

Вариант 37 Пакетный менеджер

Разработать приложение - пакетный менеджер (базу данных о находящихся на некотором узле пакетах программ).

Разновидности пакетов:

- 1) Пакет, содержащий основную функциональность пользовательского или системного приложения. Все подобные пакеты должны быть в явном виде доступны пользователю через некоторый интерфейс хранилища. Характеризуется признаком установки пакета, именем файла пакета, номером последней версии, номером установленной версии (если он установлен), именем издателя, ссылками на требуемые для его функционирования другие пакеты.
- 2) Пакет со вспомогательным ПО (библиотеками). Характеристики те же, но данный пакет должен устанавливаться в системе только при наличии хотя бы одного основного пакета, ссылающегося на него.
- 3) "Пустой" пакет --- связан с конкретным пакетом, но не содержит конкретной версии продукта; с этим пакетом обязательно связан пакет другого типа (основной или вспомогательный) последней версии.

Действия:

- ❖ Для пакетного менеджера:
 - инициализация списком пакетов, требуемых для первоначальной установки
 - установка требуемого пользовательского или "пустого" пакета (с автоматической установкой всех требуемых пакетов)
 - установка требуемого пользовательского или "пустого" пакета (с запросом пользователя на установку каждого требуемого пакета)
 - удаление указанного пользовательского пакета с автоматическим удалением ненужных пакетов со вспомогательным ПО или без него
 - очистка системы от неиспользуемых вспомогательных пакетов
 - глобальное обновление системы с выводом данных о каждом обновленном пакете
- ❖ Для пакета
 - добавление пакета в хранилище (* с контролем ацикличности графа зависимостей)
 - получение информации о состоянии пакета (тип, установлен или нет, текущая и последняя версии, количество связей с другими пакетами каждого типа)
 - удаление пакета из хранилища, если это возможно (полное удаление, не снос ПО)
 - обновление пакета
- ❖ Для пакета со вспомогательным ПО:
 - разделение пакета на заданное количество частей с указанием их параметров; считать, что все зависимости наследуются пакетами-наследниками
- ❖ Для непустого пакета (не связанного с пустым):
 - Объединение пакетов заданного набора с заданием параметров и объединением зависимостей (дублирующиеся зависимости устраняются)

Считать, что обновление пакета не влияет на функционал.

Порядок выполнения работы

1. На основе описания задачи определить состав классов.
2. Разработать иерархию классов и схему их взаимодействия.
3. Для каждого класса определить его состояние и необходимые методы.
4. Разработать и отладить все классы. Приложение для отладки реализовать в виде простой диалоговой программы.
5. Разработать и отладить класс, реализующий работу приложения. Предусмотреть в классе возможность сохранения таблицы в файле и ее восстановления.
6. Повторить разработку контейнерного класса (вместе с необходимыми итераторами), используя стандартную библиотеку STL. Выбор шаблона классов согласовать с преподавателем.
7. Повторить разработку контейнерного класса (вместе с необходимыми итераторами), используя собственные шаблоны классов.
8. Реализовать приложение с использованием средств оконного интерфейса.
9. Реализовать приложение, использующее разработанные классы, по дополнительному заданию.