

Понятие рефакторинга. Оптимизация программ

Оптимизация - процесс выбора наилучшего (оптимального) варианта из множества возможных или модификация системы для улучшения её эффективности.

Истинно оптимальная система в процессе оптимизации достигается далеко не всегда. Оптимизированная система обычно является оптимальной только для одной задачи или группы пользователей. Как правило, процесс оптимизации завершается до того, как достигается полная оптимальность.

«Преждевременная оптимизация — это корень всех бед». Очень важно иметь для начала озвученный алгоритм и работающий прототип.

Понятие «оптимизация» обычно подразумевает, что система сохраняет ту же самую функциональность. Однако, значительное улучшение производительности часто может быть достигнуто и с помощью удаления избыточной функциональности.

Оптимизация в основном фокусируется на одиночном или повторном времени выполнения, использовании памяти, дискового пространства, пропускной способности или некотором другом ресурсе. Это обычно требует компромиссов — один параметр оптимизируется за счёт других.

В программировании, оптимизация обычно обозначает модификацию кода и его установок компиляции для данной архитектуры для производства более эффективного ПО.

Для оптимизации требуется найти *узкое место* (англ. hotspot): критическую часть кода, которая является основным потребителем необходимого ресурса. Улучшение примерно 20 % кода иногда влечёт за собой изменение 80 % результатов. Для поиска узких мест используются специальные программы — профайлеры.

Чем больше памяти использует программа, тем быстрее она обычно выполняется.

Простейшие приёмы оптимизации:

- Инициализация объектов данных (Правильная инициализация объектов позволяет сэкономить процессорное время.)
- Программирование арифметических операций (самыми быстрыми являются операции сложения и вычитания. Более медленным является умножение, затем идёт деление)

Приоритеты оптимизации:

- **интерфейсный**, т.е. желательно заранее ВСЁ согласовать с другими участниками проекта, включая кому сколько процессорного времени на конкретном ПК.
- Если ПК не является современным и многопроцессорным, заведомо сильно ограничить суммарный CPU расход, можно утилитами типа `slowcpu`.
- согласовать и обсудить - не спеша, и каждую мелочь вплоть до того где использовать ООП, а где нет - для ускорения.
- **алгоритмический**, минус - принцип чем оптимизированней - тем менее понятно, тем больше багов; но это самый эффективный способ, тем более наиболее оптимизируемые алгоритмы уже давно оптимизированы и можно посмотреть на их реализацию.

Рефакторинг — процесс изменения внутренней структуры программы, не затрагивающий её внешнего поведения и имеющий целью облегчить понимание её работы. В основе рефакторинга лежит последовательность небольших эквивалентных (то есть сохраняющих поведение) преобразований.

Цель рефакторинга — сделать код программы легче для понимания.

Рефакторинг следует отличать от оптимизации производительности. Как и рефакторинг, оптимизация обычно не изменяет поведение программы, а только ускоряет ее работу. Но оптимизация часто затрудняет понимание кода, что противоположно рефакторингу.

Основные стимулы его проведения являются следующие задачи:

- необходимо добавить новую функцию, которая недостаточно укладывается в принятое архитектурное решение;
- необходимо исправить ошибку, причины возникновения которой сразу не ясны;

Сборка, внедрение и поставка ПО

Программа – это не только .exe модуль. Скорее всего он зависит от своего окружения: системные библиотеки и компоненты Windows, исполняющая среда, Dll библиотеки.

Средство создания дистрибутивов мы будем называть *инсталлятор*. Под словом *дистрибутив* понимаем распространяемые файлы программы — обычно это Setup.exe плюс, возможно, еще несколько файлов.

Bootstrapper. Эта часть программы установки запускается первой и ее задача — подготовить все необходимое для работы installation engine. В

частности, в случае использования Windows Installer, bootstrapper обеспечивает установку или обновление Windows Installer до требуемой версии.

Программа установки либо использует свой собственный installation engine, либо пользуется услугами сервиса Windows Installer. Installation engine выполняет всю основную работу: взаимодействует с пользователем через GUI, создает/удаляет папки и ключи реестра, копирует файлы, регистрирует компоненты.