Инструментальные средства разработки программного обеспечения

1. Основные понятия темы.

Программное обеспечение — совокупность программ обработки данных и необходимых для их использования документов.

Системное ПО — это совокупность программ для обеспечения работы компьютера.

Прикладное ПО — это комплекс программ для решения задач конкретной предметной области.

Инструментальное ПО — это совокупность программ для разработки, отладки и внедрения новых программ.

Технология программирования — совокупность методов и средств, используемых в процессе разработки ПО.

Жизненный цикл ПО — период времени от начала разработки до окончания эксплуатации.

Проектирование — процесс создания архитектуры и структуры программной системы.

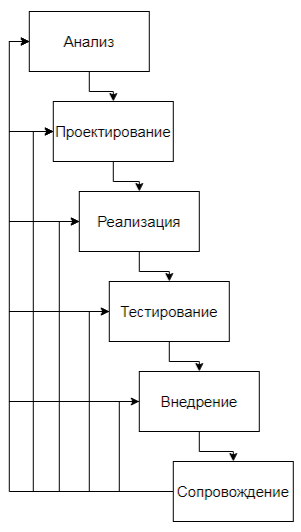


Рисунок 1 — циклическая модель проектирования ПО.

2. Описание этапов реализации программного кода.

1. Анализ — сбор и формализация требований к ПО, определение функциональных и нефункциональных спецификаций системы. Проводится анализ предметной области и построение концептуальной модели.
2. Проектирование — разработка архитектуры ПО, проектирование структуры данных, интерфейсов и алгоритмов. Создание UML-диаграмм, описание модулей и определение технологического стека.
3. Реализация — кодирование программных модулей, компиляция исходного кода и модульное тестирование. Разработка пользовательского интерфейса и интеграция компонентов системы.
4. Тестирование — выявление и документирование дефектов, проверка соответствия требованиям и стандартам качества.
5. Внедрение — развертывание ПО в целевой среде, миграция данных и конфигурирование системы. Проведение приемочного тестирования и обучение конечных пользователей.
6. Сопровождение — мониторинг работы системы, устранение дефектов и выпуск патчей. Адаптивная и перфективная модификация ПО на основе обратной связи от эксплуатации.

3. Системы программирования включают:

1. Компилятор (транслятор).
2. Интегрированную среду разработки программ (не всегда).
3. Отладчик.
4. Средства оптимизации кода программ; - набор библиотек.
5. Редактор связей.
6. Сервисные средства (утилиты) (для работы с библиотеками, текстовыми и двоичными файлами).
7. Справочные системы.
8. Систему поддержки и управления продуктами программного комплекса.

4. Функции современных компиляторов.

Современные компиляторы преобразуют код в машинный, выполняют лексический, синтаксический и семантический анализ, а также оптимизируют программу для повышения производительности и уменьшения размера исполняемого файла, а также осуществляют линковку, объединяя скомпилированные модули в единый исполняемый файл.

5. Современные средства программирования.

Python — язык программирования общего назначения с простым синтаксисом для веб-разработки, анализа данных и автоматизации.

C++ — компилируемый язык для создания высокопроизводительных приложений, таких как игры, операционные системы и драйверы.

MS Visual Studio — интегрированная среда разработки от Microsoft для программирования на C++, C#, Python и других языках.

Oracle — коммерческая система управления реляционными базами данных для крупных корпоративных приложений.

MS SQL Server — система управления базами данных от Microsoft для хранения и обработки структурированных данных.

MySQL — популярная бесплатная система управления базами данных с открытым исходным кодом, часто для веб-приложений.

6. Этапы проектирования приложений.

1. Сбор и анализ требований — определение целей проекта, функциональности и ожиданий заказчика и пользователей.
2. Планирование — оценка сроков, ресурсов, бюджета и выбор технологического стека.
3. Проектирование архитектуры и UI/UX — создание общей структуры системы, интерфейсов и проектирование пользовательского опыта.
4. Прототипирование — создание упрощенной рабочей модели для проверки ключевых идей.
5. Непрерывная разработка — написание кода по частям, часто с использованием Agile-методологий.
6. Тестирование — проверка качества приложения на наличие ошибок и соответствие требованиям.
7. Развертывание — выгрузка приложения на рабочие серверы и его запуск для пользователей.
8. Сопровождение и обновление — исправление ошибок, техническая поддержка и добавление нового функционала.

7. Нотации и средства для этапа проектирования.

Нотация — это система обозначений, предназначенная для описания структуры системы, элементов данных, этапов обработки; может включать графы, диаграммы, таблицы, схемы алгоритмов, формальные и естественные языки. Например, метод JSP реализуется с помощью нотации, базирующейся на применении четырех базовых конструкций данных. Современной нотацией методологии SADT является IDEF0.

Средства — это инструментарий для поддержки методов, помогающий пользователям при создании и редактировании графического проекта в интерактивном режиме, способствующий организации проекта в виде иерархии уровней абстракции, выполняющий проверки соответствия компонентов. Например, средством, поддерживающим метод JSP, является SmartDraw. IDEF0 поддерживается средством BPwin.

8. Вывод.

🤔🤔🤔