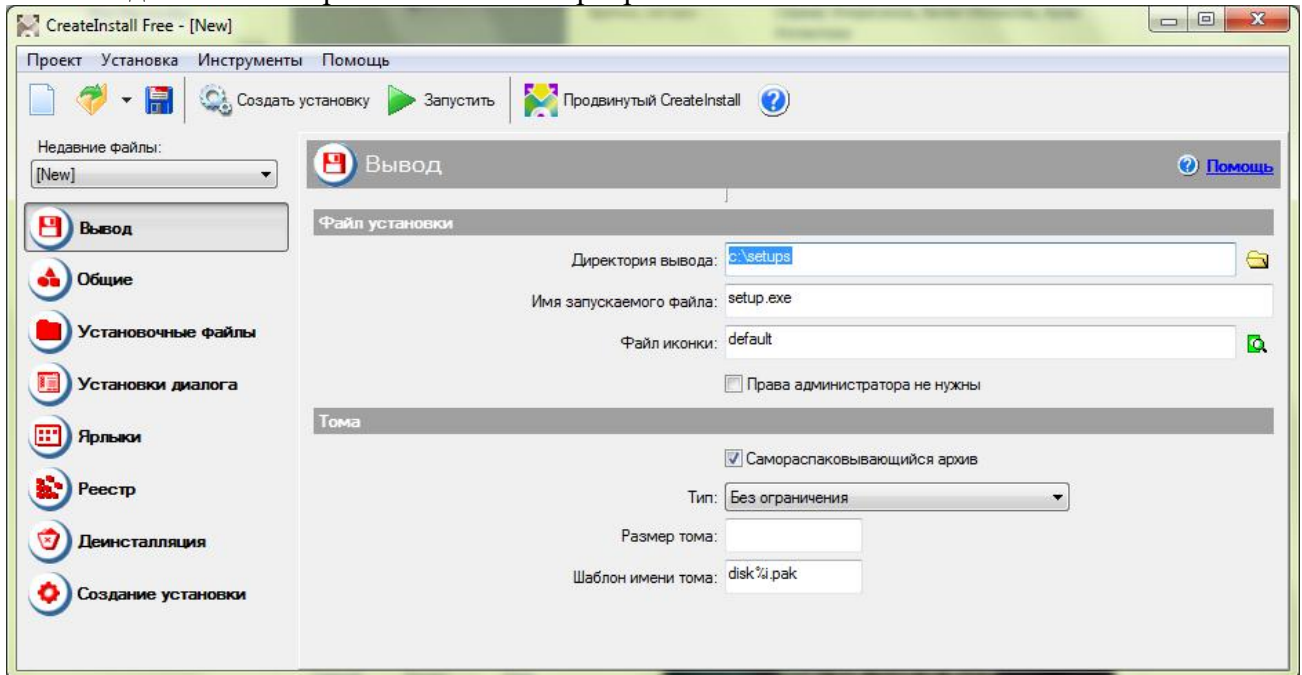


Рисунок 1 – Окно Quick CreateInstall

Код проекта не предназначен для самостоятельного редактирования, переноса в IDE-среду, экспорта. Хотя язык Gentee имеет отличный потенциал: как минимум, это переменные и функции, условные выражения и синтаксис, базирующийся на C, C++ и Java.

Существует 3 редакции программы – полная, light (простая) и бесплатная.

Интерфейс и справка доступны на русском языке.

Advanced Installer

Advanced Installer основывается на технологии Windows Installer, позволяя создавать msi-, exe- и других видов дистрибутивов. Этому способствует продуманный интерфейс и работа с проектами. В Advanced Installer можно обнаружить немало возможностей, которых нет в других подобных комплексах.

