**Лабораторная работа №4**

**Тема:** Изучение методологий структурного проектирования ПО.

**Цель работы:**

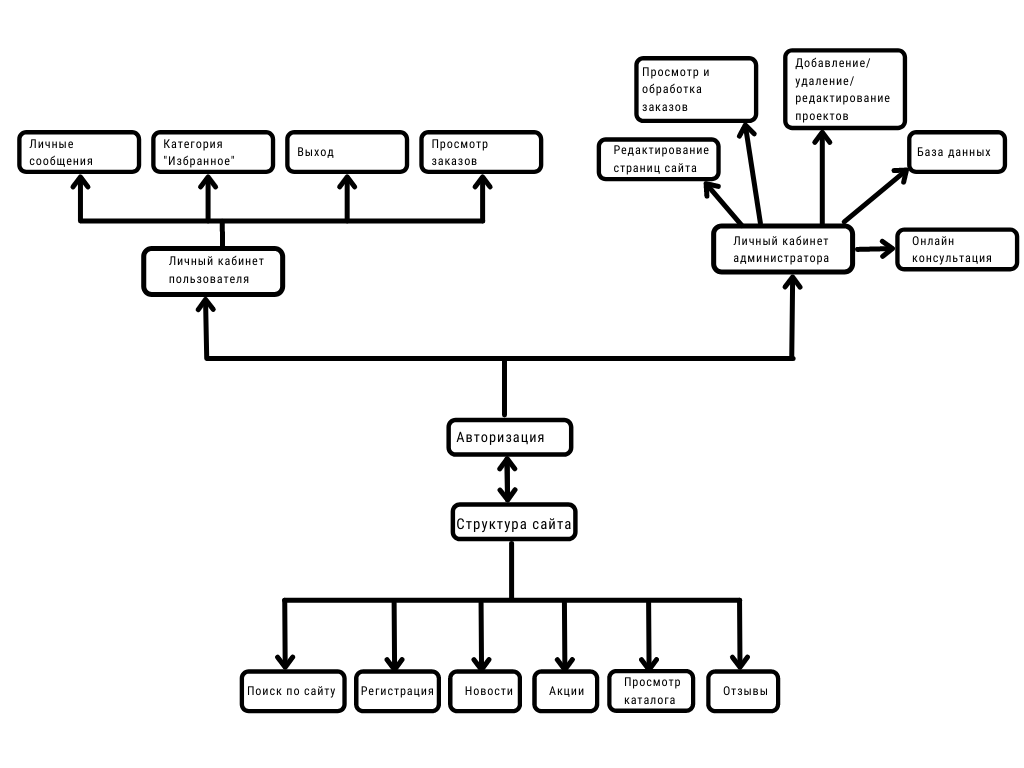
1. Изучить основные подходы, используемые при проектировании программного

продукта структурным способом.

2. Научиться использовать нисходящее и восходящее проектирование для разбиения

исходной задачи на подзадачи с их последующей детализацией на следующих уровнях.

**Задание 1**



**Контрольные вопросы:**

1. Методы структурного проектирования представляют собой комплекс технических и организационных принципов системного проектирования.

Типичными методами структурного проектирования являются:

* нисходящее проектирование, кодирование и тестирование программ;
* модульное программирование;
* структурное проектирование (программирование) и др.

Достоинства структурного программирования:

* Программы любой сложности и размера можно получать на основе соответствующего сочетания расширенного базиса управляющих структур: следования, ветвления, организации циклов и вызова подпрограмм; при этом все перечисленные структуры допускают только один вход и один выход.
* программа может быть проанализирована проверкой ее структуры, позволяя обнаруживать в ней ошибки уже на стадии проектирования

Недостатки структурного программирования:

* невозможно полностью отказаться от меток и операторов безусловного перехода при создании сложных ПП
* структурное программирование предоставляет способ разработки функций, которые могли служить блоками для построения программ. Однако ни гибкость этого метода, ни масштабы использования не позволяли существенно ускорить массовое программирование.
* локальные модификации могли нарушить работоспособность всей системы.
* далеко не все задачи поддаются алгоритмическому описанию и тем более алгоритмической декомпозиции, как того требует структурное программирование.

2. Структурное программирование представляет собой совокупность рекомендуемых технологических приемов, охватывающих выполнение всех этапов разработки ПП и основывается на следующих принципах:

* нисходящей разработки, рекомендующей на всех этапах вначале определять наиболее общие моменты, а затем поэтапно выполнять детализацию.
* сквозного структурного контроля, предполагающего проведение содержательного контроля всех этапов разработки.
* структурного программирования, рекомендующего определенные структуры алгоритмов и стиль программирования.

3. Стратегии структурного проектирования:

Общие стратегии

Это обычно часто упоминаемые и общепринятые стратегии:

“разделяй-и-властвуй” и пошаговое уточнение

проектирование “сверху-вниз” и “снизу-вверх”

абстракция данных и сокрытие информации

итеративный и инкрементальный подход

и другие...

Функционально-ориентированное или структурное проектирование

Это один из классических методов проектирования, в котором декомпозиция сфокусирована на идентификации основных программных функций и, затем, детальной разработке и уточнении этих функций “сверху-вниз”. Структурное проектирование, обычно, используется после проведения структурного анализа с применением диаграмм потоков данных и связанным описанием процессов. Исследователи предлагают различные стратегии и метафоры или подходы для трансформации DFD в программную архитектуру, представляемую в форме структурных схем.

Объектно-ориентированное проектирование

Представляет собой множество методов проектирования, базирующихся на концепции объектов. Данная область активно эволюционирует с середины 80-х годов, основываясь на понятиях объекта (сущности), метода (действия) и атрибута (характеристики). Здесь главную роль играют полиморфизм и инкапсуляция, в то время, как в компонентно-ориентированном подходе большее значение придается мета-информации.

Проектирование на основе структур данных

В данном подходе фокус сконцентрирован в большей степени на структурах данных, которыми управляет система, чем на функциях системы. Инженеры по программному обеспечению часто вначале описывают структуры данных входов (inputs) и выходов (outputs), а, затем, разрабатывают структуру управления этими данными (или, например, их трансформации).

Компонентное проектирование

Программные компоненты являются независимыми единицами, которые обладают однозначно определенными (well-defined) интерфейсами и зависимостями (связями) и могут собираться и развертываться независимо друг от друга. Данный подход призван решить задачи использования, разработки и интеграции таких компонент с целью повышения повторного использования активов (как архитектурных, так и в форме кода).