**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.

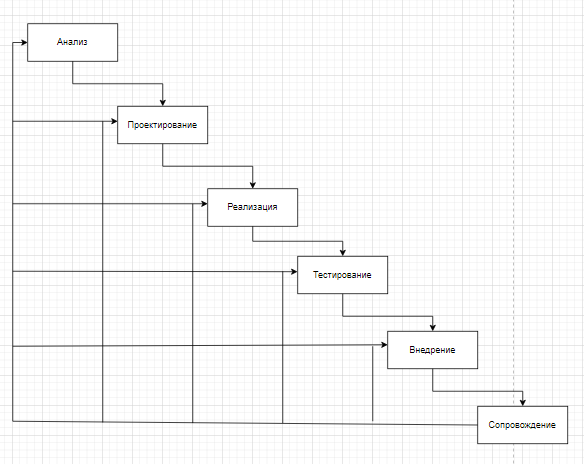
Проектирование — процесс создания архитектуры и структуры программной системы.

Рисунок 1 — циклическая модель проектирования ПО.

**2. Oписание этапов реализации программного кода.**

Анализ — сбор и формализация требований к ПО, определение функциональных и нефункциональных спецификаций системы. Проводится анализ предметной области и построение концептуальной модели.

Проектирование — разработка архитектуры ПО, проектирование структуры данных, интерфейсов и алгоритмов. Создание UML-диаграмм, описание модулей и определение технологического стека.

Реализация — кодирование программных модулей, компиляция исходного кода и модульное тестирование. Разработка пользовательского интерфейса и интеграция компонентов системы.

Тестирование — выявление и документирование дефектов, проверка соответствия требованиям и стандартам качества.

Внедрение — развертывание ПО в целевой среде, миграция данных и конфигурирование системы. Проведение приемочного тестирования и обучение конечных пользователей.

Сопровождение — мониторинг работы системы, устранение дефектов и выпуск патчей. Адаптивная и перфективная модификация ПО на основе обратной связи от эксплуатации.

# **3. Схема процесса описания реализации программного кода**

Современные системы программирования представляют собой комплекс взаимосвязанных компонентов, включающий:

* интегрированную среду разработки, объединяющую текстовый редактор и отладчик;
* компиляторы для преобразования исходного кода;
* библиотеки с готовыми программными модулями;
* системы контроля версий (такие как Git) для отслеживания изменений;
* вспомогательные инструменты: сборщики проектов и утилиты для работы с файлами;
* редактор связей;
* справочные системы;
* средства управления программными проектами.

**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** — интерпретируемый язык высокого уровня, отличающийся простотой синтаксиса, легкостью чтения кода и широкой областью применения.

**C++** — компилируемый язык программирования общего назначения, созданный как расширение языка Си и характеризующийся высокой производительностью.

**VSS** — служба теневого копирования томов в ОС Windows, обеспечивающая создание резервных копий данных.

**Microsoft Visual Studio** — комплексная среда разработки, предоставляющая полный набор инструментов для создания, тестирования и развертывания приложений.

**Oracle**  — многомодельная СУБД, предназначенная для управления большими объемами структурированных и неструктурированных данных.

**MySQL** — широко распространенная СУБД с открытым исходным кодом, ориентированная на эффективное хранение и организацию структурированных данных.

### **6. Этапы проектирования приложений**

Процесс проектирования приложений состоит из последовательных стадий: анализа требований, формирования концепции, разработки архитектурного решения и создания технической документации.

**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**

Мы выполнили практическую работу, мы молодцы.