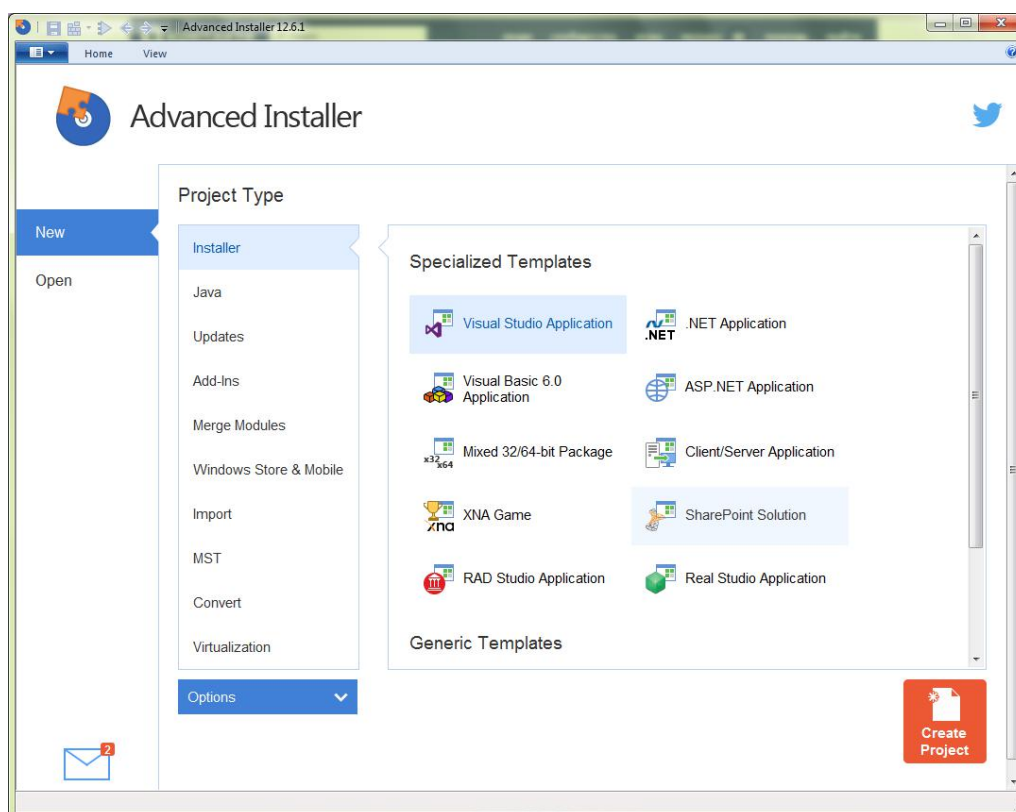


Рисунок 2 – Окно Advanced Installer

Примечательно, прежде всего, разнообразие проектов: сюда входят инсталляторы, Java-установщики, обновления, дополнения, модули слияния и другие. В разделе меню **Installer** собраны команды импорта проектов из Visual Studio, RAD Studio, Real Studio, Visual Basic. Здесь раскрывается потенциал Advanced Installer во взаимодействии с IDE-средами.

Для каждого из выбранных типов проекта предусмотрен детальный мастер настройки. Есть общие шаблоны – Simple, Enterprise, Architect или Professional. Большая часть проектов доступна только для определенных типов лицензии, общедоступные проекты обозначены как None в графе License Required.

Как уже сказано, при создании проекта можно воспользоваться пошаговым мастером, где, в частности, доступен выбор способа распространения пакета, языков локализации, настройка пользовательского интерфейса, ввод текста лицензии и другие опции. Advanced Installer позволяет выбрать вариант распространения программы – оставить данные без компрессии, разделить на CAB-архивы, сохранить в MSI и др., добавить цифровую подпись, потребовать ввод серийного номера и т. д.

Главное окно Advanced Installer (редактор проекта), в простом режиме отображения (Simple), содержит несколько секций:

- Product Information (Информация о продукте) – ввод сведений о продукте, параметры установки.
 - Requirements (Требования) – указание аппаратных и системных требований, зависимостей ПО.
- Также имеется возможность создания пользовательских условий.
- Resources (Ресурсы) – редактор ресурсов (файлов и ключей реестра).
 - Deployment (Развертывание) – выбор типа распространения продукта. Это может быть MSI, EXE или веб-инсталлятор. Для MSI, EXE ресурсы можно поместить отдельно от инсталлятора.
 - System Changes – переменные среды.

При выборе ресурсов могут использоваться файлы, ключи реестра, переменные окружения, конфигурационные ini, драйверы, базы данных и переводы. С помощью модулей объединения можно добавить и другие ресурсы, такие как сервисы, разрешения, ассоциации и др.

Для выполнения более сложных задач позволяется использовать пользовательские действия, EXE, DLL или скрипты, написанные на C, C++, VBS или JS. Для создания сценариев предусмотрен удобный редактор.

Однако следует отметить, что в режиме Simple доступна лишь малая часть разделов. Работая с Advanced Installer в ознакомительном режиме, есть смысл зайти в настройки и переключиться в другой режим работы с проектом. После этих действий становятся доступны новые подразделы редактора.

Практическая часть

Задание 1. С помощью системы создания инсталляторов создайте из программы, созданной на

лабораторной работе № 6, установочный файл.

Задание 2. Выполните тестирование удобства установки.

Задание 3. Выполните тестирование конфигурации оборудования.

Задание 4. Выполните тестирование восстановления.

Задание 5. Выполните тестирование удобства эксплуатации при помощи соседа.

Задание 6. Результаты выполнения практического задания запишите в отчет.

Контрольные вопросы

1. Что является целью тестирования программ?
2. Какие подходы к тестированию вы знаете? В чем они заключаются?
3. Обоснуйте необходимость создания инсталляторов программ.

Содержание отчета

1. Тема.
2. Цель.
3. Оборудование.
4. Результат выполнения практического задания.
5. Ответы на контрольные вопросы.
6. Вывод.